UNIVERSITÉ PARIS XIII - SORBONNE PARIS NORD

STELLANTIS

École doctorale Sciences, Technologies, Santé Galilée

Méthodes d'obtention de garantie des performances d'un système d'intelligence artificielle appliqué à la compression d'images pour la conception de véhicules autonomes

Methods for obtaining performance guarantees from an artificial intelligence system applied to image compression for the design of autonomous vehicles

---

THÈSE DE DOCTORAT
présentée par

Jérémy JASPAR

Laboratoire du Traitement et Transport de l'Information (L2TI)

pour l'obtention du grade de
DOCTEUR EN SCIENCES POUR L'INGENIEUR

soutenue le 24 mars 2025 devant le jury d'examen constitué de :

BEGHDADI, Azeddine, USPN L2TI, Président du jury
ROUMY, Aline, INRIA, Rapportrice
ZAHARIA, Titus, Télécom SudParis, Rapporteur
FOGELMAN-SOULIÉ, Françoise, Hub France IA, Examinatrice
CHAARI, Lotfi, Toulouse INP, Examinateur
VIENNET Emmanuel, USPN L2TI, Directeur de thèse
GUALANDRIS David, Stellantis, Co-encadrant de thèse

Dédicace

À mon défunt père.

Remerciements

Je tiens tout d'abord à exprimer ma profonde gratitude à mon directeur de thèse, Emmanuel Viennet, pour sa confiance, sa disponibilité et son accompagnement constant au cours de ces trois années. Sa rigueur scientifique, la clarté de ses conseils et la pertinence de ses orientations ont été essentiels à la bonne conduite de ce travail. Je le remercie tout particulièrement pour sa patience, son écoute, ainsi que pour sa réactivité face à mes sollicitations, parfois urgentes et de dernière minute. Il a su m'accorder son soutien dans les moments de doute, tout en me laissant l'autonomie nécessaire pour développer mes idées et ma propre méthode de travail. J'ai énormément appris à ses côtés, tant sur le plan scientifique qu'humain, et je lui en suis profondément reconnaissant.

Je remercie très sincèrement les membres du jury qui m'ont fait l'honneur de participer à l'évaluation de cette thèse. M. Azedinne Beghdadi, président du jury, pour son expertise, la clarté de ses remarques et la qualité de nos échanges lors de la soutenance. Mme Aline Roumy et M. Titus Zaharia, rapporteurs, pour l'attention qu'ils ont portée à ce travail, la pertinence de leurs commentaires, et la richesse de leurs retours écrits, qui m'ont permis de prendre du recul sur mes contributions et d'en affiner la présentation. M. Chaari Lotfi, examinateur, pour ses observations constructives et pour l'intérêt qu'il a porté à ce projet de recherche. Leur engagement, leur écoute et la qualité de leurs interventions ont été très appréciés et ont grandement enrichi cette soutenance.

Je souhaite exprimer ma reconnaissance envers mes encadrants industriels chez Stellantis. M. Fabien Figueres, qui m'a accueilli et guidé avec beaucoup de gentillesse durant ma première année, a su nourrir ma réflexion à travers des discussions stimulantes et des remarques toujours pertinentes. Je suis heureux de le compter aujourd'hui parmi mes amis, et je lui souhaite le meilleur pour sa nouvelle vie à Bordeaux.

M. David Gualandris, qui a ensuite repris le flambeau avec implication et enthousiasme, m'a également apporté un précieux soutien, tant humain que professionnel. Encadrant engagé et membre du jury, il a été présent à mes côtés tout au long de ces trois années. Toujours à l'écoute, disponible pour m'accompagner dans mes travaux, il m'a guidé avec une implication remarquable. Il a été un soutien important, y compris dans les moments les plus difficiles, et j'ai énormément appris à ses côtés. Je lui adresse un grand merci.

Enfin, M. Jean-Louis Sauvaget, pour son suivi rigoureux, ses conseils techniques avisés et ses apports précieux concernant l'application Field Monitoring, toujours disponible pour répondre à mes questions et me former avec patience et pédagogie.

Un remerciement tout particulier à Mme Françoise Fogelman-Soulié, dont l'exigence intellectuelle, la justesse des remarques et la pertinence des questionnements ont été une véritable force motrice tout au long de cette thèse. Ses interventions ont souvent permis de faire avancer mes réflexions dans la bonne direction. Je la remercie également très sincèrement pour son accueil chaleureux lors de la réunion que nous avons pu organiser chez elle, un moment précieux de collaboration et de convivialité, autour de café et de croissants, qui restera un beau souvenir dans ce parcours.

Je remercie chaleureusement toute l'équipe VENG de Stellantis, qui m'a accueilli avec chaleur et intégré avec convivialité dans ses projets comme dans ses traditions. Merci pour les échanges autour d'un café, les jeudis croissants, les repas partagés, les pots de départ, les soutiens du quotidien et les nombreux moments de franche rigolade. J'ai eu une chance incroyable de rejoindre une équipe aussi humaine et dynamique.

Je tiens également à adresser un remerciement tout particulier à M. David Routier et M. Vincent Herbert pour la confiance et le soutien qu'ils m'ont témoigné tout au long de ces trois années. Leur appui a été déterminant, notamment pour le financement du projet de génération d'images simulées et pour l'opportunité professionnelle offerte au sein de Stellantis. Leur reconnaissance et leur engagement ont grandement compté pour moi et m'ont encouragé à mener ce travail à son terme.

Je remercie également tout particulièrement M. Nicolas De Falci pour sa présence constante, sa disponibilité et sa bonne humeur. Un immense merci aussi pour sa venue à ma soutenance, malgré le long déplacement que cela impliquait : ce geste m'a sincèrement touché. J'espère avoir la chance de retrouver, dans la suite de mon parcours, d'autres collègues aussi attentionnés, et je lui souhaite une magnifique retraite.

Un grand merci également à M. Jean-Christophe Carniel, pour nos discussions passionnantes sur l'aviation, les voyages, et pour son humour toujours au rendez-vous.

Merci également à Mme Sandrine Loze pour son rôle, souvent en coulisses mais ô combien essentiel, dans l'encadrement des doctorants au sein de Stellantis. Son aide précieuse et son efficacité ont grandement facilité mon parcours.

Je n'oublie pas non plus les membres de l'équipe Software X, et en particulier M. David Bertrand, grâce à qui j'ai pu accéder à l'infrastructure de calcul indispensable à mes recherches, notamment les ressources GPU, sans lesquelles ce projet n'aurait pu aboutir.

Un mot également pour M. Matthieu Donain, qui a été un fantastique maître de stage et dont le soutien en amont du projet a, en partie, permis que je puisse réaliser cette thèse. Je lui en suis sincèrement reconnaissant.

Je souhaite aussi remercier du fond du cœur ma famille, et tout particulièrement ma mère, Édith Labyod, et mon défunt père, Bernard Jaspar, pour leur soutien indéfectible, leur présence rassurante et leur aide concrète durant les périodes les plus difficiles. Merci également à mon frère, ainsi qu'à mes oncles, tantes, cousins et cousines, pour leurs encouragements, leur écoute et leur soutien constant.

Enfin, je remercie mes amis pour leur patience et leur soutien moral. Une pensée toute particulière pour M. Émilien Bodiot, docteur en mathématiques, pour les nombreux conseils précieux, les discussions intellectuellement stimulantes et les moments partagés. Ces échanges ont été fondamentaux, tant sur le plan mathématique qu'humain, et m'ont permis de garder confiance dans mes idées.

Un immense merci à toutes celles et ceux qui ont, de près ou de loin, contribué à cette aventure.

Résumé

Ces dernières années, l'intelligence artificielle (IA) a transformé de nombreux secteurs, notamment l'automobile, la médecine, la finance et l'aéronautique, en améliorant des tâches complexes telles que la détection d'anomalies ou la prise de décision. Cependant, les systèmes d'IA, en particulier ceux basés sur l'apprentissage profond, manquent de garanties solides en matière de fiabilité, ce qui limite leur adoption dans des applications critiques comme la conduite autonome ou le diagnostic médical. Le principal défi réside dans l'imprévisibilité de ces "boîtes noires" et dans l'absence de données couvrant la totalité des scénarios possibles. Les simulateurs et modèles génératifs comme les GANs ou modèles de diffusion offrent des perspectives pour générer des données synthétiques réalistes et élargir le spectre des cas de test, mais des questions sur leur réalisme et leur pertinence persistent.

Dans ce contexte, la thèse, menée en collaboration avec le laboratoire L2TI et Stellantis, vise à développer une méthodologie pour garantir les performances des systèmes d'IA dans des scénarios critiques, en combinant approches statistiques et génération de données synthétiques. L'étude porte sur un cas concret dans la conduite autonome, intégrant un module de compression d'image et un module de détection d'objets, pour évaluer l'impact des distorsions générées par la compression sur la performance de détection. En utilisant des données réelles et synthétiques, cette méthodologie permet de définir des métriques d'erreur, d'identifier les scénarios critiques, et d'estimer la probabilité de dépassement de seuils de tolérance. L'objectif est de garantir que la qualité des données compressées reste suffisante pour des analyses fiables, minimisant ainsi les risques d'erreurs.

Abstract

In recent years, artificial intelligence (AI) has transformed numerous sectors, including automotive, healthcare, finance, and aerospace, by enhancing complex tasks such as anomaly detection and decision-making. However, AI systems, particularly those based on deep learning, lack strong reliability guarantees, limiting their adoption in critical applications like autonomous driving or medical diagnosis. The main challenge lies in the unpredictability of these "black boxes" and the absence of data covering all possible scenarios. Simulators and generative models, such as GANs or diffusion models, present promising solutions for generating realistic synthetic data and expanding the spectrum of test cases, but questions about their realism and relevance remain.

In this context, the thesis, conducted in collaboration with the L2TI laboratory and Stellantis, aims to develop a methodology to guarantee the performance of AI systems in critical scenarios by combining statistical approaches and synthetic data generation. The study focuses on a concrete case in autonomous driving, involving an image compression module and an object detection module, to evaluate the impact of compression-induced distortions on detection performance. Using real and synthetic data, this methodology defines error metrics, identifies critical scenarios, and estimates the probability of exceeding tolerance thresholds. The objective is to ensure that the quality of compressed data remains sufficient for reliable analyses, thereby minimizing error risks.

Publications

Workshops

— Jérémy Jaspar, Emmanuel Viennet, David Gualandris, Jean-Louis Sauvaget et Françoise Fogelman-Soulié, "Using Synthetic Images to Improve and Test Object Detection in the context of the Autonomous Vehicle", in IEEE 12th European Workshop on Visual Information Processing (EUVIP), 2024

Brevets

— Jérémy Jaspar, Emmanuel Viennet, David Gualandris, Jean-Louis Sauvaget et Françoise Fogelman-Soulié, "Procédé et dispositif de compression avec pertes et relative aux zones d'intérêts d'images numériques utilisant des paramètres déterminés à partir d'un réseau de neurones", 2023, N° FR2304591, en instance.

Table des matières

Remerciements . . . 3
Résumé . . . 5
Publications . . . 7
Table des matières . . . 10
Nomenclature . . . 12

1 Introduction **13**
1.1 Contexte et motivation . . . 13
1.2 Application industrielle . . . 15
1.3 Contribution de la thèse . . . 17
1.4 Structure de la thèse . . . 19

2 État de l'art pour la garantie d'IA **21**
2.1 Introduction . . . 21
2.2 Vérification formelle . . . 22
2.2.1 Solveurs SMT . . . 23
2.2.2 Arithmétique d'Intervalles et Propagation Symbolique . . . 24
2.2.3 Programmation Linéaire . . . 25
2.2.4 Branch and Bound (BaB) . . . 27
2.2.5 Relaxation Linéaire ou Convexe . . . 28
2.2.6 Interprétation Abstraite . . . 29
2.2.7 Constante de Lipschitz . . . 31
2.2.8 Apprentissage sous contraintes . . . 32
2.2.9 Conclusion . . . 33
2.3 Certification statistique . . . 34
2.3.1 Randomized Smoothing . . . 34
2.3.2 Analyse PAC-Bayes . . . 36
2.3.3 Conformal Prediction . . . 39
2.3.4 Conclusion . . . 42
2.4 Conclusion . . . 43
2.5 Récapitulatif de l'état de l'art pour la garantie d'IA . . . 44

3 État de l'art pour l'application industrielle **46**
3.1 Introduction . . . 47
3.2 Compression apprise par deep learning . . . 47
3.2.1 Introduction . . . 47
3.2.2 Métriques . . . 50
3.2.3 Modèles Entropiques . . . 52
3.2.4 Approches basées sur les auto-encodeurs . . . 56
3.2.5 Approches basées sur les modèles de contexte . . . 58
3.2.6 Approches multi-échelles . . . 59
3.2.7 Approches basées sur l'attention et le contenu . . . 61
3.2.8 Approches basées sur les Transformers . . . 63
3.2.9 Approches basées sur les modèles génératifs . . . 64
3.2.10 Autres approches . . . 67
3.2.11 Approches à taux de compression variable . . . 69

3.2.12 Approches ROI & orientées tâches . . . 73
3.2.13 Conclusion . . . 77
3.3 Détection d'objets par deep learning . . . 77
3.3.1 Introduction . . . 78
3.3.2 Métriques . . . 79
3.3.3 Approches Two-stage . . . 82
3.3.4 Approches Single-stage . . . 87
3.3.5 Approches basées sur les Transformers . . . 91
3.3.6 Conclusion . . . 94
3.4 Génération d'image . . . 94
3.4.1 Métriques . . . 95
3.4.2 Approches à base de VAE . . . 97
3.4.3 Approches à base de GAN . . . 98
3.4.4 Approches à base de Diffusion . . . 101
3.4.5 Génération dans le domaine automobile . . . 105
3.4.6 Génération d'images critiques . . . 110
3.4.7 Conclusion . . . 113
3.5 Conclusion . . . 113
3.6 Récapitulatif de l'état de l'art pour l'application industrielle . . . 115

4 Stratégies d'obtention de garanties 118
4.1 Introduction . . . 118
4.2 Méthodes d'estimation statistique . . . 120
4.2.1 Estimation par Bootstrap . . . 120
4.2.2 Notions de probabilité . . . 122
4.3 Méthodologie pour l'obtention de garanties de performances . . . 123
4.3.1 Identification de l'ODD et acquisition de données . . . 124
4.3.2 Estimation Statistique et Probabiliste du Risque d'Erreur . . . 127
4.3.3 Génération de données critiques aux modules d'IA . . . 133
4.3.4 Interprétation de la méthode . . . 136
4.3.5 Comment rajouter des données ? . . . 138
4.4 Conclusion et perspectives . . . 138

5 Application industrielle : définition du système, conception et analyse des données multi-sources 140
5.1 Introduction . . . 140
5.2 Définition du système et de l'ODD . . . 141
5.3 Acquisition et analyse de données . . . 145
5.3.1 Données naturelles . . . 145
5.3.2 Données Simulées . . . 168
5.3.3 Étude sur l'utilisation de données simulées . . . 174
5.3.4 Récapitulatif de l'ensemble des jeux de données . . . 179
5.3.5 Conclusion et perspectives . . . 179

6 Application industrielle : estimation de garanties 181
6.1 Introduction . . . 181
6.2 Apprentissage des modèles . . . 182
6.2.1 Entrainement de la compression . . . 182
6.2.2 Entrainement de la Détection . . . 190
6.2.3 Entrainement de la Diffusion . . . 191
6.2.4 Conclusion . . . 195
6.3 Analyse de l'effet de la compression sur la détection . . . 195
6.4 Estimation des probabilités d'erreur . . . 200
6.4.1 Estimation intra-datasets . . . 202
6.4.2 Estimation inter-datasets . . . 206
6.4.3 Conclusion . . . 214
6.5 Optimisation des performances . . . 215
6.5.1 Réentrainer la détection . . . 216

6.5.2 Ré-entrainer la compression . . . 220
6.6 Génération d'images par diffusion . . . 222
6.6.1 Maximiser l'erreur liée à la compression . . . 224
6.6.2 Maximiser l'erreur de détection induite par la compression . . . 227
6.7 Conclusion et perspectives . . . 237

7 Vers une generalisation de la methodologie **238**
7.1 Flexibilité et limites de la méthodologie . . . 239
7.2 Recommandations pour l'adaptation . . . 240
7.3 Perspectives . . . 242
7.4 Conclusion . . . 243

8 Conclusion et perspectives **245**
Références . . . 247
Table des figures . . . 278
Liste des tableaux . . . 287

Nomenclature

$C \times H \times W$ dimensions des images dans le format PyTorch (C canaux, hauteur H, largeur W). Si C est omis, sa valeur par défaut est $C = 3$.

AC Arithmetic Coding

ANS Asymmetric Numeral System

BaB Branch and Bound

BPG Better Portable Graphics

bpp bits per pixel

CNN Convolutional Neural Network

CP Conformal Prediction

CRC Conformal Risl Control

ERM Empirical Risk Minimizer

FID Frechet Inception Distance

FPS Frame per second

GAN Generative Adversarial Network

HEIF High Efficiency Image File Format

HEVC High Efficiency Video Coding

IP Integer Programming

IS Inception Score

LDM Latent Diffusion Model

LP Linear Programming

LSTM Long Short Term Memory (réseau de neurones)

mAP mean Average Precision

MILP Mixed Integer Linear Programming

MIP Mixed Integer Programming

MLP Multilayer Perceptron

MPEG Moving Picture Experts Group

MS-SSIM Multi-Scale Structural Similartiy Index

MSE Mean Squarred Error

ODD Domaine de coneption foctionnelle (Operational Design Domain)

PAC Probably Approximately Correct

PSNR Peak Signal to Noise Ratio

ReLU Rectified Linear Unit

RoI Region of Interest

SDP Semi-Definite Programming

SMT Satisfiability Modula Theories

TIDE Type-Informed Detection Error

TP, FN, FP, MC True Positive, False Negative, False Positive, Miss-classification

VAE Variational Auto Encodeur

VVC Versatile Video Coding

Chapitre 1

Introduction

Contents

1.1 Contexte et motivation 13
1.2 Application industrielle 15
1.3 Contribution de la thèse 17
1.4 Structure de la thèse 19

1.1 Contexte et motivation

Ces dernières années, l'essor des méthodes dites d'intelligence artificielle (IA) a déclenché une révolution dans notre société, touchant de nombreux aspects de la vie quotidienne et transformant en profondeur divers secteurs, notamment le monde industriel. Par exemple, dans l'industrie automobile avec les nouvelles fonctionnalités de conduite autonome [aut, 2020], dans le domaine médical avec des outils d'aide au diagnostic ou à la détection d'anomalies physiologiques assistés par l'IA [Paul et al., 2021], dans la finance avec la détection automatique des fraudes [Bao et al., 2024], dans le commerce en ligne avec des systèmes de recommandation [Zhou et al., 2023], ou encore dans l'aéronautique [Boniol et al., 2020]. L'IA joue également un rôle clé dans l'industrie avec des applications comme la maintenance prédictive [Serradilla et al., 2022]. Dans certains cas, l'IA dépasse même les compétences humaines, excédant nos capacités dans des tâches complexes nécessitant une analyse rapide et précise de grandes quantités de données. À l'avenir, il est probable que cette expansion continue, affectant de nouveaux secteurs et renforçant l'intégration de l'IA dans tous les domaines de la société. Grâce à ces avancées, l'IA promet d'améliorer l'efficacité des processus, de réduire les coûts, d'optimiser la prise de décision et de proposer des solutions innovantes aux défis complexes auxquels nos sociétés sont confrontées.

Cependant, malgré leurs avancées impressionnantes, les méthodes d'intelligence artificielle manquent encore de garanties solides sur leur fiabilité. Les systèmes d'IA, en particulier ceux basés sur l'apprentissage automatique (Deep Learning), sont souvent perçus comme des "boîtes noires", composées de millions, voire de milliards de paramètres, dont le comportement peut être difficile à prédire dans toutes les situations possibles. Cela pose un risque pour les applications critiques où une défaillance peut avoir des conséquences graves. Par exemple, dans la conduite autonome, un modèle non fiable pourrait ne pas réagir correctement à des situations rares, entraînant des accidents potentiellement mortels. De même, dans le domaine médical, une erreur dans le diagnostic automatisé pourrait conduire à des traitements inappropriés, mettant en danger la vie des patients. C'est pourquoi il est impératif de développer des méthodes permettant de prouver que ces systèmes se comporteront de manière sûre et fiable dans toutes les situations.

L'apprentissage des modèles d'IA nécessite des ensembles de données de très grande taille, comptant des millions, voire des milliards d'exemples, dont la constitution est extrêmement coûteuse. Si l'on disposait d'exemples de test couvrant, sans biais, l'ensemble des scénarios possibles que le système pourrait rencontrer, il serait envisageable de garantir son bon fonctionnement. Cependant, pour des applications comme la conduite de véhicules autonomes, les situations potentielles sont si nombreuses et variées qu'il est difficile de toutes les couvrir par des données réelles. Face à cette difficulté, les industriels ont souvent recours à des simulateurs pour générer des scénarios de test. Toutefois, ces simulateurs, bien que pratiques, sont complexes à concevoir et ne produisent pas

toujours des données aussi réalistes que celles issues du monde réel, ce qui complique encore l'évaluation précise des performances des systèmes d'IA dans des conditions réelles. Ces dernières années, les modèles de *deep learning* dits génératifs ont connu un essor spectaculaire, offrant de nouvelles perspectives. Ces modèles, comme les réseaux génératifs adverses (GANs) [Goodfellow et al., 2014, Karras et al., 2019] et les modèles de diffusion [Ho et al., 2020, Rombach et al., 2022], sont capables de créer de nouvelles données réalistes en apprenant la distribution de données d'origine. Ils peuvent générer des images, du texte, des vidéos, ou même des scénarios simulés de manière très convaincante. Par exemple, les GANs [Kim et al., 2021, Wang et al., 2018c] peuvent produire des images de haute qualité, difficilement distinguables de celles capturées par des capteurs réels, tandis que les modèles de diffusion [Wang et al., 2024c, Zhao et al., 2024a, Li et al., 2024b, Hu et al., 2023, Gao et al., 2024, Su et al., 2024], utilisés notamment dans la création d'images complexes, montrent des résultats impressionnants dans la génération de détails fins. Ces prouesses techniques pourraient être une solution envisageable pour pallier le manque de données dans certaines applications critiques. En créant des ensembles de données synthétiques réalistes, ces modèles permettraient de couvrir un plus large éventail de scénarios de test, contribuant ainsi à améliorer la robustesse et la sécurité des systèmes d'IA. Cependant, ces méthodes posent des questions complexes comme comment caractériser le réalisme de ces images.

Cette thèse est menée dans le cadre d'une collaboration entre le Laboratoire L2TI (Laboratoire du Traitement et Transport de l'Information) de l'Université Sorbonne Nord et Stellantis l'un des plus grands constructeurs automobiles au monde, né de la fusion entre PSA Group et Fiat Chrysler Automobiles, regroupant des marques emblématiques telles que Peugeot, Citroën, Fiat, Jeep, et Maserati. L'objectif de la thèse est de développer une méthodologie pour obtenir des garanties de performance en vue d'évaluer la fiabilité des modules d'Intelligence Artificielle basés sur des modèles d'apprentissage profond, particulièrement dans des applications critiques. Cette méthodologie alliera des approches statistiques pour établir des bornes sur les taux d'erreurs et des techniques de génération d'exemples tests via des simulateurs et des modèles génératifs, avec une attention particulière aux scénarios rares et limites pour le système étudié. À partir de cas concrets, tels que l'utilisation de l'IA pour la reconnaissance d'objets dans des chaînes de traitement d'image comprenant une étape de compression et de décompression, des estimateurs seront développés pour garantir la fiabilité des modules d'IA. L'objectif est de vérifier que, même après compression, la qualité des données reste suffisante pour que les algorithmes de détection d'objets conservent leur niveau de performance. Toutefois, l'accent sera mis sur l'évaluation et l'assurance de la fiabilité des modèles existants, et utilisera des modèles de l'état de l'art sans nécessairement chercher à les améliorer. La méthodologie développée au cours de ce travail pourra être adaptée à d'autres domaines de l'IA industrielle, tels que le transport, l'aéronautique ou la médecine.

1.2 Application industrielle



FIGURE 1.1 – Le véhicule autonome (illustration générée par IA).

Les véhicules autonomes représentent l'une des avancées technologiques les plus prometteuses du secteur automobile, avec pour objectif de permettre aux voitures de naviguer sans intervention humaine tout en assurant sécurité, efficacité et confort. L'idée centrale est de créer des systèmes de conduite automatisés capables de percevoir leur environnement, de prendre des décisions et de contrôler le véhicule en fonction des situations rencontrées sur la route. Les applications de cette technologie sont nombreuses, allant de la réduction des accidents de la route à l'amélioration de la fluidité du trafic, en passant par une mobilité plus accessible pour les personnes à mobilité réduite. Il existe différents niveaux d'autonomie, définis par la *Society of Automotive Engineers* (SAE) [sae, 2021], allant du niveau 0 (absence totale d'automatisation) au niveau 5 (autonomie complète) :

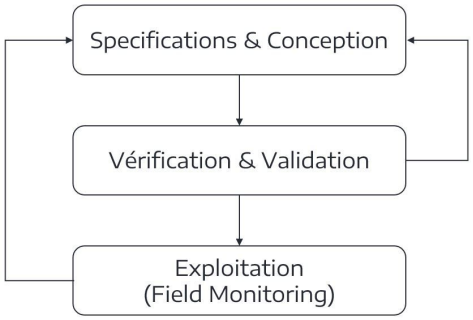
- Niveau 0 : Aucune automatisation. Le conducteur humain est responsable de toutes les fonctions du véhicule.
- Niveau 1 : Assistance au conducteur. Le véhicule peut prendre en charge une seule fonction automatisée, comme le contrôle de la vitesse (régulateur adaptatif) ou l'assistance à la direction, mais le conducteur doit rester actif et garder le contrôle.
- Niveau 2 : Automatisation partielle. Le véhicule peut contrôler à la fois la direction et l'accélération/freinage dans certaines situations, mais le conducteur doit surveiller la route et être prêt à reprendre le contrôle à tout moment.
- Niveau 3 : Automatisation conditionnelle. Le véhicule peut gérer toutes les fonctions de conduite dans des conditions spécifiques, appelées ODD (*Operational Design Domain*), qui désignent un cadre bien défini (ex : certaines routes, conditions météorologiques) dans lequel le véhicule peut opérer de manière autonome. En dehors de cet ODD, le conducteur doit reprendre le contrôle.
- Niveau 4 : Automatisation élevée. Le véhicule est capable de conduire de manière autonome dans un ODD bien défini sans intervention humaine, même si le conducteur ne répond pas à une demande de reprise de contrôle. Cependant, en dehors de cet ODD, le véhicule ne fonctionne pas en mode autonome.
- Niveau 5 : Automatisation complète. Le véhicule est totalement autonome et n'a besoin d'aucune intervention humaine, quels que soient l'environnement ou les conditions (ODD illimité).

La notion d'ODD est cruciale dans la définition des niveaux 3 et 4, car elle précise les limites dans lesquelles le véhicule peut fonctionner de manière sûre et autonome. L'ODD inclut des facteurs tels que la localisation, les types de routes, la météo et même l'heure du jour, et sert de cadre pour évaluer la sécurité et les capacités du

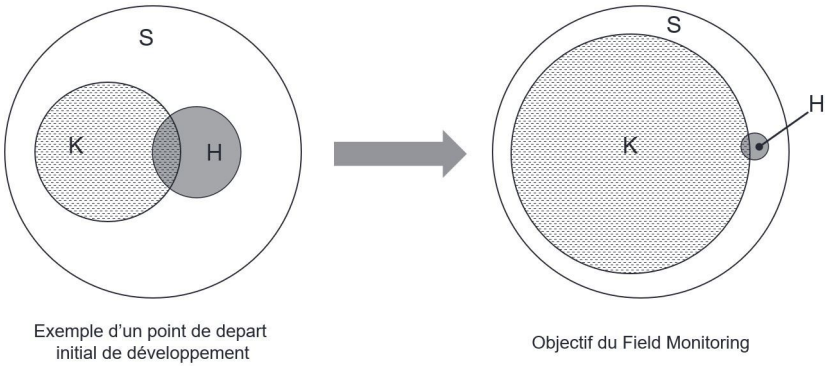
système autonome.

Les technologies embarquées, telles que les capteurs LiDAR, les caméras, les radars et les systèmes de traitement par intelligence artificielle, permettent aux véhicules de détecter et d'interpréter l'environnement pour garantir la sécurité et la performance du système. Actuellement, les véhicules de Stellantis sont principalement équipés de systèmes de conduite autonome de niveau 2. Cela inclut des fonctionnalités avancées d'aide à la conduite (ADAS) comme le maintien de voie, le régulateur de vitesse adaptatif et l'assistance au stationnement. Cependant, pour atteindre les niveaux 3 et supérieur, l'autonomie des véhicules s'accompagne de défis importants, notamment en matière de validation de la sécurité et de fiabilité dans des environnements variés et complexes.

L'ISO 21448, intitulée "Véhicules routiers – Sécurité de la fonction attendue (SOTIF)", a été développée par l'Organisation Internationale de Normalisation (ISO), une organisation indépendante regroupant plus de 160 organismes nationaux de normalisation. Elle est responsable de l'élaboration de normes internationales couvrant divers secteurs techniques, industriels et commerciaux. L'ISO 21448 se concentre spécifiquement sur la sécurité des systèmes de conduite autonome. L'une des recommandations est la mise en place d'un *Field Monitoring*. Ce système a pour but d'être embarqué sur les véhicules et de surveiller le bon fonctionnement des systèmes autonomes après les phases de conception et de validation (cf. Figure 1.2a). Il devra être capable d'acquérir les images et données recueillies dans certaines situations spécifiques, de stocker ces images et données, et enfin de les transférer vers des centres de données pour être ultérieurement analysé. En identifiant, interprétant et corrigeant les dysfonctionnements, le *Field Monitoring* permet de réduire les risques associés aux scénarios dangereux non anticipés et d'atténuer les scénarios connus à risque (cf. Figure. 1.2b). De plus, ce mécanisme de surveillance continue est crucial pour adapter régulièrement le système aux changements de l'environnement.



(a) Phases de développement et de suivi d'une fonction de conduite autonome.



(b) Acquérir des données pour réduire les risques associées aux scénarios dangereux (S : ensemble de tous les scénarios possibles, K : ensemble des scénarios connus et H : ensemble des scénarios dangereux).

FIGURE 1.2 – Objectif du Field Monitoring (Source ISO 21448 [iso, 2022]).

Ce système de *Field Monitoring* sera déployé dans des millions de véhicules, générant ainsi une quantité colossale de données, notamment des vidéos (ou ensembles d'images), qu'il faudra transférer et stocker. Cependant, ces transferts et ce stockage peuvent représenter un coût financier considérable, notamment en raison des

volumes énormes de données à traiter. Pour réduire ces coûts, l'utilisation de technologies de compression haute performance est indispensable. Les techniques récentes de compression, basées sur l'apprentissage profond, surpassent aujourd'hui les algorithmes classiques tels que JPEG ou MPEG, en adaptant la compression au contenu sémantique des images. Ces méthodes permettent de conserver plus d'informations pertinentes, tout en réduisant la taille des fichiers. Toutefois, les artefacts ou le bruit générés par ces techniques de compression peuvent avoir un impact négatif sur les algorithmes de reconnaissance de motifs, qui sont essentiels pour analyser les scènes et les objets capturés. Ces distorsions supplémentaires pourraient rendre certaines images inutilisables, ce qui entraînerait des retards critiques dans l'identification et la correction des scénarios défaillants. Cela pourrait retarder l'ajustement des modèles et potentiellement conduire à des accidents qui auraient pu être évités. Ainsi, il est primordial de s'assurer que ces distorsions ne dépassent pas un certain seuil, afin de garantir que la qualité des données reste suffisante pour permettre une reconnaissance précise des motifs et une analyse fiable des incidents, y compris les erreurs rares, qui doivent impérativement être prises en compte.

Le *Field Monitoring* fonctionne selon une chaîne de traitement d'images (cf. Figure 1.3) basée sur des événements déclencheurs, appelés *Triggers*, qui activent le système de surveillance lorsqu'une situation anormale ou critique est détectée. Ces *triggers* incluent, par exemple : une demande de reprise de contrôle par le conducteur (avec ou sans défaillance du système), le déclenchement de l'ABS (système antiblocage des roues), une incohérence dans la détection de trajectoire, un freinage brusque, un faible temps avant collision (quasi-déclenchement du freinage d'urgence automatique, AEB), la détection de choc (déploiement de l'airbag), ou encore une crevaison. Lorsque le *Field Monitoring* est activé par l'un de ces événements, il enregistre une fenêtre temporelle de 10 secondes, avec une fréquence d'au minimum 5 images par seconde, centrée autour du moment déclencheur ([-5s avant l'événement, +5s après l'événement]). Ces images sont ensuite compressées sous forme de fichier(s) binaire(s) pour réduire la taille des données et faciliter leur transfert. Les fichiers compressés sont ensuite envoyés vers un centre de données Stellantis pour être stockés. Dans une phase ultérieure, ces fichiers binaires sont récupérés et décompressés pour restaurer les images d'origine. Les images ainsi reconstruites passent alors par des systèmes d'analyse automatique, tels que des détecteurs d'objets ou des algorithmes de reconnaissance de scènes, afin d'identifier les éléments pertinents de l'environnement et d'analyser les causes qui ont conduit à l'événement critique.

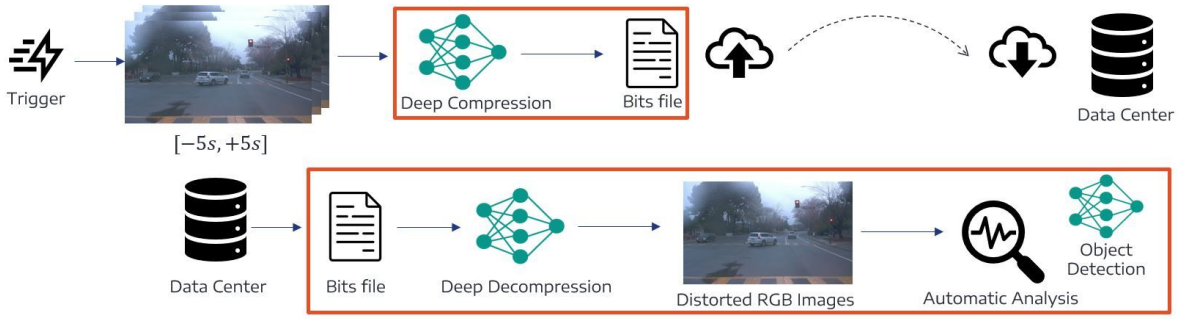


FIGURE 1.3 – Fonctionnement complet du *Field Monitoring*. L'application étudiée dans cette thèse concerne les zones encadrées en rouge.

En conclusion, notre étude de cas porte sur une chaîne de traitement d'image intégrant deux modules basés sur l'apprentissage profond (zones encadrées en rouge dans la Figure 1.3) : un modèle de compression d'image et un modèle de détection d'objets. L'objectif est d'appliquer notre méthodologie d'obtention de garanties de performances pour s'assurer que l'impact des distorsions générées par la compression sur la détection ne dépassent pas un seuil de tolérance prédéfini. Cela est essentiel pour garantir que la qualité des données compressées reste suffisante pour permettre une détection d'objets fiable et précise. En validant ce seuil, nous visons à minimiser les risques de dégradations qui pourraient nuire à l'analyse des scènes, assurant ainsi que la chaîne de traitement conserve son efficacité.

1.3 Contribution de la thèse

Les contributions de cette thèse se concentrent sur le développement d'une méthodologie permettant d'estimer des garanties de performance pour des systèmes intégrant un ou plusieurs modèles d'apprentissage profond,

avec une application spécifique sur une fonction de conduite autonome. Cette approche repose sur des études statistiques appliquées à de vastes ensembles de données de test, qui doivent être représentatives et diversifiées pour couvrir l'ensemble des situations possibles définies par le domaine de conception fonctionnelle (ODD) de la fonction étudiée.

Les étapes de la méthodologie sont les suivantes :

1. Définition du système et d'une métrique d'erreur : La première contribution consiste à formaliser le système d'IA étudié en définissant clairement ses composants, ses limites, ainsi qu'une métrique d'erreur spécifique liée à la performance que l'on souhaite garantir. Cette métrique permet d'évaluer la performance globale du système selon des critères de sécurité et de fiabilité.
2. Identification de l'ODD et de sa complexité : l'ODD représente le cadre dans lequel le système est conçu pour opérer en toute sécurité. Plus cet ODD est vaste et complexe, plus la tâche de validation devient ardue. Pour des systèmes simples, comme la détection de défauts sur des pièces industrielles, la définition de l'ODD peut être relativement aisée. En revanche, pour des applications plus complexes comme la conduite autonome, l'identification des situations possibles nécessite une expertise approfondie et la collaboration avec des experts métiers pour définir des scénarios couvrant des cas fréquents et des cas critiques.
3. Acquisition de données représentatives de l'ODD : l'efficacité de la méthodologie repose sur l'utilisation de données variées provenant de différentes sources.
   - Données réelles : capturées par des capteurs en situation réelle, ces données sont précieuses pour évaluer le comportement du système dans des conditions authentiques.
   - Données synthétiques simulées : issues de simulateurs, elles permettent de tester le système dans divers environnements complexes et d'évaluer son comportement sur des scénarios rares, qui seraient impossibles à reproduire dans le monde réel en raison de coûts trop élevés ou de contraintes éthiques. Cependant, il est crucial de vérifier que leur utilisation est pertinente, c'est-à-dire que le comportement du système est similaire à celui observé avec des données réelles.
   - Données synthétiques générées : créées via des modèles génératifs (par exemple, des GANs ou des modèles de diffusion), ces données doivent atteindre un niveau de réalisme suffisant pour être utilisées efficacement. Leur validité repose sur leur capacité à reproduire des conditions visuelles proches des observations réelles. Et similaire aux données simulées, il faudra vérifier que le comportement du système sur ces données est similaire à celui observé avec des données réelles.

   Plus la diversité des exemples de données est grande, plus les estimations de performance seront précises et représentatives des situations potentielles. Cela inclut la couverture de cas rares et critiques, qui sont souvent les plus difficiles à identifier et à modéliser.
4. Estimation statistique et probabiliste des garanties de performance : l'objectif est de fournir une évaluation quantitative du risque que le système dépasse un seuil d'erreur donné. Pour ce faire, des quantités statistiques, telles que la probabilité d'erreur de la forme $P(e(x) \geq \epsilon)$, où x représente l'entrée du système, e est la fonction d'erreur et ϵ le seuil d'erreur acceptable, sont calculées à partir des données disponibles en tenant compte des disparités entre les ensembles de données et des incertitudes liées à la distribution des scénarios dans l'ODD.
5. Optimisation du système et réduction des risques d'erreur : une fois les probabilités d'erreur et les incertitudes estimées, il est possible d'optimiser les performances du système par des étapes de ré-apprentissage. Cette optimisation consiste à réentraîner les modèles d'apprentissage profond avec des ensembles de données plus larges et plus diversifiés, notamment pour les scénarios identifiés comme critiques ou présentant un risque d'erreur élevé. L'ajout de ces données permet de réduire le risque de défaillance dans les contextes complexes, en renforçant la robustesse du modèle pour mieux gérer les cas non anticipés ou rares.

La méthodologie sera appliquée à une étude de cas intégrant une chaîne de traitement d'image dans un système de conduite autonome. Cette chaîne comprend un module de compression d'image et un module de détection d'objets, tous deux basés sur des modèles d'apprentissage profond. L'objectif est d'évaluer l'impact des artefacts et du bruit générés par le module de compression sur la performance globale de la détection d'objets. Cela nécessite d'estimer la probabilité qu'un certain seuil d'erreur soit dépassé et de s'assurer que la qualité des images compressées reste suffisante pour garantir une détection d'objets fiable et sécurisée. Nous démontrerons ainsi que les données jouent un rôle primordial dans l'estimation des garanties de performance, car plus le volume et la diversité des données utilisées sont importants, plus l'estimation est précise. En particulier, les données synthétiques revêtent une importance capitale, car elles permettent d'accéder à des scénarios rares ou

difficiles à capturer en situation réelle, comme des conditions météorologiques extrêmes ou des situations de trafic complexes. Ces données, générées par des simulateurs ou des modèles de *deep learning*, élargissent le champ d'étude en créant des exemples de test variés et contrôlés. Nous montrerons ainsi leur utilité cruciale pour compléter les données réelles et couvrir des cas d'usage non observés, renforçant ainsi la robustesse des estimations de performance. De plus, nous montrerons que la compression d'images par apprentissage profond surpasse les algorithmes classiques tels que JPEG ou WebP en adaptant la compression au contenu sémantique des images, ce qui réduit considérablement la perte d'information. Enfin, pour optimiser ces garanties et minimiser le risque d'erreur, différentes stratégies seront explorées, notamment le réentraînement des modèles avec des ensembles de données enrichis. Cette approche permettra d'améliorer la robustesse du système de compression-détection en prenant en compte les cas extrêmes et les scénarios critiques identifiés au cours de l'analyse.

Cette méthodologie intégrée permettra de valider les performances du système global tout en considérant les interactions entre les différents modules et les incertitudes inhérentes aux conditions d'utilisation réelles. La méthodologie proposée pourra ensuite être étendue à d'autres fonctions de systèmes autonomes et à des applications critiques dans divers domaines industriels.

1.4 Structure de la thèse

La structure de la thèse est organisée autour de plusieurs chapitres qui guident le lecteur depuis l'état de l'art jusqu'à l'application pratique de la méthodologie d'obtention de garanties de performances pour des systèmes intégrant des modèles d'apprentissage profond.

À la suite de cette introduction du sujet et de la problématique de la thèse, le chapitre 2 présente un état de l'art détaillé sur les méthodes existantes pour garantir la fiabilité des systèmes d'intelligence artificielle. Il explore les différentes approches théoriques et pratiques, comme les méthodes de vérification formelle, les techniques d'estimation statistique et les approches empiriques visant à évaluer les performances des systèmes d'IA dans des environnements variés, tout en mettant en lumière les limites théoriques et techniques.

Ensuite, dans le chapitre 3, nous ferons un état de l'art spécifique au cas d'application choisi pour cette thèse qui explore trois domaines technologiques essentiels. Le premier est constitué des techniques de compression d'images par *deep learning*, qui adaptent la compression au contenu sémantique des images pour maximiser le compromis entre qualité visuelle et taux de compression. Plusieurs architectures de compression seront explorées, chacune avec des caractéristiques distinctes. Le second domaine concerne les algorithmes de détection d'objets par apprentissage profond. Ces algorithmes ont pour objectif de localiser et classifier les objets dans une image avec une grande précision et en un temps relativement court. Cependant, chaque architecture présente des compromis différents en termes de vitesse, de précision et de robustesse face aux perturbations, notamment lorsqu'il s'agit d'analyser des images compressées. Le troisième domaine concerne les modèles de génération d'images, tels que les GANs (*Generative Adversarial Networks*) [Goodfellow et al., 2014] et les modèles de diffusion [Ho et al., 2020] qui sont actuellement les techniques les plus avancées pour produire des images synthétiques réalistes.

Le chapitre 4 est consacré à la définition de la stratégie d'obtention de garanties de performance pour des modèles de *deep learning*. Il débute par une présentation de quelques notions de probabilité, telles que la Loi des Grands Nombres (LGN) et le théorème de l'Espérance Totale, qui, sous certaines conditions, permettent d'estimer des espérances statistiques. Ces concepts fondamentaux servent de base théorique pour l'estimation des erreurs et la quantification du risque d'échec des modèles étudiés. Ensuite, des méthodes d'estimation statistique, comme la méthode par bootstrap, seront présentées pour évaluer l'incertitude des résultats obtenus.

Dans les chapitres 5 et 6, la méthodologie du chapitre 4 est mise en œuvre dans une étude de cas industrielle portant sur une chaîne de traitement d'images dans un système de conduite autonome. La chaîne inclut un module de compression d'image et un module de détection d'objets, tous deux basés sur des modèles d'apprentissage profond. L'objectif est de s'assurer que les artefacts générés par la compression n'affectent pas excessivement les performances de détection. Une fois les garanties de performance estimées, des méthodes d'optimisation seront explorées pour réduire les risques d'erreurs identifiés.

Enfin, dans le chapitre 7, nous développerons les enseignements tirés des expériences réalisées sur l'application, ainsi que les étapes vers l'élaboration d'une méthodologie générale visant à garantir les performances des modules d'IA. Le chapitre 8 conclura en présentant les points majeurs de cette étude, ainsi que ses résultats, tout en discutant des orientations futures de la recherche.

Chapitre 2

État de l'art pour la garantie d'IA

Contents

2.1 Introduction . . . 21
2.2 Vérification formelle . . . 22
2.2.1 Solveurs SMT . . . 23
2.2.2 Arithmétique d'Intervalles et Propagation Symbolique . . . 24
2.2.3 Programmation Linéaire . . . 25
2.2.4 Branch and Bound (BaB) . . . 27
2.2.5 Relaxation Linéaire ou Convexe . . . 28
2.2.6 Interprétation Abstraite . . . 29
2.2.7 Constante de Lipschitz . . . 31
2.2.8 Apprentissage sous contraintes . . . 32
2.2.9 Conclusion . . . 33
2.3 Certification statistique . . . 34
2.3.1 Randomized Smoothing . . . 34
2.3.2 Analyse PAC-Bayes . . . 36
2.3.3 Conformal Prediction . . . 39
2.3.4 Conclusion . . . 42
2.4 Conclusion . . . 43
2.5 Récapitulatif de l'état de l'art pour la garantie d'IA . . . 44

2.1 Introduction

L'enjeu de certifier les systèmes intégrant de l'intelligence artificielle (IA) est motivé par la nécessité de garantir la sécurité et la fiabilité des systèmes critiques, tels que ceux utilisés dans la conduite autonome, le domaine médical ou l'aéronautique. Cependant, bien que les termes "certifier" et "garantir" soient souvent employés de manière interchangeable dans la littérature, il est essentiel de rappeler qu'une certification formelle implique l'intervention d'une autorité extérieure. Celle-ci est délivrée par des organismes compétents, comme des autorités gouvernementales ou des instances de régulation telles que l'Union Européenne, et uniquement lorsque le système répond à des critères rigoureusement définis. Ainsi, la responsabilité de la certification incombe à ces autorités régulatrices, et non aux ingénieurs ou aux entreprises. Dans notre travail, nous nous concentrons sur la notion de garantie de performance, qui représente l'une des conditions indispensables à l'obtention d'une certification. Elle englobe deux aspects fondamentaux : d'une part, la vérification du bon fonctionnement du système dans toutes les situations possibles, ou à défaut, la réduction des probabilités de défaillance à un niveau acceptable ; d'autre part, sa robustesse, c'est-à-dire sa capacité à conserver des performances satisfaisantes en présence de perturbations. Notre étude se penche spécifiquement sur l'analyse des scénarios pouvant conduire à une défaillance, en cherchant à estimer une probabilité du type $P(e \geq \epsilon)$, avec e une mesure de l'erreur d'un module d'IA et ϵ un seuil d'acceptabilité. Garantir le bon fonctionnement des modules d'IA constitue un défi majeur, principalement en raison de l'opacité de ces modèles, souvent qualifiés de "boîtes noires", et du phénomène de la

malédiction de la dimensionnalité. Ce dernier fait référence à l'augmentation exponentielle du nombre de configurations possibles à mesure que la dimension des entrées croît. Comme le souligne [Balestriero et al., 2021], cela implique que l'apprentissage en haute dimension conduit inévitablement à de l'extrapolation, ce qui complique l'évaluation des performances sur des données non observées.

À ce jour, il n'existe pas de méthode universelle permettant de vérifier de manière exhaustive et systématique les performances des systèmes d'intelligence artificielle, en particulier pour les architectures complexes telles que les réseaux de neurones profonds. Les approches disponibles se concentrent généralement sur des aspects spécifiques, comme la robustesse face aux perturbations d'entrée, la stabilité des prédictions ou le respect de contraintes opérationnelles. Toutefois, leur efficacité dépend fortement de la nature du système étudié, de l'architecture du réseau, de sa taille et de sa complexité, ce qui limite leur applicabilité dans certains contextes industriels et critiques où des garanties rigoureuses sont nécessaires. Cette diversité d'approches soulève également des questions quant à leur capacité à couvrir l'ensemble des scénarios opérationnels possibles et à offrir des garanties interprétables et exploitables pour les systèmes étudiés.

Dans ce chapitre, nous proposons un état de l'art des méthodes existantes pour l'obtention de garanties de performances des systèmes intégrant des modules d'IA, en distinguant les méthodes de vérification formelle, qui offrent des garanties déterministes, et les méthodes de certification statistique, qui fournissent des estimations probabilistes des performances.

2.2 Vérification formelle

La vérification formelle des réseaux de neurones est un processus qui consiste à s'assurer qu'un réseau de neurones satisfait à certaines propriétés de sécurité, ce qui est crucial dans les applications où l'intelligence artificielle doit être fiable et sécurisée. Plus précisément, nous cherchons à vérifier si la sortie d'un réseau de neurones f respecte certaines contraintes basées sur les plages d'entrées. Par exemple, étant donné un réseau de neurones et des plages d'entrées définies par une norme L_p (souvent utilisée pour les images), l'objectif est de déterminer la plage de sorties correspondante, notée F, et de vérifier si elle satisfait une propriété de sécurité donnée P.

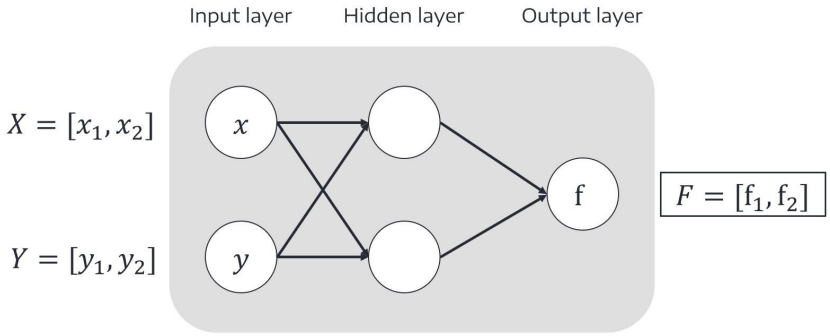
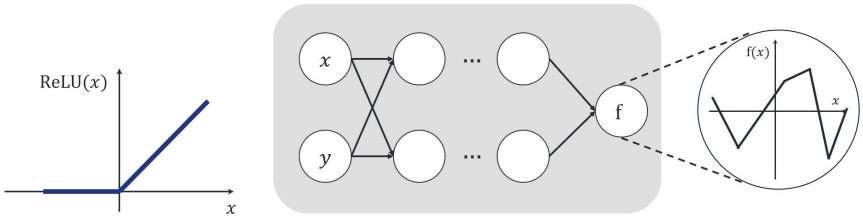


FIGURE 2.1 – Illustration de la vérification formelle d'un réseau de neurones avec X et Y représentant les plages d'entrées, et F la plage de sortie correspondante.

Dans la vérification formelle, un réseau de neurones est considéré comme *Safe* si la plage de sorties F respecte la propriété P. À l'inverse, le réseau est considéré comme *Unsafe* si une partie de la sortie viole P. Une propriété de sécurité courante peut être une condition telle que $f > 0$, indiquant que toutes les sorties doivent être supérieures à zéro. Cependant, l'analyse de F est complexe et exigeante en termes de ressources de calcul, en particulier lorsque des composants non linéaires comme la fonction d'activation ReLU (*Rectified Linear Unit*) sont impliqués.

La fonction ReLU, définie par $\text{ReLU}(x) = \max(x, 0)$, se compose de deux segments linéaires (cf. Figure 2.2a). Comme les réseaux de neurones contiennent souvent de nombreuses unités ReLU, la sortie devient une combinaison de ces segments linéaires, augmentant de manière exponentielle avec le nombre de fonctions ReLU. Cela rend le suivi de toutes les combinaisons linéaires possibles un problème NP-difficile (cf. Figure 2.2b).

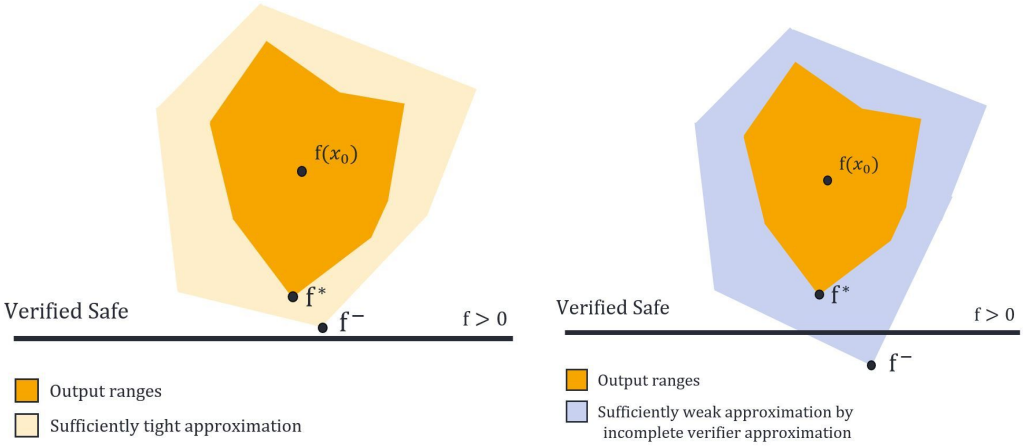


(a) Fonction ReLU. (b) Illustration des non-linéarités sur la sortie du réseau.

FIGURE 2.2 – Effet des fonctions non-linéaires comme ReLU sur la sortie des réseaux de neurones.

Pour résoudre ce problème, les outils de vérification formelle utilisent deux types de vérificateurs :

- **Les vérificateurs complets** visent à prouver ou réfuter la propriété de sécurité avec certitude, à condition d'avoir suffisamment de temps. Ces vérificateurs explorent toutes les combinaisons linéaires possibles, mais en raison de la nature NP-difficile du problème, ils peuvent devenir très coûteux en termes de calcul, notamment pour les réseaux de grande taille.
- **Les vérificateurs incomplets**, quant à eux, utilisent des approximations pour améliorer l'efficacité. Ces vérificateurs sont généralement plus rapides, mais peuvent générer des faux positifs, c'est-à-dire qu'ils peuvent signaler à tort qu'un réseau est *Unsafe* alors qu'il est en réalité *Safe*. Malgré cela, les vérificateurs incomplets sont souvent suffisants en pratique lorsqu'une approximation correcte est acceptable.



(a) Méthodes complètes. (b) Méthodes incomplètes.

FIGURE 2.3 – Illustration des méthodes de vérification formelles.

Dans le cadre de la vérification, la "solidité" (*soundness*) garantit que la borne inférieure f^- de la fonction reste toujours inférieure ou égale à la valeur limite réelle f^*. Si un vérificateur incomplet prouve que la borne inférieure de la sortie du réseau satisfait la propriété de sécurité P, nous pouvons être confiants qu'il n'y a pas de violations.

2.2.1 Solveurs SMT

Les premières méthodes développées de vérification formelle se concentrent sur les réseaux de neurones simples, tels que les MLP (*Multi-Layer Perceptron*) ou CNN (*Convolutional Neural Network*) à base de ReLU, et font appel à des **solveurs SMT (*Satisfiability Modulo Theories*)**. Ces derniers sont utilisés pour résoudre des problèmes de satisfiabilité impliquant des contraintes logiques ainsi que des théories mathématiques spécifiques. Ils consistent à déterminer s'il existe une affectation des variables qui rend une formule logique vraie. Plus formellement, un solveur SMT résout des formules de la forme :

$$\phi = \exists x_1, ..., x_n \text{ tel que } T(x_1, ..., x_n) \tag{2.1}$$

où ϕ est une formule booléenne et T est une théorie, par exemple l'arithmétique linéaire ou non linéaire, ou la théorie des ensembles. Par exemple, dans le cadre des attaques adversariales pour un réseau de neurones f, le

problème peut être formulé de la manière suivante :

$$\phi = \exists \delta \text{ avec } \| \delta \|_p \leq \epsilon \text{ tel que } f(x+\delta) \neq y \tag{2.2}$$

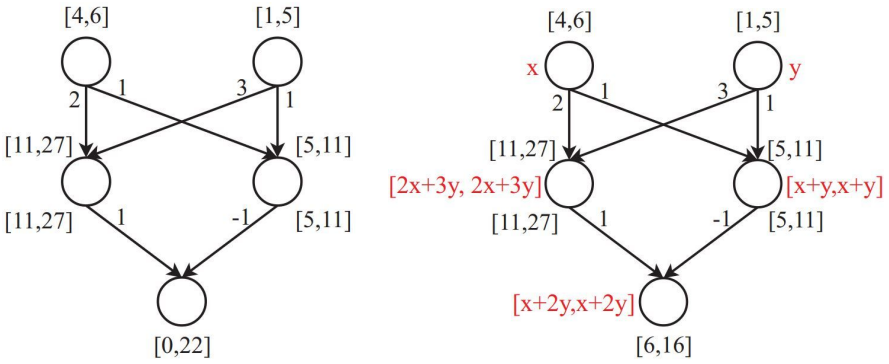
L'objectif du SMT sera donc de parcourir toutes les valeurs possibles de δ afin de trouver une solution qui satisfait la propriété, ou de conclure que la propriété est "non satisfaite" si aucune valeur de δ ne la viole. Le fonctionnement d'un solveur SMT repose sur la combinaison de deux composants : un solveur SAT (*Satisfiability solver*) et des solveurs de théories. Le solveur SAT vérifie la satisfiabilité d'une formule booléenne ϕ, exprimée en termes de variables propositionnelles. Ensuite, le solveur de théories vérifie la validité des contraintes imposées par la théorie T sur les variables interprétées (par exemple, entiers, réels, etc.). Quelques exemples d'outils solveurs SMT incluent Z3 (Microsoft), CVC4/CVC5 (Stanford/NYU), MathSAT (FBK Trento), Yices (SRI International), et Alt-Ergo (Université Paris-Saclay). Ces outils sont largement utilisés dans la vérification formelle de systèmes, l'analyse de programmes, et la synthèse de logiciels, car ils permettent de traiter des formules complexes impliquant des interactions entre logique booléenne et théories mathématiques.

Les solveurs SMT ont été adaptés pour vérifier les réseaux neuronaux en encodant le problème de vérification tel que leur robustesse face aux perturbation sous forme de contraintes logiques pouvant être résolues algorithmiquement : [Katz et al., 2017] ont introduit Reluplex, une extension de l'algorithme du simplexe capable de gérer la fonction d'activation par morceaux linéaires ReLU, courante dans les réseaux neuronaux profonds. Reluplex a démontré la faisabilité de l'utilisation de solveurs SMT pour vérifier les propriétés de sécurité des réseaux neuronaux, en particulier celles impliquant des contraintes linéaires et des activations ReLU. Sur le même principe, [Huang et al., 2017] applique des solveurs SMT à des données d'entrée de type image, avec des dimensions pouvant atteindre $3 \times 224 \times 224$ (comme celles d'ImageNet [Deng et al., 2009]), et à des classifieurs de grande taille tels que VGG16, un réseau neuronal convolutif comptant $138,357,544$ paramètres. L'objectif est de vérifier la robustesse du modèle face à de petites perturbations. Dans les cas où le solveur trouve un contre-exemple, celui-ci correspond à une attaque adversariale. Cependant, pour des modèles et des images de cette ampleur, la recherche peut devenir extrêmement longue et parfois même ne pas aboutir (*timeout* généralement entre 20 et 30 minutes), en raison de la complexité de l'espace de recherche et du modèle. Pour surmonter ces limitations, des méthodes telles que [Katz et al., 2019, Wu et al., 2024b, Elboher et al., 2020] ont introduit un cadre basé sur l'abstraction, qui utilise des sur-approximations pour simplifier les réseaux neuronaux. Ces approches transforment le réseau original en un réseau abstrait plus petit, en conservant les propriétés essentielles. Si une propriété est vérifiée pour ce réseau abstrait, elle l'est également pour le réseau d'origine. Cela permet de réduire considérablement la complexité calculatoire du processus de vérification par SMT, rendant l'analyse plus scalable pour des réseaux et des données de plus grande taille.

En conclusion, bien que les solveurs SMT aient été parmi les premières méthodes utilisées pour résoudre les problèmes de vérification formelle des réseaux neuronaux, leur applicabilité diminue à mesure que la complexité des modèles et des données augmente. Les réseaux avec davantage de couches, des fonctions d'activation non linéaires complexes et des données d'entrée à haute dimension élargissent l'espace de recherche, rendant ces méthodes impraticables sans recours à des techniques d'approximation ou de simplification.

2.2.2 Arithmétique d'Intervalles et Propagation Symbolique

Les méthodes d'**Arithmétique d'Intervalles (*Interval Arithmetic*)** et de **Propagation Symbolique (*Symbolic Propagation*)** visent à calculer des bornes sur les sorties des réseaux neuronaux étant donné des entrées bornées, fournissant des garanties sur le comportement du réseau (cf. Figure 2.4). Par exemple, [Wang et al., 2018b] ont introduit ReluVal, qui utilise l'arithmétique d'intervalles et la propagation symbolique pour analyser formellement la sécurité des réseaux neuronaux. ReluVal emploie un raffinement itératif en bissectant les intervalles d'entrée, les divisant uniformément en sous-intervalles pour réduire la surestimation. Cette méthode est parallélisable et a démontré des performances supérieures par rapport à des outils antérieurs comme Reluplex (SMT). Dans le même esprit, [Wang et al., 2018a] ont développé Neurify, qui combine l'analyse symbolique des intervalles avec des techniques de relaxation linéaire. Neurify introduit une nouvelle méthode appelée "relaxation linéaire symbolique" et utilise un raffinement des contraintes dirigé pour identifier les nœuds intermédiaires surestimés, en scindant sur des points critiques (par exemple, zéro) pour améliorer l'efficacité de la vérification. Cette approche équilibre efficacement la précision des bornes avec la tractabilité calculatoire.



(a) Propagation d'intervalle.

(b) Propagation d'intervalle symbolique.

FIGURE 2.4 – Exemples montrant (a) une extension d'intervalle où l'intervalle de sortie est très large, car elle ignore l'interdépendance des variables d'entrée, (b) l'utilisation de l'analyse d'intervalle symbolique pour suivre certaines des dépendances (source [Wang et al., 2018b]).

Similaires aux méthodes par solveurs SMT, ces méthodes se distinguent par une rapidité légèrement supérieure et une capacité accrue à identifier des contre-exemples. Cependant, elles sont davantage sujettes aux faux positifs. Cela signifie qu'une donnée peut être incorrectement considérée comme ne satisfaisant pas la propriété alors qu'elle la vérifie effectivement, ce qui réduit la fiabilité des résultats obtenus.

2.2.3 Programmation Linéaire

D'autres approches de vérification proposent d'utiliser une méthode d'optimisation appelée **Programmation Linéaire (*Linear Programming*, LP)**, qui est une technique d'optimisation mathématique ayant pour objectif de maximiser ou minimiser une fonction objective linéaire, sous réserve d'un ensemble de contraintes linéaires d'égalité et d'inégalité. Appliquées aux réseaux de neurones, ces méthodes permettent d'estimer l'intervalle de sortie du réseau à partir d'un intervalle d'entrée donné, en calculant des bornes inférieure et supérieure sur la sortie du réseau. Les variables dans un problème de LP sont continues et peuvent prendre n'importe quelle valeur dans la région définie par les contraintes. La forme générale d'un problème de LP est la suivante :

$$\begin{aligned} \text{Minimiser ou Maximiser} \quad & c^T x \\ \text{sous les contraintes} \quad & Ax \leq b \\ & x \geq 0 \end{aligned}$$

où x est un vecteur de variables continues, c est un vecteur de coefficients pour la fonction objective, A et b sont respectivement une matrice et un vecteur de contraintes. ***Integer Programming* (IP)** est similaire à la LP mais exige que certaines ou toutes les variables de décision prennent des valeurs entières : $x \in \mathbb{Z}^n$. ***Mixed Integer Programming* (MIP)** combine les éléments de LP et IP, où certaines variables sont contraintes à être entières tandis que d'autres peuvent être continues. Cela permet de modéliser des situations plus complexes où des décisions discrètes (par exemple, oui/non, activé/désactivé) sont combinées avec des variables continues. La forme générale du problème d'optimisation est la suivante :

$$\begin{aligned} \text{Minimiser ou Maximiser} \quad & c^T x + d^T y \\ \text{sous les contraintes} \quad & Ax \leq b \\ & x \in \mathbb{Z}^n \\ & y \geq 0 \end{aligned}$$

***Mixed Integer Linear Programming* (MILP)** est essentiellement un autre terme pour MIP lorsque toutes les relations (fonction objectif et contraintes) sont linéaires. En pratique, MIP et MILP sont souvent utilisés de manière interchangeable, surtout lorsque le contexte précise que les relations sont linéaires.

Par exemple, [Ehlers, 2017] introduit PLANET, une méthode qui formalise la vérification des réseaux neuronaux en utilisant des approximations linéaires des fonctions ReLU et MaxPool. En formulant le problème sous forme de satisfaisabilité modulaire (SMT) ou de programmation linéaire en nombres entiers (ILP), PLANET

exploite des solveurs existants pour valider des propriétés critiques des réseaux neuronaux. Cette approche permet de vérifier des réseaux contenant jusqu'à $1,300$ neurones sur des images MNIST ($1 \times 28 \times 28$). Cependant, bien qu'elle améliore la scalabilité par rapport aux méthodes précédentes, elle reste limitée et sujette à des *timeouts* pour des propriétés de vérification complexes. De manière similaire, SHERLOCK [Dutta et al., 2018] combine une recherche globale basée sur la programmation linéaire en nombres entiers mixtes (MILP) avec une recherche locale utilisant des algorithmes de branchement et de coupe. Cette combinaison explore efficacement l'espace d'entrée, permettant la vérification des propriétés sur des réseaux atteignant jusqu'à $1,500$ neurones avec MNIST, et surpassant Reluplex (SMT) en termes de performances. Des travaux ultérieurs, comme ceux présentés dans [Tjeng et al., 2019, Anderson et al., 2020], ont affiné les formulations MILP, augmentant leur extensibilité et leur efficacité. Ces méthodes ont été appliquées avec succès à des réseaux beaucoup plus grands, comprenant jusqu'à $100,000$ neurones, tout en réduisant les temps de calcul par un facteur de deux à trois par rapport à Reluplex, ouvrant ainsi la voie à des analyses sur des modèles et données plus complexes.

Par ailleurs, certaines méthodes, telles que [Dvijotham et al., 2018, Bunel et al., 2020a, Palma et al., 2021], améliorent la formulation MILP via une décomposition ou relaxation lagrangienne. Les problèmes d'optimisation MILP peuvent souvent être reformulés pour inclure des contraintes complexes qui, dans le cadre de la relaxation lagrangienne, sont intégrées dans la fonction objective sous forme de termes de pénalité proportionnels au degré de violation des contraintes dualisées. Cette reformulation permet de simplifier le problème initial en un problème lagrangien, souvent plus facile à résoudre. En effet, le problème lagrangien peut être décomposé en sous-problèmes indépendants, et fournit une borne supérieure pour un problème de maximisation. Pour illustrer, si l'on reprend une formulation MIP précédente, on peut introduire un terme lagrangien en relaxant les contraintes $Ax \leq b$. La formulation du problème lagrangien devient alors :

$$\begin{aligned} \text{Maximiser} \quad & \inf_{x \in \mathbb{Z}^n, y \geq 0} c^T x + d^T y + \lambda^T (b - Ax) \\ \text{sous les contraintes} \quad & \lambda \geq 0 \end{aligned}$$

où λ est le vecteur des multiplicateurs de Lagrange. Cette approche consiste donc à formuler la vérification comme un problème d'optimisation, visant à identifier la plus grande violation possible d'une spécification donnée. Une relaxation lagrangienne de ce problème fournit une borne supérieure sur le pire cas de violation.

Une variante de la programmation linéaire (LP), appelée Programmation Semi-Définie (*Semi-Definite Programming*, SDP), offre également des alternatives intéressantes. C'est une méthode d'optimisation convexe qui consiste à minimiser (ou maximiser) une fonction linéaire sous des contraintes définies par des matrices semi-définies positives. Le formalisme mathématique est le suivant :

$$\begin{aligned} & \min_X \langle C, X \rangle \\ \text{sous les contraintes} \quad & \langle A_i, X \rangle = b_i, \quad i = 1, ..., m \\ & X \succeq 0 \end{aligned}$$

où X est une matrice semi-définie positive, C et A_i sont des matrices symétriques, $\langle ., . \rangle$ représente le produit scalaire matriciel, et b_i sont des constantes. La contrainte $X \succeq 0$ indique que X doit être semi-définie positive (c'est-à-dire que toutes ses valeurs propres sont non négatives). Dans le cadre de la vérification de réseaux de neurones, la SDP est utilisée pour encadrer et certifier les propriétés d'un réseau, telles que la robustesse face à des attaques adversariales ou des perturbations d'entrée. La SDP permet de relaxer les non-linéarités (tel que ReLU) en les approximant par des contraintes convexes. La programmation semi-définie offre des relaxations plus serrées que la programmation linéaire en capturant les interactions quadratiques entre les variables (par exemple, $l_j \geq x_j \geq u_j$ si et seulement si $(x_j - l_j)(x_j - u_j) \leq 0$, ce qui donne la contrainte quadratique $x_j^2 \leq (l_j + u_j)x_j - l_j u_j$), ce qui permet une meilleure précision pour les réseaux de petite ou moyenne taille.

Par exemple, les articles [Raghunathan et al., 2018, Dathathri et al., 2020, Ma and Sojoudi, 2021] proposent des approches pour certifier la robustesse des réseaux neuronaux face aux exemples adversariaux. Ces méthodes fournissent des bornes supérieures traitables pour des réseaux de taille modérée. En capturant des interactions complexes que les relaxations basées sur la programmation linéaire ne peuvent pas représenter, elles permettent d'obtenir des certifications plus précises. La clé de cette approche réside dans la reformulation du problème d'optimisation. Les contraintes non linéaires, telles que celles introduites par les fonctions ReLU, sont remplacées par des contraintes linéaires et quadratiques, transformant ainsi le problème initial en un Programme Quadratiquement Contraint Quadratique (*Quadratically Constrained Quadratic Program*, QCQP). Ce QCQP peut ensuite

être détendu en un programme semidéfini (SDP). Ces techniques permettent parfois d'adopter des formulations économes en mémoire, ce qui étend leur applicabilité à des réseaux plus grands. Par exemple, elles permettent la certification de réseaux tels que des MLP et CNN comprenant environ $1,000$ neurones sur des images CIFAR-10 ($3 \times 32 \times 32$), ou encore d'un VAE sur MNIST ($1 \times 28 \times 28$) avec un décodeur convolutionnel contenant jusqu'à $10,000$ neurones au total. Cependant, pour les réseaux de grande taille, les bornes obtenues tendent à être moins précises. Donc, comparée à la programmation linéaire (LP), la précision est meilleure, mais le coût de calcul est plus élevé, ce qui limite la taille des réseaux pouvant être traités.

La méthode LP et ses variantes améliorent les approches précédentes en optimisant le processus de recherche, réduisant ainsi les temps de calcul et augmentant l'efficacité de la vérification. Toutefois, malgré ces avancées, les temps de calcul restent long pour les réseaux neuronaux de très grande taille ou les architectures modernes complexes. Le choix entre les différentes variantes repose sur un compromis entre précision et scalabilité, en fonction des besoins spécifiques de l'application et des contraintes du système à vérifier.

2.2.4 Branch and Bound (BaB)

Dans la suite, la méthode du ***Branch and Bound*** **(BaB)** [Bunel et al., 2018] s'est imposée comme une approche puissante pour traiter les problèmes d'exactitude et de robustesse. Cette méthode permet d'explorer systématiquement l'espace des entrées ou des configurations du réseau afin de vérifier des propriétés spécifiques, tout en gérant efficacement la complexité combinatoire inhérente à ces réseaux. La méthodologie du *Branch and Bound* repose sur la décomposition récursive du problème initial en sous-problèmes plus petits (branchements) et sur l'utilisation de bornes inférieures et supérieures (bornes) pour élaguer les branches qui ne conduiront pas à une solution optimale ou satisfaisante. Mathématiquement, cela implique la construction d'un arbre de recherche où chaque nœud représente un sous-espace de l'ensemble des entrées possibles, avec des estimations des valeurs minimales et maximales de la fonction objectif ou des contraintes à vérifier :

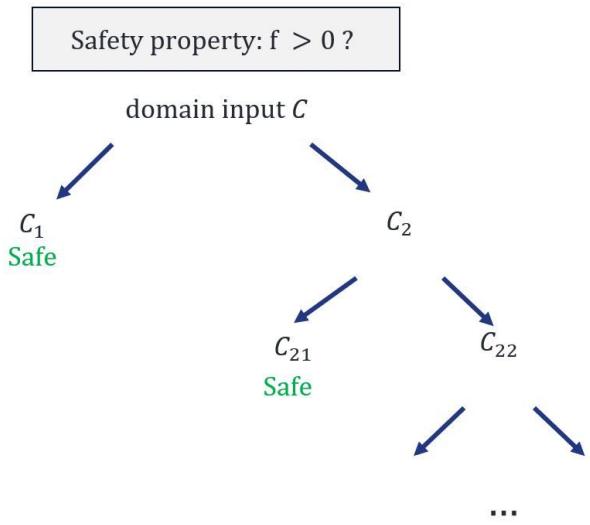


FIGURE 2.5 – Processus du Branch and Bound.

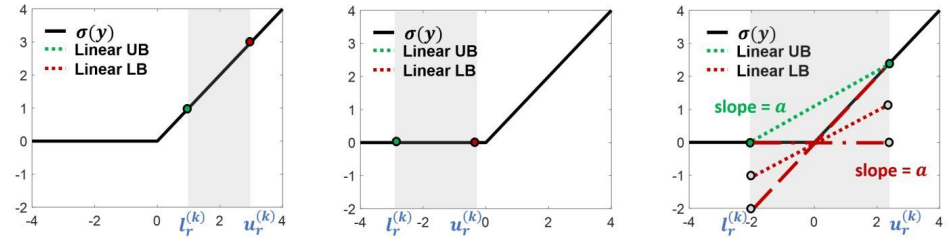
Par exemple, [Henriksen and Lomuscio, 2020] ont développé VeriNet, une méthode qui combine la propagation symbolique d'intervalles, la relaxation linéaire et BaB pour vérifier efficacement les réseaux neuronaux. VeriNet optimise la stratégie de raffinement en priorisant les branches présentant le plus fort potentiel d'amélioration des bornes de sortie et intègre une recherche adversariale pour identifier rapidement des contre-exemples. Cette recherche repose sur une approche locale basée sur les gradients, permettant de localiser des contre-exemples valides en divisant le nœud ayant le plus d'impact sur les bornes de sortie. VeriNet est jusqu'à 10 fois plus rapide que les approches précédentes et est capable de vérifier des réseaux comprenant jusqu'à $50,000$ neurones sur les images de CiFAR-10 ($3 \times 32 \times 32$). De même, [Bunel et al., 2020b] ont encore amélioré les méthodes BaB en les intégrant avec la MIP et en utilisant des mécanismes de notation pour prioriser les nœuds ReLU lors du branchement. Cette priorisation permet de concentrer les ressources calculatoires sur les parties du réseau les plus critiques, améliorant ainsi l'efficacité globale. De nombreuses autres approches cherchent à optimiser le

mécanisme de branchement afin de réduire le nombre de branches nécessaires et d'accélérer la vérification. Par exemple, [Lu and Kumar, 2019] proposent une méthode utilisant des graphes neuronaux (GNN) pour imiter les heuristiques de branchement fortes dans BaB, [Botoeva et al., 2020] exploitent les relations de dépendance entre les nœuds ReLU pour affiner les décisions de branchement, ou encore [Henriksen and Lomuscio, 2021] introduit des fonctions de score de scission qui estiment l'impact d'une scission sur un nœud, permettant des choix de branchement plus éclairés et une vérification plus rapide. Enfin, [Shi et al., 2024] ont étendu les méthodes BaB à des non-linéarités générales dans les graphes computationnels avec GenBaB. Leur approche adapte BaB pour gérer divers types de fonctions d'activation, notamment sigmoïde, tanh, sinus, GeLU, LSTM. Ils introduisent une heuristique innovante de branchement appelée Propagation de Bornes avec Raccourcis (*Bound Propagation with Shortcuts*, BBPS), qui permet de gérer efficacement la complexité liée à ces non-linéarités générales.

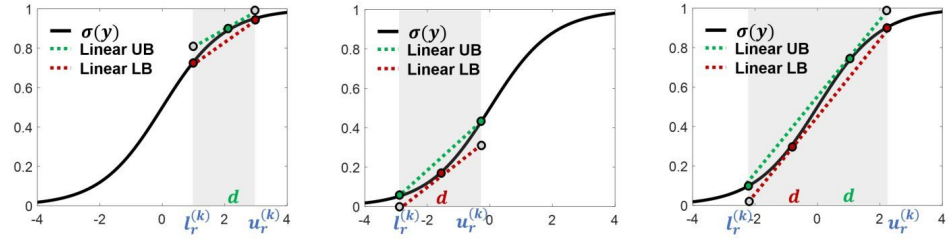
La méthode BaB apporte donc des améliorations significatives à la vérification des réseaux neuronaux. Elle permet non seulement une exploration plus rapide et efficace de l'espace des solutions, mais aussi une accélération de la recherche de contre-exemples. De plus, elle rend certaines méthodes initialement incomplètes complètement vérifiables, élargissant ainsi leur applicabilité.

2.2.5 Relaxation Linéaire ou Convexe

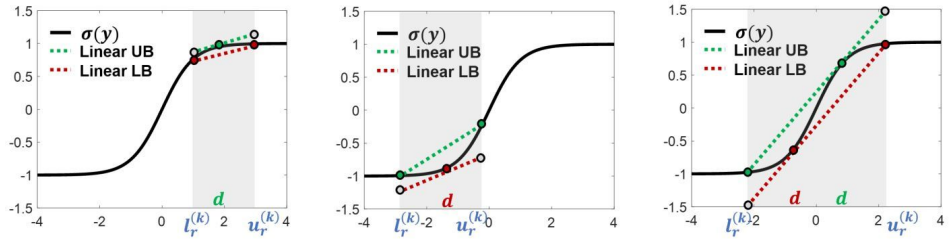
La **Relaxation Linéaire ou Convexe (*Convex or Linear Relaxation*)** consiste à approximer les composants non linéaires d'un réseau neuronal (comme les fonctions d'activation ReLU, sigmoïde, etc.) par des contraintes linéaires ou convexes. L'idée est de remplacer les parties non linéaires du réseau par des sur-approximations linéaires ou convexes qui englobent tous les comportements possibles de ces composants sur un certain domaine d'entrée. Introduite par [Zhang et al., 2018a], CROWN est une méthode efficace pour la certification de la robustesse des réseaux neuronaux qui utilise des sélections adaptatives sur les approximations linéaires. Elle introduit des bornes linéaires adaptatives (CROWN-Ada) et des bornes quadratiques (CROWN-Quad) pour rechercher des bornes inférieures certifiées plus serrées en sélectionnant de manière adaptative des substituts appropriés pour chaque activation de neurone (cf. Figure 2.6).



(a) ReLU.



(b) Sigmoid.



(c) Tanh.

FIGURE 2.6 – Relaxations adaptative proposées par CROWN [Zhang et al., 2018a].

De multiples autres méthodes adoptent cette approche, tels que [Weng et al., 2018a, Singh et al., 2019a, Boopathy et al., 2019, Lyu et al., 2020, Ferrari et al., 2021, Zhang et al., 2022b] conduisant à des bornes plus serrées et à une vérification plus efficace. Les avancées ont continué avec des méthodes comme α, β-CROWN [Xu et al., 2020, Wang et al., 2021c], qui intègrent BaB avec une analyse de perturbation basée sur la relaxation linéaire (*Linear Relaxation based Perturbation Analysis*, LiRPA). En optimisant LiRPA, ces méthodes améliorent les performances de BaB et permettent des calculs massivement parallèles sur les GPU, les rendant adaptées à la vérification de réseaux neuronaux à grande échelle.

Les méthodes de relaxation linéaire ou convexe offrent une capacité accrue à gérer un plus grand nombre de fonctions non linéaires, ce qui les rend particulièrement adaptées à des réseaux neuronaux complexes. De plus, l'intégration de la méthode Branch-and-Bound (BaB) permet de transformer ces approches en solutions de vérification complètes et efficaces. Cela place des frameworks tels que α, β-CROWN au rang des méthodes state-of-the-art. Cependant, la complétude n'est pas garantie pour certaines fonctions d'activation non linéaires autres que ReLU, et bien que certains de ces vérificateurs, notamment ceux accélérés par GPU, surpassent les approches précédentes en termes de rapidité et de scalabilité, les vérificateurs basés sur la méthode BaB restent contraints à des modèles relativement petits, comme ceux contenant quelques couches linéaires et convolutives.

2.2.6 Interprétation Abstraite

Similaire aux relaxations linéaires et convexes, l'**Interprétation Abstraite (*Abstract Interpretation*)** est une technique issue de l'analyse statique des programmes, utilisée pour raisonner sur tous les comportements possibles d'un programme sans l'exécuter. Appliquée aux réseaux neuronaux, elle consiste à propager des "abstractions" des valeurs à travers le réseau, plutôt que des valeurs précises, afin de sur-approximer l'ensemble des sorties possibles pour un ensemble donné d'entrées (cf. Figure 2.7).

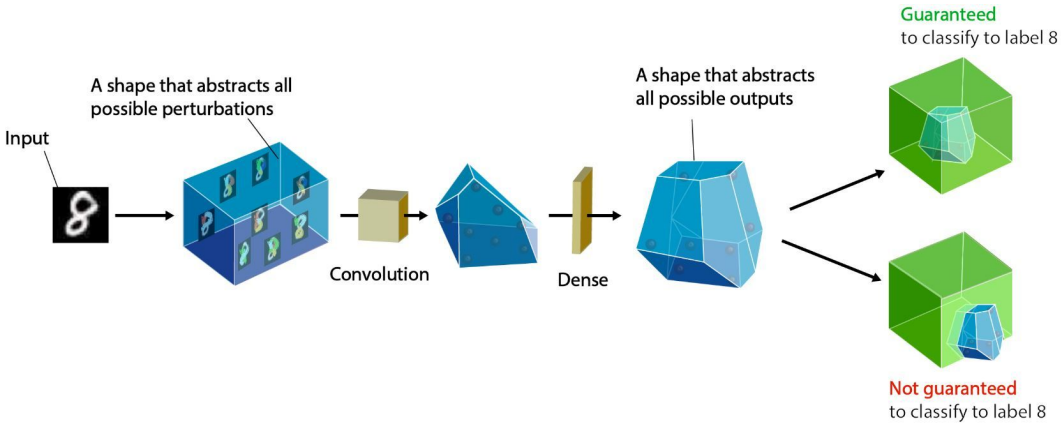


FIGURE 2.7 – Illustration de la vérification par Interprétation Abstraite (source [eta,]).

L'interprétation abstraite est une méthode classique permettant une sur-approximation solide et précise d'un ensemble potentiellement non borné ou infini de comportements d'un programme. L'idée clé de ce cadre repose sur la définition de ce que l'on appelle des transformateurs abstraits pour les instructions utilisées par le programme (par exemple, l'arithmétique affine, les fonctions ReLU, etc.). Ces transformateurs approximent le comportement de chaque instruction en définissant son effet sur un domaine abstrait. Un domaine abstrait est simplement un ensemble d'éléments abstraits (des approximations). Un défi majeur lors de la définition des transformateurs abstraits réside dans le compromis entre la scalabilité (rapidité du calcul de l'approximation par le transformateur) et la précision (perte de précision engendrée). Une fois les transformateurs définis, l'analyse via l'interprétation abstraite procède en les exécutant sur le programme en question (par exemple, un réseau de neurones) et en calculant une approximation finale (un point fixe). La propriété pertinente peut ensuite être vérifiée sur cette approximation finale : si la propriété peut être prouvée, elle est alors garantie pour toute entrée concrète du programme, sinon, il est possible soit que la propriété soit vraie, mais que l'abstraction soit trop grossière pour la prouver (faux positif), soit qu'elle ne soit effectivement pas vérifiée.

Par exemple, l'abstraction Zonotope est un ensemble convexe défini par :

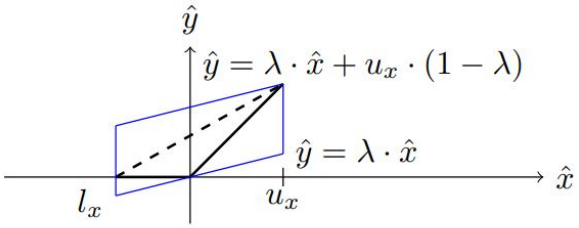
$$Z(c, G) = \left\{ c + \sum_{p}^{i=1} \alpha_i g_i \ \text{ avec } \ \alpha_i \in [-1, 1] \right\} \tag{2.3}$$

où $c \in \mathbb{R}^n$ est le centre du zonotope, $G = [g_1, ...g_p] \in \mathbb{R}^{n \times p}$ est une matrice dont les colonnes g_i sont les générateurs du zonotope, et p est le nombre de générateurs. Le zonotope $Z(c, G)$ représente l'ensemble des points accessibles en combinant linéairement les générateurs g_i, avec des coefficients α_i contraints dans l'intervalle $[-1, 1]$. L'objectif est de propager des ensembles abstraits (zonotopes) à travers les couches d'un réseau neuronal, tout en garantissant une approximation conservatrice. Pour une couche affine $h(x) = Wx + b$, où $W \in \mathbb{R}^{m \times n}$ est la matrice de poids et $b \in \mathbb{R}^m$ est le biais. Si l'entrée x est représentée par un zonotope $Z(c_x, G_x)$, la sortie $h(x)$ est donnée par un nouveau zonotope $Z(c_h, G_h)$, où :

$$c_h = W c_x + b, \quad G_h = W G_x \tag{2.4}$$

Cela correspond à l'application linéaire de la transformation affine sur le centre c_x et les générateurs G_x. Pour la fonction d'activation ReLU ($\text{ReLU}(x) = \max(0, x)$), la propagation est plus complexe, car ReLU n'est pas linéaire. On doit partitionner le domaine pour considérer les differentes possibilités lié à ReLU. On considère chaque dimension x_i de l'entrée x. Si $Z(c_x, G_x)$ est l'entrée :

- si $c_x^i + \| G_x^i \|_1 < 0$, alors $\text{ReLU}(x_i) = 0$ partout.
- si $c_x^i - \| G_x^i \|_1 > 0$, alors $\text{ReLU}(x_i) = x_i$.
- si $c_x^i - \| G_x^i \|_1 < 0 < c_x^i + \| G_x^i \|_1$, alors x_i traverse à la fois des régions activées et désactivées, la sortie ne peut donc pas être capturée exactement par l'abstraction zonotope et des approximations sont donc nécessaires (cf. Figure 2.8)



(a) $0 \leq \lambda \leq \frac{u_x}{u_x - l_x}$

FIGURE 2.8 – Exemple d'approximation zonotopique pour la fonction ReLU paramétrées par la pente λ, avec l_x et u_x les bornes du zonotope d'entrée $Z(c_x, G_x)$ (source [Singh et al., 2018]).

L'interprétation abstraite offre un cadre compact et efficace pour analyser le comportement des réseaux neuronaux en sur-approximant leurs calculs à l'aide de domaines abstraits. Par exemple, AI^2 [Gehr et al., 2018], applique cette technique pour certifier la sécurité et la robustesse des réseaux, gérant des fonctions d'activation comme ReLU, sigmoïde et tanh sur des CNN de petite taille (jusqu'à 53,000 neurones). Cette méthode se révèle bien plus rapide et performante que Reluplex (SMT). De même, DeepZ [Singh et al., 2018] utilise des transformateurs de zonotope abstrait spécialisés pour traiter ces fonctions d'activation et RefineZono [Singh et al., 2019b] affine cette approche en intégrant des techniques MILP et LP pour améliorer la précision des résultats. DeepPoly [Singh et al., 2019c], introduit par les mêmes auteurs, combine des polyèdres en virgule flottante avec des intervalles, offrant une certification plus précise des réseaux neuronaux. BBPoly [Zhong et al., 2021] améliore encore DeepPoly en introduisant des résumés de blocs et des heuristiques de rétro-substitution bornée, permettant une vérification à plus grande échelle. Enfin, GPUPoly [Müller et al., 2021] pousse ces avancées en tirant parti des GPU pour des implémentations massivement parallélisées, prouvant la robustesse d'un réseau de 1 million de neurones (ResNet34) en seulement 34 ms sur CIFAR-10 ($3 \times 32 \times 32$).

En conclusion, l'interprétation abstraite est particulièrement adaptée aux couches linéaires ou presque linéaires, avec des approximations conservatrices garantissant que l'ensemble calculé contient toujours les vraies sorties possibles. Cependant, lors de la relaxation, notamment pour les ReLU, le zonotope peut devenir trop large, réduisant la précision. De plus, le nombre de générateurs p peut croître exponentiellement avec la profondeur du réseau, rendant l'approche difficilement extensible aux architectures très profondes.

2.2.7 Constante de Lipschitz

Une autre méthode pour garantir la robustesse consiste à estimer la **Constante de Lipschitz** d'un réseau neuronal qui fournit des informations sur sa sensibilité aux perturbations d'entrée. D'après [Weng et al., 2018c], pour une fonction $f(x)$ continûment différentiable définie sur un ensemble convexe, borné et fermé $S \sqsubseteq \mathbb{R}^d$, la fonction est Lipschitzienne avec une constante L_q si l'inégalité suivante est vérifiée pour tout $x, y \in S$:

$$|f(x) - f(y)| \leq L_q \parallel x - y \parallel_p \tag{2.5}$$

La constante de Lipschitz L_q est liée à la norme maximale du gradient de la fonction $f(x)$, où $L_q = \max\{\parallel \nabla f(x) \parallel_p : x \in S\}$, et la relation entre les normes p et q est donnée par $\frac{1}{p} - \frac{1}{q} = 1$. Dans le cadre de la classification, la distorsion adversariale minimale δ est bornée par β_L, définie comme suit :

$$\parallel \delta \parallel_p \leq \beta_L = \min_{j \neq c} \frac{f_c(x_0) - f_j(x_0)}{L_j^q} \tag{2.6}$$

où $f_c(x_0)$ est le score de la classe prédite c, $f_j(x_0)$ est le score d'une autre classe j, et L_j^q est la constante de Lipschitz pour la fonction $f_c(x) - f_j(x)$ dans la norme p. Cette borne inférieure β_L garantit que pour toute perturbation δ, si $\parallel \delta \parallel_p \leq \beta_L$, aucun exemple adversarial ne peut être trouvé. En d'autres termes, la fonction $g(x) = f_c(x) - f_j(x)$ autour de x_0 peut être visualisée comme étant bornée par un "cône" formé par deux lignes de pentes $\pm L_q$ (cf. Figure 2.9).

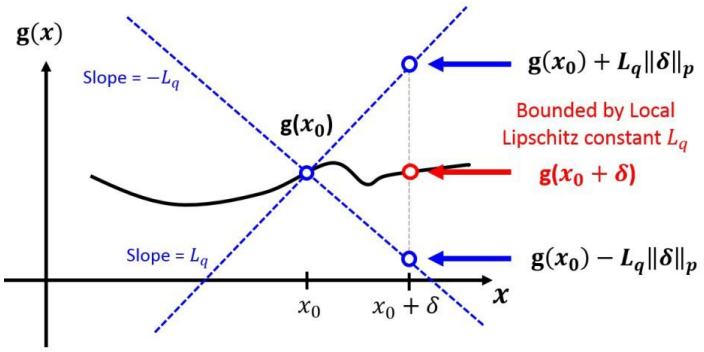


FIGURE 2.9 – Interpretation de la constante de Lipschitz (source [Weng et al., 2018c]).

Cependant, il est NP-difficile de calculer cette constante exactement pour des fonctions non linéaires complexes comme les réseaux neuronaux. Ainsi, des efforts récents se concentrent sur l'estimation de bornes supérieures serrées de cette constante pour des architectures de réseaux neuronaux. Ces estimations sont obtenues soit via MIP (*Mixed Integer Programming*) ou SDP (*Semi-Definit Programming*). Cependant, le coût calculatoire augmente considérablement avec la profondeur du réseau.

Par exemple, CLEVER [Weng et al., 2018c] estime une borne inférieure sur la perturbation adversariale minimale en s'appuyant sur la théorie des valeurs extrêmes. CLEVER est particulièrement adapté aux réseaux neuronaux convolutionnels et résiduels de grande taille, comptant plus de 100,000 neurones, et a été appliqué avec succès sur des jeux de données comme MNIST ($1 \times 28 \times 28$) et CIFAR-10 ($3 \times 32 \times 32$). Par rapport à des méthodes basées sur SMT, telles que Reluplex, CLEVER offre une évaluation significativement plus rapide. Une extension de CLEVER [Weng et al., 2018b] améliore cette méthode permettant ainsi d'obtenir un score de robustesse indépendant des attaques. Cette extension a été déployée pour évaluer des modèles de grande envergure comme DenseNet-121, entraînés sur le jeu de données ImageNet ($3 \times 256 \times 256$), démontrant sa capacité à s'adapter aux réseaux complexes et aux données haute résolution. LipSDP, proposé par [Fazlyab et al., 2019], introduit un cadre d'optimisation convexe pour calculer des bornes supérieures garanties sur la constante de Lipschitz des réseaux de neurones avec précision et efficacité. Les fonctions d'activation des réseaux sont interprétées comme des gradients de fonctions convexes satisfaisant certaines propriétés décrites par des contraintes quadratiques. Ce problème d'estimation est reformulé en un programme semi-défini (SDP), ce qui permet d'obtenir des bornes garanties tout en conservant une bonne scalabilité. LipSDP est applicable à des modèles pouvant atteindre 50,000 neurones, avec un temps de calcul de 12 minutes pour établir une borne précise sur la constante de Lipschitz.

Comparée aux méthodes précédentes, l'estimation de la constante de Lipschitz offre une approche plus générale et théoriquement rigoureuse pour évaluer la robustesse globale des réseaux neuronaux, sans se limiter à une propagation locale d'incertitudes. L'approche par constante de Lipschitz garantit des bornes précises et applicables globalement à l'ensemble des entrées, en encapsulant la sensibilité du modèle. Toutefois, elle manque de scalabilité.

2.2.8 Apprentissage sous contraintes

Au-delà de la vérification des réseaux existants, on peut également adapter les réseaux neuronaux via leur structure ou des contraintes durant leur entrainement pour qu'ils vérifient intrinsèquement une certaine propriété (**Apprentissage sous contraintes**, *Constraint Learning*).

Par exemple, les méthodes [Mirman et al., 2018, Gowal et al., 2019, Zhang et al., 2020a, Mueller et al., 2023] exploitent la propagation des bornes et les relaxations pour renforcer la robustesse des réseaux neuronaux pendant l'entraînement. DiffAI [Mirman et al., 2018] intègre l'interprétation abstraite différentiable pour produire des réseaux prouvablement robustes en utilisant des transformateurs abstraits adaptés à la descente de gradient et CROWN-IBP [Zhang et al., 2020a] combine la propagation des bornes d'intervalles avec des relaxations linéaires, augmentant à la fois l'efficacité et la précision des vérifications. Enfin, [Mueller et al., 2023] optimisent la propagation des bornes en se concentrant sur un sous-ensemble clé de la région d'entrée adversariale, réduisant les approximations tout en améliorant l'efficacité de l'entraînement certifié. Alors que les méthodes [Wong and Kolter, 2018, Xiao et al., 2019, Boopathy et al., 2021, Shi et al., 2021, Zhang et al., 2021a, De Palma et al., 2022, Pfrommer et al., 2023a] entraînent les réseaux de manière à ce qu'ils vérifient certaines

propriétés, telles que la stabilité des activations ReLU, la parcimonie des poids, ou encore le respect de la continuité Lipschitz, qui facilitent leur vérifiabilité. Ces approches intègrent des mécanismes comme des régularisations explicites, des approximations linéaires ou convexes, et des architectures conçues pour garantir des propriétés robustes et vérifiables, tout en réduisant la complexité des processus de certification. COLT [Balunovic and Vechev, 2020], propose d'inclure à la fois le vérificateur et l'adversaire dans la boucle d'entraînement. À chaque itération, le vérificateur certifie le réseau en utilisant une relaxation convexe, tandis que l'adversaire tente de trouver des entrées à l'intérieur de cette relaxation qui font échouer la vérification, assurant que le réseau est robuste tout en étant vérifiable. Comprendre la dynamique de l'entraînement est crucial pour obtenir des modèles robustes et vérifiables. Les articles [Lee et al., 2021, Jovanovi'c et al., 2021] mettent en avant l'importance de la continuité et de la sensibilité du relief de la fonction de perte, elles soulignent leur rôle clé dans la construction de modèles certifiables et robustes. Ces travaux montrent que la précision des approximations et des gradients relaxés influence directement la formation du paysage de la fonction de perte. Ils révèlent également un paradoxe : des relaxations trop serrées peuvent paradoxalement nuire à la robustesse certifiée, ce qui met en évidence la nécessité d'un équilibre entre rigueur des approximations et continuité du relief.

En conclusion, les méthodes d'apprentissage sous contraintes se distinguent par leur approche proactive, visant à intégrer des garanties directement dans le processus d'entraînement des modèles. En favorisant des propriétés comme la robustesse certifiable, elles réduisent le besoin d'une vérification coûteuse a posteriori, typique des approches SMT, MILP ou d'interprétation abstraite. Ces méthodes sont particulièrement efficaces pour gérer des architectures modernes à grande échelle et des données complexes, où les approches traditionnelles peuvent échouer en raison de leur complexité calculatoire. Les méthodes d'apprentissage sous contraintes sont donc complémentaires aux méthodes de vérification formelle précédentes. Cependant, ces méthodes peuvent entraîner une réduction des performances nominales, une augmentation de la complexité de l'entraînement, ainsi qu'une convergence parfois moins efficace.

2.2.9 Conclusion

La vérification formelle des réseaux de neurones est un processus qui consiste à garantir que ces modèles respectent certaines propriétés critiques, en particulier dans des applications où la fiabilité et la sécurité sont primordiales. Elle vise à vérifier si la sortie d'un réseau de neurones f satisfait à des contraintes définies sur ses plages d'entrée. Les approches de vérification formelle, telles que celles basées sur les solveurs SMT, la programmation mixte en nombres entiers (MILP), l'interprétation abstraite ou l'estimation de constantes de Lipschitz, permettent de certifier des propriétés telles que la robustesse aux perturbations ou la stabilité des prédictions, apportant un haut niveau de confiance dans des scénarios spécifiques. Ces méthodes se révèlent utiles dans des cas bien définis, comme notre cadre d'étude sur la chaîne compression-détection. Cependant, malgré leur pertinence et leur sophistication, ces méthodes souffrent de limitations significatives qui empêchent leur généralisation pour garantir les performances dans tous les contextes. En effet, elles sont souvent adaptées uniquement à des réseaux de petite ou moyenne taille ou à des architectures simples, les garanties fournies sont limitées à des scénarios spécifiques ou des perturbations bien caractérisées (comme des bruits gaussiens ou adversariaux). Elles échouent à couvrir des situations plus complexes ou déterministes, telles que les distorsions causées par la compression d'images. Enfin, ces méthodes sont souvent coûteuses en termes de calcul, ce qui les rend impraticables pour des modèles modernes complexes ou des données de grande dimension.

Dans notre cadre d'application, qui consiste en une chaîne de compression d'image suivie de détection d'objets, l'objectif est de garantir que l'impact des distorsions induites par la compression sur la détection reste sous un seuil de tolérance prédéfini. Autrement dit, nous cherchons à nous assurer que ces distorsions ne compromettent pas significativement les performances du modèle de détection. Les modèles de compression et de détection sont hautement complexes, et leurs comportements sont difficiles à formaliser intégralement. De plus, les images utilisées sont de haute dimension, ce qui complique davantage l'analyse et la vérification des propriétés de robustesse. Dans ce contexte, les méthodes de vérification formelle ne s'adaptent pas bien à des perturbations déterministes (c'est-à-dire issues de transformations fixes et reproductibles) comme celles causées par la compression, contrairement aux perturbations stochastiques ou adversariales généralement étudiées. Elles ne visent pas à garantir une robustesse globale sur toutes les données possibles mais se limitent à des espaces d'entrée restreints et elles peinent à traiter des modèles complexes et des données réalistes de haute dimension comme celles utilisées dans notre chaîne compression-détection.

En conclusion, bien que les méthodes de vérification formelle soient des outils puissants pour analyser des scénarios restreints et fournir des garanties rigoureuses, elles ne permettent pas de développer une méthodologie universelle pour garantir les performances dans des contextes généraux. Dans notre cas spécifique de chaîne

compression-détection, elles s'avèrent inapplicables en raison de la complexité des modèles et des données utilisées, ainsi que de la nature des distorsions étudiées.

2.3 Certification statistique

Les méthodes de certification statistique fournissent des garanties probabilistes sur les performances ou les propriétés des systèmes intégrant des modules d'intelligence artificielle (IA). Contrairement aux méthodes de vérification formelle, qui visent à prouver de manière exhaustive des propriétés déterministes, ces méthodes s'appuient sur des analyses empiriques et des hypothèses statistiques pour évaluer les performances. Elles permettent d'estimer, avec un certain niveau de confiance, la probabilité qu'un système respecte une propriété donnée ou qu'il fonctionne correctement dans un ensemble de conditions spécifiées. Ces approches sont particulièrement adaptées à des contextes où une évaluation exhaustive est impraticable en raison de la complexité ou de la taille des systèmes étudiés. Elles offrent un cadre flexible pour analyser des systèmes complexes, tout en tenant compte des incertitudes liées aux données et aux modèles utilisés. Certaines de ces méthodes se révèlent particulièrement pertinentes pour notre application industrielle *Field Monitoring*, qui vise à garantir la robustesse d'un module de détection d'objets face aux perturbations induites par un module de compression d'images. Cependant, leur portée reste limitée lorsqu'il s'agit de développer une méthodologie plus générale d'obtention de garanties de performances pour des modules d'IA.

2.3.1 Randomized Smoothing

Une méthode de certification statistique qui semble pertinente pour notre application est le ***Randomized Smoothing*** [Cohen et al., 2019]. Cette approche consiste à transformer un modèle de deep learning, tel qu'un classifieur, en une version dite "lissée" (*smoothed*), offrant des garanties probabilistes sur sa robustesse face à des perturbations. Soit un classifieur de base $f : \mathbb{R}^d \to \mathcal{Y}$, où $\mathcal{Y}$ est l'ensemble des classes possibles. La méthode construit un nouveau classifieur lissé $g : \mathbb{R}^d \to \mathcal{Y}$ tel que :

$$g(x) = \arg\max_{c \in \mathcal{Y}} P(f(x+\epsilon) = c), \tag{2.7}$$

où $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ est un bruit gaussien indépendant avec une variance σ^2. Pour certifier la robustesse de g contre des perturbations adversariales de norme $||\delta||_p$, on considère la probabilité qu'une perturbation ne modifie pas la classification majoritaire. Si $g(x)$ prédit la classe c_A pour un point x, on peut certifier la robustesse avec le rayon R si la probabilité suivante est vérifiée :

$$P(f(x+\epsilon) = c_A) > P(f(x+\epsilon) = c_B) \quad \forall c_B \neq c_A \tag{2.8}$$

On peut prouver que si cette condition est remplie, alors $g(x+\delta)$ reste invariant pour toutes les perturbations adversariales δ telles que $||\delta||_p \leq R$, où :

$$R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B})) \tag{2.9}$$

avec $\underline{p_A}$ la probabilité minimale de prédiction correcte sur la classe c_A, $\overline{p_B}$ la probabilité maximale de prédiction correcte sur la classe c_B et Φ^{-1} la fonction quantile inverse de la distribution normale standard (cf. Figure 2.10). Dans la pratique, les probabilités $P(f(x+\epsilon) = c)$ sont estimées par une méthode de Monte Carlo, où plusieurs échantillons $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$ sont générés et la proportion de prédictions pour chaque classe est calculée. Le lissage garantit que si le classifieur lissé g attribue une classe à une certaine entrée, cette classe reste robuste à des perturbations de norme $||\delta||_p \leq R$. La taille du rayon R dépend de la variance du bruit ajouté σ et des probabilités estimées des classes.

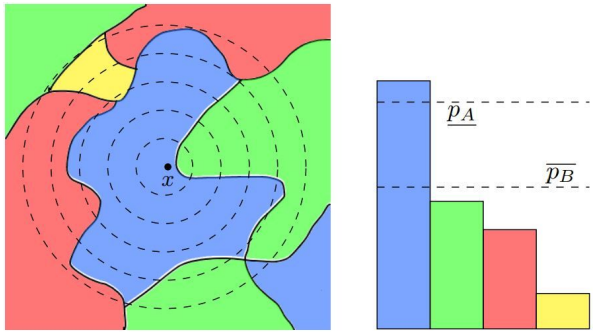
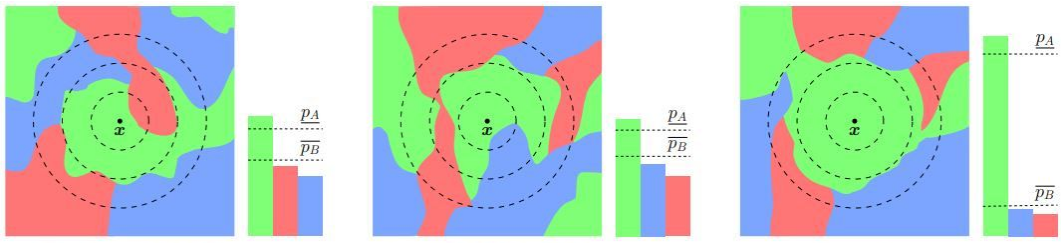


FIGURE 2.10 – Évaluation du classifieur lissé pour une entrée x. À gauche : les régions de décision du classifieur de base f sont représentées par différentes couleurs. Les lignes pointillées indiquent les niveaux de la distribution $\mathcal{N}(x, \sigma^2 I)$. À droite : la distribution $f(\mathcal{N}(x, \sigma^2 I))$, $\underline{p_A}$ est une borne inférieure de la probabilité de la classe principale, et p_B est une borne supérieure de la probabilité de chaque autre classe. Ici, $g(x)$ est "bleu" (source [Cohen et al., 2019]).

Le *randomized smoothing* a été initialement introduit par [Cohen et al., 2019] comme un mécanisme de défense certifié, où l'utilisation de bruit gaussien isotrope (c'est-à-dire sans biais directionnel dans l'ensemble des dimensions considérées) permet de créer un classifieur lissé offrant une robustesse L_2. L'une des contributions majeures de cette étude est le concept de rayon certifié, garantissant la robustesse des prédictions lors de l'inférence et ouvrant ainsi la voie à de futures avancées dans ce domaine.

Depuis sa création, la méthode de *randomized smoothing* a été enrichie par de nombreuses extensions et améliorations visant à étendre son champ d'application et à accroître son efficacité. Les travaux présentés dans les articles [Fischer et al., 2020, Alfarra et al., 2022a, Li et al., 2021, Eiras et al., 2021, Rumezhak et al., 2023, Labarbarie et al., 2022] ont permis d'élargir les types de perturbations pour lesquels des garanties de robustesse peuvent être obtenues. Ces perturbations incluent des transformations géométriques, des déformations de champs de vecteurs, des translations, des rotations, des changements d'échelle, ainsi que des perturbations isotropes et anisotropes. Cette diversité rend la méthode applicable à un éventail plus large de contextes et de défis pratiques. Parallèlement, plusieurs travaux ont cherché à optimiser les rayons de robustesse certifiés. Par exemple, [Zhai et al., 2019, Feng et al., 2020, Jeong et al., 2021, Anderson and Sojoudi, 2022] proposent des techniques d'entraînement, telles que l'entraînement adversarial, pour améliorer la scalabilité et la robustesse certifiée des modèles. D'autres, comme [Horváth et al., 2022a], utilisent des modèles d'ensemble pour réduire la variance des prédictions et augmenter les rayons certifiables (cf. Figure 2.11).



(a) Individual Model 1 (b) Individual Model 2 (c) Ensemble

FIGURE 2.11 – Illustration du paysage de prédiction des modèles de base et du modèle d'ensemble, où les couleurs représentent les classes (source [Horváth et al., 2022a]).

De manière complémentaire, [Alfarra et al., 2022b, Anderson et al., 2022] optimisent les stratégies de lissage pour minimiser la variance tout en maximisant les rayons de certification. Pour généraliser la certification à davantage de normes L_p, les articles [Dvijotham et al., 2020, Kou et al., 2022] introduisent des extensions permettant de gérer des mesures de lissage arbitraires en s'appuyant sur les f-divergences. Ces divergences, qui incluent la divergence de Kullback-Leibler, la divergence de Jensen-Shannon et la distance Total Variation, offrent une base théorique solide pour évaluer la robustesse des classifieurs lissés. La formule générale des f-divergences, définie ci-dessous, élargit les capacités du *randomized smoothing* à des perturbations et contextes

variés.

$$D_f(P\|Q) = \int_{\mathcal{X}} f\left(\frac{p(x)}{q(x)}\right) q(x)\, dx \tag{2.10}$$

où $p(x)$ et $q(x)$ sont les densités de probabilité de P et Q respectivement, et f est une fonction convexe telle que $f(1) = 0$ (en générale la fonction logarithmique). Pour concilier robustesse et précision dans un cadre déterministe, [Mueller et al., 2021] introduit une architecture compositionnelle appelée ACE, combinant un modèle principal (F_{Core}) et un modèle certifiable (F_{Certify}) avec un mécanisme de sélection certifiable (F_{Select}). Cette idée a été étendue par [Horváth et al., 2022b], qui applique le *randomized smoothing* aux composants d'ACE pour créer l'architecture ACES :

$$F_{\text{ACES}}(x) = \overline{F}_{\text{Select}}(x) \cdot \overline{F}_{\text{Certify}}(x) + (1 - \overline{F}_{\text{Select}}(x)) \cdot F_{\text{Core}}(x) \tag{2.11}$$

permettant une robustesse accrue avec des compromis optimisés ($\overline{F}_{\text{Select}}$ et $\overline{F}_{\text{Certify}}$ sont les versions lissées). Le *randomized smoothing* a également été adapté à des tâches spécifiques, comme la détection d'objets. Les articles [Chiang et al., 2020, Kumar and Goldstein, 2021] proposent des modifications adaptées aux caractéristiques de ces tâches complexes. De plus, [Kumar et al., 2020] ajoute une dimension de confiance aux classes prédites en calculant un score moyen des prédictions, tandis que [Carlini et al., 2023] intègre des modèles de diffusion pour améliorer le rayon de certification.

En conclusion, bien que la méthode *Randomized Smoothing* présente des avantages pour certifier la robustesse des modèles face à des perturbations aléatoires, elle ne répond pas pleinement aux besoins spécifiques de recherche de garanties pour les modules d'IA dans notre contexte. La méthode est conçue pour certifier la robustesse des modèles à grande échelle, indépendamment de leur architecture interne, et peut être appliquée sans nécessiter de modifications des modèles, ce qui est un avantage notable pour des systèmes complexes comme notre chaîne compression-détection. Cependant, *Randomized Smoothing* repose sur l'hypothèse de perturbations aléatoires (typiquement gaussiennes), ce qui diffère fondamentalement des distorsions déterministes et spécifiques engendrées par la compression d'image. Cette divergence limite l'applicabilité directe de la méthode à notre cas d'étude. De plus, la nature déterministe des distorsions liées à la compression empêche l'échantillonnage multiple des perturbations pour une même image. L'application de *Randomized Smoothing* à des espaces d'entrée complexes, tels que les images compressées en haute résolution, peut également poser des défis en termes de ressources de calculs majeurs. La méthode exige de nombreuses évaluations du modèle sous différentes perturbations, ce qui est prohibitif pour des tâches coûteuses comme la détection d'objets. Enfin, les garanties offertes par cette méthode concernent les formes lissées des modèles et non pas les modèles eux-mêmes. De plus, ces garanties sont locales, c'est-à-dire spécifiques à chaque point d'entrée, et ne permettent pas d'évaluer les performances globales du modèle sur des scénarios variés, tels que ceux définis par l'ODD. Ainsi, bien que le *Randomized Smoothing* puisse être un outil complémentaire pour certifier la robustesse locale, il ne constitue pas une solution globale pour garantir les performances de la chaîne compression-détection et, par conséquent, pour répondre aux objectifs de la thèse.

2.3.2 Analyse PAC-Bayes

Une autre méthode de certification statistique se nomme l'**Analyse PAC-Bayes**. L'analyse PAC-Bayes, introduite pour la première fois dans [McAllester, 1999] en 1998, constitue un cadre puissant pour quantifier la généralisation et la robustesse des modèles d'apprentissage automatique. Cette approche repose sur le principe "*Probably Approximately Correct*" (PAC) et l'application de la théorie bayésienne pour fournir des bornes sur la performance des modèles en fonction de l'erreur empirique et de la complexité du modèle.

Selon [Alquier, 2024], étant donné un modèle f, l'erreur de généralisation, ou risque, est définie par :

$$R(\theta) = \mathbb{E}_{(X,Y)\sim P}\left[l\left(f_\theta(X), Y\right)\right] \tag{2.12}$$

où (X, Y) représente les données d'entrée et les annotations sous la distribution P, l est la fonction de perte, et f_θ est le prédicteur paramétré par θ. L'objectif de l'apprentissage est de minimiser cette erreur, bien que dans la pratique, on minimise l'erreur empirique :

$$r(\theta) = \frac{1}{n}\sum_{i=1}^{n} l\left(f_\theta(X_i), Y_i\right) \tag{2.13}$$

avec n la taille de l'échantillon. Le minimiseur du risque empirique (*Empirical Risk Minimizer*, ERM) est défini comme :

$$\hat{\theta}_{ERM} = \text{argmin}_{\theta \in \Theta} r(\theta) \tag{2.14}$$

Les bornes PAC fournissent une garantie sur la probabilité que l'erreur de généralisation $R(\hat{\theta}_{ERM})$ soit bornée par une quantité calculable basée sur l'erreur empirique. En utilisant des outils tels que l'inégalité de Markov, l'inégalité de Hoeffding et la borne de Chernoff, on peut obtenir la borne PAC suivante :

$$\mathbb{P}\left(R\left(\hat{\theta}_{ERM}\right) \leq \inf_{\theta \in \Theta} r(\theta) + C\sqrt{\frac{\log \frac{M}{\epsilon}}{2n}} \right) \geq 1 - \epsilon \tag{2.15}$$

où $M = \text{card}(\Theta)$, n est la quantité de données et C est une constante dépendant des valeurs que peut prendre la fonction de perte l. L'équation 2.15 correspond à une borne empirique signifiant que, pour tout θ, $R(\theta)$ est majoré par une quantité empirique calculable à partir des données observées. Cela permet de définir l'ERM comme le minimiseur de cette borne. Lorsque l'espace des paramètres Θ n'est pas fini, le terme $\log(M)$ est remplacé par la dimension de Vapnik-Chervonenkis [Vapnik and Chervonenkis, 2015] (VC-dim), qui mesure la complexité de Θ. La VC-dimension d'un ensemble de fonctions de classification (ou d'un modèle) peut être définie comme le plus grand nombre de points qui peuvent être complètement séparés de toutes les manières possibles par les fonctions de cet ensemble. En termes plus simples, la dimension VC mesure la capacité d'un modèle à classifier différentes configurations de données. Par exemple, si on prend un modèle de classification linéaire dans un espace bidimensionnel (comme une ligne dans un plan), la VC-dimension est 3 (cf. Figure 2.12). Cela signifie que l'algorithme peut séparer de manière correcte et complète jusqu'à 3 points dans n'importe quelle disposition, mais pas 4. Un modèle avec une VC-dimension trop élevée peut être trop flexible et surapprendre, tandis qu'une VC-dimension trop basse peut indiquer un modèle trop simple qui sous-apprend. Une autre métrique de quantification de complexité est la complexité de Rademacher [Bartlett and Mendelson, 2003] (elle fournit une mesure quantitative de la complexité d'une classe de fonctions, aidant à comprendre si une classe est susceptible de surapprendre).

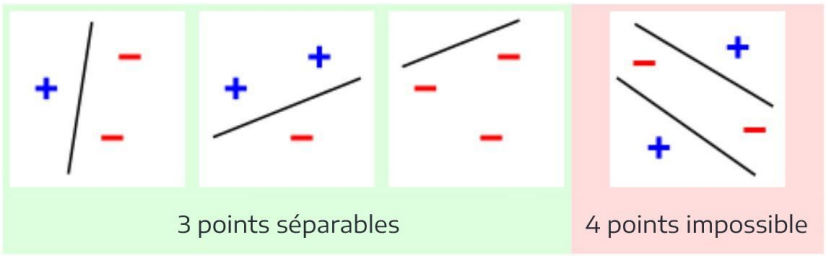


FIGURE 2.12 – Détermination de la VC-dim pour un classifieur linéaire dans un espace bidimensionnel, VC-dim= 3 (source [Wikipedia, 2024]).

On peut distinguer deux types principaux de bornes PAC :

- Bornes PAC numériques : Elles fournissent une estimation directe de l'erreur de généralisation, de la forme

$$\mathbb{P}\left(R\left(\hat{\theta}_{ERM}\right) \leq 0.12 \right) \geq 0.99$$

- Bornes PAC oracles : Bien qu'elles ne puissent pas être directement calculées car elles dépendent de la fonction R inconnue, elles donnent une idée de la proximité entre $R\left(\hat{\theta}_{ERM}\right)$ et sa valeur minimale possible

$$\mathbb{P}\left(R\left(\hat{\theta}_{ERM}\right) \leq \inf_{\theta \in \Theta} R(\theta) + r_n(\epsilon) \right) \geq 1 - \epsilon$$

où le reste $r_n(\epsilon) \to 0$ lorsque $n \to \infty$. Néanmoins, ces bornes sont très intéressantes car elles indiquent à quel point on peut s'attendre à ce que $R(\hat{\theta}_{ERM})$ soit proche de la plus petite valeur possible de R.

Les bornes PAC-Bayes sont une extension du cadre PAC classique, intégrant des principes bayésiens pour fournir des garanties probabilistes plus fines sur l'erreur de généralisation des modèles d'apprentissage. Ces bornes exploitent une mesure de probabilité a priori sur l'espace des paramètres pour quantifier la complexité du modèle et ainsi mieux évaluer sa capacité à généraliser. Le principe fondamental des bornes PAC-Bayes est de considérer non pas un unique estimateur ponctuel, mais une distribution de probabilité sur l'ensemble des paramètres possibles du modèle. Cette approche probabiliste permet d'évaluer la performance moyenne du modèle sur cette distribution, offrant ainsi une mesure plus robuste de sa capacité de généralisation. Pour cela, on introduit une mesure de probabilité a priori π définie sur l'espace des paramètres Θ, qui représente nos croyances initiales avant l'observation des données. Pour formaliser cette idée, on définit une mesure de probabilité dépendant des données, notée $\hat{\rho}$, qui associe à chaque jeu de données possible une distribution de probabilité sur Θ :

$$\hat{\rho} : \bigcup_{n=1}^{\infty} (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{P}(\Theta) \tag{2.16}$$

où $\mathcal{P}(\Theta)$ est l'ensemble de toutes les distributions de probabilité sur Θ, signifiant que pour chaque jeu de données possible, quelle que soit sa taille, la fonction $\hat{\rho}$ associe une distribution de probabilité sur Θ. Cette fonction $\hat{\rho}$ permet de modéliser comment nos croyances sur les paramètres du modèle sont mises à jour en fonction des données observées. Dans ce cadre PAC-Bayes, plusieurs types de risques peuvent être considérés :

- Risque d'un estimateur aléatoire : $R(\tilde{\theta})$, où $\tilde{\theta}$ est un ensemble possible de paramètres du modèle échantillonné selon la distribution $\hat{\rho}$. Cela représente l'erreur de généralisation d'un modèle choisi aléatoirement en suivant $\hat{\rho}$.
- Risque moyen : $\mathbb{E}_{\theta\sim\hat{\rho}}[R(\theta)]$, qui est l'espérance du risque sur la distribution $\hat{\rho}$. Ce risque moyen offre une mesure globale de la performance attendue du modèle sur l'ensemble des paramètres possibles selon la distribution $\hat{\rho}$.
- Risque de l'estimateur agrégé $R(f_{\hat{\rho}})$, où on moyenne les predicteurs $f_{\hat{\rho}}(\cdot) = \mathbb{E}_{\theta\sim\hat{\rho}}[f_\theta(\cdot)]$ donnant un nouveau predicteur.

Un objectif central de l'approche PAC-Bayes est de dériver des bornes aussi serrées que possible pour ces risques, afin d'obtenir des garanties solides sur la capacité de généralisation du modèle. Ces bornes intègrent non seulement le risque empirique observé sur les données d'entraînement, mais aussi une pénalité liée à la complexité du modèle, souvent mesurée par la divergence de Kullback-Leibler (KL) entre la distribution a posteriori ρ et π.

Par exemple, la borne de Catoni [Catoni, 2003] est une illustration emblématique des bornes PAC-Bayes. Elle établit une relation précise entre le risque moyen sous ρ et le risque empirique, tout en intégrant une pénalité pour la complexité du modèle. Pour établir cette borne, on commence par fixer une mesure de probabilité a priori $\pi \in \mathcal{P}(\Theta)$, reflétant nos croyances initiales sur les paramètres avant l'observation des données. Ensuite, pour tout $\lambda > 0$ et tout $\epsilon \in (0, 1)$, la borne de Catoni affirme qu'avec une probabilité d'au moins $1 - \epsilon$, l'inégalité suivante est vérifiée pour toute mesure de probabilité $\rho \in \mathcal{P}(\Theta)$:

$$\mathbb{E}_{\theta\sim\rho}[R(\theta)] \leq \mathbb{E}_{\theta\sim\rho}[r(\theta)] + \frac{\lambda C^2}{8n} + \frac{\mathrm{KL}(\rho \parallel \pi) + \log\frac{1}{\epsilon}}{\lambda} \tag{2.17}$$

avec $\mathbb{E}_{\theta\sim\rho}[R(\theta)]$ le risque moyen réel sur la distribution ρ, $\mathbb{E}_{\theta\sim\rho}[r(\theta)]$ le risque empirique moyen calculé sur les données d'entraînement, $\mathrm{KL}(\rho \parallel \pi)$ est la divergence de Kullback-Leibler entre ρ et π, quantifiant la "distance" entre la distribution a posteriori et le prior, C une constante dépendant de la fonction de perte utilisée et n la taille de l'échantillon de données.

L'étude [Alquier, 2024] offre un survol des applications des bornes PAC-Bayes empiriques, mettant en évidence la course aux bornes les plus serrées, y compris celles appliquées aux modèles d'apprentissage profond.

Dans [Rivasplata et al., 2020], une inégalité PAC-Bayes de base pour les noyaux stochastiques est présentée, permettant de dériver diverses extensions des bornes PAC-Bayes connues et des nouvelles bornes. Pour ce qui est de la robustesse face aux perturbations adversariales, [Viallard et al., 2021] adapte les bornes PAC-Bayes pour estimer à quel point un modèle sera invariant à des perturbations imperceptibles des entrées en phase de test. Plutôt que de se concentrer sur une analyse du pire cas du risque, l'approche utilise le cadre PAC-Bayes pour borner le risque moyen des perturbations sur l'ensemble des hypothèses. Deux mesures du risque adversarial sont définies : le risque moyen, qui estime la probabilité de mauvaise classification d'un exemple perturbé, et le risque adversarial moyennement maximal, qui considère la probabilité qu'au moins une perturbation entraîne une erreur de classification. Ces mesures, adaptées au cadre PAC-Bayes et liées à la robustesse adversariale classique,

permettent de dériver des bornes valables pour tout type d'attaque et peuvent être minimisées algorithmiquement pour entraîner un classifieur à vote majoritaire robuste aux attaques. Les résultats empiriques montrent que ce cadre offre des garanties de généralisation non vides tout en assurant une protection efficace contre les attaques adversariales. Pour les auto-encodeurs variationnels (VAEs), [Chérief-Abdellatif et al., 2022] analyse leur capacité de reconstruction sur des données de test non vues, en utilisant des arguments issus de la théorie PAC-Bayes. L'article offre des bornes de généralisation sur l'erreur théorique de reconstruction et met en lumière l'effet de régularisation des objectifs des VAEs. Une avancée majeure dans l'obtention de bornes plus serrées est réalisée par [Jang et al., 2023], qui affine la stratégie de preuve en utilisant le cadre "*Coin Betting*", cadre théorique utilisé dans l'apprentissage en ligne et la théorie des jeux financiers pour modéliser la prise de décision en situation d'incertitude. Il implique de parier de manière répétée sur le résultat d'un lancer de pièce, avec pour objectif de développer une stratégie qui maximise la richesse au fil du temps tout en gérant le risque. Cette approche, issue des algorithmes de jeux en ligne, permet de dériver des inégalités de concentration valides simultanément pour toutes tailles d'échantillons, ce qui constitue une première dans le domaine. Pour les fonctions de pertes à queue lourde ("*Heavy-tailed*"), [Haddouche and Guedj, 2023] introduit des bornes PAC-Bayes exploitant des supermartingales et une extension de l'inégalité de Markov. Cette approche unifie divers cadres PAC-Bayesiens et fournit des bornes valides pour les contextes d'apprentissage batch et en ligne ("*Online*", méthode pour l'apprentissage dans des environnements où les données arrivent continuellement et où une adaptation rapide du modèle est nécessaire). Enfin, [Chugg et al., 2023] présente une méthodologie unifiée pour dériver des bornes PAC-Bayes uniformes dans le temps. Contrairement aux travaux antérieurs, ces bornes sont valides pour tout moment d'arrêt de l'apprentissage et ne nécessitent pas une taille d'échantillon fixe. Cette contribution permet de relâcher les hypothèses traditionnelles, y compris celles concernant les fonctions de perte non stationnaires et les données non i.i.d.

La méthode PAC-Bayes cherche donc à fournir des garanties probabilistes sur la généralisation des modèles, c'est-à-dire leur capacité à maintenir une bonne performance sur des données non vues. La méthode repose sur une borne qui encadre l'erreur de généralisation (test) en fonction de l'erreur observée sur les données d'entraînement, corrigée par un terme régularisateur qui dépend de la divergence entre la distribution a posteriori sur les paramètres du modèle et une distribution a priori, ainsi que de la taille de l'ensemble d'entraînement. La borne PAC-Bayes peut s'écrire sous la forme d'un compromis entre l'ajustement sur les données d'entraînement et la complexité du modèle : plus le modèle est complexe (grande KL-divergence entre les distributions a priori et a posteriori), plus le terme de régularisation pénalise la borne. Cela rend la méthode particulièrement utile pour comprendre les mécanismes de généralisation des modèles, en équilibrant flexibilité et contrôle du surapprentissage. Cependant, bien que PAC-Bayes offre un cadre rigoureux, son applicabilité pratique reste limitée aux modèles relativement simples ou aux tâches de faible dimension. Par exemple, [Chérief-Abdellatif et al., 2022] rapporte des expériences sur des réseaux de neurones VAE avec seulement 3 couches, 400 neurones par couche cachée et des activations ReLU, appliqués au dataset MNIST ($1 \times 28 \times 28$). Ces paramètres modestes mettent en lumière la difficulté d'étendre PAC-Bayes à des architectures modernes complexes ou à des données de haute dimension, où les bornes deviennent rapidement trop lâches pour être significatives. Des efforts sont en cours pour améliorer la pertinence de PAC-Bayes, notamment en affinant les choix de distribution a priori ou en réduisant la dépendance à la complexité explicite du modèle, mais ces avancées restent encore en grande partie limitées à des contextes expérimentaux contrôlés ou simplifiés. Dans le cadre de la thèse, cette méthode aurait pu prétendre correspondre à la méthodologie que nous cherchons, car elle vise à fournir des garanties générales sur les performances des modèles. Toutefois, elle se heurte à plusieurs obstacles pratiques qui limitent son applicabilité. La complexité des modèles modernes, combinée à la haute dimension des données utilisées rend les bornes PAC-Bayes peu informatives ou inutilisables dans ce contexte. De plus, la difficulté à définir une distribution a priori pertinente pour des réseaux complexes complique davantage son intégration dans une méthodologie robuste et scalable. Par conséquent, bien que théoriquement prometteuse, la méthode PAC-Bayes n'est pas adaptée à l'obtention de garanties de performance dans un cadre appliqué tel que la chaîne compression-détection étudiée dans cette thèse.

2.3.3 Conformal Prediction

Ensuite, parmi les méthodes de certification statistique, il est pertinent de mentionner une méthode récente nommée ***Conformal Prediction*** (CP). La *Conformal Prediction*, souvent aussi référencée *Split Conformal Prediction*, est une méthode simple et flexible pour générer des ensembles de prédiction pour n'importe quel modèle d'apprentissage automatique. L'approche commence par un modèle déjà entrainé (comme un classifieur) que l'on désigne par f. Ensuite, des ensembles de prédiction (groupes de labels potentiels) sont créés pour ce

modèle à l'aide d'un ensemble de données de calibration supplémentaire. Formellement, comme expliqué dans [Fontana et al., 2023, Angelopoulos and Bates, 2023], supposons que nous ayons des images en entrée, chacune appartenant à l'une des K classes, et un classifieur $f(x)$ qui retourne des probabilités estimées (scores softmax) pour chaque classe. On réserve un ensemble modéré de données indépendantes et identiquement distribuées (i.i.d.), ou de manière plus générale échangeables (c'est-à-dire la probabilité conjointe de l'ensemble des observations reste la même quel que soit l'ordre des observations), non vues pendant l'entraînement, par exemple, 500 paires d'images et de classes $(X_1, Y_1), ..., (X_n, Y_n)$, pour la calibration. L'objectif est de construire un ensemble de prédiction $C(X_{test}) \subset \{1, ..., K\}$ valide, satisfaisant la couverture marginale :

$$1 - \alpha \leq \mathbb{P}\left(Y_{test} \in C(X_{test})\right) \leq 1 - \alpha + \frac{1}{n+1} \tag{2.18}$$

où (X_{test}, Y_{test}) est un point de test issu de la même distribution, et α est le taux d'erreur choisi par l'utilisateur.



FIGURE 2.13 – Trois exemples de plus en plus difficiles de la classe écureuil fauve et les ensembles de prédiction (c'est-à-dire, $C(X_{test})$) générés par la prédiction conforme (source [Angelopoulos and Bates, 2023]).

Pour construire C, une étape de calibration est réalisée en définissant le score conforme (*conformal score*) $s_i = 1 - f(X_i)_{Y_i}$, c'est-à-dire 1 moins la sortie softmax de la classe réelle, qui est élevé lorsque le modèle est loin de la vérité. Ensuite, le quantile empirique $\hat{q}$ est calculé à partir des scores conformes des données de calibration, permettant de créer l'ensemble $C(X_{test}) = \{y : f(X_{test})_y \geq 1 - \hat{q}\}$, garantissant que l'ensemble de prédiction respecte la couverture désirée, indépendamment du modèle ou de la distribution des données (cf. Figure 2.14).

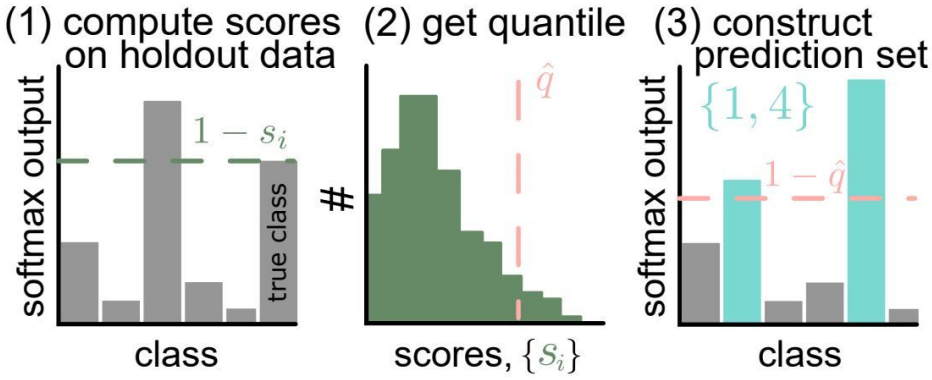


FIGURE 2.14 – Processus pour créer l'ensemble $C(X_{test})$ (source [Angelopoulos and Bates, 2023]).

Bien que la garantie soit toujours respectée, l'utilité des ensembles de prédiction dépend principalement de la fonction de score utilisée. La *conformal prediction* peut être appliquée à tout modèle préentraîné pour produire des ensembles contenant la vérité terrain avec une probabilité choisie par l'utilisateur. Les extensions de la méthode incluent :

- Prédiction conforme équilibrée par groupe (*Group-Balanced Conformal Prediction*) : pour assurer des taux d'erreur égaux à travers des sous-groupes de données (ex. groupes raciaux ou de genre).
- Prédiction conforme conditionnelle à la classe (*Class-Conditional Conformal Prediction*) : pour obtenir une couverture spécifique pour chaque classe.
- Contrôle du risque conforme (*Conformal Risk Control*) : nous avons utilisé la prédiction conforme pour construire des ensembles de prédiction qui bornent la non-couverture $\mathbb{P}\left(Y_{test} \notin C(X_{test})\right) \leq \alpha$. Cependant, pour de nombreux problèmes d'apprentissage automatique, la notion naturelle d'erreur n'est pas la

non-couverture. La prédiction conforme peut également fournir des garanties de la forme

$$\mathbb{E}[l\left(C(X_{test}), Y_{test}\right)] \leq \alpha, \tag{2.19}$$

pour toute fonction de perte bornée l qui diminue à mesure que C s'agrandit.

- Détection d'anomalies (*Outlier Detection*) : qui identifie si des points de test ne proviennent pas de la même distribution.
- Prédiction conforme sous décalage de données (*Conformal Prediction Under Data Shift*) : en pondérant davantage les scores conformes récents pour s'adapter aux changements dans la distribution des données.

La *Conformal Prediction* est donc une méthode qui permet de générer des ensembles de prédiction garantissant des intervalles de confiance pour tout modèle d'apprentissage, indépendamment de sa structure ou de la distribution des données.

L'article [Angelopoulos et al., 2021] introduit un cadre permettant de calibrer les modèles d'apprentissage automatique de manière à ce que leurs prédictions satisfassent des garanties statistiques explicites pour des échantillons finis. Les algorithmes de calibration fonctionnent avec n'importe quel modèle sous-jacent et une distribution inconnue de génération de données, sans nécessiter de réentraînement du modèle. La méthode "*Learn then Test*" permet d'octroyer des garanties statistiques aux modèles de prédiction complexes sans faire d'hypothèses sur leur fonctionnement interne ni sur la distribution réelle des données. Cette approche lie l'inférence prédictive sans distribution aux concepts bien établis des tests d'hypothèses multiples pour garantir la fiabilité des modèles d'apprentissage automatique. Cette méthode est particulièrement utile dans des domaines tels que les sciences médicales, où [Vazquez and Facelli, 2022] l'a appliquée pour évaluer la précision des prédictions individuelles en contexte clinique. Pour les contextes où le nombre de données étiquetées par classe est limité, [Ding et al., 2023] propose la prédiction conforme conditionnelle par clusters, qui regroupe les classes ayant des scores conformes similaires pour améliorer la couverture. L'article [Gibbs et al., 2024] se concentre sur la construction d'ensembles de prédiction non dépendants de la distribution avec des garanties conditionnelles sur des échantillons finis, quantifiant l'incertitude des modèles modernes sans hypothèses sur la distribution. Dans le domaine de la détection d'objets, [Timans et al., 2024] a introduit une méthode de prédiction conforme en deux étapes pour obtenir des intervalles d'incertitude garantis pour les boîtes englobantes (cf. Figure 2.15). Cette approche conditionne les prédictions des boîtes sur la classe de l'objet et adapte les intervalles d'incertitude en fonction de la taille de l'objet, permettant ainsi une couverture plus équilibrée.

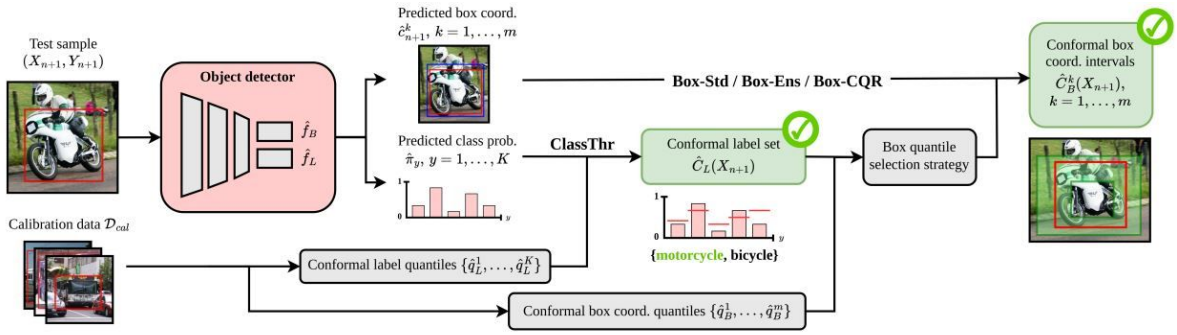


FIGURE 2.15 – Schéma de l'approche proposée par [Timans et al., 2024].

Comme décrit précédemment, le *Conformal Risk Control* (CRC) étend la *Conformal Prediction* pour contrôler l'espérance de toute fonction de perte monotone. L'article [Angelopoulos et al., 2024] a généralisé cette approche pour inclure des extensions telles que le contrôle des risques sous changement de distribution et l'utilisation d'U-statistiques (statistiques qui peuvent s'écrire comme la moyenne empirique d'une fonction symétrique à m variables prise sur toutes les sous-parties de taille m d'un échantillon). De même, [Xu et al., 2023b] a appliqué le CRC aux tâches de classification ordinale (quand les classes sont ordonnées, par exemple l'ensemble {"très négatif", "négatif", "neutre", "positif", "très positif"}), proposant des fonctions de perte adaptées pour construire des ensembles de prédiction qui contrôlent le risque à un niveau désiré. Des applications spécifiques telles que la détection d'objets dans le domaine ferroviaire ont démontré la pertinence du cadre CRC pour évaluer les performances des modèles et garantir des bornes d'incertitude formelles ([Andeol et al., 2023]). Dans le domaine de la génération d'image par modèle de diffusion, [Teneggi et al., 2023] a appliqué le CRC à des problèmes de régression image-à-image en utilisant une optimisation convexe pour minimiser la longueur des intervalles tout en

garantissant une couverture pixel par pixel sur des observations futures. Enfin, [Cohen et al., 2024] a récemment introduit une méthode CRC basée sur la validation croisée, améliorant l'efficacité des prédicteurs d'ensemble lorsque les données sont limitées, et offrant des garanties théoriques sur le risque moyen.

En conclusion, la *Conformal Prediction* (CP) et le *Conformal Risk Control* (CRC) sont des méthodes récentes et puissantes pour garantir la fiabilité des prédictions issues de modèles d'apprentissage machine, en particulier dans des contextes critiques où le contrôle des erreurs est essentiel. Le CRC se distingue par sa capacité à contrôler l'espérance des pertes tout en maintenant des taux d'erreur maîtrisés. En s'appuyant sur le cadre de la prédiction conforme, le CRC fournit des garanties probabilistes sur des quantités agrégées, telles que le risque attendu ou le taux d'erreur, et permet de calibrer des modèles complexes sans nécessiter de réentraînement, tout en étant capable de gérer les changements de distribution. Dans le contexte de cette thèse, le CRC pourrait jouer un rôle pour garantir que les dégradations causées par la compression restent sous un seuil acceptable, offrant ainsi un contrôle précis et fiable des risques associés. Cependant, la mise en œuvre de ces méthodes peut être coûteuse en termes de calcul, en particulier lorsque le système implique de nombreux scénarios ou classes. De plus, le CRC repose sur l'hypothèse d'indépendance et d'identique distribution (iid) des données, une condition qui peut être difficile à satisfaire dans des systèmes hétérogènes tels que ceux des véhicules autonomes. Enfin, l'efficacité du CRC dépend fortement de la qualité, de la quantité et de la diversité des données de calibration disponibles, ce qui pourrait limiter son application dans des environnements où les données sont limitées ou peu représentatives.

2.3.4 Conclusion

Les méthodes de certification statistique offrent donc un cadre probabiliste pour garantir les performances des systèmes intégrant des modules d'intelligence artificielle. Contrairement aux méthodes de vérification formelle, qui visent à prouver des propriétés de manière exhaustive et déterministe, ces approches reposent sur des analyses empiriques et des hypothèses statistiques. Elles permettent d'évaluer, avec un certain niveau de confiance, la probabilité qu'un système respecte une propriété donnée ou qu'il fonctionne correctement dans un ensemble de conditions spécifiées. Ces méthodes sont particulièrement utiles lorsque l'évaluation exhaustive est impraticable en raison de la taille, de la complexité, ou des incertitudes liées aux données et aux modèles utilisés. Parmi l'état de l'art, on distingue trois méthodes qui se démarquent.

La première est le *Randomized Smoothing*. C'est une méthode conçue pour certifier la robustesse locale des modèles en présence de perturbations aléatoires, telles que des bruits gaussiens. Son avantage réside dans son indépendance vis-à-vis de l'architecture interne des modèles, rendant son application possible pour des systèmes complexes sans nécessiter de modifications majeures. Cependant, son applicabilité est limitée dans le cadre de perturbations déterministes, comme celles causées par la compression d'images. De plus, les ressources de calcul nécessaires pour générer des garanties robustes sur des espaces d'entrée complexes, tels que les images haute résolution, limitent son utilisation. Enfin, les garanties offertes sont locales, c'est-à-dire spécifiques à des données d'entrée individuelles et ne s'étendent généralement pas à l'ensemble des données ou à des scénarios définis par un ODD. Cette méthode est pertinente vis-à-vis de notre application industrielle, *Field Monitoring*, qui vise à garantir la robustesse d'un module de détection d'objets face aux perturbations induites par un module de compression d'images. Cependant, elle ne constitue pas une approche adaptée à une méthodologie plus générale d'obtention de garanties de performances pour des modules d'IA.

La seconde est la méthode PAC-Bayes. Elle fournit des garanties probabilistes sur la généralisation des modèles en encadrant leur erreur sur des données non vues à l'aide de bornes calculées à partir de la divergence KL entre distributions a priori et a posteriori sur les paramètres du modèle. Bien qu'elle soit théoriquement rigoureuse et utile pour analyser la capacité de généralisation, elle reste difficilement applicable aux modèles modernes complexes et aux données de haute dimension. Les bornes deviennent rapidement trop lâches pour être informatives dans des contextes réels impliquant des architectures avancées. Dans le cadre de cette thèse, bien qu'intéressante pour comprendre les performances globales, cette méthode ne répond pas aux contraintes de complexité et de scalabilité de notre chaîne compression-détection.

Enfin, la troisième est la *Conformal Prediction* (CP), qui permet de calibrer des prédictions avec des garanties probabilistes sur des quantités agrégées telles que le taux d'erreur ou le risque attendu. En particulier, une extension, *Conformal Risk Control* (CRC), est adaptée pour contrôler l'espérance des pertes tout en maintenant des taux d'erreur prédéfinis. Cette méthode offre une flexibilité importante, notamment en permettant une calibration sans réentraînement et en gérant les changements de distribution des données. Dans le contexte de cette thèse, CRC pourrait être particulièrement pertinent pour garantir que les dégradations induites par la compression d'images restent sous un seuil acceptable. Cependant, comme pour CP, l'efficacité de CRC dépend

fortement de la qualité et de la représentativité des données de calibration disponibles. De plus, son hypothèse d'indépendance et de distribution identique (iid) des données peut limiter son applicabilité dans des systèmes hétérogènes, tels que ceux des véhicules autonomes.

En conclusion, bien que les méthodes de certification statistique apportent des garanties flexibles et probabilistes adaptées à des systèmes complexes, elles ne sont pas encore parfaitement adaptées à la mise en œuvre d'une méthodologie générale pour garantir les performances dans des contextes variés. Dans le cas spécifique de cette thèse, visant à analyser et garantir la robustesse d'une chaîne compression-détection, ces méthodes offrent des outils complémentaires mais insuffisants pour couvrir toutes les dimensions de performance et de fiabilité requises. Des adaptations spécifiques ou des approches hybrides pourraient être nécessaires pour combler les lacunes identifiées.

2.4 Conclusion

À ce jour, il n'existe pas encore de méthode générale permettant de garantir de manière universelle que des modules d'IA satisfont certaines propriétés spécifiques. Malgré leur nombre et leur puissance, les méthodes permettant d'obtenir des garanties, incluant à la fois les techniques de vérification formelle et les approches de certification statistique, restent très limitées (cf. Tableaux 2.1 et 2.2). Ces limitations dépendent fortement de la complexité des modules d'IA, de la dimension des données, du type de garanties souhaitées, ainsi que du type de perturbations envisagées, entre autres facteurs. En particulier, la malédiction des dimensions représente un obstacle majeur : à mesure que la dimension des données augmente, le volume de l'espace des possibles croît de façon exponentielle, rendant les méthodes de vérification et de certification moins efficaces et plus coûteuses. Cette complexité accrue complique considérablement l'analyse, la rendant impraticable pour les modèles d'IA à grande échelle et nécessitant des approches plus robustes et innovantes pour surmonter ces défis. Il n'existe pas encore de méthode universelle pour garantir les modules d'IA, en particulier dans le contexte des réseaux de neurones profonds. La recherche se dirige vers une combinaison de techniques formelles, de certifications statistiques et de méthodes d'entraînement robustes. Les défis restent considérables, notamment l'extensibilité de ces approches, la gestion de la complexité des modèles modernes et l'adaptation à des environnements réels imprévisibles. Dans notre cas, si nous considérons un système composé de modules d'IA, nous souhaitons obtenir une estimation aussi précise que possible d'une probabilité d'erreur de la forme $P(\text{erreur} \geq \epsilon)$. Aucune des méthodes listées dans l'état de l'art n'est adaptée aux types de modules d'IA que nous souhaitons garantir, car elles sont trop limitées. Dans notre cadre applicatif, la chaîne compression-détection, nous pourrions utiliser l'une des méthodes précédemment citées pour garantir la robustesse du modèle de détection vis-à-vis des distorsions induites par le modèle de compression. Cependant, la compression crée des distorsions qui sont déterministes pour chaque image et ne ressemblent à aucune distribution de probabilité standard (le bruit n'est ni gaussien ni d'une autre forme classique), et pas nécessairement borné. De plus, nos modèles de compression et de détection sont complexes et profonds, comme nous le verrons par la suite, et comme nous travaillons avec des images de grande dimension ($3 \times 640 \times 1280$), notre espace d'entrée est très vaste. Les méthodes pour obtenir des garanties statistiques semblent donc être les plus envisageables dans notre contexte applicatif.

2.5 Récapitulatif de l'état de l'art pour la garantie d'IA

TABLE 2.1 – Tableau récapitulatif de l'état de l'art des méthodes de Vérification Formelles.

Méthodes	**Caractéristiques**	**Articles**
Solveurs SMT	rigueur garantie, adaptabilité aux architectures simples ($\sim 1,000$ neurones), scalabilité limitée ($> 10,000$ neurones ou données $> 100 \times 100$ pixels), temps de calcul exponentiel.	[Katz et al., 2017, Huang et al., 2017, Katz et al., 2019, Wu et al., 2024b, Elboher et al., 2020]
Arithmétique d'intervalles et propagation symbolique	Scalabilité supérieure à SMT et plus rapide pour des réseaux avec jusqu'à $\sim 50,000$ neurones, perte de précision si profondeur du réseau augmente (relaxation excessive des intervalles) causant faux positifs.	[Wang et al., 2018b, Wang et al., 2018a]
Programmation linéaire	réduction temps de calcul par rapport SMT, applicable à réseaux de taille moyenne ($\sim 100,000$ neurones), variantes (MILP, SDP, etc.) permettent d'équilibrer précision et scalabilité, temps de calcul élevé pour réseaux de grande taille (> 1 million de neurones), approximations nécessaires peuvent réduire la précision des garanties.	[Ehlers, 2017, Dutta et al., 2018, Tjeng et al., 2019, Anderson et al., 2020, Dvijotham et al., 2018, Bunel et al., 2020a, Palma et al., 2021, Raghunathan et al., 2018, Dathathri et al., 2020, Ma and Sojoudi, 2021]
Branch and bound	garantit des résultats exacts pour la vérification de propriétés formelles, Possibilité d'accélération grâce à des heuristiques de branchement, scalabilité améliorée, complexité exponentielle avec la taille et la profondeur du réseau, limité à $\sim 100,000$ neurones	[Bunel et al., 2018, Henriksen and Lomuscio, 2020, Bunel et al., 2020b, Lu and Kumar, 2019, Botoeva et al., 2020, Henriksen and Lomuscio, 2021, Shi et al., 2024]
Relaxation linéaire ou convexe	vérifications rapides grâce à des approximations, scalabilité raisonnable pour des réseaux de petite à moyenne taille ($\sim 100,000$ neurones), approximations convexes assurent des garanties, perte de précision	[Zhang et al., 2018a, Weng et al., 2018a, Singh et al., 2019a, Boopathy et al., 2019, Lyu et al., 2020, Xu et al., 2020, Wang et al., 2021c, Ferrari et al., 2021]
Interprétation abstraite	propagation rapide des intervalles, gèrent réseaux de grande taille jusqu'à plusieurs millions de neurones (ResNet-34), compatibles avec des implémentations parallèles (GPU), approximations sûres, approximations peuvent devenir trop larges, explosion combinatoire, complexité mémoire	[eta, , Gehr et al., 2018, Singh et al., 2018, Singh et al., 2019b, Singh et al., 2019c, Zhong et al., 2021, Müller et al., 2021]
Constante de Lipschitz	applicables à divers types de modèles et fonctions d'activation, bornes rigoureuses, évalue la sensibilité du modèle, réseaux de taille modérée ($\sim 50,000$ neurones), limitation dimensionnelle	[Weng et al., 2018c, Weng et al., 2018b, Fazlyab et al., 2019]
Apprentissage sous contraintes	modèles prouvablement robustes, divers types de réseaux (CNN) et données (ImageNet), performance nominale réduite, complexité entrainement accrue	[Mirman et al., 2018, Gowal et al., 2019, Zhang et al., 2020a, Mueller et al., 2023, Wong and Kolter, 2018, Xiao et al., 2019, Boopathy et al., 2021, Shi et al., 2021, Zhang et al., 2021a, De Palma et al., 2022, Pfrommer et al., 2023a, Balunovic and Vechev, 2020, Lee et al., 2021, Jovanovi'c et al., 2021]

TABLE 2.2 – Tableau récapitulatif de l'état de l'art sur la Certification Statistique.

Méthodes	**Caractéristiques**	**Articles**
Randomized smoothing	Simple, reposant sur des perturbations gaussiennes, s'applique efficacement à des modèles complexes et de grande taille, fournit des bornes probabilistes sur la robustesse des modèles, limité aux perturbations aléatoires, bornes certifiées peuvent être lâches, requiert nombreuses évaluations pour estimations précises	[Cohen et al., 2019, Zhai et al., 2019, Fischer et al., 2020, Gluch and Urbanke, 2020, Feng et al., 2020, Kumar et al., 2020, Dvijotham et al., 2020, Kou et al., 2022, Alfarra et al., 2022a, Levine and Feizi, 2021, Jeong et al., 2021, Li et al., 2021, Mueller et al., 2021, Horváth et al., 2022b, Horváth et al., 2022a, Chiang et al., 2020, Kumar and Goldstein, 2021, Eiras et al., 2021, Rumezhak et al., 2023, Labarbarie et al., 2022, Alfarra et al., 2022b, Anderson et al., 2022, Anderson and Sojoudi, 2022, Carlini et al., 2023, Pfrommer et al., 2023b]
Analyze PAC-Bayes	Garanties probabilistes sur la généralisation des modèles, applicable à divers modèles profonds, bornes obtenues peuvent être trop lâches, hypothèses fortes spécifiques sur la distribution des données ou les priors, calculs coûteux	[McAllester, 1999, Catoni, 2003, Alquier, 2024, Rivasplata et al., 2020, Viallard et al., 2021, Chérief-Abdellatif et al., 2022, Jang et al., 2023, Haddouche and Guedj, 2023, Chugg et al., 2023]
Conformal prediction	Fiabilité des prédictions, applicable à une large variété de modèles et de types de données, utilisé comme post-traitement sur n'importe quel modèle prédictif, suppose que les données de test sont identiquement distribuées comme les données d'entraînement, élevé pour les grands ensembles de données ou les modèles complexes, difficile à appliquer dans des environnements où la distribution des données change rapidement	[Fontana et al., 2023, Angelopoulos and Bates, 2023, Vazquez and Facelli, 2022, Angelopoulos et al., 2021, Timans et al., 2024, Gibbs et al., 2024, Ding et al., 2023, Angelopoulos et al., 2024, Xu et al., 2023b, Andeol et al., 2023, Teneggi et al., 2023, Cohen et al., 2024]

Chapitre 3

État de l'art pour l'application industrielle

Contents

3.1 Introduction . . . 47
3.2 Compression apprise par deep learning . . . 47
3.2.1 Introduction . . . 47
3.2.2 Métriques . . . 50
3.2.3 Modèles Entropiques . . . 52
3.2.4 Approches basées sur les auto-encodeurs . . . 56
3.2.5 Approches basées sur les modèles de contexte . . . 58
3.2.6 Approches multi-échelles . . . 59
3.2.7 Approches basées sur l'attention et le contenu . . . 61
3.2.8 Approches basées sur les Transformers . . . 63
3.2.9 Approches basées sur les modèles génératifs . . . 64
3.2.10 Autres approches . . . 67
3.2.11 Approches à taux de compression variable . . . 69
3.2.12 Approches ROI & orientées tâches . . . 73
3.2.13 Conclusion . . . 77
3.3 Détection d'objets par deep learning . . . 77
3.3.1 Introduction . . . 78
3.3.2 Métriques . . . 79
3.3.3 Approches Two-stage . . . 82
3.3.4 Approches Single-stage . . . 87
3.3.5 Approches basées sur les Transformers . . . 91
3.3.6 Conclusion . . . 94
3.4 Génération d'image . . . 94
3.4.1 Métriques . . . 95
3.4.2 Approches à base de VAE . . . 97
3.4.3 Approches à base de GAN . . . 98
3.4.4 Approches à base de Diffusion . . . 101
3.4.5 Génération dans le domaine automobile . . . 105
3.4.6 Génération d'images critiques . . . 110
3.4.7 Conclusion . . . 113
3.5 Conclusion . . . 113
3.6 Récapitulatif de l'état de l'art pour l'application industrielle . . . 115

3.1 Introduction

Comme présenté dans le Chapitre 1, notre application concerne une chaîne de traitement d'images intégrant deux modules clés basés sur l'apprentissage profond : un modèle de compression d'images et un modèle de détection d'objets (cf. Figure 1.3). L'objectif central est de garantir que les distorsions introduites par la compression n'affectent pas de manière significative la précision et la fiabilité du détecteur d'objets. En appliquant une méthodologie pour obtenir des garanties de performances, nous cherchons à valider que l'impact des distorsions reste en dessous d'un seuil de tolérance prédéfini, assurant ainsi que la qualité des données compressées est suffisante pour maintenir une détection efficace et cohérente.

Dans ce chapitre, nous explorons les trois aspects fondamentaux de cette chaîne de traitement d'images. Premièrement, nous nous concentrerons sur les méthodes de compression d'images basées sur l'apprentissage profond. Ces approches cherchent à optimiser un compromis entre débit de compression et distorsion, mais nous devrons aussi prendre en compte la vitesse d'inférence, un critère crucial pour les applications en (quasi) temps réel. Ensuite, nous présenterons un état de l'art sur les méthodes de détection d'objets, en nous intéressant particulièrement aux avancées offertes par les réseaux de neurones profonds, tels que les modèles two-stage, single-stage et ceux basés sur des Transformers. Nous analyserons leurs forces et leurs limites dans le contexte de tâches de vision par ordinateur complexes. Enfin, comme la validation des performances repose fortement sur des ensembles de données, nous aborderons les méthodes génératives de création d'images. Ces modèles, qui continuent d'évoluer rapidement, permettent non seulement de générer des données supplémentaires pour enrichir les ensembles de test, mais aussi de cibler des scénarios spécifiques pour explorer les limites du système. Ces trois domaines, compression, détection et génération d'images, sont au cœur des avancées de l'apprentissage profond et font l'objet d'un développement intense et continu, contribuant à repousser les frontières des performances dans le traitement et l'analyse d'images.

3.2 Compression apprise par deep learning

Dans cette section, nous réaliserons un état de l'art sur les méthodes de compression d'image par deep learning, en les classant en fonction de leur architecture. Nous nous concentrons ici exclusivement sur la compression d'images, et non sur celle de vidéos, afin de simplifier l'étude. Ce choix se justifie par la facilité d'obtention et de génération d'images, comparé aux vidéos. De plus, dans le cadre de l'application du *Field Monitoring*, l'application enregistre des séquences temporelles de 10 secondes avec une fréquence d'acquisition minimale de 5 images par seconde. La transition de la compression d'image à celle de vidéo, qui prendrait en compte l'information temporelle pour potentiellement améliorer les processus de compression, pourra faire l'objet d'une étude ultérieure. Ce passage à la compression vidéo permettrait d'exploiter les relations temporelles entre images successives, optimisant ainsi l'efficacité de compression tout en réduisant les redondances au sein des séquences. Pour sélectionner les modèles les plus adaptés à notre application, nous retiendrons trois critères principaux : la qualité de reconstruction, qui évalue la fidélité de l'image reconstruite par rapport à l'originale, le taux de compression, qui représente le gain en stockage obtenu, et enfin la vitesse d'inférence, essentielle pour une utilisation en temps réel.

3.2.1 Introduction

La compression d'image est un processus qui vise à réduire la taille des fichiers d'image tout en maintenant une qualité visuelle acceptable. Elle est essentielle dans de nombreuses applications, car elle permet d'économiser de l'espace de stockage, de réduire les coûts de transmission de données et d'accélérer le chargement des images. La compression d'image repose sur le principe de réduction des redondances présentes dans les données d'une image, et elle peut être divisée en deux grandes catégories : compression sans perte qui permet de réduire la taille d'une image sans perdre aucune information, ce qui signifie que l'image d'origine peut être parfaitement reconstruite (elle offre souvent des taux de compression plus faibles par rapport à la compression avec perte), et la compression avec perte dans laquelle certaines informations non essentielles sont supprimées de manière irréversible, permettant ainsi une réduction de taille beaucoup plus importante.

Dans les deux types de compression, une étape essentielle est le codage entropique. Cette technique repose sur l'analyse des fréquences des différentes valeurs dans l'image compressée pour réduire davantage la taille du fichier et permet de convertir efficacement les informations en format binaire, optimisant ainsi leur stockage et leur transmission. Voici les principales méthodes de codage entropique utilisées :

- Codage Huffman : Cette méthode consiste à attribuer des codes binaires plus courts aux valeurs les plus fréquentes et des codes plus longs aux valeurs plus rares. Elle utilise un arbre binaire pour représenter les fréquences des symboles, optimisant ainsi la longueur moyenne des codes sans perdre d'information. Le codage Huffman est efficace pour les données à faible diversité de symboles, bien qu'il soit moins performant que d'autres méthodes pour des distributions plus complexes.
- Système numérique asymétrique (*Asymmetric Numeral System*, ANS) : ANS est un système de codage entropique moderne qui offre une compression plus efficace pour les flux de données. Contrairement au codage Huffman, ANS fonctionne avec des nombres en base élevée et peut coder une séquence de symboles en un seul entier, optimisant ainsi la mémoire utilisée. ANS est largement utilisé dans des applications modernes comme la compression de fichiers multimédias, car il permet une compression proche de la limite théorique tout en étant plus rapide et plus compact que d'autres systèmes.
- Codage arithmétique (*Arithmetic Coding*, AC) : Le codage arithmétique représente la séquence entière d'un message comme une fraction dans l'intervalle [0,1], en subdivisant cet intervalle en fonction de la probabilité de chaque symbole. Plus un symbole est fréquent, plus son intervalle est large. Ce système de codage est extrêmement efficace pour les données avec des distributions de probabilité complexes, offrant souvent une compression supérieure au codage Huffman.
- Codage par intervalle (*Range Coding*) : Le codage par intervalle est une variante simplifiée du codage arithmétique, utilisant des opérations de précision fixe pour réduire les exigences en calcul tout en offrant des performances similaires à l'AC. Il encode les symboles dans un intervalle qui se rétrécit à chaque étape en fonction de la probabilité des symboles. Le codage par intervalle est souvent utilisé en raison de sa rapidité et de son efficacité, bien qu'il puisse être moins précis pour des compressions très fines.

En utilisant ces méthodes, le codage entropique contribue à maximiser l'efficacité de la compression en minimisant la redondance statistique des données d'image compressées. Les méthodes de compression d'image conventionnelles, bien établies depuis plusieurs décennies, utilisent divers algorithmes pour réduire la taille des images tout en maintenant une qualité visuelle acceptable. Les principales méthodes regroupent JPEG [Wallace, 1992], JPEG-2000 [Rabbani and Joshi, 2002], Webp [web, 2010], BPG [bpg, 2014] ou encore HEIF [Hannuksela et al., 2015].

Le format JPEG [Wallace, 1992], introduit en 1992 par Wallace, est l'un des standards de compression d'image les plus populaires et largement utilisés. Sa popularité tient à sa faible complexité calculatoire et à son efficacité à coder des images tout en tenant compte du système visuel humain. Le flux de compression JPEG divise l'image en blocs de 8×8 pixels, qui subissent une transformation en cosinus discrète (*Discrete Cosine Transform*, DCT) pour passer du domaine spatial au domaine fréquentiel. Cette transformation permet une meilleure concentration de l'énergie de l'image. Les coefficients DCT sont ensuite quantifiés via une table de quantification, réduisant les coefficients et créant des blocs DCT plus clairsemés. Ces coefficients quantifiés sont réarrangés en ordre zigzag pour le codage par longueur de séquence, optimisant le codage des séquences de zéros consécutifs. Enfin, les coefficients sont codés séparément en utilisant des techniques de codage entropique telles que le codage Huffman ou le codage arithmétique. Le processus inverse est appliqué pour la décompression (cf. Figure 3.1).

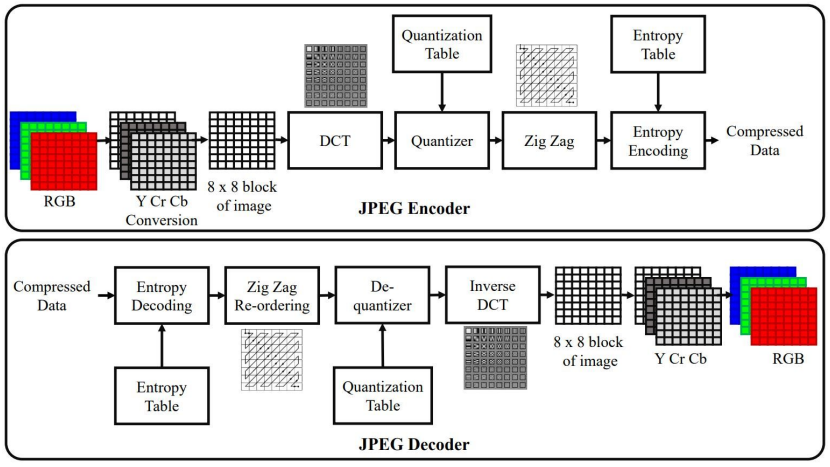


FIGURE 3.1 – Schéma de la compression et décompression JPEG (source [Jamil et al., 2023]).

JPEG-2000 [Rabbani and Joshi, 2002], introduit en 2001, est une version améliorée de JPEG qui utilise une transformation en ondelettes discrètes (*Discrete Wavelet Transform*, DWT) au lieu de la DCT. Cette transformation permet une meilleure gestion de la compression progressive et une meilleure qualité d'image pour des taux de compression élevés. Le codage entropique dans JPEG-2000 utilise des méthodes similaires à celles de JPEG, incluant le codage Huffman, RLE (*Run-Length Encoding*), et le codage arithmétique (AC).

WebP [web, 2010] est un format d'image matricielle développé par Google en 2010, spécifiquement pour le Web. Il utilise des technologies de codage intra-image provenant du codec vidéo VP8 [VP8, 2008], incluant la prédiction intra, le filtrage en boucle et la quantification adaptative par bloc. Ces techniques permettent une efficacité de codage supérieure à celle de JPEG, avec un meilleur compromis entre la qualité et la taille du fichier.

Better Portable Graphics (BPG) [bpg, 2014], basé sur le codec vidéo HEVC (*High Efficiency Video Coding*), offre une compression plus efficace que JPEG en utilisant des technologies similaires au codec VP8 de WebP. Le format BPG est conçu pour améliorer le codage intra-image, ce qui le rend adapté aux images haute qualité avec une taille de fichier réduite.

Enfin, développé par le *Moving Picture Experts Group* (MPEG), le *High Efficiency Image File Format* (HEIF) [Hannuksela et al., 2015] est un format conteneur pour stocker des images individuelles et des séquences d'images numériques. Par défaut, les images HEIF sont encodées avec le codec intra HEVC, ce qui offre une efficacité de compression supérieure. Depuis 2017, Apple a adopté HEIF comme format par défaut pour les photos sur iPhone. HEIF peut également supporter les outils de codage intra de la norme VVC (*Versatile Video Coding*), positionnant HEIF comme un codec conventionnel à la pointe de la compression d'image.

Ces méthodes de compression conventionnelles sont encore largement utilisées aujourd'hui en raison de leur efficacité et de leur adaptabilité à diverses applications. Cependant, avec l'avènement de l'intelligence artificielle, de nouvelles méthodes de compression d'image par deep learning ont émergé, offrant des perspectives prometteuses pour des compressions encore plus performantes. Ces techniques exploitent les réseaux de neurones pour apprendre automatiquement des représentations compactes des images, permettant une meilleure adaptation aux caractéristiques visuelles spécifiques de chaque image.

Compte tenu des contraintes pratiques sur la bande passante et le stockage, la compression d'images avec perte est largement adoptée pour minimiser le débit binaire nécessaire à la représentation d'une image donnée, en tolérant un certain niveau de distorsion. Le cadre de compression consiste généralement en une paire encodeur-décodeur. Étant donné une image d'entrée x avec sa distribution p_x, l'encodeur utilise une transformation de codage $\mathcal{E}$ et une fonction de quantification Q pour générer un code discret $\hat{y}$ de la manière suivante :

$$\hat{y} = Q(y) = Q\left(\mathcal{E}(x; \theta_{\mathcal{E}})\right) \tag{3.1}$$

où $\theta_{\mathcal{E}}$ désigne les paramètres de l'encodeur obtenus au cours de l'apprentissage. Pour obtenir la représentation en pixels de l'image, le décodeur correspondant $\mathcal{D}$ reconstruit l'image $\hat{x}$ à partir du code $\hat{y}$ comme suit :

$$\hat{x} = \mathcal{D}(\hat{y}; \theta_{\mathcal{D}}) = \mathcal{D}\left(Q\left(\mathcal{E}(x; \theta_{\mathcal{E}})\right); \theta_{\mathcal{D}}\right), \tag{3.2}$$

où $\theta_{\mathcal{D}}$ représente les paramètres de $\mathcal{D}$. Deux types de métriques, à savoir la distorsion D et le débit R, donnent lieu à l'optimisation taux-distorsion $R + \lambda D$, le problème central de la compression d'images avec perte. Le terme de distorsion D mesure la différence entre l'image reconstruite et l'image d'origine, souvent évaluée via des métriques de fidélité ou perceptuelles comme suit :

$$D = \mathbb{E}_{x \sim p_x}\left[d(x, \hat{x})\right], \tag{3.3}$$

où d représente la fonction de distorsion, souvent le MSE (*Mean Squared Error*) ou le MS-SSIM (*Multi-Scale Structural Similarity Index*) définis ultérieurement dans la Section 3.2.2. Le terme de débit R représente le nombre de bits nécessaires pour encoder $\hat{y}$, et est limité par les contraintes d'entropie. Pour le calculer, le modèle de probabilité a priori $p_{\hat{y}}(\hat{y})$ (également appelé modèle d'entropie) de la représentation $\hat{y}$ est crucial. La distribution marginale réelle de $\hat{y}$, qui dépend des distributions des images, est inconnue. Nous l'estimons donc par la distribution a priori. Les distributions a priori peuvent être formulées par un modèle paramétrique, et les paramètres sont appris pour s'adapter aux données. Étant donné le modèle d'entropie, la borne inférieure du débit est déterminée par l'entropie de la distribution a priori discrète de $\hat{y}$. Les débits réels R obtenus par un modèle d'entropie bien conçu sont seulement légèrement supérieurs à l'entropie :

$$R = \mathbb{E}_{x \sim p_x}\left[-\log_2 p_{\hat{y}}(\hat{y})\right], \tag{3.4}$$

Formellement, le problème peut être résolu en minimisant l'optimisation suivante avec un coefficient de compromis λ :

$$\hat{\theta}_{\mathcal{E}}, \hat{\theta}_D, \hat{\theta}_p = \text{argmin}_{\theta_{\mathcal{E}}, \theta_D, \theta_p} R + \lambda D, \tag{3.5}$$

où θ_p représente les paramètres du modèle d'entropie. Les paramètres optimaux $\hat{\theta}_E, \hat{\theta}_D, \hat{\theta}_p$ permettent au modèle d'atteindre une bonne performance taux-distorsion globale pour l'image x suivant $x \sim p_x$. Différentes valeurs de λ indiquent divers compromis taux-distorsion selon les exigences des applications. Pour une grande partie des méthodes basées sur l'apprentissage, plusieurs modèles doivent être entraînés pour différents compromis taux-distorsion. Cependant, il existe des architectures à débit variable, mais ces dernières sont généralement beaucoup plus gourmandes en temps.

3.2.2 Métriques

Il existe de nombreuses métriques permettant d'évaluer les performances de la compression d'image. Ces dernières nous seront utiles pour sélectionner les modèles pour notre application. Nous en énumérons quelques-unes dans la liste suivante :

— **MSE** (Mean Squared Error) : Calculé en prenant la moyenne des carrés des différences entre les pixels correspondant des deux images :

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [x(i,j) - \hat{x}(i,j)]^2 \tag{3.6}$$

avec m et n les nombre de pixels sur les deux dimensions spatiales des images, x l'image d'origine et $\hat{x}$ l'image comprimée. Un MSE élevé indique une grande différence entre les deux images, ce qui signifie que l'image reconstruite est de mauvaise qualité par rapport à l'image originale.

— **PSNR** (Peak Signal to Noise Ratio) : basé sur le rapport entre la puissance maximale d'un signal et la puissance du bruit qui affecte la fidélité de sa représentation. Il est calculé à partir du MSE selon la formule suivante :

$$\text{PSNR} = 10. \log_{10}(\frac{\text{MAX}^2}{\text{MSE}}) \tag{3.7}$$

où MAX est la valeur maximale possible d'un pixel dans l'image et l'unité est le décibel (dB). Dans notre cas, en Pytorch, les images sont normalisées et chaque pixel $i \in [0, 1]$, donc MAX= 1 et on obtient la formule suivante :

$$\text{PSNR} = -10. \log_{10}(\text{MSE}) \tag{3.8}$$

Un PSNR élevé indique que l'image reconstruite est très proche de l'image originale, c'est-à-dire que le bruit ou les distorsions sont faibles. Bien que le MSE et le PSNR soient utiles pour quantifier la qualité d'image, ils ne prennent pas en compte la perception visuelle humaine. Deux images peuvent avoir des PSNR similaires mais être perçues très différemment par un observateur humain.

— **SSIM** (Structural Similarity Index) : Conçu pour prendre en compte les caractéristiques perceptuelles et structurelles des images. Il compare trois aspects différents entre deux images : la Luminance (compare la luminosité moyenne des deux images), le Contraste (compare la variance ou l'écart-type des deux images) et la Structure (compare les corrélations locales entre les deux images). La formule du SSIM pour deux images x et y est :

$$\text{SSIM}(x,y) = [l(x,y)]^\alpha \cdot [c(x,y)]^\beta \cdot [s(x,y)]^\gamma \tag{3.9}$$

où $l(x,y)$, $c(x,y)$ et $s(x,y)$ sont les comparaisons de luminance, de contraste, et de structure, respectivement (les exposants α, β et γ sont généralement fixés à 1). Les sous-composants sont définis comme :

$$\begin{aligned} l(x,y) &= \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \\ c(x,y) &= \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \\ s(x,y) &= \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \end{aligned} \tag{3.10}$$

où μ_x et μ_y sont les moyenne des images, σ_x et σ_y sont les écart-types, σ_{xy} est la covariance, et C_1, C_2, C_3 sont de petites constantes pour éviter les divisions par zéro. Le SSIM est compris entre 0 et 1, une valeur proche de 1 indiquant une très haute similarité entre les images.

— **MS-SSIM** (Multi-Scale Structural Similarity Index) : est une extension de la métrique SSIM, utilisée pour évaluer la qualité d'images en tenant compte des structures et des détails perceptuellement importants. Contrairement au SSIM, qui opère à une seule échelle, le MS-SSIM évalue la similarité à travers plusieurs échelles, ce qui le rend plus sensible aux variations de structures à différentes résolutions. Les principales étapes de calcul sont le découpage en échelles multiples, le calcul du SSIM à chaque échelle et enfin la combinaison des résultats via une moyenne pondérée. Le MS-SSIM est aussi compris entre 0 et 1, une valeur proche de 1 indiquant une très haute similarité entre les images. Dans le domaine de la compression d'image, comme le PSNR, le MS-SSIM est souvent exprimé en décibel via la formule l'échelle décibel via la formule $-10.\log_{10}(1 - \text{MS-SSIM})$.

— **LPIPS** [Zhang et al., 2018b] (Learned Perceptual Image Patch Similarity) : similaire au MS-SSIM, c'est une métrique pour évaluer la qualité d'image en termes de similarité perceptuelle. Elle repose sur des réseaux de neurones convolutifs (CNN) pré-entraînés pour extraire des caractéristiques des images. La métrique calcule la différence entre les caractéristiques extraites de l'image d'origine et de l'image comprimée. Elle utilise une combinaison de différentes couches du réseau pour évaluer les écarts perceptuels à plusieurs niveaux de détail (voir l'illustration ci-dessous).

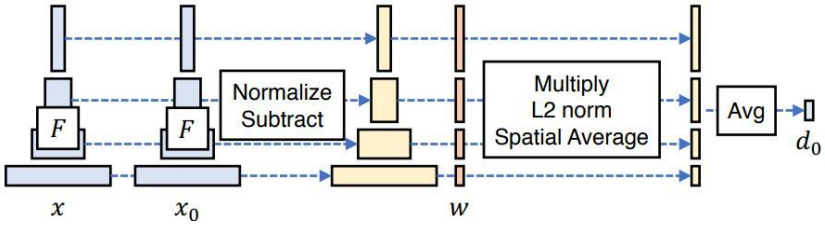


FIGURE 3.2 – Pour calculer une distance d_0 entre deux zones x et x_0, étant donné un réseau de neurones F (souvent VGG [Simonyan and Zisserman, 2015] ou AlexNet [Krizhevsky et al., 2012]), nous commençons par calculer les embeddings profonds, normaliser les activations dans la dimension des canaux, mettre à l'échelle chaque canal par le vecteur w, et prendre la distance l_2. Nous faisons ensuite une moyenne à travers la dimension spatiale et à travers toutes les couches (source [Zhang et al., 2018b]).

— **bpp** (bits per pixel) : Mesure qui indique le nombre moyen de bits utilisés pour représenter chaque pixel d'une image compressée. Le bpp est calculé de la manière suivante :

$$\text{bpp} = \frac{\text{taille du fichier en bits}}{\text{largeur de l'image} \times \text{hauteur de l'image}} \tag{3.11}$$

Les modèles de compression d'images appris (*Learned Image Compression*) sont évalués et comparés à l'aide de courbes qui mettent en relation des métriques de qualité visuelle, telles que le PSNR ou le MS-SSIM, souvent exprimées en décibels (dB), avec le débit de compression mesuré en bits par pixel (bpp). En complément, le temps d'inférence nécessaire pour l'encodage et le décodage constitue également un critère d'évaluation important, en particulier pour des applications nécessitant un traitement rapide ou en temps réel. Les courbes PSNR(bpp) et MS-SSIM(bpp) permettent d'apprécier le compromis entre le taux de compression et la qualité des images reconstruites. Un modèle est considéré comme performant lorsqu'il atteint des valeurs élevées de PSNR ou MS-SSIM pour un bpp donné, démontrant sa capacité à préserver une qualité visuelle optimale tout en réduisant significativement la taille des données. Un exemple illustratif de ces courbes est présenté dans la Figure 3.3, mettant en évidence les différences de performances entre plusieurs approches.

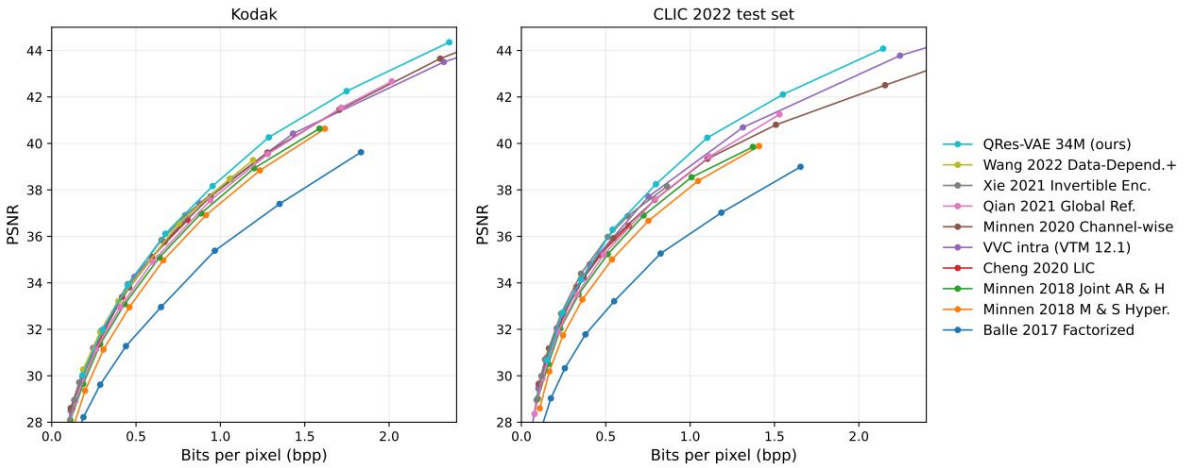


FIGURE 3.3 – Exemple de courbes taux-distorsion. Les méthodes incluent QRes-VAE [Duan et al., 2023], Wang 2022 Data-Depend.+ [Wang et al., 2022a], Xie 2021 Invertible Enc. [Xie et al., 2021], Qian 2021 Global Ref. [Qian et al., 2021], Minnen 2020 Channel-wise [Minnen and Singh, 2020], VVC intro (VTM 12.1), Cheng 2020 LIC [Cheng et al., 2020a], Minnen 2018 Joint AR & H [Minnen et al., 2018a], Minnen 2018 M & S Hyper [Ballé et al., 2018] et Balle 2017 Factorized [Ballé et al., 2016] (source [Duan et al., 2023]).

Les datasets utilisés pour l'apprentissage et l'évaluation jouent un rôle crucial dans ces comparaisons, car ils influencent directement les capacités généralisantes des modèles. Parmi les datasets couramment employés figurent Kodak [Kod,], Tecknick [Asuni and Giachetti, 2014], CLIC (Challenge on Learned Image Compression) [CLI,], DIV2K [Agustsson and Timofte, 2017], et Open Images [Kuznetsova et al., 2020]. Ces ensembles de données offrent une variété de résolutions et de contenus visuels adaptés à différents scénarios. Par exemple, Kodak, composé de 24 images au format PNG (non compressées) de dimensions $3\times512\times768$, est souvent utilisé comme benchmark en raison de sa taille modérée et de la diversité des scènes proposées. DIV2K, en revanche, est apprécié pour ses images haute résolution, qui permettent de mieux évaluer la capacité des modèles à préserver les détails fins après compression. Pour l'entraînement, des datasets plus larges tels que Flickr2K [Fli,] ou COCO [Lin et al., 2014] sont souvent utilisés pour garantir une couverture étendue des scénarios visuels. Ces courbes de performance, associées à des benchmarks standardisés, offrent un cadre d'évaluation objectif et comparatif des modèles de compression. Elles permettent de mettre en évidence leurs forces et leurs faiblesses, facilitant ainsi leur optimisation et leur adoption dans des applications pratiques exigeantes.

On remarquera dans la Figure 3.3 que pour de faibles débits, en dessous de 1.0 bpp, les performances des différents modèles semblent tendre à se stabiliser, atteignant une sorte de "plafond de verre". En d'autres termes, les gains en termes de qualité de reconstruction deviennent marginaux, même en augmentant la complexité du modèle. Dans ce contexte, les temps d'inférence pour l'encodage et le décodage prennent une importance cruciale et deviennent le critère clé de sélection des modèles.

3.2.3 Modèles Entropiques

Une grande quantité de recherches porte sur les modèles d'entropie pour la compression d'images apprise. Un codage entropique idéal nécessite une estimation précise de la distribution jointe des éléments dans les représentations latentes $\hat{y}$. Dans les premiers travaux, ces éléments sont supposés indépendamment distribués pour simplifier la conception. Cependant, même avec des encodage $\mathcal{E}$ optimisés, il reste difficile d'éliminer la redondance spatiale dans les cartes latentes des images. Par conséquent, une variété de modèles d'entropie sont proposés pour réduire davantage la redondance dans le code latent. Ces méthodes incluent l'analyse statistique sur un ensemble de données, la prédiction ou l'analyse contextuelle, et l'utilisation d'un hyperprior appris pour la modélisation d'entropie. Le modèle d'entropie fournit une estimation de la probabilité pour tous les éléments, et l'espérance des log-vraisemblances constitue la limite des débits binaires pour encoder ces éléments. Dans la plupart des travaux, un codage arithmétique est utilisé pour encoder de manière quasi-sans-perte les symboles des représentations latentes. Chacune de ces approches apporte des avantages spécifiques dans la réduction de la redondance et l'amélioration de l'efficacité du codage :

— **Modèle entropique Factorisé (*Factorized Model*)** : Le modèle Factorisé estime la probabilité de chaque élément du code latent indépendamment des autres. Étant donné un code latent y, composé de N élé-

ments $y = (y_1, ..., y_N)$, le modèle factorisé suppose une distribution indépendante pour chaque y_i. La distribution de probabilité jointe $p(y)$ est factorisée comme suit :

$$p(y) = \prod_{i=1}^{N} p(y_i) \tag{3.12}$$

où $p(y_i)$ est estimé par un modèle paramétrique, par exemple une distribution gaussienne ou une autre distribution paramétrique. Les paramètres du modèle $p(y_i) = f(y_i; \theta)$ sont ajustés en fonction des données d'entraînement.

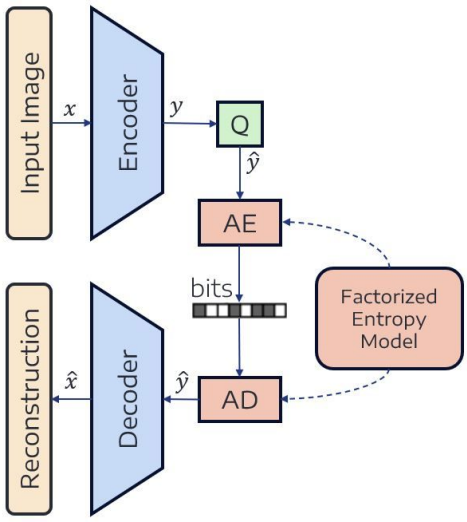


FIGURE 3.4 – Modèle entropique Factorisé. Q représente la quantification, et AE et AD représentent respectivement l'encodeur arithmétique et le décodeur arithmétique.

Le modèle factorisé est facile à implémenter et rapide à calculer, ce qui le rend adapté aux systèmes à faible latence ou aux environnements avec des ressources limitées. Cependant, en ignorant les dépendances spatiales entre les éléments latents, il perd en précision dans l'estimation des probabilités et ne capture pas les redondances spatiales, ce qui peut augmenter le débit binaire et limiter son efficacité pour la compression de haute qualité.

— **Modèle entropique *Hyperprior*** : Le modèle Hyperprior introduit une variable latente auxiliaire z pour capturer des informations statistiques supplémentaires sur le code latent y. L'idée est que z (l'hyper-latent) contient des informations globales ou contextuelles sur y, et est utilisé pour affiner la distribution de y. La distribution jointe de y et z est donnée par :

$$p(y, z) = p(z) \cdot p(y|z). \tag{3.13}$$

Pour encoder cette information, on utilise un Hyper-Encoder, noté $h_{\mathcal{E}}$, qui transforme y en z :

$$z = h_{\mathcal{E}}(y; \theta_{h_{\mathcal{E}}}), \tag{3.14}$$

où $\theta_{h_{\mathcal{E}}}$ représente les paramètres du Hyper-Encoder. Ce réseau capture les statistiques globales de y et produit z, une représentation comprimée qui facilite la modélisation de y. Ensuite, z est lui-même compressé et transmis, puis utilisé pour décoder y. Un Hyper-Decoder, noté h_D, est utilisé pour transformer z en les paramètres de la distribution conditionnelle de y. La distribution conditionnelle de y est généralement paramétrée par $\mu(z)$ et $\sigma(z)$, c'est-à-dire une loi normale, calculée par l'Hyper-Decoder :

$$p(y|z) = \prod_{i=1}^{N} \mathcal{N}\left(\mu_i(z), \sigma_i(z)\right)(y_i) \quad \text{avec} \quad \mu(z), \sigma(z) = h_D(z; \theta_{h_D}), \tag{3.15}$$

et θ_{h_D} sont les paramètres du Hyper-Decoder. En utilisant z, le modèle affine ainsi l'estimation de $p(y)$ avec des informations additionnelles sur les dépendances et redondances au sein de y. Le modèle est

entrainé "end-to-end" en adaptant la fonction de perte de la manière suivante :

$$L = R + \lambda D = \mathbb{E}_{x\sim p_x}[-\log_2 p_{\hat{y}|\hat{z}}(\hat{y}|\hat{z})] + \mathbb{E}_{x\sim p_x}[-\log_2 p_{\hat{z}}(\hat{z})] + \lambda D \tag{3.16}$$

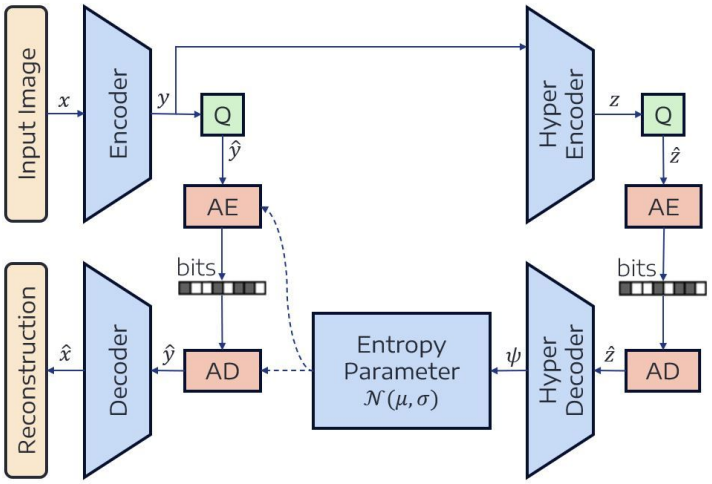


FIGURE 3.5 – Modèle entropique Hyperprior. Q représente la quantification, et AE et AD représentent respectivement l'encodeur arithmétique et le décodeur arithmétique.

L'utilisation d'un hyperprior z permet de capturer des informations globales qui améliorent l'estimation de la distribution de y, modélisant mieux les variations latentes et réduisant ainsi le débit binaire nécessaire pour la compression. Bien qu'il soit souvent plus précis que le modèle factorisé, le modèle hyperprior est plus complexe et demande davantage de ressources, car la transmission des paramètres de z dans le flux binaire augmente la taille du modèle et la latence, ce qui peut poser des problèmes pour les applications en temps réel ou sur des appareils aux capacités limitées.

— **Modèle entropique Contextuel (*Context Model*)** : Le modèle Contextuel prend en compte la dépendance spatiale des éléments dans y en utilisant des éléments déjà encodés pour estimer les probabilités des éléments restants. Pour un élément y_i, le modèle contextuel utilise une fonction de probabilité conditionnelle basée sur les éléments $y_{<i}$ déjà encodés (ceux précédant y_i dans un certain ordre, comme l'ordre raster pour les pixels). La probabilité jointe est alors exprimée comme :

$$p(y) = \prod_{i=1}^{N} p(y_i|y_{<i}), \tag{3.17}$$

où $p(y_i|y_{<i})$ est la probabilité de y_i étant donné les valeurs de y précédentes. Cette probabilité conditionnelle est souvent modélisée par un réseau convolutionnel masqué ou un modèle autoregressif qui permet de prédire y_i à partir des valeurs adjacentes.

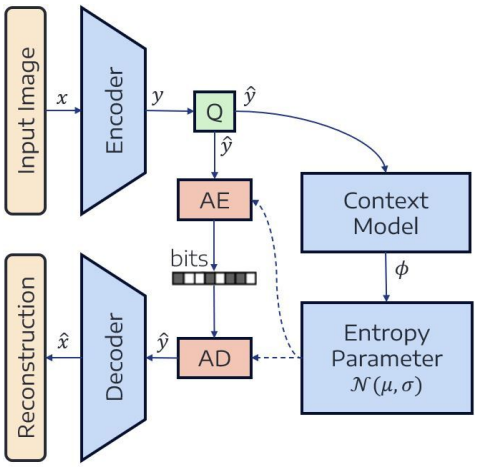


FIGURE 3.6 – Modèle entropique Contextuel. Q représente la quantification, et AE et AD représentent respectivement l'encodeur arithmétique et le décodeur arithmétique.

Le modèle contextuel utilise les éléments encodés précédemment pour prédire les suivants, capturant ainsi les dépendances spatiales et réduisant les redondances pour une compression plus efficace. Il est particulièrement adapté aux images présentant des corrélations spatiales, permettant un encodage précis en fonction du contexte immédiat. Cependant, il est coûteux en calculs, car il doit estimer des probabilités conditionnelles, ce qui peut ralentir le codage et le décodage. Son implémentation requiert des mécanismes complexes, tels que des réseaux convolutionnels masqués ou des architectures autoregressives, augmentant ainsi la complexité du modèle.

— **Modèle entropique Hybride** : Dans certaines architectures, le Hyperprior et le Context Model sont combinés pour obtenir des estimations encore plus précises. Dans ce cas, la probabilité de y est donnée par :

$$p(y|z) = \prod_{i=1}^{N} p(y_i|y_{<i}, z), \tag{3.18}$$

où z fournit des informations globales, tandis que $y_{<i}$ apporte un contexte spatial local pour chaque y_i. Ces modèles permettent d'améliorer la précision de la compression en capturant différents types de dépendances dans les données.

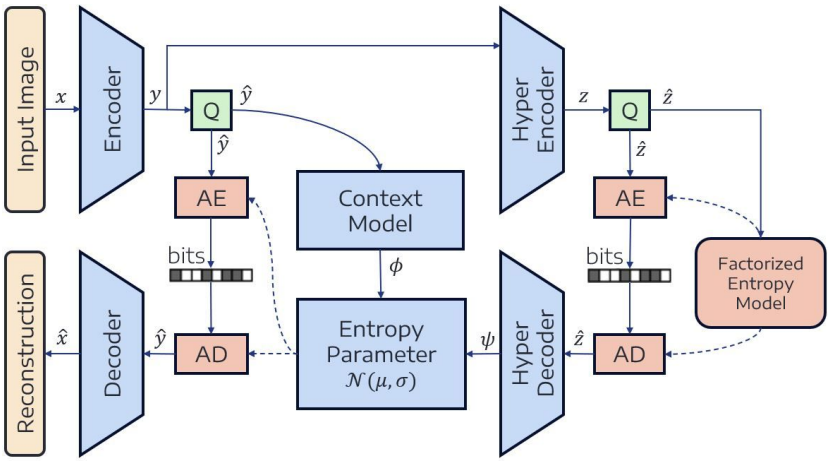


FIGURE 3.7 – Modèle entropique Hybride. Q représente la quantification, et AE et AD représentent respectivement l'encodeur arithmétique et le décodeur arithmétique (source [Minnen et al., 2018b]).

Bien que cette approche minimise le débit binaire et améliore la qualité de reconstruction, elle est très coûteuse en calcul et en mémoire, en raison de la gestion simultanée des informations globales et

locales. La complexité et le besoin d'encoder et de transmettre z peuvent limiter son utilisation dans des applications aux ressources ou aux délais restreints.

En résumé, les méthodes existantes visent à fournir un encodage flexible et un modèle d'entropie précis, tous deux basés sur des réseaux neuronaux et entraînables de bout en bout. Le choix du modèle dépendra du compromis entre la précision de compression, la taille du modèle, et le temps de calcul selon les besoins de l'application.

3.2.4 Approches basées sur les auto-encodeurs

Les auto-encodeurs, introduits pour la compression d'image, exploitent des réseaux neuronaux afin de réduire la dimensionnalité tout en préservant la qualité visuelle des images. La compression par auto-encodeurs s'appuie sur un processus d'encodage et de décodage : un encodeur réduit les informations de l'image en une représentation compacte, tandis qu'un décodeur reconstruit une version approchée de l'image initiale. Cette technique, qui comprend des variantes telles que les auto-encodeurs variationnels, a été optimisée dans plusieurs travaux pour obtenir de meilleures performances en matière de compression.

Par exemple, en 2017, [Ballé et al., 2017] introduit une méthode de compression d'image basée sur un autoencodeur variationnel. Cette approche utilise une transformation d'analyse non linéaire, un quantificateur uniforme et une transformation de synthèse non linéaire. La compression et la décompression sont réalisées à travers trois couches successives de filtres convolutifs, associées à des fonctions d'activation non linéaires. Grâce à une optimisation conjointe de l'ensemble du modèle en utilisant une variante de la descente de gradient stochastique, cette méthode améliore la performance de compression en termes de compromis taux-distorsion. En 2018, la méthode [Cheng et al., 2018] présente une architecture de compression d'images avec pertes utilisant un auto-encodeur convolutionnel (CAE) pour améliorer l'efficacité de codage. Premièrement, une architecture CAE novatrice remplace les transformations conventionnelles et est entraînée à l'aide d'une fonction de perte de débit-distorsion. Deuxièmement, pour une représentation plus compacte en énergie, une analyse en composantes principales (PCA) est appliquée aux cartes de caractéristiques produites par le CAE, suivie d'une quantification et d'un codage entropique pour générer les codes. Le modèle est conçu pour des images YCbCr et utilise le codeur entropique JPEG-2000 pour décomposer le latent en plans de bits, avec un codeur arithmétique binaire adaptatif. Ce dernier surpasse les performances de JPEG-2000. Dans la même année, la publication [Zhou et al., 2018] présente un cadre de compression conçu pour les faibles débits binaires. Basé sur un autoencodeur variationnel, le modèle comprend une transformation non linéaire de l'encodeur, un quantificateur uniforme, une transformation non linéaire du décodeur, et un module de post-traitement. La probabilité a priori de la représentation compressée est modélisée par une distribution de Laplace, en utilisant un hyperprior auto-encodeur qui est entraîné conjointement avec l'auto-encodeur de transformation. Chaque caractéristique latente y est modélisée selon une distribution de Laplace centrée en zéro avec une déviation standard σ_i de la forme $p_{y|z}(y|z) = \prod_i \text{Laplacian}(0, \sigma^2)(y_i)$ (similaire à l'équation 3.15 mais avec une distribution Laplacienne) Le module de post-traitement, basé sur des convolutions, permet de réduire davantage les distorsions, atténuant ainsi les artefacts et flous causés par la compression à faible débit. De manière similaire, [Ballé et al., 2018] introduit un hyperprior appelé "*Scale Hyperprior*" dans le cadre des auto-encodeurs variationnels. Ce *Scale Hyperprior* permet de capturer efficacement les dépendances spatiales dans la représentation latente en modélisant le latent selon une loi normale centrée en 0, c'est-à-dire $p_{y|z}(y|z) = \prod_i \mathcal{N}(0, \sigma^2)(y)$. Cela offre une compression plus efficace et des images reconstruites de meilleure qualité. La méthode est optimisée conjointement avec l'auto-encodeur sous-jacent, renforçant ainsi le compromis entre le taux de compression et la qualité de l'image (cf. Figure 3.8).

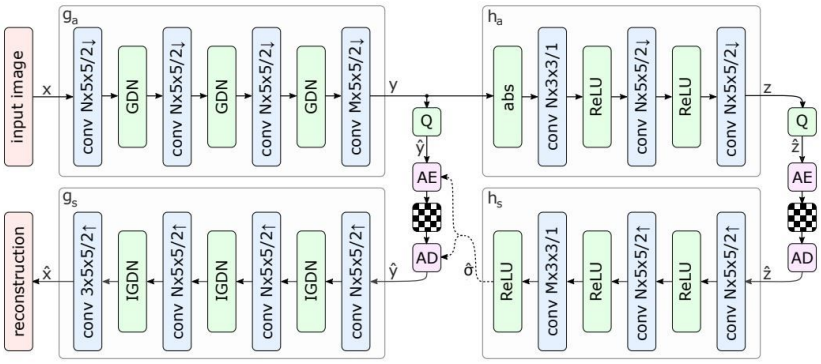


FIGURE 3.8 – Architecture du réseau du modèle Hyperprior. Comme décrit dans la Section 3.2.3, le côté gauche montre une architecture d'auto-encodeur d'image, et le côté droit correspond à l'auto-encodeur implémentant l'hyperprior. Q représente la quantification, et AE et AD représentent respectivement l'encodeur arithmétique et le décodeur arithmétique (source [Ballé et al., 2018]).

Pour le modèle entropique, on peut aussi considérer une mixture de plusieurs types de distribution, comme le fait [Fu et al., 2023]. En effet, l'article propose un modèle de mixture plus flexible, nommé *Gaussian-Laplacian-Logistic Mixture Model* (GLLMM), pour les représentations latentes. Ce modèle s'adapte avec plus de précision et d'efficacité aux différents contenus d'images et aux diverses régions d'une même image, tout en maintenant une complexité constante. Concernant la conception du réseau d'encodage/décodage, l'article introduit également des blocs résiduels concaténés (*Concatenated Residual Blocks*, CRB), où plusieurs blocs résiduels sont connectés en série avec des connexions de raccourci additionnelles, optimisant ainsi la capacité du modèle sans alourdir sa complexité. Enfin, [Brummer and De Vleeschouwer, 2021] propose un schéma de compression utilisant un unique auto-encodeur convolutionnel et plusieurs distributions a priori apprises, fonctionnant comme une compétition d'experts pour offrir une performance de compromis débit-distorsion comparable à celle obtenue avec un prior paramétré, mais avec une complexité de codage et de décodage d'entropie réduite. Cette approche utilise le principe de compétition d'experts pour minimiser le coût en bits de chaque localisation spatiale dans la représentation latente quantifiée. Pendant l'entraînement, seule la distribution prioritaire optimale est mise à jour à chaque emplacement spatial, ce qui améliore la spécialisation des distributions. Les tables de la fonction de distribution cumulative (CDF) sont fixées à la fin de l'entraînement, permettant ainsi, lors du test, d'assigner la table CDF au plus faible coût binaire pour chaque localisation spatiale. Le compromis débit-distorsion obtenu est similaire à celui des distributions paramétrées, tandis que le codage d'entropie est grandement simplifié en s'appuyant sur des tables de consultation (LUT) au lieu de calculs analytiques (cf. Figure 3.9).

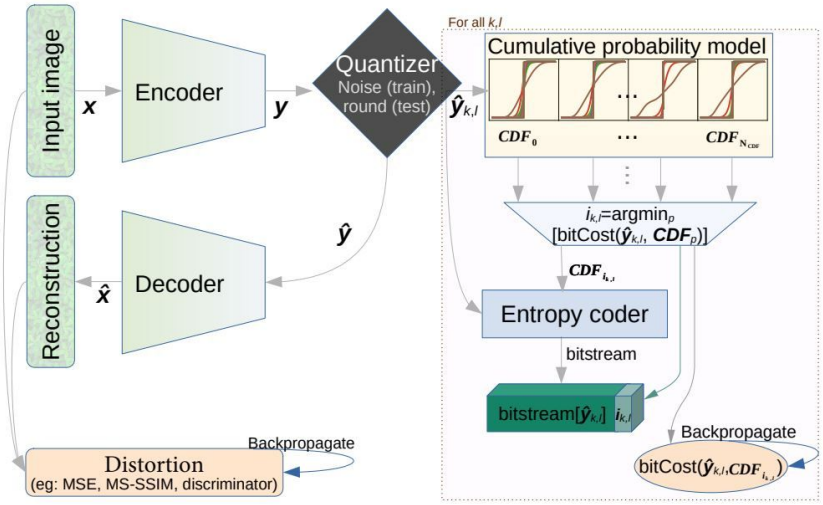


FIGURE 3.9 – Schéma de compression AE avec concurrence des distributions a priori. Les indices i désignent les indices des tables CDF qui minimisent le nombre de bits pour chaque dimension spatiale latente. L'entrainement se fait selon Loss = Distortion + λbitCost (source [Brummer and De Vleeschouwer, 2021]).

Les méthodes de compression d'images basées sur les autoencodeurs offrent des avantages notables, notamment une capacité à produire des représentations compressées plus compactes et de meilleure qualité visuelle par rapport aux méthodes traditionnelles telles que JPEG 2000. Grâce aux transformations non linéaires, à l'optimisation conjointe pour le taux-distorsion et à l'incorporation de distributions a priori, ces modèles parviennent à capturer les détails et les dépendances spatiales de manière plus efficace. Toutefois, leurs performances demeurent encore assez moyennes comparées aux compresseurs conventionnels.

3.2.5 Approches basées sur les modèles de contexte

Les modèles de contexte en compression d'images jouent un rôle crucial dans l'optimisation de la performance de codage. Comme décrit dans la Section 3.2.3 (cf. Figure 3.6), les modèles exploitent les dépendances spatiales et canaux dans les représentations latentes pour améliorer l'efficacité de codage. Le contexte, ici, fait référence aux informations tirées des pixels ou des caractéristiques voisines qui permettent de mieux prédire les valeurs de la représentation latente compressée, réduisant ainsi le nombre de bits nécessaires pour décrire ces valeurs.

Par exemple, [Lee et al., 2018] propose un modèle de contexte adaptatif pour la compression d'image optimisée de bout en bout. Ce modèle exploite deux types de contextes : les contextes consommant des bits et ceux qui n'en consomment pas, en fonction de l'allocation de bits nécessaire. En s'appuyant sur ces contextes, le modèle parvient à estimer plus précisément la distribution de chaque représentation latente, grâce à une forme généralisée des modèles d'approximation. Cette approche améliore ainsi les performances de compression, surpassant les codecs traditionnels comme BPG et JPEG 2000 dans les tests expérimentaux. D'autres approches ont exploré différents types de prior, comme dans [Minnen et al., 2018b], qui examine les priors autorégressifs, hiérarchiques, ainsi que leurs combinaisons, permettant d'améliorer la précision de codage en intégrant des informations de contexte à plusieurs échelles. Il propose un modèle qui combine modèles entropiques contextuels et hyperprior donnant des performances supérieures à JPEG-2000 et WebP. L'article [Minnen and Singh, 2020] propose deux améliorations pour la compression d'image : le conditionnement par canal (*channel-conditioning*) et la prédiction des résidus latents (*latent residual prediction*). Ces techniques permettent d'obtenir de meilleures performances taux-distorsion par rapport aux modèles de contexte adaptatifs existants tout en réduisant le traitement séquentiel, souvent responsable de la lenteur du décodage. L'architecture développée, hautement parallélisable, parvient à surpasser les modèles adaptatifs au contexte récent et montre des performances supérieures au codec BPG. Le modèle *Cross Channel Context* [Ma et al., 2021] propose une nouvelle organisation des latents en groupes de canaux, traités séquentiellement pour tirer parti des dépendances entre canaux. Associé à un modèle de prior autorégressif et hiérarchique, ce modèle améliore significativement la précision de la prédiction tout en optimisant le débit de bits. Pour répondre aux limites de traitement spatial, le modèle *Checkerboard Context Model* [He et al., 2021] réorganise l'ordre de décodage des pixels en utilisant des convolutions masquées (**Masked Convolution**), offrant une approche parallélisable sans sacrifier la précision du modèle de contexte, ce qui représente une avancée importante pour les applications nécessitant un traitement rapide (cf. Figure 3.10).

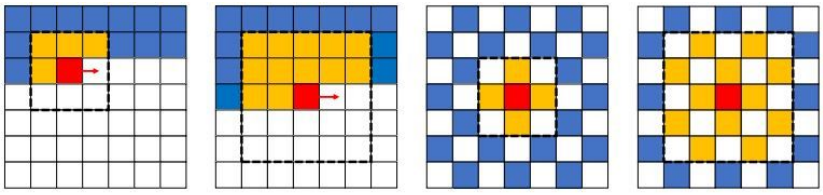


FIGURE 3.10 – Convolutions masquées modélisant le contexte causal spatial. Les blocs rouges représentent les éléments à encoder/décoder. Les latents aux emplacements jaunes et bleus sont actuellement visibles (ils sont tous visibles pendant l'encodage, et ceux qui ont été décodés sont visibles pendant le décodage). Une modélisation de contexte peut être effectuée en utilisant une convolution masquée centrée sur l'emplacement rouge et convoluant uniquement avec les latents jaunes. (a)(b) Instances 3×3 et 5×5 du modèle de contexte en série largement utilisé dans la littérature. Ce modèle limite l'efficacité en termes de calcul. (c)(d) Modèle de contexte en damier proposé avec une taille de noyau de 3×3 et 5×5. Après le décodage de tous les éléments d'ancrage (bleus et jaunes), le calcul du contexte pour tous les non-ancrages peut être exécuté en parallèle (source [He et al., 2021]).

En 2022, le modèle *Causal Contextual Prediction* [Guo et al., 2022] introduit une approche causale qui sépare les latents en différents canaux et utilise des relations entre canaux pour créer des contextes adjacents informatifs. En plus, un modèle global de prédiction causale permet de trouver des points de référence pour des prédictions plus précises. Le modèle CCCMGLF (*Cross Channel Context Model and Generalized Loop Filter*) [Ma et al., 2022a] améliore encore les performances en intégrant un filtre en boucle généralisé, optimisé conjointement avec le modèle de contexte, réduisant ainsi davantage les artefacts de compression. L'article [Shizhan, 2023] propose une approche novatrice de compression d'image appelée *Merge-Then-Estimate* (MTE). Ce modèle traite les problèmes liés à la modélisation du contexte spatial et à la fusion inadéquate des sources a priori. Il fusionne d'abord uniformément les informations a priori disponibles en un "token de prior" grâce à un module d'intégration pour chaque position spatiale dans la représentation latente quantifiée. De plus, il introduit un modèle de contexte adaptatif au contenu (*Content-aware Context Model*) pour capturer dynamiquement les dépendances de la représentation en cours de codage avec les a priori voisins. L'article [Wang et al., 2023b] présente des améliorations en efficacité de compression grâce à la technique ELIC. En introduisant une propriété de compactage de l'information comme biais inductif, cette approche renforce le modèle d'entropie adaptatif de manière conditionnelle par canal. En le combinant avec un modèle de contexte spatial, le modèle SCCTX (*Space-Channel ConTeXt*) est proposé pour estimer l'entropie de manière multidimensionnelle, réduisant efficacement le débit binaire. Le modèle ELIC, qui utilise des blocs résiduels empilés pour la transformation non linéaire, dépasse les performances du codec VVC en termes de PSNR et MSSSIM. Pour capturer les dépendances globales et locales de manière équilibrée, MIXLIC (*Mixing Global and Local Context Model*) [Ruan et al., 2023b] propose en 2023 un module combinant un extracteur de contexte global et un extracteur local en parallèle, capturant ainsi des dépendances de contexte à des échelles variées. En 2023 également, le modèle *Context-Based Trit-Plane Coding* [Jeon et al., 2023] propose un module de réduction de taux basé sur le contexte pour estimer avec précision les probabilités des éléments latents en utilisant un codage par trit-planes. Ce module exprime le tenseur latent centré et quantifié dans un système numérique ternaire, par le biais d'un module de découpage en trit-planes, avant de l'encoder en binaire. Un schéma de réentraînement du décodeur est également proposé pour optimiser le compromis entre le taux et la distorsion, améliorant ainsi l'efficacité de la compression progressive. Enfin, [Sui et al., 2024] introduit une approche de prédiction du contexte progressif du coin vers le centre, en utilisant un module d'attention croisée à longue portée pour capturer les informations sémantiques distantes dans l'encodage et le décodage. Cette architecture permet de mieux appréhender les dépendances globales et d'améliorer la performance en termes de taux-distorsion.

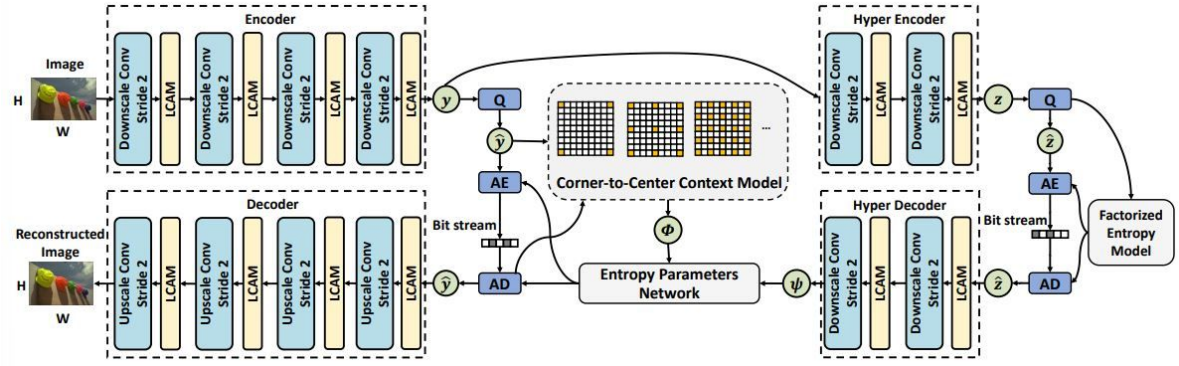


FIGURE 3.11 – Pipeline de compression d'image appris proposé par [Sui et al., 2024].

Les modèles entropiques contextuels apportent une amélioration significative du compromis entre compression et distorsion en tirant parti des dépendances spatiales et de canal dans les représentations latentes. Cependant, cette précision accrue s'accompagne souvent d'une augmentation du temps de traitement, car les modèles contextuels nécessitent généralement des calculs séquentiels complexes. Cette limitation pose un défi pour leur intégration dans des applications en temps réel, où la rapidité de traitement est cruciale.

3.2.6 Approches multi-échelles

Les méthodes de compression multi-échelle (ou "*multi-scale*") visent à exploiter les informations présentes dans les images à différentes résolutions ou niveaux de détail. Le concept repose sur le principe que les caractéristiques visuelles importantes d'une image, comme les contours ou les textures, apparaissent différemment

selon l'échelle d'observation. En exploitant cette variabilité, les méthodes multi-échelle peuvent mieux comprimer les images en allouant des bits de manière plus efficace, en fonction des niveaux de détail nécessaires pour une qualité visuelle optimale.

En 2017, l'article [Rippel and Bourdev, 2017] présente une architecture auto-encodeur avec une analyse pyramidale pour extraire les caractéristiques à plusieurs échelles. Cette approche inclut un module de codage adaptatif et une régularisation de la longueur attendue des codes, ce qui permet une compression en temps réel adaptée aux caractéristiques de l'image. L'article [Hu et al., 2020] présente un cadre de compression d'image basé sur des hyper-priors hiérarchiques pour analyser l'image en profondeur et réduire plus efficacement la redondance spatiale, améliorant ainsi le compromis taux-distorsion. Ce cadre utilise une hyper-représentation multi-couche pour modéliser les dépendances spatiales sans modèle de contexte élémentaire, capturant les redondances dans un contexte plus large pour une estimation plus précise de l'entropie (cf. Figure 3.12). Le *Signal-Preserving Hyper Transform*, conçu pour préserver les informations tout en réduisant les redondances spatiales, permet de construire un modèle hyper-prior multi-couche. En complément, un sous-réseau d'agrégation d'informations exploite les informations encodées des différentes couches, fournissant des hyper-latents comme information auxiliaire pour améliorer la qualité de reconstruction.

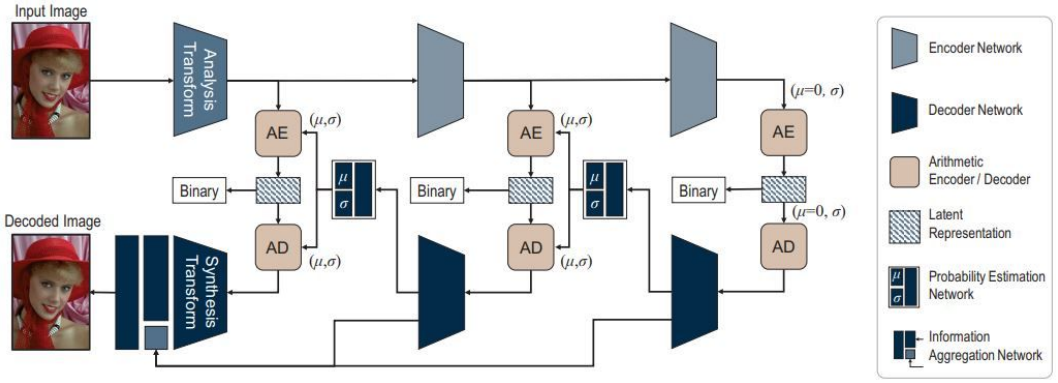


FIGURE 3.12 – Architecture proposée par [Hu et al., 2020]. La distribution de probabilité de la couche la plus interne de l'hyperprior est approximée par une distribution gaussienne à moyenne nulle, où les valeurs d'échelle σ sont des paramètres entraînables indépendants par canal et partagés spatialement.

Une autre approche en 2020, iWave [Ma et al., 2020], adapte les principes de la transformation en ondelettes pour la compression d'images. Bien que les ondelettes soient un outil puissant pour l'analyse multi résolution, les images naturelles ne répondent pas toujours aux hypothèses théoriques des signaux idéaux (par exemple, la continuité ou la régularité des variations locales). iWave propose donc une transformation en ondelettes inspirée par un schéma de levée (*lifting scheme*), où le filtre de prédiction est une CNN entraînée, permettant une transformation multi-échelle mieux adaptée aux images naturelles. En 2020, iWave est étendu par iWave++ [Ma et al., 2022b], un schéma de compression versatile optimisé de bout en bout. iWave++ permet une compression sans perte ou avec pertes, en ajustant simplement l'échelle de quantification, et propose un moteur de codage entropique progressif et un module de déquantification pour la compression avec pertes. L'évolution de cette famille se poursuit en 2023 avec iWavePro [Xue et al., 2023a], qui améliore iWave++ en intégrant un codage entropique mixture de gaussiennes, une transformation affine de type ondelettes, et un entraînement en ligne. Ce modèle avancé améliore la précision de compression tout en assurant une exécution efficace sur les architectures matérielles modernes.

En 2021, l'article [Gao et al., 2021] propose une méthode innovante de rétroprojection avec fusion d'attributs multi-échelle et attentionnelle pour renforcer la puissance de représentation. Cette méthode de rétroprojection réajuste l'estimation actuelle en établissant des connexions de rétroaction entre les attributs de haut et bas niveau de manière attentionnelle et discriminative. L'approche décompose l'image d'entrée pour traiter séparément les différentes composantes fréquentielles, puis réassemble les latents dérivés grâce à un module de double attention, permettant de manipuler explicitement les détails dans les régions d'intérêt. Un nouveau schéma d'entraînement est également proposé pour réduire le résidu de quantification latent. Cette méthode a démontré des performances supérieures au codec VVC. En 2022, l'approche [Liu et al., 2022b] propose de fusionner les informations spatiales et contextuelles à plusieurs échelles, permettant de reconstruire des images compressées de manière plus fidèle à l'original, et ainsi d'augmenter la qualité visuelle. Enfin, récemment, le modèle QRes-

VAE [Duan et al., 2023] réinvente les VAE basés sur ResNet pour la compression avec pertes. En adaptant les variables latentes avec une quantification et un codage entropique efficace, ce modèle propose un schéma de compression de type "coarse-to-fine" (du plus grossier au plus détaillé) qui permet un encodage et un décodage parallèles, accélérant ainsi le processus sur les GPU et surpassant les méthodes antérieures en termes de compression d'images naturelles avec pertes (cf. Figure 3.13).

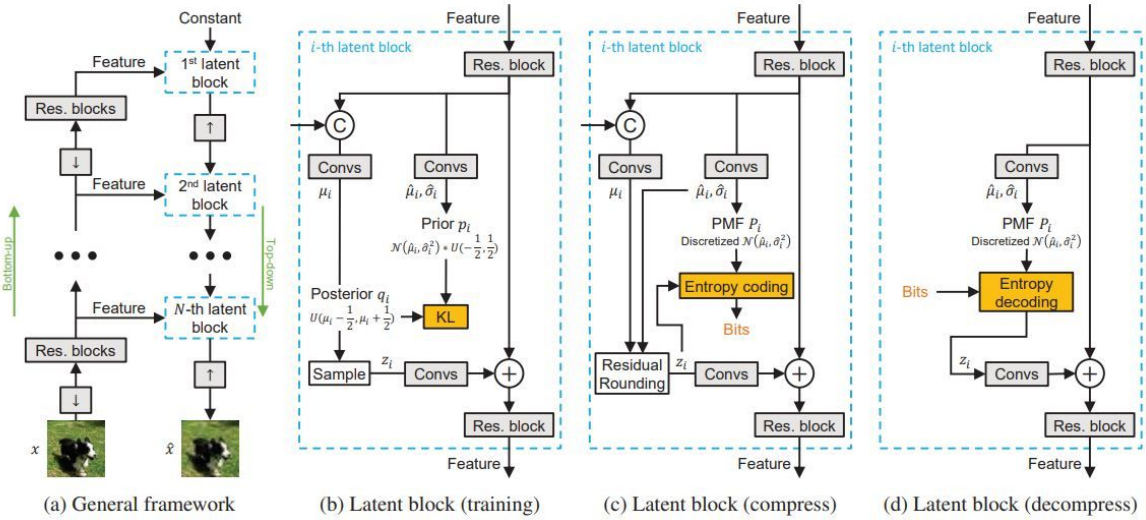


FIGURE 3.13 – Vue d'ensemble de QRes-VAE. En (a), le cadre général des ResNet VAEs. Les blocs latents se comportent différemment pour l'entraînement, la compression et la décompression, illustrés en (b), (c) et (d) respectivement (source [Duan et al., 2023]).

Les méthodes multi-échelle offrent une grande flexibilité pour les systèmes de compression d'images. En adaptant l'architecture des réseaux et les schémas de codage à chaque niveau de détail, ces approches permettent de concilier la qualité visuelle et l'efficacité de compression, tout en offrant des solutions adaptées aux contraintes de temps réel et aux applications nécessitant des taux de compression variables.

3.2.7 Approches basées sur l'attention et le contenu

Les méthodes de compression d'images basées sur l'attention et le contenu se concentrent sur l'allocation adaptative des ressources de codage en fonction de l'importance des différentes parties d'une image. Le concept clé derrière ces méthodes est d'analyser le contenu de l'image pour identifier les zones critiques (ex : les détails et les bords) et de leur accorder une priorité de codage, afin de réduire la distorsion tout en maintenant un débit de bits minimal. Cela implique souvent l'utilisation de mécanismes d'attention pour concentrer les ressources de codage sur les zones les plus importantes de l'image, ainsi que de modèles basés sur le contenu pour ajuster dynamiquement le taux de compression en fonction de la complexité locale.

En 2018, [Mentzer et al., 2018] propose de modéliser directement l'entropie de la représentation latente en utilisant un modèle de contexte : un CNN 3D qui apprend un modèle de probabilité conditionnelle de la distribution latente de l'autoencodeur. Cette méthode utilise les modèles de contexte, traditionnellement employés pour améliorer les taux de codage, comme terme d'entropie dans l'optimisation. La méthode est généralisée pour des réseaux spatialement adaptatifs, avec une carte d'importance permettant de concentrer le débit binaire sur les régions essentielles de l'image compressée. La même année, [Li et al., 2018] introduit une approche de compression pondérée par le contenu. Ici, le taux de bits alloué varie en fonction du contenu local de l'image. Une carte d'importance pondérée par le contenu guide l'allocation de bits, où la somme de cette carte d'importance sert d'estimation continue de l'entropie pour contrôler le taux de compression (cf. Figure 3.14). Cette approche permet un codage plus précis, en adaptant dynamiquement le niveau de compression aux zones les plus significatives de l'image.

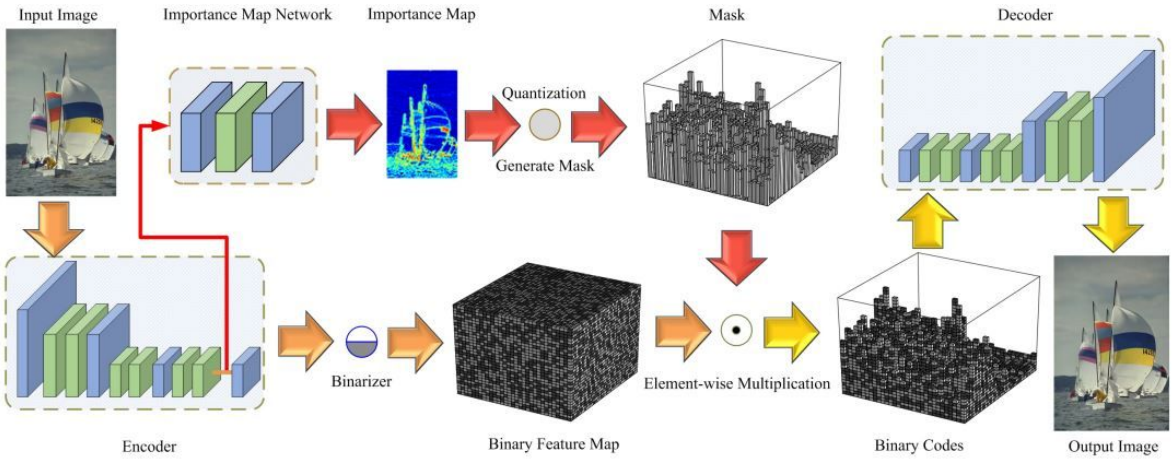


FIGURE 3.14 – Illustration de l'architecture CNN pour la compression d'image pondérée par le contenu proposé par [Li et al., 2018].

En 2020, [Cheng et al., 2020b] propose une modélisation plus flexible de l'entropie des codes latents en utilisant des distributions de type mixture de gaussiennes discrétisées. Cette technique, associée à des modules d'attention, permet d'obtenir des modèles d'entropie plus précis pour les représentations latentes de l'auto-encodeur, ce qui améliore la performance en termes de débit-distorsion. L'article [Chen et al., 2021] s'appuie sur un auto-encodeur variationnel (VAE) optimisé pour la compression avec pertes, intégrant une attention non-locale et une modélisation contextuelle améliorée. Ce modèle, nommé NLAIC, optimise l'utilisation de l'attention non-locale pour capter des relations à longue distance dans l'image, ce qui permet d'allouer efficacement des bits aux zones de l'image les plus importantes. En 2022, [Wang et al., 2022b] propose un module d'attention axé sur la réduction de l'entropie (*information entropy reduction attention*, IERA), qui diminue l'entropie des cartes de caractéristiques lors du codage et du décodage, pour une meilleure efficacité de compression. Ce module d'IERA améliore l'efficacité de la compression en minimisant la quantité d'information redondante dans les caractéristiques d'image encodées. La même année, [Wang and Wang, 2022] propose un hyperprior basé sur une mixture gaussienne discrétisée pour modéliser plus précisément la distribution des codes latents et ainsi améliorer l'estimation du taux de compression. Un module de masquage est également introduit pour renforcer la capacité d'extraction de caractéristiques de l'encodeur et allouer les débits binaires de manière adaptative. De plus, un module d'attention et un bloc résiduel sont intégrés pour optimiser la performance de compression. Enfin, [Tang et al., 2023] introduit une méthode de compression d'images intégrant l'attention par graphe et un réseau de neurones convolutifs asymétrique (ACNN). L'ACNN renforce les caractéristiques locales clés et réduit le coût d'entraînement du modèle, tandis que l'attention par graphe permet de surmonter les limitations des CNN pour modéliser les dépendances à longue portée. Cette approche d'auto-attention régule la distribution de bits, ce qui permet une allocation plus rationnelle et une compression améliorée (cf. Figure 3.15).

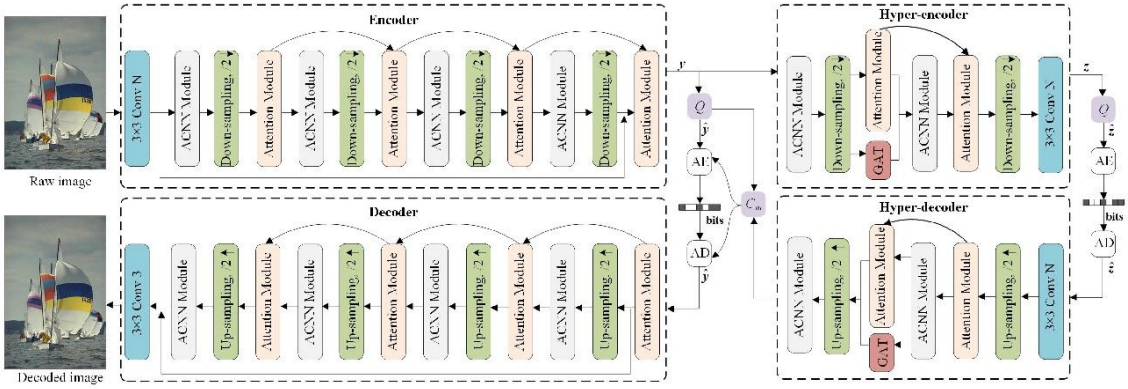


FIGURE 3.15 – Vue d'ensemble du modèle proposé par [Tang et al., 2023].

Ces méthodes de compression par attention et basées sur le contenu montrent une avancée significative

dans l'optimisation de la qualité visuelle des images compressées. Elles permettent une allocation de bits plus intelligente en adaptant les ressources de compression aux régions d'importance de l'image, et, en intégrant l'attention, elles maximisent l'efficacité de codage tout en minimisant la perte de qualité dans les détails visuels critiques.

3.2.8 Approches basées sur les Transformers

Les méthodes de compression d'images basées sur les Transformers [Vaswani et al., 2017] sont apparues récemment comme une alternative prometteuse aux approches traditionnelles de compression par convolution en permettant une modélisation plus sophistiquée des dépendances globales et locales dans les images. Les modèles Transformer, initialement conçus pour la traduction automatique et le traitement du langage naturel, sont capables de capturer des relations à longue portée entre les éléments d'une séquence, un avantage qui a été adapté avec succès à la compression d'images pour capturer la structure spatiale et contextuelle au sein des données visuelles.

Par exemple, l'article [Zhu et al., 2022] introduit SwinT-ChARM, qui repose sur les Swin Transformers [Liu et al., 2021], exploitant des fenêtres décalées pour capturer des relations spatiales hiérarchiques. Ce modèle de compression basé sur des transformations non linéaires démontre une meilleure efficacité que les réseaux convolutionnels (CNN) tout en réduisant le nombre de paramètres et en raccourcissant le temps de décodage. SwinT-ChARM est couplé avec un modèle de prior autorégressif par canal, optimisé pour un calcul efficace. En 2022, STF (Symmetrical TransFormer) [Zou et al., 2022] propose une approche de compression utilisant des blocs de Transformers symétriques dans un encodeur de sous-échantillonnage et un décodeur de sur-échantillonnage. En combinant un mécanisme d'attention locale avec un apprentissage de caractéristiques globales, cette architecture répond aux besoins de compression d'images en capturant les informations clés aux niveaux local et global. STF introduit également une attention locale basée sur des fenêtres, adaptable et intégrable en tant que composant modulaire pour enrichir les modèles CNN et Transformer.

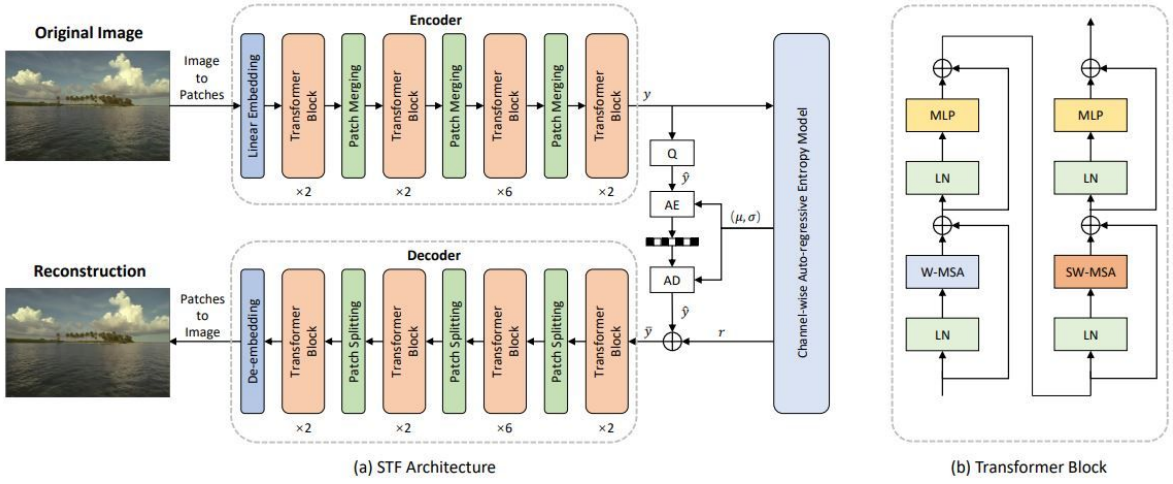


FIGURE 3.16 – (a) L'architecture de STF. La couche de fusion de patchs et la couche de division de patchs sont composées de couches linéaires et de couches de normalisation (LN). (b) W-MSA est le *window-based multi-head self-attention*, et SW-MSA est le *shifted window-based self-attention*, respectivement (source [Zou et al., 2022]).

L'approche [Liu et al., 2023a] propose une architecture hybride Transformer-CNN, dans lequel le bloc TCM (*Transformer-CNN Mixture*) intègre la capacité de modélisation locale des CNN et la modélisation non-locale des Transformers. Ce modèle utilise un module d'attention Swin-Transformer (SWAtten) pour une modélisation par canal efficace, renforçant l'efficacité de compression tout en offrant un compromis contrôlable en termes de complexité. Pour offrir une flexibilité dans les objectifs de qualité d'image, [Kao et al., 2023] propose un système de compression d'image basé sur un autoencodeur Transformer, permettant aux utilisateurs de choisir un compromis entre deux objectifs de qualité d'image à l'aide d'un modèle unique et partagé. Ce modèle utilise des « prompt tokens » pour conditionner l'autoencodeur, générés de manière adaptative selon les préférences de l'utilisateur et l'image d'entrée via un réseau de génération de prompt. Le système ajuste le processus de

codage en fonction des objectifs de qualité, en adaptant le compromis entre deux termes de distorsion (d_A et d_B) correspondant aux objectifs de qualité choisis. La fonction de perte L est formulée pour équilibrer le débit et la distorsion :

$$L = R_\lambda(-\log p(\hat{z}) - \log p(\hat{y}|\hat{z})) + \lambda d_A + (1-\lambda)d_B \tag{3.19}$$

où λ ajuste la priorité entre les deux objectifs de qualité pendant l'entraînement, permettant ainsi au modèle de s'adapter dynamiquement aux préférences de qualité (cf. Figure 3.17).

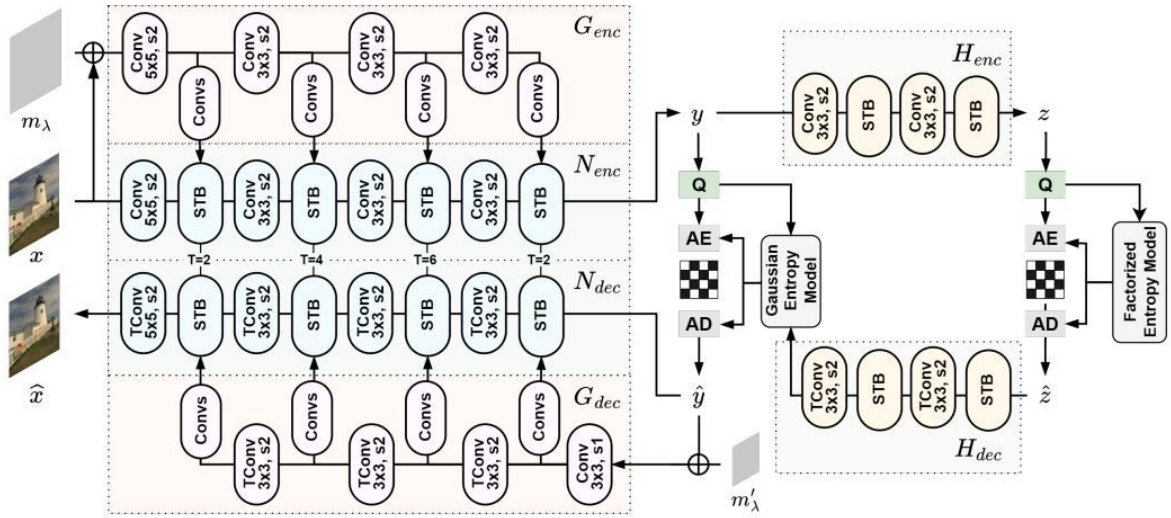


FIGURE 3.17 – Architecture proposée par [Kao et al., 2023]. G_{enc} et G_{dec} génèrent des tokens de prompt comme entrée supplémentaire afin de s'adapter à des objectifs de qualité d'image variables.

L'article [Shen and Liu, 2023] innove avec un modèle de compression basé sur un Transformer spatial-canal, dans lequel les caractéristiques latentes sont divisées en morceaux spatiaux et par canal, encodées de manière autorégressive dans un ordre par canal suivi d'un ordre spatial en zigzag. Ce modèle utilise une fenêtre glissante pour réduire la complexité calculatoire en limitant le nombre de morceaux participant au modèle d'entropie. Inspiré par l'architecture des Transformers, [Ruan et al., 2023a] présente un module d'attention simple (SAM), qui remplace l'auto-attention par des convolutions pour capturer les relations spatiales et par canal. Cette approche permet un modèle de compression efficace en bout-en-bout, alliant les avantages des Transformers à la simplicité des CNN. Enfin, S2LIC [Wang et al., 2024d] propose un modèle haute performance avec un modèle d'entropie basé sur un contexte d'attention adaptatif par canal et global (ACGC). Le modèle divise les caractéristiques latentes en différentes tranches, puis applique le modèle ACGC en utilisant un contexte de type échiquier, améliorant la vitesse de décodage et le compromis taux-distorsion. L'attention déformable est utilisée pour affiner dynamiquement les poids en fonction des relations spatiales, et un modèle Transformer SwinV2 avec blocs denses améliore la représentation non linéaire de l'image.

Ces approches illustrent la puissance des Transformers pour capturer des dépendances globales et locales, améliorant ainsi la qualité de la reconstruction et la performance du taux de compression par rapport aux architectures traditionnelles CNN. Les architectures Transformer adaptatives, flexibles et capables de s'intégrer aux modèles convolutionnels classiques, ouvrent de nouvelles perspectives dans le domaine de la compression d'images.

3.2.9 Approches basées sur les modèles génératifs

Les méthodes de compression basées sur des modèles génératifs, ou compression générative, s'appuient sur des réseaux génératifs pour compresser et reconstruire les données. Contrairement aux méthodes traditionnelles de compression, qui utilisent des codecs fixes et peu adaptatifs, la compression générative repose sur la capacité des modèles génératifs à apprendre des représentations riches et significatives de l'image ou de la vidéo, permettant ainsi une reconstruction plus fidèle et visuellement plaisante même à de très faibles taux de compression. En outre, les modèles génératifs peuvent s'adapter aux données et maintenir la robustesse face aux erreurs de

transmission, ce qui est difficile avec les techniques de codage à longueur variable.

La compression générative a été explorée en 2018 avec [Santurkar et al., 2018], qui introduit le concept même de compression utilisant des modèles génératifs. Cette approche montre que, grâce aux réseaux génératifs, GAN [Goodfellow et al., 2014] ou VAE, les reconstructions sont plus précises et visuellement agréables, même lorsque la compression atteint des niveaux très élevés. Cette technique est aussi plus résistante aux erreurs de bits que les schémas de codage conventionnels, ce qui en fait une solution prometteuse pour les canaux de transmission bruités. Par exemple, [Tschannen et al., 2018] propose d'optimiser le compromis débit-distorsion tout en préservant la distribution des données de l'entraînement dans les échantillons reconstruits. Le système de compression générative obtenu permet d'ajuster la reconstruction selon le taux de bits, interpolant entre la modélisation générative des données d'entraînement et la reconstruction fidèle des échantillons. Cette méthode explore notamment l'utilisation de GANs de Wasserstein et d'auto-encodeurs de Wasserstein pour résoudre ce problème d'optimisation, en développant une analyse théorique et empirique de ces systèmes de compression. En 2020, [Mentzer et al., 2020] étudie la combinaison de réseaux adversariaux génératifs (GANs) et de compression apprise pour atteindre une compression générative d'images avec pertes de haute fidélité. Ce travail explore les architectures et stratégies d'entraînement pour les générateurs et discriminateurs, incluant des pertes perceptuelles (comme LPIPS, voir Section 3.2.2). Formellement, ils proposent de combiner les fonctions de perte pour entraîner un GAN conditionnel et un modèle de compression appris. L'entraînement d'un GAN conditionnel repose sur une compétition entre deux réseaux neuronaux, le générateur G et le discriminateur D, avec une certaine condition s, selon les équations :

$$L_G = \mathbb{E}_{y\sim p_Y}\left[-\log(D(G(y,s),s))\right] \tag{3.20}$$

$$L_D = \mathbb{E}_{y\sim p_Y}\left[-\log(1-D(G(y,s),s))\right] + \mathbb{E}_{x\sim p_{X|s}}\left[-\log(D(x,s))\right] \tag{3.21}$$

Comme vu précédemment, pour entraîner un compresseur appris, on minimise le compromis suivant :

$$L_{EG} = \mathbb{E}_{x\sim p_X}\left[\lambda r(y) + d(x,x')\right] \tag{3.22}$$

où $r(y)$ est le taux de compression et $d(x,x')$ la distorsion entre l'image d'origine x et l'image reconstruite x'. En fusionnant les deux types d'apprentissage et en se basant sur l'architecture de la Figure 3.18, l'optimisation du compresseur génératif proposé dans [Mentzer et al., 2020] correspond à la minimisation des deux termes suivants :

$$L_{EGP} = \mathbb{E}_{x\sim p_X}\left[\lambda r(y) + d(x,x') - \beta\log(D(x',y))\right] \tag{3.23}$$

$$L_D = \mathbb{E}_{x\sim p_X}\left[-\log(1-D(x',y))\right] + \mathbb{E}_{x\sim p_X}\left[-\log(D(x,y))\right] \tag{3.24}$$

où λ, β sont des hyperparamètres qui contrôlent le compromis entre les différents termes. Contrairement aux méthodes antérieures, cette approche permet de produire des reconstructions visuellement similaires à l'image d'origine sur un large éventail de taux de compression et peut être appliquée à des images haute résolution.

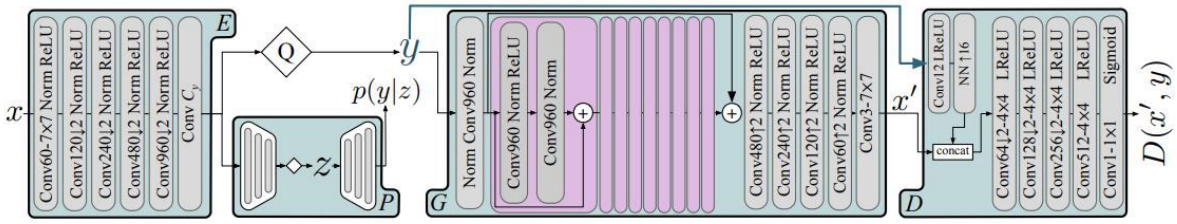


FIGURE 3.18 – Architecture de HiFiC (source [Mentzer et al., 2020])

En 2023, une approche plus avancée est introduite avec [Jiang et al., 2023], qui propose une compression guidée par le texte en utilisant des informations sémantiques pour améliorer les performances de compression. Cette méthode adopte des modules d'attention image-texte et de complément d'image-requête pour fusionner les caractéristiques des images et du texte. En plus, elle introduit une perte de cohérence sémantique multimodale pour garantir des reconstructions sémantiquement complètes (cf Figure 3.19).

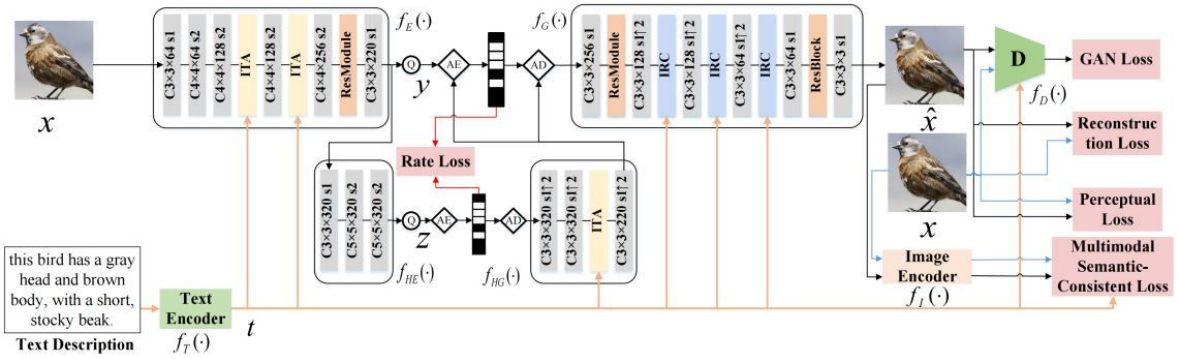


FIGURE 3.19 – Architecture proposée par [Jiang et al., 2023]. AE et AD représentent l'encodage et le décodage arithmétique, et Q correspond à la quantification. ITA est introduit pour fusionner les caractéristiques d'image et de texte basées sur le mécanisme d'attention, et IRC est conçu pour utiliser de manière adaptative les caractéristiques textuelles en complément sémantique de l'image.

L'article [Yang and Mandt, 2023] propose un modèle de diffusion conditionnelle pour la compression d'images avec pertes. Contrairement aux méthodes basées sur des auto-encodeurs variationnels (VAE) où le décodeur est déterministe, cette méthode utilise un modèle de diffusion conditionnel pour le décodeur, introduisant une variable latente supplémentaire pour le contenu de l'image. Pendant le processus de décompression, les informations de texture de l'image sont générées par synthèse, permettant une reconstruction plus riche en détails. Enfin, [Careil et al., 2024] propose PerCo, un modèle de diffusion itératif pour une compression à très bas débit avec un réalisme parfait. Cette méthode conditionne le processus de décodage sur une représentation quantifiée de l'image ainsi qu'une description globale de l'image pour fournir un contexte additionnel. L'encodeur LDM transforme une image RGB en un espace latent pour le modèle de diffusion. L'hyper-encodeur mappe ensuite l'image vers un hyper-latent de résolution spatiale réduite, qui est ensuite quantifié par vecteur et représenté sous forme de flux de bits en utilisant un codage uniforme. Le modèle de description d'image génère une description textuelle de l'image d'entrée, qui est compressée sans perte, puis traitée par un encodeur de texte pour conditionner le modèle de diffusion. Le modèle de diffusion reconstruit l'image d'entrée dans son espace latent en se basant sur les sorties de l'encodeur de texte et de l'hyper-encodeur. Enfin, le décodeur LDM ramène la reconstruction latente dans l'espace des pixels RGB. Malgré un débit ultra-bas, cette approche maintient une capacité de reconstruction d'images réalistes, repoussant les limites de la compression générative (cf. Figure 3.20).

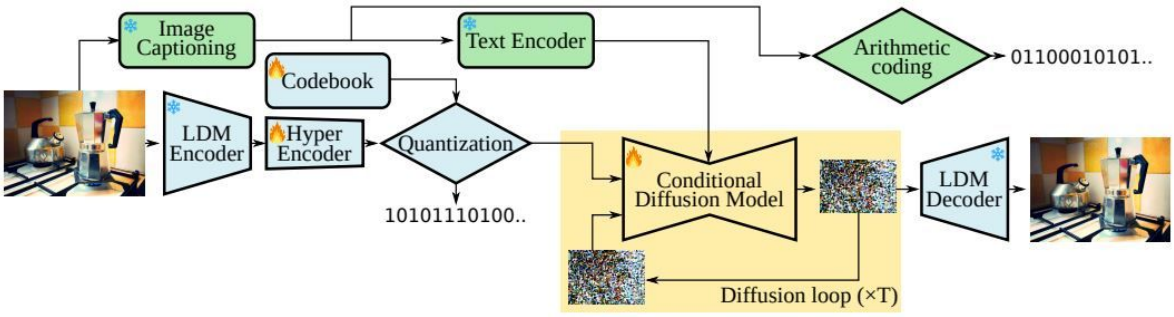


FIGURE 3.20 – Vue d'ensemble de PerCo (source [Careil et al., 2024]).

En somme, les méthodes de compression par modèle génératif démontrent une capacité sans précédent à produire des reconstructions de haute qualité même à des taux de compression extrêmes. En intégrant des techniques de génération d'images avancées telles que les GANs et les modèles de diffusion, elles promettent d'offrir des solutions de compression résilientes, adaptées aux images à très faible débit, et capables de maintenir la fidélité visuelle de manière robuste face aux erreurs de transmission. Cependant, ces méthodes sont souvent coûteuses en calcul, nécessitant des réseaux profonds et de lourdes ressources matérielles pour l'entraînement et l'inférence, ce qui les rend moins adaptées aux environnements à faibles ressources ou aux applications en temps réel. En outre, leur complexité peut introduire des difficultés de stabilité lors de l'entraînement, notamment pour les GANs, et elles peuvent produire des artefacts indésirables ou des distorsions perceptuelles lorsqu'elles tentent

de recréer des détails complexes ou dans des situations de compression extrême.

3.2.10 Autres approches

De nombreuses autres méthodes, bien que moins populaires, méritent également d'être soulignées.

Par exemple, [Ho et al., 2021a] introduit l'utilisation de *augmented normalizing flows* (ANF) pour la compression avec pertes. ANF ajoute un bruit indépendant à l'entrée pour permettre une transformation plus lisse vers l'espace latent, généralisant ainsi le cadre de compression VAE (Variational Autoencoder). Cette approche est étendue par ANFIC [Ho et al., 2021b], qui utilise une hiérarchie de VAE empilés pour augmenter la capacité du modèle et exploite l'inversibilité de l'ANF pour offrir un contrôle flexible de la qualité sans modifier les réseaux d'encodage et de décodage. Pour limiter la perte d'information, [Xie et al., 2021] propose un réseau d'encodage inversible qui utilise des réseaux neuronaux inversibles (INN), permettant une reconstruction plus précise avec moins de pertes, améliorant ainsi la qualité de la compression.

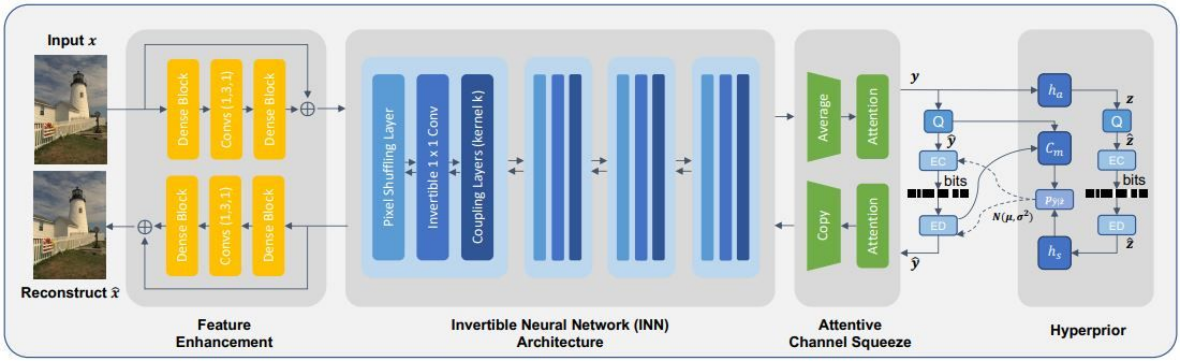


FIGURE 3.21 – Vue d'ensemble du réseau de codage inversible proposé par [Xie et al., 2021].

En 2023, LLIC [Jiang et al., 2024b] introduit des convolutions avec de grands noyaux et une allocation adaptative des bits par canal. Cette approche réduit les redondances tout en conservant une complexité modérée, générant des facteurs d'importance des canaux en fonction du contenu.

Dans [Prativadibhayankaram et al., 2023], une architecture divise la tâche de compression en deux sous-tâches : l'apprentissage de l'information structurelle depuis le canal de luminance et l'apprentissage de la couleur depuis les canaux de chrominance. Le modèle comporte deux branches distinctes pour traiter séparément les composantes de luminance et de chrominance (cf. Figure 3.22). Pour optimiser la fidélité des couleurs, la fonction de perte utilise la métrique de différence de couleur CIEDE2000, favorisant ainsi une meilleure préservation de la couleur dans l'image compressée selon la fonction de coût suivante :

$$L = R + \lambda_1 \cdot \text{MSE}(.) + \lambda_2 \cdot (1 - \text{MS-SSIM}(.)) + \lambda_3 \cdot \Delta_{00}^{12}(.), \tag{3.25}$$

où Δ_{00}^{12} est une métrique de différence de couleur CIEDE2000 pour la fidélité des couleurs.

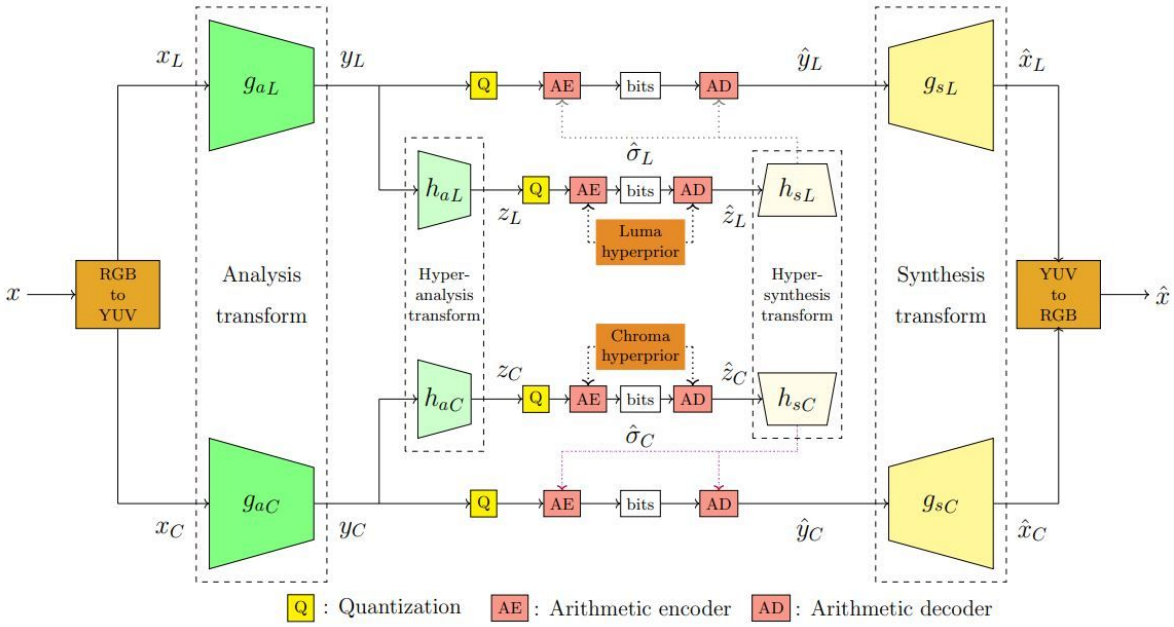


FIGURE 3.22 – Architecture du réseau du modèle d'apprentissage des couleurs proposé par [Prativadibhayankaram et al., 2023]. La partie supérieure représente la branche de luminance et la partie inférieure la branche de chrominance.

L'article [Luo et al., 2023] propose une technique de compression allégée par élagage dans le module hyperprior ("pruning"). Cette méthode élimine une grande partie des paramètres et des calculs sans dégrader les performances, simplifiant ainsi les modèles de compression. Pour l'optimisation de la quantification du latent, [Zhang and Wu, 2023] présente un schéma de quantification vectorielle par réseau (*Lattice Vector Quantization*, LVQ) associé à un mappage de compandage adaptatif spatialement (*Adaptive Companding*, AC). Cette approche exploite mieux les dépendances entre caractéristiques que la quantification scalaire uniforme tout en restant presque aussi simple en termes de calcul. Contrairement aux méthodes de compression end-to-end qui utilisent principalement des quantificateurs scalaires uniformes, ce schéma tire parti des propriétés optimales de la quantification vectorielle pour une meilleure efficacité de compression, avec une adaptabilité accrue grâce au mappage de compandage adaptatif. Pour répondre aux besoins de compression spatiale échelonnable, [Park et al., 2023] introduit une méthode flexible de compression spatiale échelonnable, permettant un ajustement de l'échelle pendant l'inférence en utilisant une méthode de prédiction entre couches basée sur des représentations neuronales implicites. L'article [Inazu and Kimata, 2024] se concentre sur les images RGBA, en proposant une architecture dédiée pour les images avec transparence. Ce modèle utilise un encodage séparé pour les canaux RGB et alpha, et une fonction de perte spécialement conçue pour tenir compte des caractéristiques distinctes du canal alpha. L'étude [Fu et al., 2024] propose un modèle de compression d'image avec deux réseaux de codage ramifiés pour apprendre indépendamment une représentation latente en basse résolution et en haute résolution de l'image d'entrée, représentant de manière discriminative les informations globales et locales. Le modèle utilise un modèle entropique auto-régressif par canal en parallèle pour le codage et le décodage de ces représentations. Cette approche améliore le compromis taux-distorsion et rend le codage et le décodage environ deux fois plus rapides. Pour équilibrer la complexité et les performances, la représentation latente haute résolution sert de condition pour la basse résolution, apportant des informations globales et réduisant ainsi la redondance entre les deux représentations. L'approche [Zhang and Lin, 2024] propose un modèle de compression d'image optimisé de bout en bout, basé sur une transformation orientée par fréquence, surpassant tous les codecs traditionnels, y compris VVC. Ce modèle comprend quatre composants : l'échantillonnage spatial, la transformation orientée par fréquence, l'estimation de l'entropie et la fusion sensible à la fréquence. La transformation orientée par fréquence décompose le signal de l'image en bandes de fréquences distinctes, ce qui correspond aux concepts interprétables par l'humain. Exploitant l'hypothèse de non-recouvrement, ce modèle permet un codage évolutif en transmettant sélectivement certaines composantes de fréquence. Enfin, [Pakdaman et al., 2024] intègre la notion de *Just Noticeable Distortion* (JND) dans le processus d'apprentissage de la compression d'image. Trois méthodes d'optimisation perceptuelle sont introduites pour intégrer JND : la perte JND au niveau pixel (*Pixel-Wise JND Loss*, PWL), qui assure la fidélité pixel par pixel pour reproduire les caractéristiques JND, la perte JND au

niveau image (*Image-Wise JND Loss*, IWL), qui vise des niveaux de dégradation imperceptibles globalement, et la perte JND au niveau des caractéristiques (*Feature-Wise JND Loss*, FWL), qui aligne les caractéristiques des images reconstruites avec les caractéristiques perceptuellement significatives.

Ces méthodes montrent des progrès significatifs en termes de qualité de reconstruction, de flexibilité et d'adaptation à des exigences spécifiques, telles que la prise en charge de la transparence, la réduction de la complexité et l'amélioration de l'efficacité perceptuelle. Cependant, elles peuvent impliquer des calculs complexes et nécessitent souvent des ressources élevées pour être déployées efficacement.

3.2.11 Approches à taux de compression variable

Jusqu'à présent, les modèles étudiés sont à compression fixe après entraînement, c'est-à-dire que pour une image donnée, le compresseur ne produit qu'une version reconstruite avec un seul taux de compression, et pour obtenir un autre taux de compression, il est nécessaire de réentraîner le modèle avec un hyperparamètre λ différent. À l'inverse, les méthodes de compression d'images à taux variable permettent de contrôler dynamiquement le taux de compression pour répondre à des exigences spécifiques en termes de qualité visuelle et de débit de données. Cela est particulièrement avantageux dans des applications où les conditions de transmission ou les contraintes de stockage fluctuent, nécessitant un ajustement du taux de compression sans réentraînement du modèle pour chaque nouveau taux. Voici un aperçu des principales méthodes de compression à taux variable dans le domaine de la compression d'images, comprenant les approches multi-échelle, les couches adaptatives (slimmable layers), les cartes de qualité, les multiplicateurs de Lagrange, ainsi que d'autres techniques avancées.

La compression multi-échelle repose sur l'idée de décomposer l'image en plusieurs niveaux de détail pour coder efficacement les informations essentielles. En utilisant des structures hiérarchiques, l'image est compressée couche par couche, avec une première couche qui encode le contenu de base et les couches suivantes qui affinent la reconstruction en codant les erreurs résiduelles. Cette approche permet d'obtenir plusieurs niveaux de qualité ou de débits de compression à partir d'un seul modèle, chaque couche apportant des détails supplémentaires en fonction des besoins. Par exemple, l'article [Jia et al., 2019] propose un cadre de compression d'images basé sur un auto-encodeur évolutif (SAE) conçu avec des couches hiérarchiques de codage. Chaque couche est un auto-encodeur optimisé de bout en bout : la première couche (base) code le contenu et la texture globale de l'image, tandis que les couches suivantes (enhance) codent les erreurs de reconstruction de manière itérative au niveau du pixel. Ce cadre permet d'éviter de former plusieurs modèles pour différents taux de compression, les couches SAE pouvant être combinées pour produire des flux compressés à différents débits (cf. Figure 3.23).

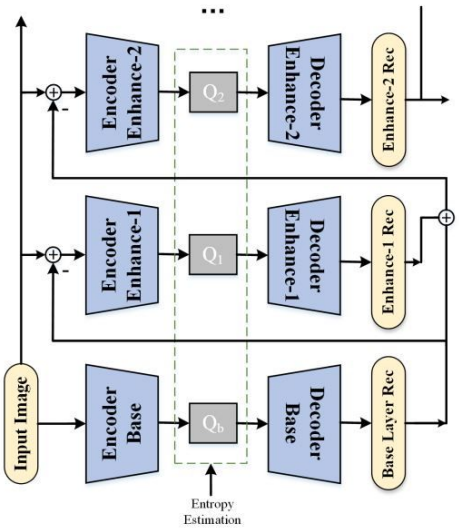


FIGURE 3.23 – Le cadre de la compression d'image basée sur le SAE proposé par [Jia et al., 2019].

De même, l'article [Cai et al., 2019] propose un modèle de compression d'images multi-échelle basé sur un réseau de transformation CNN et une allocation de bits adaptative selon le contenu. Le réseau de transformation décompose l'image en plusieurs échelles, optimisant ainsi le compromis taux-distorsion (RD) sur chaque échelle.

Des algorithmes d'allocation de bits sont fournis pour déterminer l'échelle optimale pour chaque bloc d'image en fonction d'un taux cible ou de la complexité du contenu, ce qui permet un contrôle efficace et adaptatif du débit.

Les auto-encodeurs compressifs *slimmable* sont conçus pour ajuster dynamiquement la capacité du modèle en fonction des besoins de compression. Dans cette méthode, les couches du réseau peuvent être activées ou désactivées selon les contraintes de débit de données, permettant ainsi de gérer différents niveaux de taux sans nécessiter plusieurs modèles. Le modèle ajuste sa complexité en sélectionnant les couches nécessaires pour atteindre le niveau de compression souhaité, ce qui optimise le compromis débit-distorsion pour chaque configuration.

Par exemple, l'article [Yang et al., 2021] introduit des auto-encodeurs compressifs ajustables (SlimCAEs), où le taux de bits (R) et la distorsion (D) sont optimisés conjointement pour différentes capacités. Une fois entraînés, les encodeurs et décodeurs peuvent être exécutés à des capacités variées, offrant ainsi des taux de compression et des complexités ajustables, selon des compromis RD spécifiques pour chaque configuration.

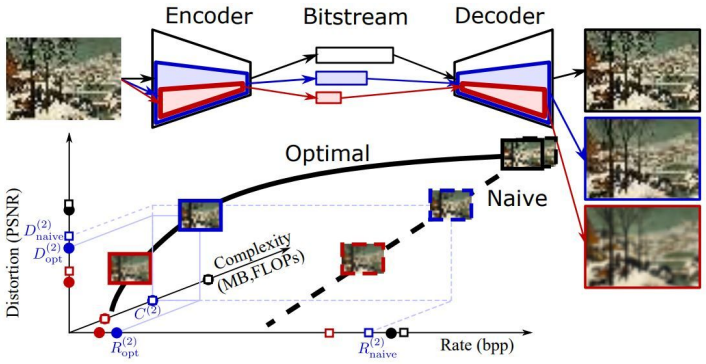


FIGURE 3.24 – Approche proposée par [Yang et al., 2021].

De même, [Tao et al., 2023] propose des auto-encodeurs compressifs (CAEs) avec une adaptation spatiale des convolutions et des opérations de normalisation, permettant une transformation non linéaire par bloc pour allouer les ressources de calcul selon une carte de capacité. Un modèle de routage léger prédit la carte de capacité pour optimiser les blocs CAE selon les cibles RD et la complexité (JUMC), assurant ainsi une allocation adaptative des ressources. Enfin, [Zhang et al., 2023f] propose un codec d'image flexible et complet, ELFIC, avec des couches ajustables (*Slimmable Layer*), permettant un contrôle du taux et de la complexité du décodage. ELFIC utilise une transformation non linéaire (NFFT) pour optimiser la performance RD, une allocation de complexité de décodage adaptée à l'instance (IDCA) pour ajuster le compromis qualité-complexité, et un algorithme d'optimisation RD-complexité qui maximise la qualité visuelle sous des contraintes spécifiques de taux et de complexité pour l'ensemble du cadre de compression.

Les méthodes basées sur des cartes de qualité ou des multiplicateurs de Lagrange ajustent la distribution des bits dans l'image en fonction de l'importance visuelle de chaque région. La carte de qualité est une matrice de pondération qui attribue plus de bits aux zones importantes et moins de bits aux zones moins critiques, améliorant ainsi l'efficacité de la compression. Le multiplicateur de Lagrange est une technique de contrôle des coûts qui introduit un paramètre dans la fonction de perte pour ajuster le compromis entre le débit de bits et la distorsion. En ajustant ce multiplicateur, il est possible d'affiner la qualité et le taux de compression, permettant ainsi de contrôler précisément le compromis RD (débit-distorsion).

Par exemple, en 2019, [Choi et al., 2019] introduit un cadre de compression d'image à taux variable basé sur un auto-encodeur conditionnel. Contrairement aux méthodes de compression basées sur l'apprentissage qui nécessitent des réseaux séparés pour chaque taux de compression, cette méthode utilise un seul modèle en adaptant le taux via deux paramètres de contrôle : le multiplicateur de Lagrange et la taille des bins de quantification. En ajustant le multiplicateur de Lagrange, le modèle peut gérer une adaptation de taux grossière, tandis que la taille des bins permet un réglage plus fin. En 2021, [Song et al., 2021b] propose un algorithme de compression d'image conditionné par une carte de qualité en valeurs réelles. Cette carte de qualité guide l'allocation de bits en fonction des besoins spatiaux de chaque pixel, optimisant ainsi la compression de l'image entière. L'architecture utilise une transformation des caractéristiques adaptatives, basée sur la spatialité comme information préalable,

permettant au modèle de surpasser les performances des modèles à taux fixe sur une gamme de débits. Cette approche offre également la capacité de compression consciente de la tâche, en estimant une carte de qualité spécifique à la tâche à partir d'un modèle pré-entraîné lors de l'encodage.

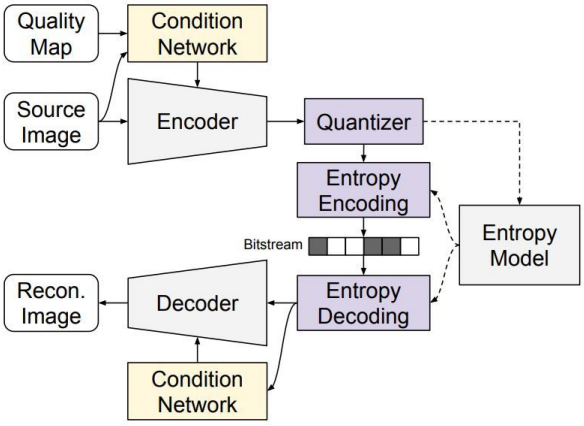


FIGURE 3.25 – Architecture proposée par [Song et al., 2021b].

L'article [Cai et al., 2022a] aborde la question de la compression à taux variable haute fidélité avec un module de transformation d'activation réversible (*Invertible Activation Transformation*, IAT). Le module IAT est intégré dans un réseau neuronal inversible basé sur un modèle à taux unique, où le niveau de qualité (QLevel) est introduit pour générer des tenseurs de mise à l'échelle et de biais, permettant ainsi un contrôle fin du taux variable tout en préservant la fidélité de l'image. En 2023, l'article [Shi et al., 2023] propose un réseau de compression à taux variable utilisant un hyperprior de profondeur variable, entraîné avec une carte de qualité uniforme générée par le paramètre de compromis débit-distorsion. Cela permet de compenser les informations latentes pour différents niveaux de qualité. L'article [Tong et al., 2023] introduit un cadre de compression à taux variable basé sur l'erreur de quantification. Le QVRF utilise un régulateur de quantification univarié et des multiplicateurs de Lagrange prédéfinis pour contrôler l'erreur de quantification dans les représentations latentes, permettant une large gamme de taux de compression au sein d'un même modèle. De manière similaire, [Liang et al., 2023] propose une méthode de compression d'images à taux variable guidée par l'importance spatiale. Le modèle inclut une unité d'activation spatiale (*Spatial Gating Unit*, SGU) pour apprendre une carte d'importance spatiale Λ et un réseau de mise à l'échelle spatial (*Spatial Scaling Network*, SSN) qui utilise cette carte pour guider la mise à l'échelle des caractéristiques et l'allocation de bits, optimisant ainsi le taux en fonction de l'importance spatiale.

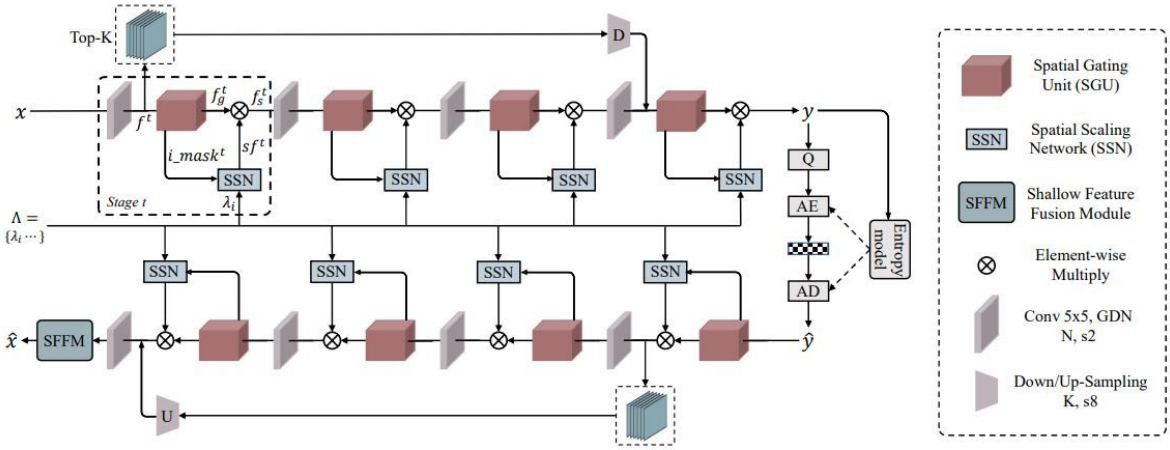


FIGURE 3.26 – Vue d'ensemble de SigVIC (source [Liang et al., 2023]).

Enfin, [Guo et al., 2024] propose un algorithme de compression à taux variable basé sur une agrégation contextuelle bi-résolution, qui divise les représentations latentes en composantes haute résolution (HR) et basse résolution (LR). L'algorithme combine cette approche avec un modèle de contexte spatial-canal (*BSC context*

model), intégrant un modèle en damier et un regroupement de canaux pour couvrir une large gamme de débits.

Plusieurs autres techniques ont été développées pour la compression à taux variable. En 2021, [Lu et al., 2021] introduit un schéma de compression progressif, dans lequel des niveaux de quantification imbriqués et un ordre d'importance des éléments latents sont utilisés pour ajuster le niveau de détail de manière continue. Chaque niveau de quantification affine les latents du plus grossier au plus précis, avec un ordre basé sur les priorités de taux-distorsion. L'article [Sun et al., 2021] propose un réseau de compression appelé *Interpolation Variable-Rate* (IVR) qui utilise un module d'attention d'interpolation de canaux (InterpCA) pour ajuster le taux sans perte de performance. Ce module minimaliste, qui fonctionne avec des paramètres de contrôle de taux et une interpolation linéaire, est compatible avec la plupart des modèles basés sur l'entropie. Toujours en 2021, [Cui et al., 2021] introduit un auto-encodeur variationnel asymétrique (AG-VAE) permettant une adaptation continue du taux via une interpolation exponentielle. AG-VAE utilise des unités de gain asymétriques pour ajuster le taux de manière discrète sans ajouter de complexité significative, avec un modèle d'entropie gaussienne asymétrique pour une estimation précise (cf. Figure 3.27).

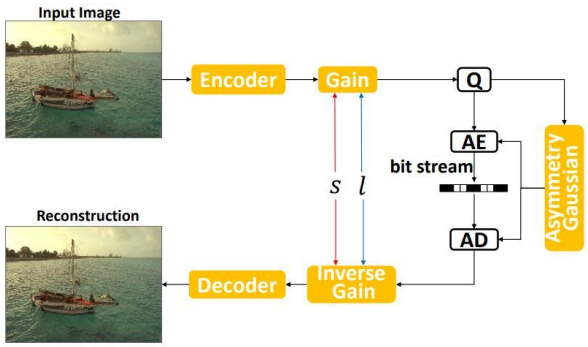


FIGURE 3.27 – Architecture de AG-VAE. L'adaptation du taux est réalisée en insérant une unité de gain après l'encodeur et une unité de gain inverse avant le décodeur. Le taux de bits peut être ajusté de manière continue en modifiant l'indice du vecteur de gain s et le coefficient d'interpolation l. Le modèle d'entropie gaussien asymétrique estime avec précision l'entropie de la représentation latente après gain et quantification (source [Cui et al., 2021]).

En 2022, [Lee et al., 2022a] propose un codec progressif basé sur des plans de trits. L'algorithme encode les éléments latents sous forme de chiffres ternaires (trits) et les ordonne selon leur importance en taux-distorsion, permettant une transmission progressive où les informations critiques sont envoyées en premier. Un réseau de post-traitement améliore la qualité des images reconstruites, en particulier pour les faibles débits. L'article [Lee et al., 2022b] propose une méthode de compression sélective basée sur une carte d'importance 3D. Cette carte d'importance, ajustée pour différents niveaux de qualité, est convertie en masque binaire pour déterminer les éléments latents essentiels à la compression, permettant une meilleure allocation de bits en fonction de la structure de l'image. De plus, [Yin et al., 2022] introduit des modules universels (*Energy-based Channel Gating* et *Bit-rate Modulator*) pouvant être intégrés dans les modèles de compression d'image. Ces modules contrôlent dynamiquement le taux de compression en fonction de l'importance des canaux latents, réduisant ainsi les FLOPs de calcul de plus de 50% dans les couches de convolution, tout en maintenant la qualité visuelle. En 2023, [Guo et al., 2023] propose un réseau adaptatif en complexité et taux de compression, CBANet, qui permet l'ajustement du taux de bits selon différents niveaux de complexité de calcul. Le réseau utilise un module adaptatif de complexité et un module adaptatif de taux pour optimiser les compromis entre complexité-distorsion et taux-distorsion. Enfin, [Qin et al., 2024] introduit le module de prompt adaptatif par couche (LPM) pour ajuster dynamiquement le taux de compression. Le LPM extrait des prompts pour les images et les caractéristiques cachées aux étapes de codage et décodage, influençant l'allocation de l'attention et des bits, et donc le ratio de compression (cf. Figure 3.28).

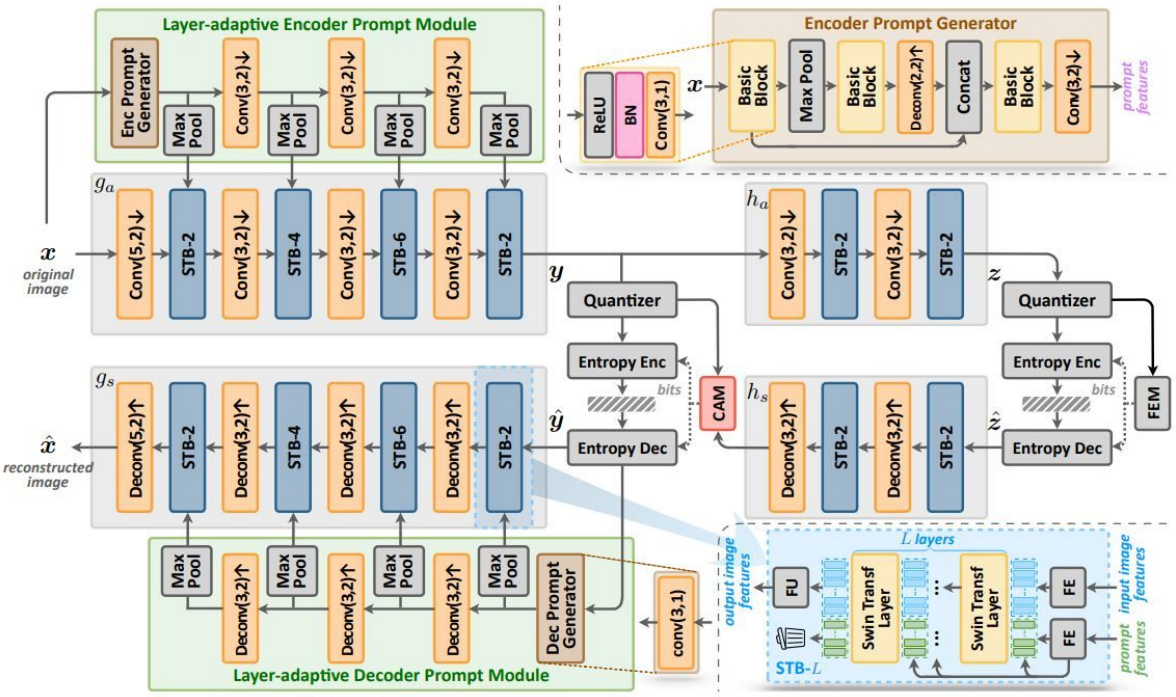


FIGURE 3.28 – Architecture proposée par [Qin et al., 2024]. CAM est le module d'attention causale, FEM est le modèle d'entropie factorisé.

Ces méthodes permettent une flexibilité et un contrôle du taux de compression, en offrant des solutions efficaces pour des environnements où la qualité, le débit de données et la complexité de calcul sont des contraintes variables. En intégrant ces différentes approches dans des réseaux d'auto-encodeurs profonds, la compression à taux variable permet des ajustements dynamiques en fonction des besoins, tout en maintenant une qualité visuelle correcte. Cependant, ces types de méthodes sont souvent moins performantes en termes de compromis taux-distorsion comparé aux méthodes à taux de compression fixe.

3.2.12 Approches ROI & orientées tâches

Il existe un dernier type de compression apprise. Ces compresseurs, étant optimisés de bout en bout, peuvent être adaptés et optimisés pour différentes tâches telles que la classification, la détection, ou la segmentation sémantique. Cette approche est connue sous le nom de compression ROI (Region of Interest) ou orientée tâches (task-oriented compression). En pratique, cela signifie que le modèle de compression peut accorder plus d'importance aux régions ou caractéristiques pertinentes pour une tâche spécifique, comme les objets d'intérêt pour la détection d'objets, tout en comprimant davantage les zones moins significatives. Cela permet d'obtenir une compression plus efficace, maximisant la précision des tâches en aval tout en réduisant les besoins en stockage et en transmission des données, rendant ces techniques particulièrement utiles dans les applications de vision par ordinateur intégrées ou à faible bande passante.

Par exemple, en 2020, [Ballé et al., 2017] propose un cadre de compression d'image basé sur l'apprentissage profond pour l'optimisation taux-distorsion dans la compression de régions d'intérêt (ROI). Ce cadre inclut un encodeur et un décodeur CNN spécifiques aux ROI, ainsi qu'un codec entropique appris. L'encodeur produit des représentations multi-échelles pour une allocation efficace des débits et génère un masque ROI implicite pour guider cette allocation. Une fonction de perte hiérarchique est également proposée pour améliorer la qualité visuelle, en préservant la fidélité des pixels dans les ROI et la similarité structurelle pour l'image entière (cf. Figure 3.29).

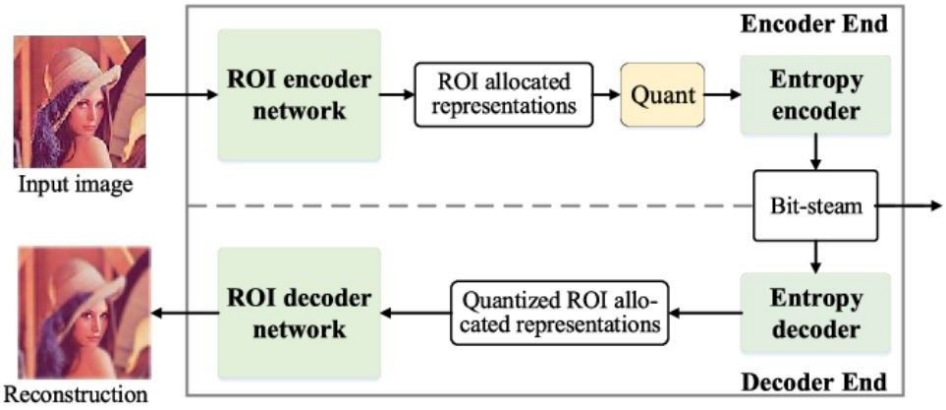


FIGURE 3.29 – Architecture proposée par [Ballé et al., 2017].

L'article [Akutsu and Naruko, 2020] démontre l'efficacité de la compression d'images avec un auto-encodeur convolutif (CAE) optimisé pour les régions d'intérêt (ROI) afin de contrôler la qualité. Cette méthode utilise des annotations (comme des boîtes englobantes ou des cartes de segmentation) pour ajuster les poids de distorsion dans la fonction de perte basée sur le MS-SSIM lors de l'entraînement du réseau de compression, permettant une meilleure qualité dans les régions d'intérêt. En 2021, [Gao et al., 2023] explore l'utilisation de la distance dans l'espace des caractéristiques profondes comme métrique, en transformant les images dans un espace latent via un réseau convolutif préentraîné, mieux aligné sur la sémantique. Une autre métrique étudiée est la distance pondérée par importance au niveau des pixels, où les poids reflètent l'importance des pixels pour la sémantique. L'article combine ces deux distances pour étudier leur complémentarité, montrant que cette métrique combinée obtient les meilleures performances. En 2022, une technique intéressante nommée *Code Editing* est proposée par [Gao et al., 2022a]. C'est une méthode de compression d'image neuronale flexible via l'édition de code, basée sur l'inférence semi-amortie et la quantification adaptative. Traditionnellement, chaque point de la courbe taux-distorsion (R-D) nécessite un couple encodeur-décodeur dédié, limitant l'applicabilité pratique. Bien que certains travaux récents permettent le contrôle du débit binaire via un codage conditionnel, ils imposent des contraintes importantes durant l'entraînement. De manière simplifiée, à partir d'un modèle préentrainé avec l'hyperparamètre λ_0, pour comprimer une image avec un autre taux de compression λ_1, on cherche le latent correspondant via l'équation suivante :

$$y^* = \text{argmin}_y L_y \quad \text{avec} \quad L_y = R + \lambda_1 D = -\log p_y(y) + \lambda_1 d(x, \mathcal{D}(y)), \tag{3.26}$$

où p_y est la probabilité associée aux latents quantifiés, $\mathcal{D}$ est le décodeur et $d(x, \mathcal{D}(y))$ la mesure de la distorsion entre l'image d'origine x et la reconstruction $\mathcal{D}(y)$. On peut étendre ce formalisme à la compression ROI de la manière suivante :

$$y^*_{ROI} = \text{argmin}_y L^{ROI}_y \quad \text{avec} \quad L^{ROI}_y = -\log p_y(y) + \lambda_1 m \circ d(x, \mathcal{D}(y)), \tag{3.27}$$

où m est un masque d'importance pour pondérer la distorsion d dans les zones spécifiques. Cette approche permet d'ajuster finement la qualité de la reconstruction dans les régions d'intérêt, tout en maintenant la flexibilité pour gérer différents niveaux de distorsion dans un même modèle. L'article [Liu et al., 2022a] vise à améliorer les performances des tâches de vision par ordinateur directement dans le domaine compressé, contournant deux inconvénients des images décodées : une complexité élevée au niveau du décodeur et une perte de précision pour les tâches de vision (comme mIoU, erreur absolue moyenne, et précision moyenne), car les images décodées sont optimisées pour la qualité perçue humaine (ex. PSNR) et non pour les besoins des tâches de vision. L'approche propose un module de portes pour sélectionner efficacement certaines caractéristiques du domaine compressé, l'utilisation de la distillation de connaissances pour améliorer la précision, et une stratégie d'entraînement supportant plusieurs tâches, y compris la compression d'image (cf. Figure 3.30).

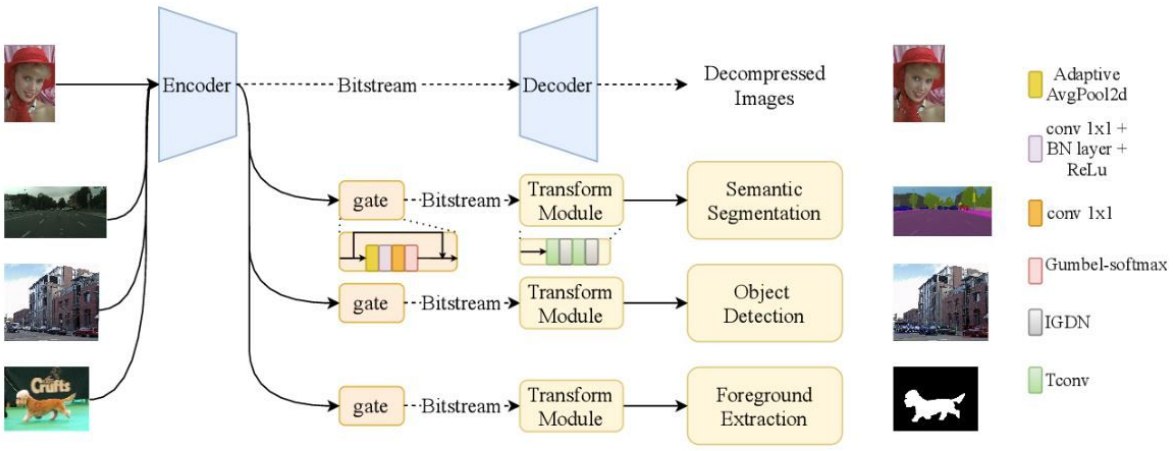


FIGURE 3.30 – Approche proposée par [Liu et al., 2022a].

En 2023, [Lei et al., 2023] propose un cadre de compression d'image profonde, nommé *Residual-Enhanced Mask-Based Progressive Generative Coding* (RMPGC), permettant un contrôle explicite des performances dans le compromis taux-distorsion-classification-perception (RDCP). Ce cadre introduit trois mécanismes indépendants pour générer une représentation latente structurée sémantiquement, capable de contrôler de manière paramétrée le débit et l'adaptation contextuelle pour des tâches hybrides de classification et de reconstruction d'image. L'article [Cao et al., 2023] propose un cadre de compression multi-tâches ajustable pour répondre aux besoins de la vision humaine et de la vision par machine, tout en équilibrant le compromis entre taux, précision et efficacité sous contraintes de bande passante, latence et ressources de calcul. La compression d'image est formulée comme un problème d'optimisation multi-tâches, où des encodeurs ajustables (*slimmable encoders*) sont utilisés pour différentes tâches de vision humaine et par machine, avec un sous-encodeur partagé et appris conjointement pour les tâches de vision par machine. Pour harmoniser la représentation latente avec les caractéristiques intermédiaires des réseaux de vision profonds, des réseaux de transformation de caractéristiques sont introduits comme décodeurs pour la compression des caractéristiques de vision par machine (cf. Figure 3.31).

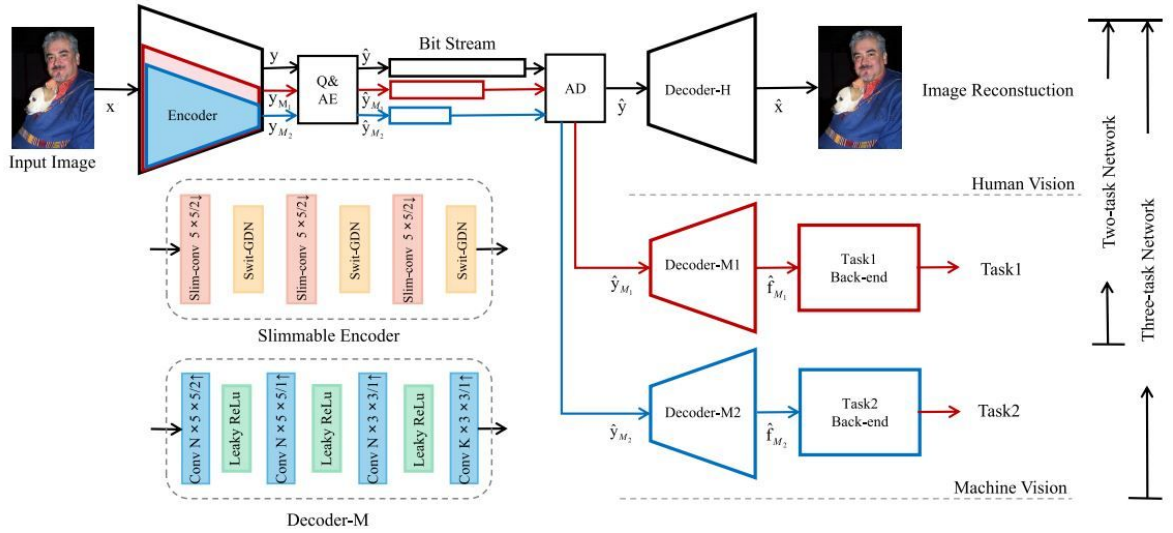


FIGURE 3.31 – Architecture proposée par [Cao et al., 2023]. "Q" représente la quantification. "AE" et "AD" désignent respectivement l'encodeur et le décodeur arithmétique.

Une autre méthode [Chen et al., 2023c] propose un codec de compression d'image basé sur des Transformers, capable de passer de la perception humaine à la perception machine sans nécessiter de réajustement du codec. Le cadre de compression, nommé TransTIC, s'inspire du *visual prompt tuning* et utilise un générateur de prompts spécifiques à chaque instance pour l'encodeur et des prompts propres aux tâches pour le décodeur, permettant une adaptation efficace aux besoins de perception machine. Plus récemment, en 2024, [Liebender et al., 2024] pro-

pose une fonction de perte personnalisée pour les régions d'intérêt (ROI) permettant d'anonymiser les visages dans les images compressées tout en conservant la détectabilité des personnes. Un autoencodeur optimisé de bout en bout est utilisé pour la compression d'image, exploitant la flexibilité des transformations d'analyse et de reconstruction afin de modifier les zones spécifiques du résultat de la compression et de rendre les visages méconnaissables. L'article [Shindo et al., 2024a] propose un modèle de compression d'image qui apprend à identifier les régions d'objets sans nécessiter de carte ROI ni de perte spécifique à une tâche. Ce modèle peut ainsi compresser des images pour divers modèles de reconnaissance. L'approche utilise des cartes ROI provenant de modèles de détection d'objets ou de dérivations manuelles pour allouer plus de bits aux régions d'objets, nécessitant donc l'image et sa carte ROI comme entrées pour optimiser la compression. L'approche [Shindo et al., 2024b] propose SA-ICM, un modèle de compression d'image pour machines centré sur l'encodage et le décodage des seules informations de contour des objets d'une image. Ce modèle utilise des masques générés par le modèle *Segment Anything Model* (SAM) pour extraire et encoder les contours, garantissant robustesse aux variations de données d'entrée. SA-ICM offre également des avantages en matière de confidentialité en supprimant les informations faciales à l'étape d'encodage. Sans nécessiter de carte ROI ou de perte spécifique à une tâche, cette méthode d'apprentissage basée sur les contours fournit une compression efficace et adaptable pour divers cas d'utilisation de reconnaissance d'image. Spécialement pour les tâches de détection d'objets, [Dai et al., 2024a] propose une méthode de compression de caractéristiques, facilement intégrable aux modèles de compression d'image appris. Le processus d'encodage comporte deux étapes : l'extraction des caractéristiques de bas niveau, suivie de la génération d'un masque pour sélectionner les régions contenant des objets, et la compression des caractéristiques masquées via un encodeur neuronal. Au décodage, un décodeur neuronal restaure la représentation compressée en caractéristiques prêtes à être directement utilisées par le modèle de détection d'objets. L'article [Li et al., 2024a] propose un cadre de réglage léger pour la compression d'image, appelé Adapt-ICMH, conçu pour optimiser l'équilibre entre la performance des tâches et le débit binaire. Adapt-ICMH utilise un adaptateur de modulation spatial-fréquence (SFMA) qui élimine simultanément les redondances non sémantiques via une modulation spatiale, et ajuste les composantes fréquentielles en renforçant celles pertinentes pour la tâche et en réduisant celles non pertinentes. Cet adaptateur, compatible avec la plupart des modèles de compression d'image appris, est plug-and-play et n'altère pas la performance des modèles pré-entraînés (cf. Figure 3.32).

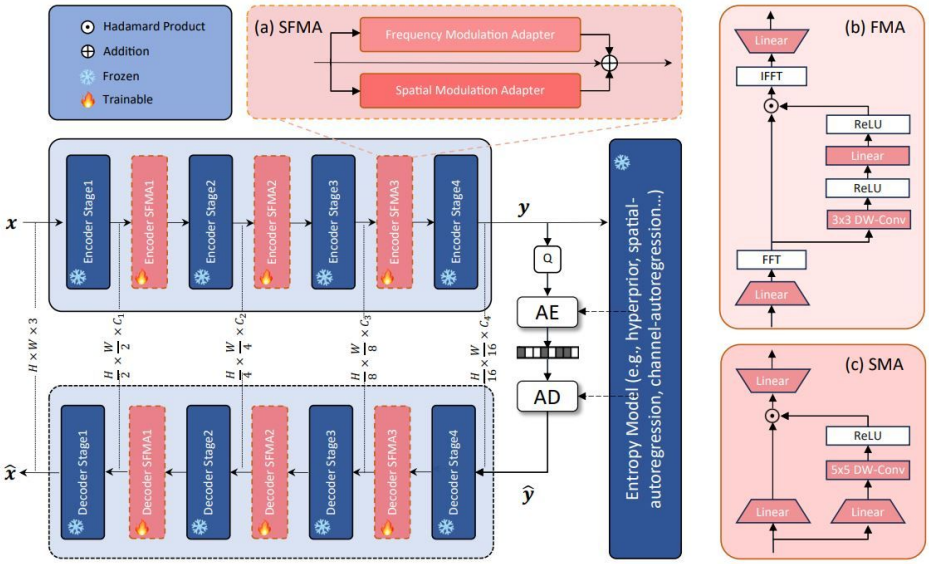


FIGURE 3.32 – Vue d'ensemble du cadre Adapt-ICMH proposé. Plusieurs adaptateurs de modulation spatial-fréquence (SFMA) sont insérés dans l'encodeur g_a et le décodeur g_s du codec de base pour l'adaptation des caractéristiques. Lors de l'adaptation à la tâche de vision par ordinateur, le codec de base est gelé et seuls ces adaptateurs sont entraînables (source [Li et al., 2024a]).

De plus, l'article [Yang et al., 2024] propose un préprocesseur adaptable qui optimise les images pour des tâches de vision par machine avant l'encodage avec un codec conçu pour la vision humaine. Ce préprocesseur préserve l'information sémantique pertinente pour chaque tâche aval tout en supprimant les informations non nécessaires pour réduire le débit binaire. En utilisant des caractéristiques sémantiques pré-extraites, il ajuste la correspondance pixel par pixel pour différents besoins de tâches, permettant ainsi une intégration fluide de mul-

tiples tâches grâce à des modulations interchangeables. Enfin, l'article [Liu et al., 2024] propose une méthode de compression d'image neural permettant aux utilisateurs de régler le débit binaire, la qualité de reconstruction d'image et la précision des tâches de machine avec un modèle unique, offrant ainsi une ultra-contrôlabilité. Cette approche introduit une perte orientée cognition dans la branche principale de compression pour entraîner un codec adapté à diverses tâches de machine, avec un débit variable ajusté par le degré de quantification dans les canaux latents. Pour améliorer la qualité des images reconstruites, une branche auxiliaire fournit des informations résiduelles à travers un flux binaire évolutif. Les deux branches combinent leurs sorties via une stratégie d'interpolation $\beta x + (1 - \beta)y$ pour un compromis optimal entre cognition et distorsion.

Les méthodes de compression ROI et orientées tâche montrent des avancées significatives en s'adaptant aux priorités des régions spécifiques de l'image et aux exigences des applications de vision par ordinateur. Ces approches permettent de maintenir la pertinence des caractéristiques critiques pour les tâches tout en optimisant le débit et la complexité, offrant ainsi des solutions de compression avancées pour une multitude d'applications. Cependant, ces modèles sont plus complexes que les méthodes de compression classiques, car ils intègrent des modules additionnels pour identifier et traiter les zones d'intérêt ou les caractéristiques spécifiques à la tâche. Cette complexité accrue entraîne une consommation plus élevée en termes de calcul et de mémoire, et présente un risque supplémentaire en cas de défaillance de ces modules.

3.2.13 Conclusion

En conclusion, il existe de nombreuses méthodes de compression d'image apprises par *deep learning*, chacune offrant ses qualités et ses limites en fonction des objectifs de compression. Ces techniques sont plus flexibles, adaptatives et performantes par rapport aux techniques classiques. Les modèles à compression fixe se révèlent généralement plus performants en termes de compromis débit-distorsion par rapport aux modèles à taux variable, ce qui les rend particulièrement adaptés aux applications nécessitant une haute fidélité visuelle à un débit de données constant. Dans notre cadre applicatif, où la rapidité de compression (presque en temps réel) est essentielle, nous privilégierons les modèles STF [Zou et al., 2022] et QRes-VAE [Duan et al., 2023]. Ces modèles répondent bien aux exigences de notre application et offrent un bon équilibre entre qualité de compression et vitesse d'exécution. En outre, nous explorerons la possibilité de modifier leurs fonctions de coût lors de l'entraînement afin de les adapter à la compression ROI, permettant ainsi de concentrer les ressources de codage sur les zones d'intérêt spécifiques. Ainsi, bien que chaque méthode de compression présente des atouts distincts, les modèles sélectionnés et les ajustements envisagés devraient répondre aux besoins spécifiques de notre application.

Néanmoins, comparés aux méthodes de compression classiques, ces modèles basés sur des réseaux de neurones restent sensibles aux perturbations et présentent des sorties imprévisibles pour certaines entrées. Par exemple, [Yu et al., 2023] démontre une attaque backdoor contre des modèles de compression d'image apprenants, en utilisant des déclencheurs dans le domaine DCT pour insérer des attaques de manière furtive. De même, l'étude [Wu et al., 2024a] propose deux paradigmes d'attaque sur le compromis débit-distorsion (SRDA et ARDA) et révèle que certains composants des modèles de compression, comme l'hyperprior et la normalisation inverse généralisée (IGDN), peuvent amplifier de manière significative les perturbations en entrée, augmentant ainsi le débit de bits ou dégradant la qualité de la compression sous attaque.

3.3 Détection d'objets par deep learning

Comme décrit dans le Chapitre 1, les images récupérées et compressées par le système de *Field Monitoring* seront ensuite automatiquement analysées afin de comprendre la scène ayant causé leur enregistrement, notamment en cas de défaillance de systèmes d'aide à la conduite. Cette analyse peut être effectuée par des réseaux de neurones à travers diverses techniques telles que la détection d'objets, la segmentation sémantique ou l'analyse de scènes. Dans notre étude, nous nous concentrerons spécifiquement sur un détecteur d'objets 2D pour ce type d'analyse.

Ainsi, dans ce chapitre, nous réaliserons un état de l'art des méthodes de détection d'objets 2D basées sur le deep learning, en les classant chronologiquement et par familles : les méthodes *Two-stage*, *Single-stage*, et celles basées sur les Transformers [Vaswani et al., 2017].

3.3.1 Introduction

La détection d'objet est une tâche essentielle en vision par ordinateur, consistant à localiser et identifier automatiquement des objets d'intérêt dans une image ou une séquence vidéo. Elle se distingue de la classification d'image en ce qu'elle fournit non seulement une catégorie pour chaque objet détecté, mais également sa position dans l'image, généralement sous forme de boîtes englobantes (cf. Figure 3.33). Cette tâche joue un rôle fondamental dans de nombreuses applications, notamment les véhicules autonomes, la surveillance, l'analyse de scènes et l'interaction homme-machine.

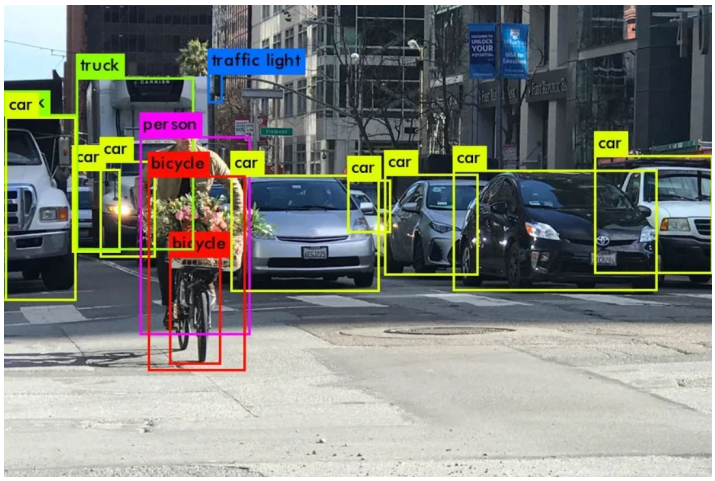


FIGURE 3.33 – Illustration de la détection d'objets.

Les modèles de détection d'objets peuvent être classés en trois grandes catégories. Un réseau qui dispose d'un module distinct pour générer des propositions de régions est appelé un détecteur *Two-stage*. Ces modèles identifient un nombre arbitraire de propositions d'objets dans une image lors de la première étape, puis les classifient et les localisent dans la seconde. Étant donné que ces systèmes comportent deux étapes distinctes, ils prennent généralement plus de temps pour générer des propositions, ont une architecture complexe et manquent parfois de contexte global. Les détecteurs en une seule étape, ou *Single-stage*, classifient et localisent les objets en une seule étape à l'aide d'un échantillonnage dense. Ils utilisent des boîtes ou des points clés prédéfinis de différentes échelles et ratios d'aspect pour localiser les objets. Enfin, plus récemment, avec l'avènement des *Large Language Model* (LLM), les détecteurs basés sur des Transformers ont été développés.

Les modèles sont souvent décomposés en trois parties : Backbone, Neck et Head (cf. Figure 3.34). Le Backbone est responsable de l'extraction des caractéristiques de l'image en passant par plusieurs couches convolutives, créant ainsi des cartes de caractéristiques riches en informations visuelles. Le Neck agit comme un intermédiaire entre le Backbone et le Head, en combinant et en affinant les caractéristiques extraites. Enfin, le Head est la partie du modèle qui génère les prédictions finales, il réalise la classification des objets et la régression des boîtes englobantes, souvent via des couches fully connected ou convolutionnelles adaptées pour obtenir les coordonnées précises des objets détectés.

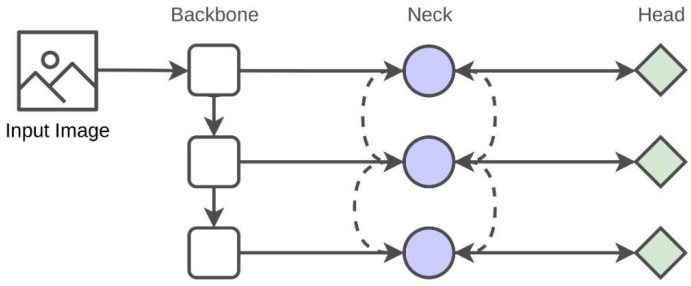


FIGURE 3.34 – Composition des détecteurs d'objets (source [Kateb et al., 2021]).

L'entraînement de modèles de détection d'objets consiste à apprendre simultanément à localiser et classifier des objets dans des images. Pour ce faire, le modèle est formé à partir d'un ensemble de données annotées,

où chaque image contient des boîtes englobantes associées aux catégories d'objets correspondantes. Pendant l'entraînement, le modèle utilise deux types de pertes : une perte de classification pour apprendre à attribuer la bonne classe à chaque objet détecté et une perte de localisation pour ajuster la position et la taille des boîtes englobantes. En minimisant ces pertes, le modèle optimise ses paramètres pour prédire des boîtes englobantes précises et les classes correspondantes, même en présence de variations de position, d'échelle et d'éclairage. Les techniques de régularisation et les données augmentées (rotations, changements de luminosité, etc.) sont souvent utilisées pour améliorer la robustesse et la généralisation du modèle. Après l'entraînement, le modèle est capable d'identifier et de localiser les objets dans de nouvelles images avec un niveau de précision et de rappel mesurable.

La détection d'objet peut être sujette à plusieurs types d'erreurs qui impactent la performance et la fiabilité du modèle :

- Faux positifs (FP) : Il s'agit d'objets détectés à tort dans des zones où il n'y en a pas. Un modèle peut, par exemple, identifier un objet inexistant dans une image, ce qui peut nuire à la fiabilité du système.
- Faux négatifs (FN) : Dans ce cas, le modèle ne détecte pas un objet qui est pourtant présent dans l'image. Ce type d'erreur est particulièrement critique dans des applications sensibles, car un objet non détecté pourrait entraîner des conséquences graves, par exemple dans la détection de piétons par un véhicule autonome.
- Erreur de classification (Miss-classification, MC) : Les objets sont parfois détectés correctement, mais mal classifiés, c'est-à-dire assignés à une mauvaise catégorie.

La sélection d'un modèle de détection d'objet repose sur plusieurs critères clés. La précision, mesurée par des métriques comme l'Average Precision (AP), définie dans la Section 3.3.2, est primordiale pour évaluer la capacité du modèle à détecter et classifier les objets avec exactitude, ce qui est essentiel pour des applications critiques. Le temps de traitement est un autre facteur important, en particulier dans des environnements où des décisions en temps réel sont requises ; la latence et le nombre d'images traitées par seconde (FPS) servent d'indicateurs de cette rapidité. La complexité du modèle, déterminée par le nombre de paramètres et les ressources de calcul nécessaires pour l'entraîner et l'exécuter, est également un critère de choix ; des modèles plus légers peuvent s'avérer avantageux pour des applications embarquées ou sur des dispositifs aux ressources limitées. La robustesse du modèle, c'est-à-dire sa capacité à bien fonctionner dans des conditions variées (comme l'éclairage, la météo ou l'angle de vue), est cruciale pour assurer des performances stables. Enfin, l'adaptabilité du modèle peut également jouer un rôle, car certains modèles sont mieux adaptés à des tâches spécifiques ou à des contextes particuliers, comme les environnements à forte densité d'objets ou avec des types d'objets bien définis, où ils peuvent surpasser des modèles plus généralistes.

3.3.2 Métriques

Plusieurs métriques ont été développées pour évaluer la performance des modèles de détection d'objets. En voici une brève liste ci-dessous.

— **MODA** (Multiple Object Detection Accuracy) : mesure l'exactitude de la détection d'objets en prenant en compte à la fois les faux positifs (FP) et les faux négatifs (FN). On utilise la formule suivante :

$$\text{MODA} = 1 - \frac{\text{FN} + \text{FP}}{\text{GT}} \tag{3.28}$$

avec GT le nombre total d'objets dans les vérités terrain (Ground Truth). Un MODA de 1 indique une détection parfaite sans faux positifs ni faux négatifs. Un MODA inférieur indique une dégradation de la performance due à des erreurs dans la détection.

— **FAF** (False Alarm Rate) :

$$\text{FAF} = \frac{\text{FP}}{\text{N}} \tag{3.29}$$

avec N le nombre total d'images ou de frames dans la séquence. Un FAF élevé signifie que le système de détection produit de nombreuses fausses alarmes.

— **mean Average Precision** (mAP) : métrique la plus standard et couramment utilisée pour évaluer la performance des modèles de détection d'objets. Le processus de calcul est décrit dans la Figure 3.35 ci-dessous.

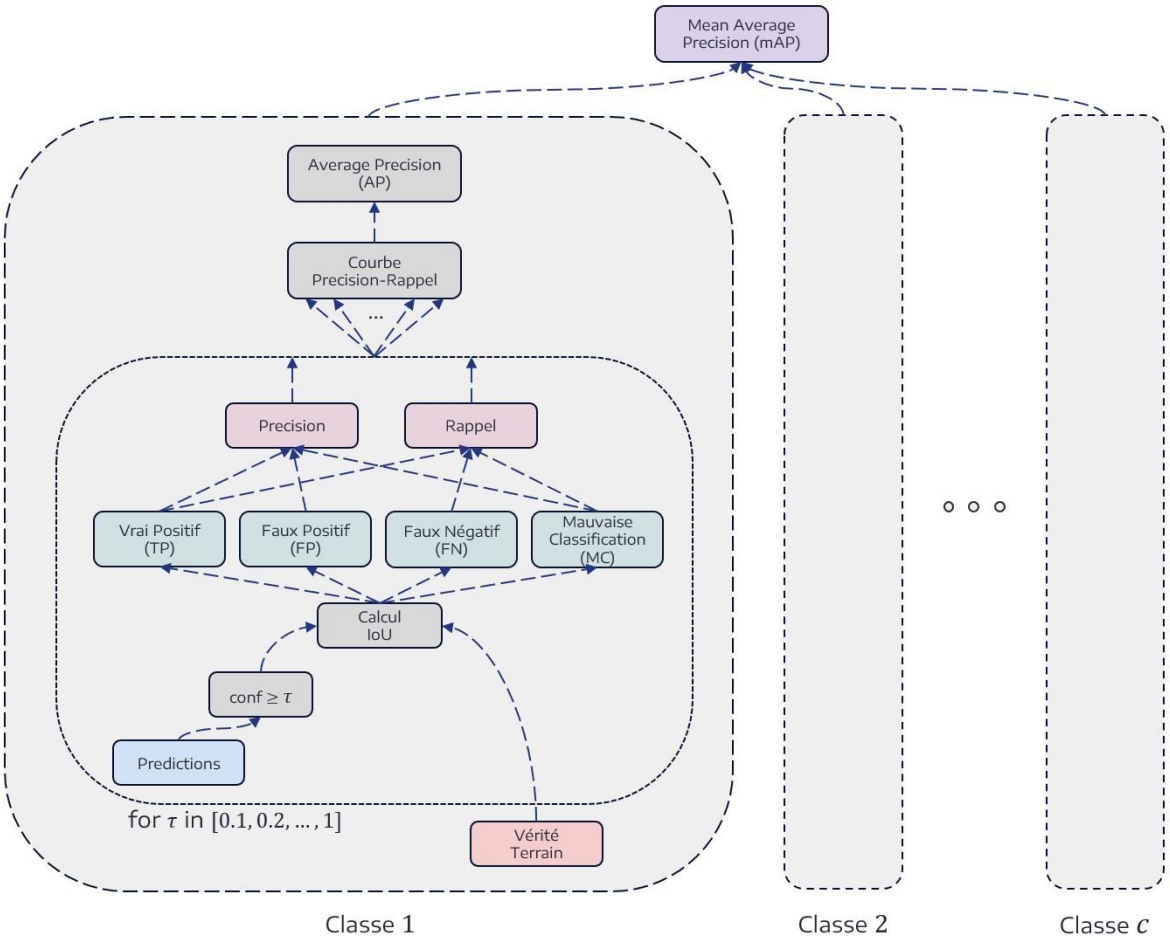


FIGURE 3.35 – Processus de calcul du mAP.

Pour une classe d'objet et différents seuils τ de score de confiance (conf), on associe les boîtes prédites par le modèle de détection à celles de la vérité terrain en utilisant l'*Intersection over Union* (IoU, cf. Figure 3.36), puis on les classifie en :

- Vrai Positif (*True Positive*, TP) : le modèle détecte un objet qui est présent dans la vérité terrain avec la bonne classe.
- Faux Positif (*False Positive*, FP) : le modèle détecte un objet qui n'est pas présent dans la vérité terrain.
- Faux Négatif (*False Negative*, FN) : le modèle ne détecte pas un objet qui est présent dans la vérité terrain.
- Mauvaise Classification (*Miss-classification*, MC) : le modèle détecte un objet qui n'est pas présent dans la vérité terrain, mais la classe prédite est incorrecte.

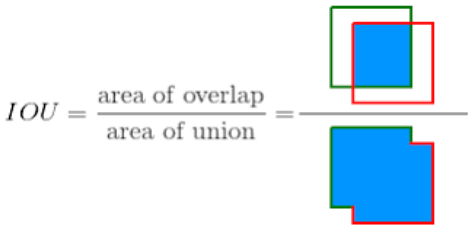


FIGURE 3.36 – *Intersection over Union* (IoU) est defini comme le rapport entre l'intersection et l'union des deux boîtes. Si ce dernier est supérieur à un certain seuil (en général 0.5) on considère que les deux boites encadrent le même objet.

Ensuite, on calcule la Précision (P) et le Rappel (R), ou en anglais Recall, définis respectivement comme la proportion des détections qui sont correctes parmi toutes les détections effectuées par le modèle et la proportion des objets réellement présents dans l'image qui ont été correctement détectés par le modèle :

$$\begin{aligned} \text{Precision (P)} &= \frac{\#\text{TP}}{\#\text{TP} + \#\text{FP} + \#\text{MC}} \\ \text{Rappel (R)} &= \frac{\#\text{TP}}{\#\text{TP} + \#\text{FN} + \#\text{MC}} \end{aligned} \tag{3.30}$$

Enfin, les paires (Précision, Rappel), obtenues avec chaque seuil τ, sont tracées pour générer la courbe Précision-Rappel (cf. Figure 3.37) puis l'AP est calculé comme l'aire sous la courbe (via une intégration).

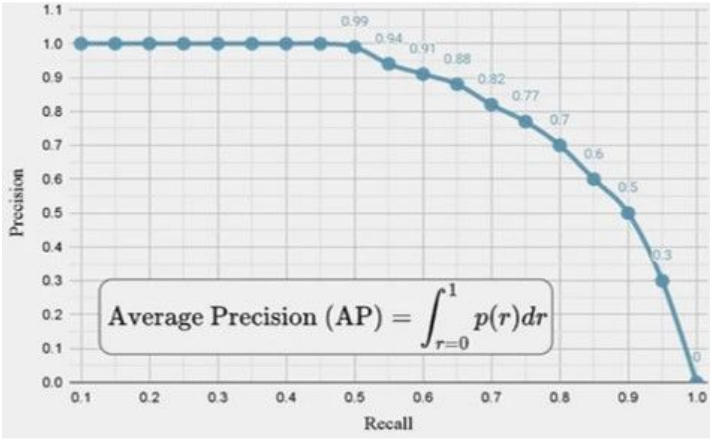


FIGURE 3.37 – Exemple de courbe Precision-Rappel (Precision-Recall).

On obtient donc une valeur d'AP par classe et quand on fait la moyenne des AP sur toutes les classes, on se réfère au mAP (mean Average Precision) :

$$\text{mAP} = \frac{1}{C} \sum_{c=1}^{C} AP_c \tag{3.31}$$

— TIDE (*Type-Informed Detection Error*) [Bolya et al., 2020] : Développé en 2020, TIDE propose une méthodologie pour catégoriser les Faux Positifs (FP) et les Faux Négatifs (FN) en six types d'erreurs distincts. Cette approche permet de créer une distribution significative des erreurs, reflétant les composantes du mAP, en classant systématiquement tous les FP et FN produits par le modèle (cf. Figure 3.38). Il est important de noter que, pour certains types d'erreurs (notamment classification et localisation), un FP peut être associé à un FN. On utilise IoUmax pour désigner le chevauchement maximal d'IoU d'un FP avec une vérité terrain (GT) de la catégorie donnée. Les seuils d'IoU pour le premier plan et l'arrière-plan sont notés respectivement t_f et t_b, fixés par défaut à 0, 5 et 0, 1, sauf indication contraire. Les erreurs sont classées comme suit :

- **Erreur de classification (Cls)** : IoUmax $\geq t_f$ pour une vérité terrain d'une classe incorrecte (c'est-à-dire localisé correctement mais classifié de manière incorrecte).
- **Erreur de localisation (Loc)** : $t_b \leq$ IoUmax $\leq t_f$ pour une vérité terrain d'une classe incorrecte (c'est-à-dire classifié et localisé de manière incorrecte).
- **Erreur de classification et de localisation (Cls+Loc)** : $t_b \leq$ IoUmax $\leq t_f$ pour une vérité terrain d'une classe incorrecte (c'est-à-dire classifié et localisé de manière incorrecte).
- **Erreur de détection dupliquée (Duplicate)** : IoUmax $\geq t_f$ pour une vérité terrain de la classe correcte, mais une autre détection avec un score plus élevé a déjà été associée à cette vérité terrain (c'est-à-dire que la détection serait correcte sans la présence d'une détection mieux notée).
- **Erreur d'arrière-plan (Bkgd)** : IoUmax $\leq t_b$ pour toutes les vérités terrain (c'est-à-dire qu'un arrière-plan est détecté comme un objet du premier plan).
- **Erreur de vérité terrain manquée (Missed)** : Toutes les vérités terrain non détectées (FN) qui ne sont pas déjà couvertes par des erreurs de classification ou de localisation.

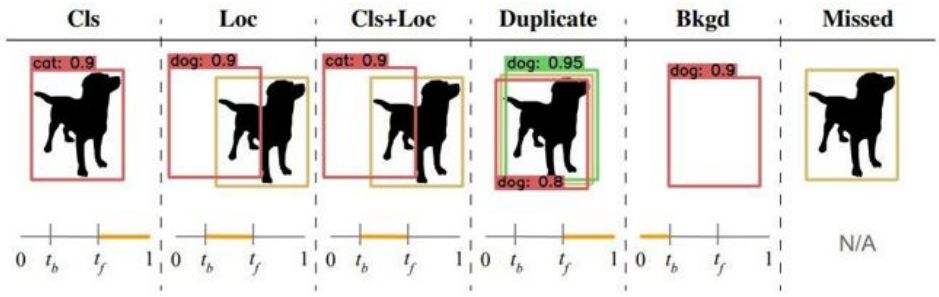


FIGURE 3.38 – Les couleurs des boîtes sont définies comme suit : ■ = détection de faux positif ; ■ = vérité terrain ; ■ = détection de vrai positif. L'IoU avec la vérité terrain pour chaque type d'erreur est indiqué par une surbrillance orange et montré dans la rangée inférieur (dans notre cas $t_b = 0,1$ et $t_f = 0,5$)

En conclusion, la détection d'objets est une tâche particulièrement complexe à évaluer en raison de la diversité des types d'erreurs (faux positifs, faux négatifs, erreurs de classification, erreurs de localisation) et du grand nombre d'hyperparamètres impliqués. Ces hyperparamètres, qu'ils soient liés aux seuils de confiance ou aux critères de correspondance entre les prédictions et les vérités terrain, impactent directement les performances des modèles.

3.3.3 Approches Two-stage

Les méthodes de détection d'objets dites "Two-Stage" constituent une approche en deux étapes qui combine la génération de régions d'intérêt (*Region of Interest*, RoI) avec la classification et la localisation de ces régions pour identifier des objets spécifiques dans une image. Ces algorithmes divisent le processus de détection en deux phases distinctes : d'abord, un module de propositions de régions identifie les zones où les objets pourraient se trouver, et ensuite, un réseau de classification et de régression affine ces propositions en assignant des classes aux objets détectés et en raffinant leurs cadres délimitants.

L'une des premières méthodes de ce type se nomme R-CNN (*Region-based Convolutional Neural Network*) [Girshick et al., 2014]. Développé en 2014, cette dernière repose sur un module de propositions de régions indépendant des classes pour convertir la détection d'objets en un problème de classification et de localisation (cf. Figure 3.39). L'image d'entrée, soustraite de sa moyenne, passe d'abord par un module de propositions de régions, produisant 2000 candidats d'objets à l'aide de la méthode *Selective Search*. Chaque candidat est ensuite normalisé et transmis à un réseau de neurones convolutif (CNN), qui extrait un vecteur de caractéristiques de dimension 4096 pour chaque proposition. Ces vecteurs sont ensuite passés à des machines à vecteurs de support (SVM) spécifiques aux classes pour obtenir des scores de confiance. Une fois la classe identifiée, l'algorithme prédit la boîte englobante de l'objet à l'aide d'un régressseur de boîtes, qui génère les coordonnées centrales, la largeur et la hauteur de la boîte. L'entraînement de R-CNN est lent et complexe, nécessitant plusieurs jours sur des petits ensembles de données.

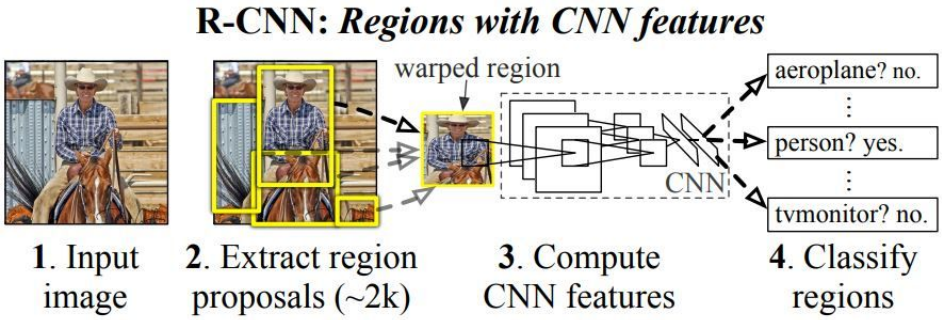


FIGURE 3.39 – Illustration de R-CNN (source [Girshick et al., 2014]).

La même année, l'article [He et al., 2014] propose SPPNet qui introduit l'utilisation d'une couche de *Spatial Pyramid Pooling* (SPP, cf. Figure 3.40) pour traiter des images de tailles ou de ratios différents. SPPNet modifie

la structure de R-CNN en déplaçant les couches de convolution avant le module de propositions de régions et en ajoutant une couche de pooling, rendant ainsi le réseau indépendant de la taille ou du ratio de l'image tout en réduisant les calculs. Le modèle utilise l'algorithme *Selective Search* pour générer des fenêtres candidates, qui sont projetées sur les cartes de caractéristiques obtenues par les couches de convolution. Ces fenêtres sont converties en représentations de longueur fixe via une couche de pooling pyramidal, puis transmises à une couche entièrement connectée et à des SVM pour la classification. SPPNet inclut une couche de post-traitement pour améliorer la localisation avec la régression de boîtes englobantes et utilise un processus d'entraînement en plusieurs étapes, en ne faisant le fine-tuning que sur les couches entièrement connectées. Ce modèle est plus rapide que R-CNN tout en offrant une précision comparable.

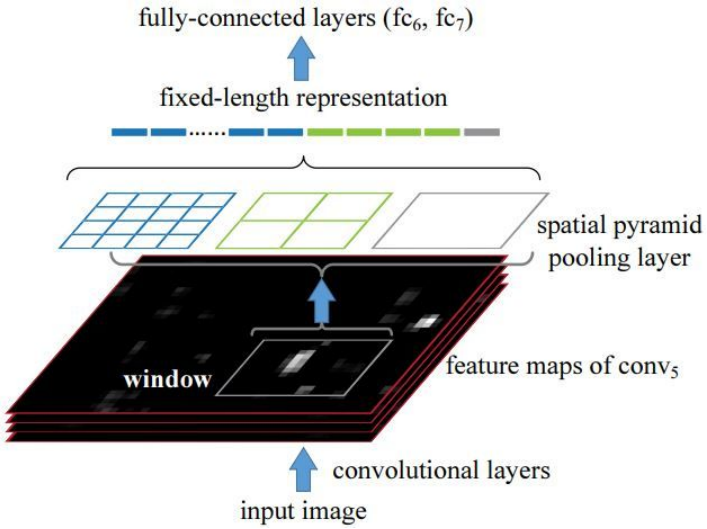


FIGURE 3.40 – Illustration de SPP. Regroupement des caractéristiques à partir de fenêtres arbitraires sur les cartes de caractéristiques. Les cartes de caractéristiques sont calculées à partir de l'image entière. Le regroupement est effectué dans les fenêtres candidates (source [He et al., 2014]).

En 2015, l'article [Girshick, 2015] présente Fast R-CNN qui introduit un système de détection d'objets unique et entraînable de bout en bout (cf. Figure 3.41). Le réseau prend en entrée une image et ses propositions d'objets, puis fait passer l'image à travers des couches de convolution, où les propositions d'objets sont projetées sur les cartes de caractéristiques obtenues. Une seule couche *RoI Pooling* est utilisée pour normaliser les régions d'intérêt, suivie de deux couches entièrement connectées, puis de deux branches : une pour la classification (via une couche SoftMax à N+1 classes) et une autre pour la régression de boîtes englobantes, optimisée avec une fonction de perte smooth L1 pour de meilleures performances. En plus d'améliorer la vitesse (146 fois plus rapide que R-CNN) et la précision, Fast R-CNN simplifie le processus d'entraînement en supprimant la pyramidal pooling et en introduisant une perte multi-tâches. Sans réseau de propositions de régions, ce modèle de détection d'objets offre une vitesse quasi en temps réel avec une précision notable.

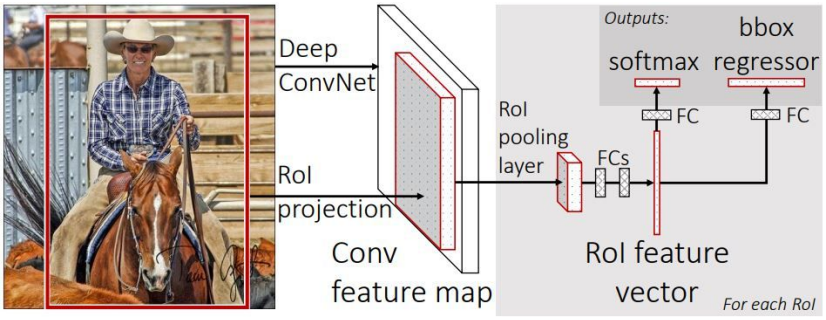
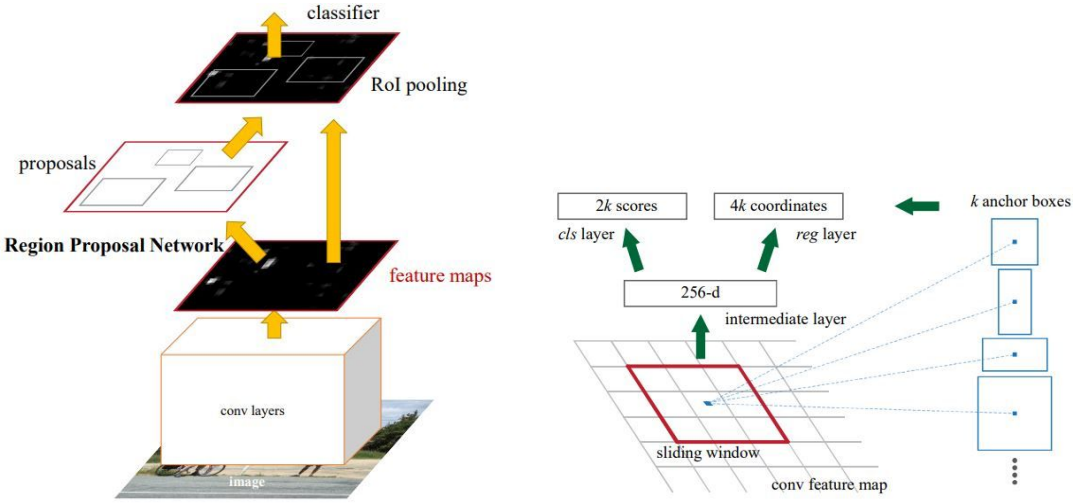


FIGURE 3.41 – Illustration de Fast R-CNN. Une image d'entrée et plusieurs régions d'intérêt (RoIs) sont introduites dans un réseau entièrement convolutionnel. Chaque RoI est mise en commun dans une carte de caractéristiques de taille fixe, puis transformée en un vecteur de caractéristiques par des couches entièrement connectées (FCs). Le réseau produit deux vecteurs de sortie par RoI : des probabilités Softmax et des décalages de régression de boîte englobante pour chaque classe (source [Girshick, 2015]).

Ensuite, [Ren et al., 2016], introduit en 2016, propose une version améliorée de Fast R-CNN qui intègre un réseau de propositions de régions (*Region Proposal Network*, RPN) comme module de propositions de régions. Le RPN prend une image en entrée et génère un ensemble de fenêtres candidates, chacune avec un score d'objectivité indiquant la probabilité de présence d'un objet. Le RPN introduit les boîtes d'ancrage (Anchor Boxes), utilisant plusieurs boîtes de différents ratios pour mieux localiser les objets. L'image est d'abord traitée par le CNN pour obtenir des cartes de caractéristiques, ensuite transmise au RPN qui produit des boîtes englobantes et leurs classifications. Les propositions sélectionnées sont ensuite projetées sur les cartes de caractéristiques de la couche CNN précédente via une couche *RoI pooling*, puis envoyées à une couche entièrement connectée pour la classification et la régression des boîtes englobantes.



(a) Architecture.

(b) Region Proposal Network (RPN).

FIGURE 3.42 – Illustration de Faster R-CNN (source [Ren et al., 2016]).

La même année, [Dai et al., 2016] introduit un réseau entièrement convolutionnel pour la détection d'objets (*Region-based Fully Convolutional Network*, R-FCN), qui optimise les calculs en partageant presque toutes les opérations dans le réseau, contrairement aux précédents détecteurs en deux étapes qui effectuent des calculs intensifs pour chaque proposition. R-FCN utilise des cartes de score sensibles à la position, qui encodent l'information spatiale relative de l'objet et sont ensuite utilisées pour affiner la localisation. R-FCN divise la région d'intérêt en une grille $k \times k$ et évalue la probabilité de chaque cellule par rapport aux classes de détection. Ces scores sont ensuite moyennés pour prédire la classe de l'objet. Le modèle combine un *Region Proposal Network* (RPN) pour identifier les propositions de régions, qui sont ensuite traitées par une couche convolutionnelle et envoyées à un classificateur et un régresseur. La couche de classification associe les cartes sensibles à la position avec les propositions de régions pour générer les prédictions, tandis que le réseau de régression produit les détails

des boîtes englobantes (cf. Figure 3.43).

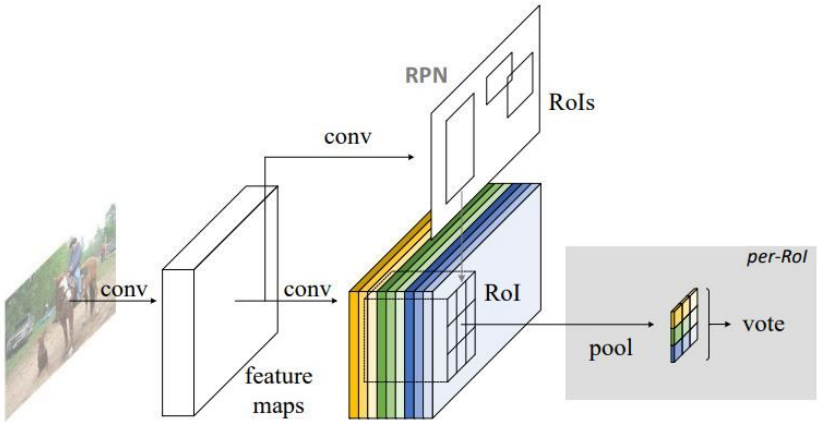


FIGURE 3.43 – Architecture générale de R-FCN. Un Réseau de Proposition de Régions (RPN) propose des RoIs candidates, qui sont ensuite appliquées sur les cartes de scores. Toutes les couches à poids ajustables sont convolutionnelles et sont calculées sur l'image entière ; le coût de calcul par RoI est négligeable (source [Dai et al., 2016]).

En 2017, l'article [Lin et al., 2017a] introduit le *Feature Pyramid Network* (FPN), une architecture en pyramide à double voie avec des connexions latérales pour construire des caractéristiques sémantiques de haut niveau à différentes échelles (cf. Figure 3.44). FPN comporte un cheminement ascendant, un réseau de convolution qui calcule une hiérarchie de caractéristiques à plusieurs échelles, et un cheminement descendant qui remonte les cartes de caractéristiques de niveau supérieur en les adaptant à une résolution plus fine. Les deux chemins sont connectés par des connexions latérales avec des convolutions 1×1 pour enrichir les informations sémantiques des caractéristiques. Cette structure permet d'obtenir des caractéristiques de haut niveau à toutes les échelles, réduisant ainsi le taux d'erreur en détection.

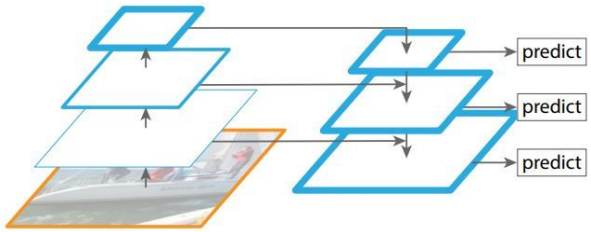


FIGURE 3.44 – Feature Pyramid Network (source [Lin et al., 2017a]).

La même année, avec Mask R-CNN [He et al., 2017] étend Faster R-CNN en ajoutant une branche supplémentaire pour la segmentation d'instances d'objets au niveau des pixels. Cette nouvelle branche est un réseau entièrement connecté appliqué sur les régions d'intérêt (RoIs), permettant de classifier chaque pixel en segments sans coût de calcul significatif. Mask R-CNN utilise l'architecture de base de Faster R-CNN pour les propositions d'objets, mais ajoute une tête de masque en parallèle aux têtes de classification et de régression de boîtes englobantes. Une différence majeure réside dans l'utilisation de la couche RoIAlign au lieu de RoIPool pour éviter les erreurs d'alignement des pixels dues à la quantification spatiale.

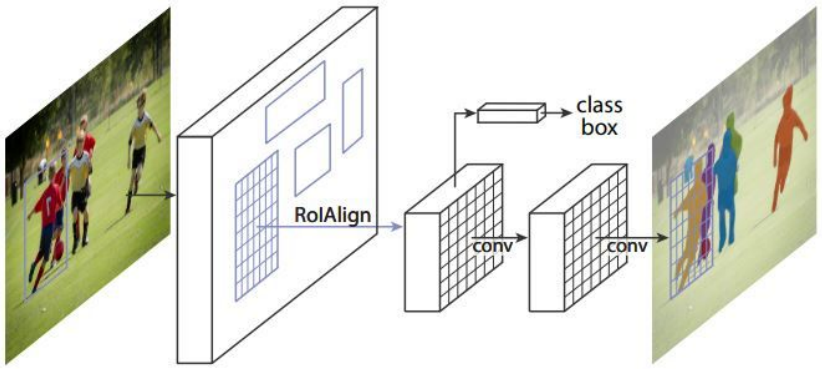


FIGURE 3.45 – Architecture de Mask R-CNN (source [He et al., 2017]).

Ensuite, en 2018, [Cai and Vasconcelos, 2018] propose Cascade R-CNN, une extension de l'architecture à deux étapes de Faster R-CNN avec une détection d'objets en plusieurs étapes. Cascade R-CNN utilise une séquence de détecteurs entraînés avec des seuils d'IoU croissants pour devenir progressivement plus sélectifs et réduire les faux positifs proches. Les détecteurs sont entraînés de manière séquentielle, en exploitant le fait que la sortie d'un détecteur constitue une bonne base pour l'entraînement du détecteur suivant, de qualité supérieure. En 2019, l'article [Qin et al., 2019] propose ThunderNet, un détecteur d'objets générique léger en deux étapes, conçu pour réduire les structures coûteuses des détecteurs de pointe.

ThunderNet utilise un backbone léger nommé SNet, spécialement conçu pour la détection d'objets, et intègre deux blocs d'architecture efficaces : le *Context Enhancement Module* (CEM) et le *Spatial Attention Module* (SAM). Le CEM combine les cartes de caractéristiques à différentes échelles pour exploiter à la fois le contexte local et global, tandis que le SAM utilise l'information issue du réseau de propositions de régions (RPN) pour affiner la distribution des caractéristiques dans la RoI warping (cf. Figure 3.46).

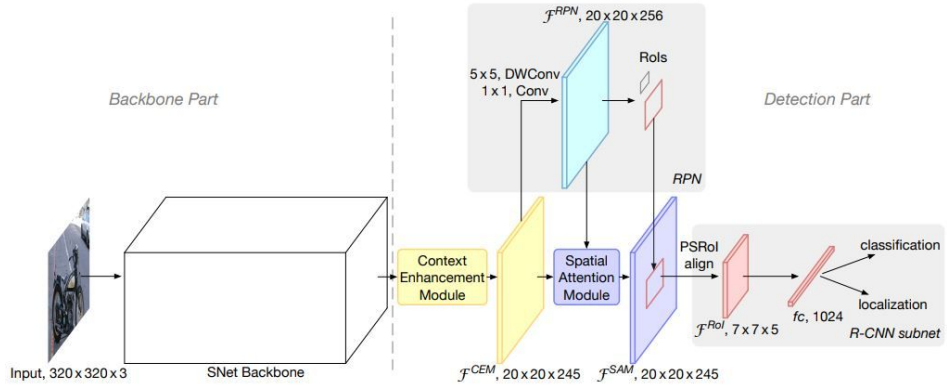


FIGURE 3.46 – L'architecture générale de ThunderNet. ThunderNet utilise une résolution d'entrée de 320×320 pixels. Le backbone SNet est basée sur ShuffleNetV2 et spécialement conçu pour la détection d'objets. Dans la partie détection, le RPN est compressé, et le sous-réseau R-CNN utilise une couche fc de dimension 1024 pour une meilleure efficacité. Le module d'amélioration du contexte exploite les informations sémantiques et contextuelles à partir de multiples échelles. Le module d'attention spatiale introduit les informations du RPN pour affiner la distribution des caractéristiques (source [Qin et al., 2019]).

Enfin, en 2021, [Qiao et al., 2021] propose DetectoRS qui introduit plusieurs innovations pour la détection d'objets. Au niveau macro, le modèle propose le *Recursive Feature Pyramid* (RFP), composé de plusieurs *Feature Pyramid Networks* (FPN) empilés, avec une connexion de rétroaction supplémentaire reliant le chemin de niveau supérieur dans le FPN à la couche de bas niveau. La sortie du FPN est traitée par une couche de *Atrous Spatial Pyramid Pooling* (ASPP) avant d'être transmise à la couche FPN suivante. Un module de fusion est utilisé pour combiner les sorties des différents modules FPN en créant une carte d'attention. Au niveau micro, le *Switchable Atrous Convolution* (SAC) régule le taux de dilatation de la convolution. Une couche de pooling moyenne avec un filtre 5×5 et une convolution 1×1 sert de fonction de commutation pour ajuster le taux de convolution atrous, permettant au réseau de détecter des objets de diverses tailles en temps réel. Le SAC est

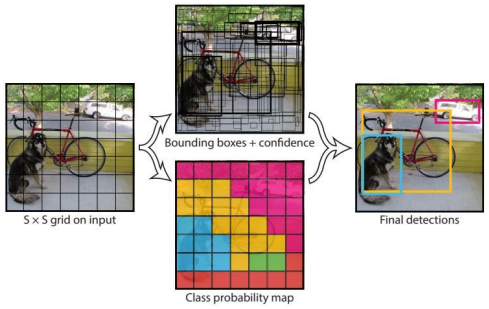
également encadré par deux modules de contexte global pour stabiliser le changement de taux de dilatation. Cependant, ce modèle n'est pas adapté pour des détections en temps réel, car il ne peut traiter qu'environ 4 images par seconde.

Les algorithmes Two-Stage ont évolué pour devenir plus rapides et précis, avec des améliorations ciblant à la fois l'efficacité et la flexibilité d'utilisation sur des images de différentes résolutions et tailles. Ces méthodes offrent une précision élevée et des possibilités d'améliorations continues en combinant différentes architectures, mais la complexité de calcul reste un défi, en particulier pour les applications nécessitant un traitement en temps réel ou sur des appareils à ressources limitées.

3.3.4 Approches Single-stage

Les méthodes de détection d'objets Single-Stage consistent à prédire simultanément les boîtes englobantes et les classes d'objets dans une image en une seule passe à travers le réseau, sans étapes intermédiaires pour proposer des régions candidates. Ces méthodes sont particulièrement adaptées aux applications en temps réel, car elles sont rapides et peuvent fonctionner sur des dispositifs à faibles ressources, bien qu'elles puissent parfois sacrifier un peu de précision par rapport aux approches en deux étapes (Two-Stage).

La famille la plus populaire de ce type de détection se nomme YOLO (*You Only Look Once*). L'article [Redmon et al., 2016] introduit en 2016 YOLOv1, qui unifie les étapes de détection d'objets en prédisant simultanément toutes les boîtes englobantes. YOLO divise l'image en une grille $S \times S$ et associe chaque cellule à la détection d'un objet, prédisant B boîtes englobantes pour chaque classe, avec un score de confiance pour C classes par cellule de grille. YOLO reformule la détection comme un problème de régression, prédisant directement les objets et leurs attributs de boîte englobante. Une cellule de la grille prédit plusieurs boîtes, chaque prédiction comprenant les coordonnées du centre (x, y), les dimensions (w, h) et le score de confiance. YOLO utilise 24 couches convolutionnelles suivies de 2 couches entièrement connectées pour les prédictions, avec des activations Leaky ReLU sauf pour la dernière couche linéaire. Pendant l'entraînement, chaque cellule de grille prédit une seule classe pour faciliter la convergence, mais peut en prédire davantage en inférence. Une fonction de coût multitâche optimise tous les composants prédits, et la suppression des non-maxima (NMS) élimine les détections multiples par classe.



(a) Processus de détection.

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$
$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

(b) Fonction de coût multitâche.

FIGURE 3.47 – Illustration du modèle YOLOv1 (source [Redmon et al., 2016]).

Ensuite, en 2017, l'article [Redmon and Farhadi, 2017] propose YOLOv2, une amélioration de YOLO avec un compromis ajustable entre vitesse et précision. YOLOv2 intègre plusieurs techniques : la normalisation par lot (*Batch Normalization*) pour améliorer la convergence, l'entraînement conjoint des systèmes de classification et de détection pour augmenter le nombre de classes détectées, la suppression des couches entièrement connectées pour accroître la vitesse, et l'utilisation de boîtes d'ancrage apprises pour améliorer le rappel et disposer de meilleures prioritisations.

Par la suite, en 2018, [Farhadi and Redmon, 2018] présente YOLOv3 une version améliorée de YOLO avec plusieurs optimisations. YOLOv3 utilise Darknet-53 comme réseau extracteur de caractéristiques (backbone), un réseau plus grand pour des prédictions plus précises. Il intègre également des techniques telles que l'augmentation de données, l'entraînement multi-échelle et la normalisation par lot. Le classificateur Softmax est remplacé

par un classificateur logistique, et YOLOv3 fait des prédictions multi-échelles, avec trois boîtes prédictives à trois échelles différentes. YOLOv3 est plus rapide que YOLOv2 tout en offrant de meilleures performances.

Plus tard, en 2020, [Bochkovskiy et al., 2020] propose YOLOv4 et introduit plusieurs améliorations sans publication formelle visant à optimiser la vitesse et la précision de détection d'objets. La nouvelle architecture modifie Darknet-53 avec des connexions partielles (CSPNet) pour réduire les calculs sans perte de précision, utilisant la fonction d'activation Mish comme base. La partie Neck intègre une version modifiée du *Spatial Pyramid Pooling* (SPP) pour augmenter le champ réceptif, ainsi qu'un *Path Aggregation Network* (PANet) modifié à la place de FPN, avec un module d'attention spatiale (SAM) adapté. Pour la détection, des ancres comme celles de YOLOv3 sont employées. YOLOv4 introduit aussi le *Bag of Freebies* (BoF) pour un entraînement avancé, utilisant la mosaïque pour augmenter les données en combinant quatre images en une seule, améliorant la détection hors contexte et réduisant les besoins en mini-lots pour la normalisation. La régularisation utilise DropBlock au lieu de Dropout et un lissage des étiquettes de classe. La perte CIoU et la normalisation cross-mini-batch (CmBN) sont également ajoutées. YOLOv4 optimise ses hyperparamètres grâce aux algorithmes génétiques pour une performance optimale.

En 2020, YOLOv5, développé par l'entreprise Ultralytics [1] sans publication formelle, apporte comme principale amélioration l'intégration du processus de sélection des boîtes d'ancrage dans le modèle, ce qui permet d'apprendre automatiquement les boîtes d'ancrage optimales pour chaque jeu de données sans configuration manuelle. Bien qu'il ne soit pas accompagné d'un article scientifique, YOLOv5 a gagné en crédibilité en étant utilisé avec succès dans de nombreuses applications.

Ensuite, plusieurs variantes ont été développées en 2021. Scaled-YOLOv4 [Wang et al., 2021a] a introduit des techniques de mise à l'échelle vers le haut pour accroître la précision au détriment de la vitesse, et vers le bas pour améliorer la vitesse en réduisant la précision, permettant ainsi une utilisation sur des systèmes embarqués à faible puissance de calcul. YOLOR [Wang et al., 2021b] (*You Only Learn One Representation*) a proposé une approche d'apprentissage multi-tâches visant à unifier les tâches de classification, de détection et d'estimation de pose au sein d'un seul modèle. En apprenant une représentation générale et en utilisant des sous-réseaux pour les tâches spécifiques, YOLOR intègre des connaissances implicites, favorisant l'optimisation des performances pour chaque tâche. Enfin, YOLOX [Ge et al., 2021], basé sur YOLOv3 d'Ultralytics, a introduit cinq améliorations principales : une architecture sans ancre, un échantillonnage centré pour identifier plusieurs positifs, une tête découplée pour séparer les tâches de classification et de régression (cf. Figure 3.48), une assignation avancée des étiquettes, et des augmentations de données avancées (MixUP et mosaïque), surpassant ainsi les performances des modèles YOLO précédents.

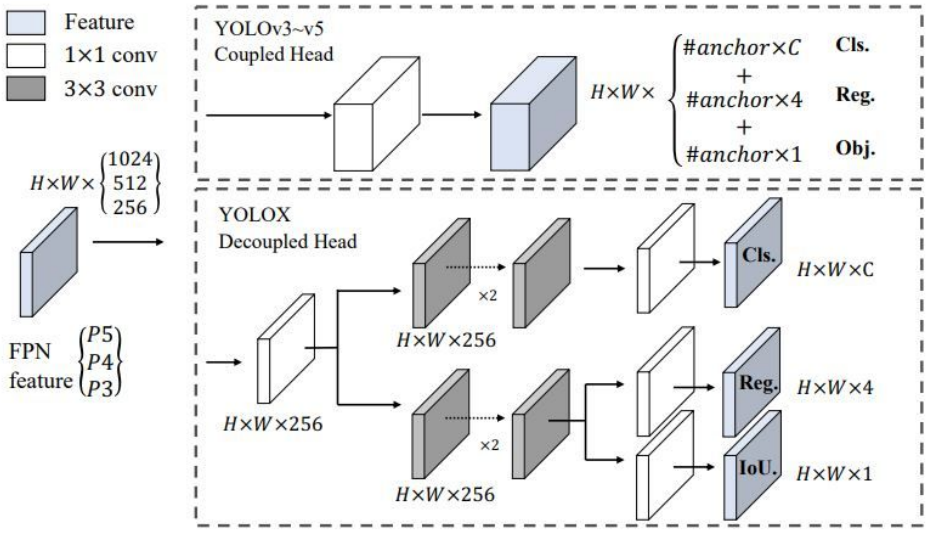


FIGURE 3.48 – Illustration de la différence entre la tête YOLOv3 et la tête découplée proposée par YOLOX (source [Ge et al., 2021]).

Plus tard, YOLOv6 [Li et al., 2022], bien que non publié officiellement, introduit plusieurs innovations. Il utilise un nouveau backbone appelé EfficientRep, offrant un parallélisme accru. Pour le Neck, il adopte un PAN

1. https ://docs.ultralytics.com/

amélioré avec des RepBlocks ou CSPStackRep Blocks pour les modèles de grande taille, et inclut une tête découplée efficace. Les pertes de classification et de régression sont également revues, utilisant la perte VariFocal pour la classification et SIoU/GIoU pour la régression. Une stratégie d'auto-distillation est appliquée aux tâches de régression et de classification, et un schéma de quantification avec RepOptimizer et une distillation par canal pour accélérer le détecteur (cf. Figure 3.49). YOLOv6 surpasse les modèles de pointe précédents, comme YOLOv5 et YOLOX.

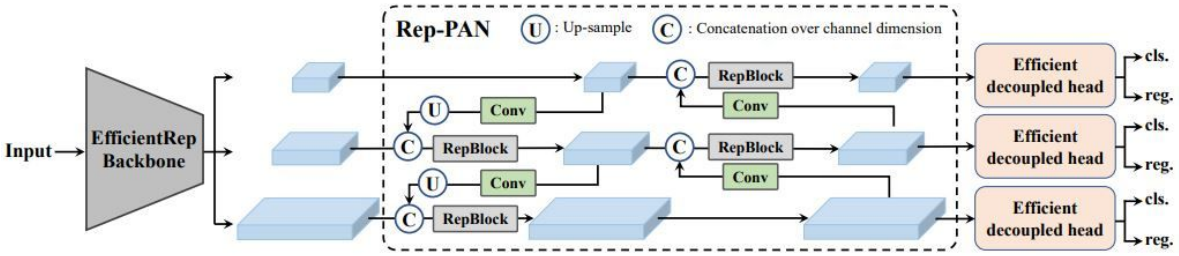


FIGURE 3.49 – Architecture de YOLOv6 (source [Li et al., 2022]).

Enfin, la dernière version publiée au démarrage de cette thèse, YOLOv7 [Wang et al., 2023a], introduit des changements architecturaux et une série de méthodes bag-of-freebies qui augmentent la précision sans affecter la vitesse d'inférence, mais en augmentant le temps d'entraînement. YOLOv7 propose un réseau d'agrégation de couches efficaces étendu (E-ELAN), permettant au modèle d'apprendre et de converger plus efficacement en optimisant le chemin de gradient le plus long et le plus court. E-ELAN combine des caractéristiques de groupes différents par un mélange et une fusion de la cardinalité pour améliorer l'apprentissage sans perturber le chemin de gradient d'origine. De plus, YOLOv7 introduit une nouvelle stratégie de mise à l'échelle pour les modèles basés sur la concaténation, ajustant la profondeur et la largeur des blocs avec le même facteur pour maintenir une structure optimale. Par rapport à YOLOv4, YOLOv7 réduit de 75% les paramètres et de 36% les calculs, tout en améliorant la précision moyenne. Par rapport à YOLOR, YOLOv7 diminue également le nombre de paramètres et la charge de calcul.

Depuis, de multiples autres variantes de YOLO ont été proposées. DAMO-YOLO [Xu et al., 2022] utilise une recherche d'architecture neuronale (NAS) pour optimiser automatiquement sa structure et intègre un Efficient-RepGFPN pour le Neck, ainsi qu'une approche ZeroHead pour la Head. YOLOv8 [Jocher et al., 2023], sorti par Ultralytics en 2023, utilise une architecture similaire à YOLOv5 avec le module C2f et un modèle sans ancre, améliorant la précision de détection avec des pertes CIoU et DFL. YOLO-NAS [yol,] se concentre sur la détection d'objets petits, en utilisant une quantification hybride pour maintenir l'équilibre entre latence et précision sur des dispositifs en périphérie. YOLOv9 [Wang et al., 2024b] introduit le concept de *Programmable Gradient Information* (PGI) pour adapter les réseaux profonds à plusieurs objectifs, accompagné du réseau léger GELAN. YOLOv10 [Wang et al., 2024a] élimine le besoin de NMS grâce à une stratégie de double assignation et propose une conception efficace de bout en bout pour améliorer les performances tout en réduisant la redondance calculatoire. Enfin, YOLOv11 [Khanam and Hussain, 2024] par Ultralytics, intègre des composants tels que le bloc C3k2 et le SPPF, permettant des améliorations en extraction de caractéristiques et des capacités étendues pour la segmentation d'instances, l'estimation de pose, et la détection d'objets orientés, tout en optimisant le compromis entre le nombre de paramètres et la précision.

Outre la grande famille YOLO, plusieurs autres modèles *Single-stage* se distinguent pour la détection d'objets. L'article [Liu et al., 2016] introduit ainsi SSD (cf. Figure 3.50), un modèle de détection en une seule étape basé sur VGG-16, enrichi de couches auxiliaires en fin de réseau pour renforcer les performances. Ces couches de convolution supplémentaires, de tailles progressivement décroissantes, permettent à SSD de détecter les petits objets dans les premières étapes du réseau, lorsque les caractéristiques extraites de l'image sont encore assez précises, tandis que les couches plus profondes gèrent les ajustements des boîtes par défaut et les ratios d'aspect. Pendant l'entraînement, chaque boîte de vérité terrain est associée aux boîtes par défaut présentant le meilleur chevauchement de Jaccard (mesure identique à l'IoU), et le modèle est affiné en conséquence. SSD s'appuie également sur un échantillonnage négatif difficile (*hard negative mining*) et une augmentation de données intensive, et utilise une somme pondérée des pertes de localisation et de confiance pour l'entraînement. Enfin, la suppression des non-maxima (NMS) est appliquée pour obtenir les résultats finaux.

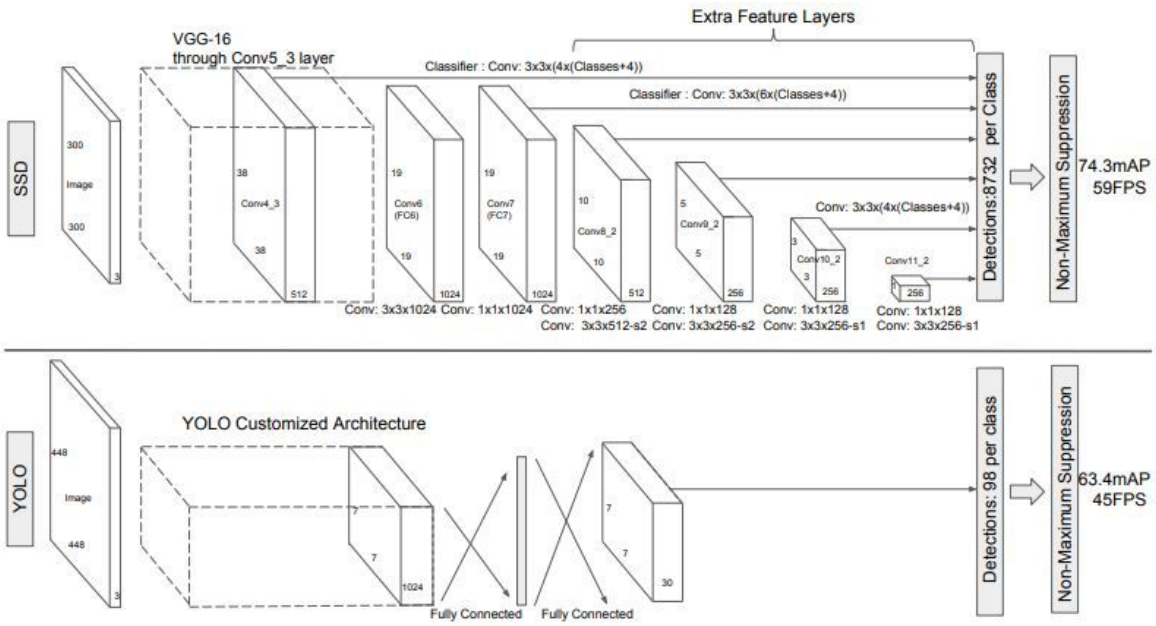


FIGURE 3.50 – Comparaison entre les modèles de détection SSD et YOLO (source [Liu et al., 2016]).

Dans le même esprit, l'article [Lin et al., 2017b] présente RetinaNet, un détecteur conçu pour résoudre le déséquilibre extrême entre les classes avant-plan (*foreground*) et arrière-plan (*background*), problème limitant souvent les performances des détecteurs Single-stage. Pour ce faire, il introduit la Focal Loss, une fonction de perte modifiée pour réduire l'impact des exemples faciles, permettant ainsi une meilleure gestion de ce déséquilibre. RetinaNet repose sur ResNet comme backbone, associé à un Feature Pyramid Network (FPN) et à deux sous-réseaux pour la classification et la régression des boîtes englobantes. Ce modèle est simple à entraîner, converge rapidement, et introduit des avancées dans l'optimisation des détecteurs grâce à sa nouvelle fonction de perte. En 2019, l'article [Tian et al., 2019] propose FCOS, un détecteur entièrement convolutionnel en une seule étape, adoptant une approche de prédiction par pixel, similaire à la segmentation sémantique. FCOS se distingue en éliminant les boîtes d'ancrage et les propositions d'objets, simplifiant ainsi les calculs et réduisant la dépendance aux hyperparamètres associés aux boîtes d'ancrage, souvent critiques pour les performances de détection. L'article [Zhou et al., 2019] innove en représentant chaque objet par un point central plutôt que par une boîte englobante. Dans CenterNet, l'image d'entrée est traitée par un réseau de neurones entièrement convolutionnel (FCN) pour générer une carte de chaleur où les pics correspondent aux centres des objets détectés. Le modèle dispose de trois têtes (*Heads*) : une pour la carte de chaleur (identifiant le centre de l'objet), une pour estimer sa taille, et une pour corriger la position du point central. Une perte multitâche, combinant les trois têtes, est rétropropagée pour l'entraînement. Lors de l'inférence, les sorties de la tête de décalage déterminent la position du point central, permettant de générer une boîte englobante sans avoir besoin de suppression des non-maxima (NMS). Dans un registre similaire, l'article [Law and Deng, 2020] introduit une approche basée sur des points clés en identifiant les boîtes englobantes via une paire de coins (coin supérieur gauche et coin inférieur droit) détectés par un réseau convolutionnel unique. En utilisant cette approche, CornerNet élimine le besoin de boîtes d'ancrage, et l'article propose également le corner pooling, un nouveau type de couche de pooling facilitant la localisation précise des coins des objets (cf. Figure 3.51).

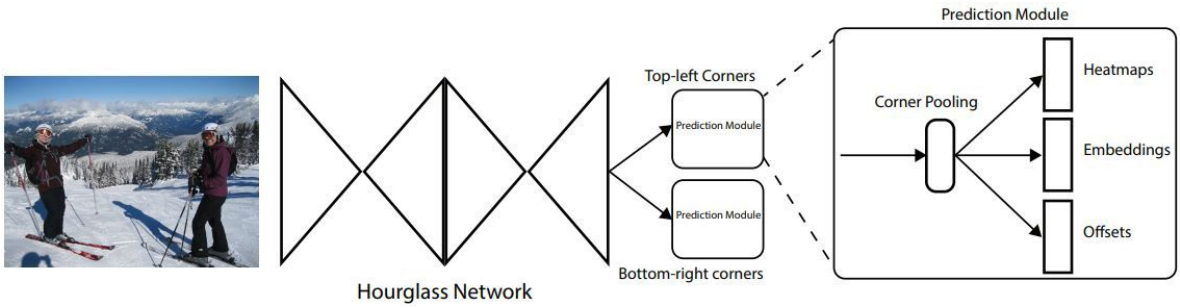


FIGURE 3.51 – Vue d'ensemble de CornerNet. Le réseau backbone est suivi de deux modules de prédiction, l'un pour les coins supérieurs gauches et l'autre pour les coins inférieurs droits. En utilisant les prédictions des deux modules, nous localisons et regroupons les coins (source [Law and Deng, 2020]).

Enfin, l'article [Tan et al., 2020] se démarque par l'introduction de fonctionnalités multi-échelles efficaces grâce au réseau pyramidal de caractéristiques bidirectionnel (BiFPN), qui utilise des poids apprenables pour connecter les caractéristiques à différentes échelles. EfficientDet propose également un coefficient de mise à l'échelle permettant de redimensionner simultanément toutes les dimensions du modèle : backbone, BiFPN, réseau de classification/boîtes et résolution. Utilisant EfficientNet comme backbone, EfficientDet empile plusieurs couches BiFPN pour extraire les caractéristiques, puis envoie chaque sortie de la dernière couche BiFPN aux réseaux de prédiction de classes et de boîtes (cf. Figure 3.52). Le modèle est entraîné avec l'optimiseur SGD, une normalisation de lot synchronisée, et utilise l'activation Swish, plus performante et efficace que ReLU.

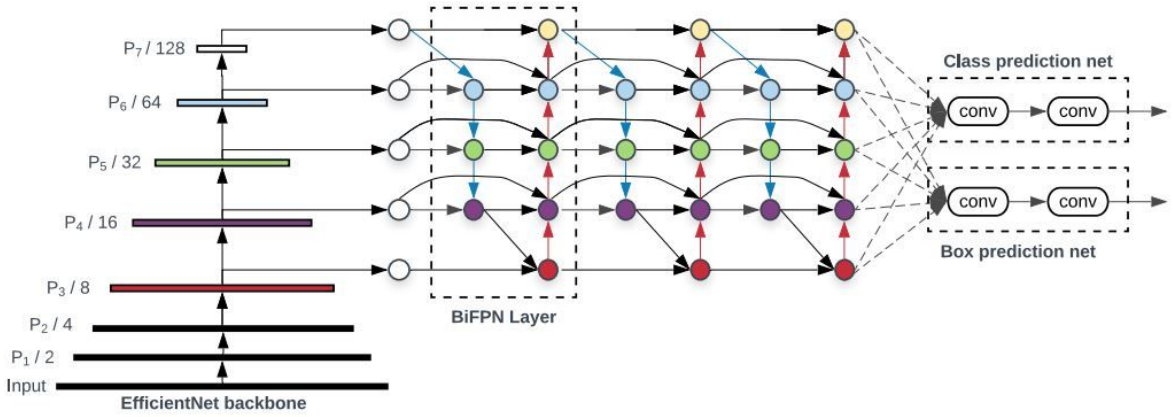


FIGURE 3.52 – Architecture de EfficientDet (source [Tan et al., 2020]).

Les modèles de détection *Single-stage* sont largement utilisés pour leurs performances en temps réel. Cependant, ils présentent parfois un compromis entre rapidité et précision, que les dernières itérations de YOLO et d'autres architectures comme EfficientDet et RetinaNet cherchent à optimiser.

3.3.5 Approches basées sur les Transformers

Les méthodes de détection d'objets basées sur les Transformers [Vaswani et al., 2017] représentent une nouvelle approche pour les tâches de vision par ordinateur, en exploitant la capacité des Transformers à modéliser des relations complexes entre les éléments de l'image. Contrairement aux modèles de détection d'objets traditionnels qui reposent souvent sur des architectures convolutives (CNN) et des composants conçus manuellement comme les ancres (anchors) et la suppression non maximale (NMS), les méthodes basées sur les Transformers abordent la détection comme un problème de prédiction d'ensembles, simplifiant ainsi le pipeline de détection.

Pour ce type de détection, la famille la plus populaire de détecteurs se nomme DETR (DEtection TRansformer) [Carion et al., 2020]. Introduit en 2020, DETR propose une nouvelle approche qui considère la détection d'objets comme un problème de prédiction d'ensemble direct, simplifiant ainsi le pipeline de détection en supprimant de nombreux composants conçus manuellement, tels que la suppression des non-maxima (NMS) et la

génération d'ancres, qui intègrent habituellement des connaissances préalables sur la tâche. Les éléments clés de DETR sont une architecture d'encodeur-décodeur basée sur des transformers et une perte globale basée sur les ensembles, qui impose des prédictions uniques grâce à un appariement bipartite. Avec un ensemble fixe de requêtes d'objets apprises, DETR analyse les relations entre les objets et le contexte global de l'image pour produire directement le jeu final de prédictions en parallèle. DETR est conceptuellement simple et ne nécessite aucune bibliothèque spécialisée, contrairement à de nombreux détecteurs modernes. Grâce aux calculs non-locaux du transformer, DETR montre des performances nettement meilleures sur les objets de grande taille, bien qu'il soit moins performant pour les petits objets. Il démontre ainsi une précision et une rapidité d'exécution comparables au modèle bien établi et optimisé Faster R-CNN.

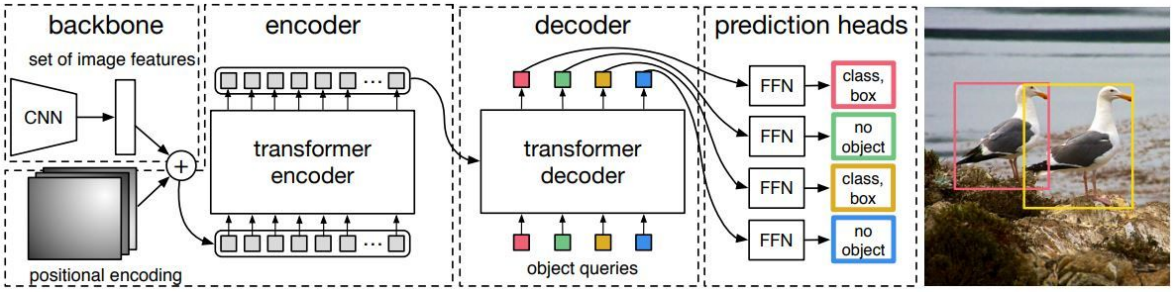


FIGURE 3.53 – Vue d'ensemble de DETR. Ce dernier utilise un réseau CNN classique en backbone pour apprendre une représentation 2D d'une image d'entrée. Le modèle aplatit cette représentation et lui ajoute un encodage positionnel avant de la passer dans un encodeur Transformer. Un décodeur Transformer prend ensuite en entrée un petit nombre fixe d'embeddings positionnels appris, appelés requêtes d'objet, et se concentre également sur la sortie de l'encodeur. Chaque embedding de sortie du décodeur est transmis à un réseau de neurones à propagation directe (FFN) partagé, qui prédit soit une détection (classe et boîte englobante), soit une classe « aucun objet » (source [Carion et al., 2020]).

Par la suite, une multitude de variantes de DETR ont été développées pour améliorer la détection d'objets. Par exemple, Deformable DETR [Zhu et al., 2021] atténue les problèmes de convergence lente et de complexité élevée de DETR en rendant le modèle plus efficace. UP-DETR [Dai et al., 2021] introduit un pré-entraînement non supervisé avec une tâche de détection de patchs aléatoires pour mieux initier l'apprentissage. Conditional DETR [Meng et al., 2021b] utilise une requête spatiale conditionnelle pour accélérer la convergence en ajustant le décodage par une attention multi-têtes. D'autres modèles, comme OV-DETR [Zang et al., 2022], permettent la détection d'objets en vocabulaire ouvert, permettant d'identifier n'importe quel objet en fonction de son nom de classe ou d'une image exemplaire. Anchor DETR [Wang et al., 2022c] propose des requêtes basées sur des points d'ancrage, ciblant les objets à proximité, tandis que Meta-DETR [Zhang et al., 2022a] est le premier détecteur d'objets de type few-shot basé sur l'exploitation des corrélations inter-classes pour améliorer la robustesse des détections. AO2-DETR [Dai et al., 2022] se concentre sur la détection d'objets orientés sans besoin d'ancres multiples ni de post-traitement complexe. Des variantes plus spécialisées incluent X-DETR [Cai et al., 2022b], intégrant une architecture vision-langage avec un alignement efficace pour les tâches de détection et de compréhension du langage, et OW-DETR [Gupta et al., 2022], qui utilise l'attention pseudo-étiquetée pour la détection d'objets en monde ouvert. DEYO [Ouyang, 2022] combine une architecture à deux étapes, fusionnant un modèle de détection classique et un modèle de type DETR, pour améliorer les performances et l'efficacité. Les extensions récentes comme DEYOv2 [Ouyang, 2023a] et DEYOv3 [Ouyang, 2023b] améliorent encore la formation et la précision par un raisonnement progressif et un entraînement étape par étape. Q-DETR [Xu et al., 2023a] propose une quantification en bits faibles pour une détection efficace, tandis que DINO [Zhang et al., 2023a] améliore la localisation grâce à une stratégie de débruitage contrastif. Group DETR [Chen et al., 2023a] introduit une assignation en groupes pour une formation plus rapide, et Cascade DETR [Ye et al., 2023] se concentre sur une attention en cascade pour une détection universelle de haute qualité. Des variantes comme Rank-DETR [Pu et al., 2024] optimisent la détection avec une architecture orientée vers le classement et une fonction de perte priorisant la précision, et Semi-DETR [Zhang et al., 2023b] introduit une approche semi-supervisée pour la détection d'objets. Des adaptations pour des applications spécifiques incluent ARS-DETR [Zeng et al., 2024], sensible aux ratios d'aspect pour la détection d'objets dans les images aériennes, et RT-DETR [Zhao et al., 2024b], un modèle DETR optimisé pour la détection en temps réel, surpassant les performances des modèles YOLO.

D'autres modèles de détecteurs intègrent également des Transformers. Par exemple, le Pyramid Vision Transformer (PVT) [Wang et al., 2021d], introduit en 2021, est un backbone basé exclusivement sur des Transformers, conçu pour les tâches de prédiction dense et remplaçant ainsi les CNN. PVT surmonte les limitations des Transformers classiques en utilisant de petits patchs d'image (4×4 pixels) pour capturer les caractéristiques haute résolution essentielles aux prédictions denses. Il incorpore une pyramide de réduction progressive, qui diminue la longueur des séquences à mesure que le réseau s'approfondit, réduisant ainsi les coûts de calcul. Un module d'attention avec réduction spatiale (Spatial Reduction Attention, SRA) optimise l'apprentissage des caractéristiques haute résolution. Combiné à DETR, PVT permet un système de détection d'objets de bout en bout sans recours aux convolutions ou aux composants manuellement conçus, comme les ancres ou la suppression des non-maxima (NMS).

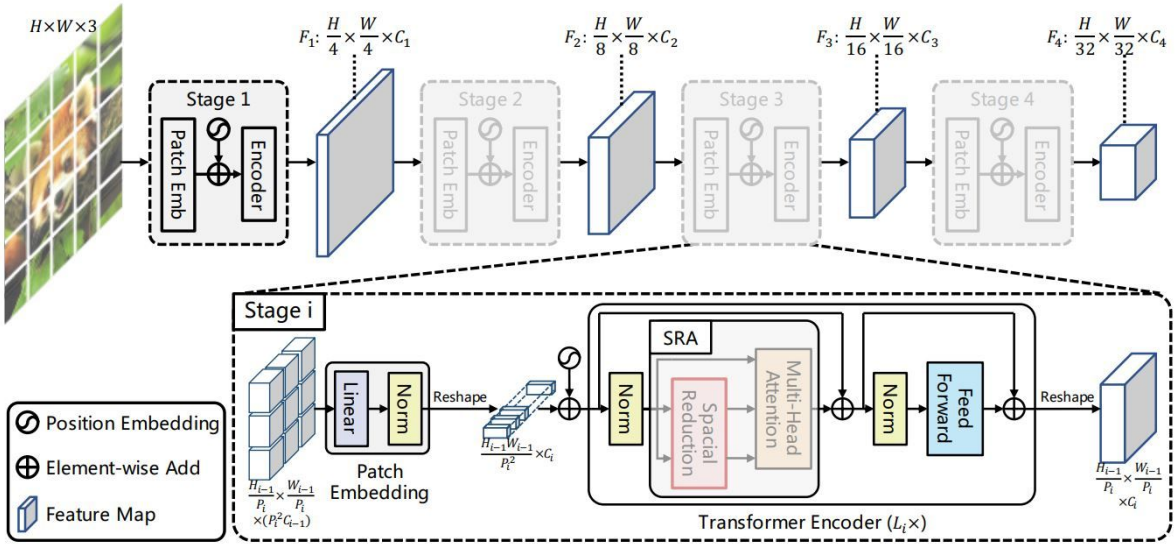


FIGURE 3.54 – Architecture générale du *Pyramid Vision Transformer* (PVT). Le modèle complet est divisé en quatre étapes, chacune composée d'une couche d'incorporation de patchs et d'un encodeur Transformer de L_i couches. Suivant une structure pyramidale, la résolution de sortie des quatre étapes diminue progressivement, passant d'une haute résolution (stride de 4) à une basse résolution (stride de 32) (source [Wang et al., 2021d]).

Un autre modèle, ViTAE [Xu et al., 2021] (*Vision Transformer Advanced by Exploring Intrinsic Inductive Bias*), propose une version avancée des Transformers de vision en intégrant un biais inductif pour modéliser les structures visuelles locales et gérer la variabilité d'échelle. Contrairement aux Transformers classiques, qui traitent les images comme des séquences 1D et nécessitent de grandes quantités de données d'entraînement pour apprendre implicitement ce biais, ViTAE utilise des modules de réduction pyramidale spatiale pour intégrer des contextes multi-échelles à travers des convolutions à différents taux de dilatation. Chaque couche Transformer de ViTAE incorpore un bloc convolutif en parallèle avec le module d'attention multi-tête, améliorant ainsi l'invariance d'échelle et la robustesse des représentations. L'extension ViTAEv2 [Zhang et al., 2023e] étend ViTAE en une conception multi-étapes avec 644 millions de paramètres, permettant un apprentissage efficace des caractéristiques multi-échelles, tout en maintenant une vitesse d'inférence rapide et une gestion de mémoire optimisée pour les images de grande taille. ViT-YOLO [Zhang et al., 2021b] est un modèle hybride combinant YOLOv4 et Transformer. Sa structure est divisée en trois parties : D'abord, le backbone MHSA-Darknet, intégrant l'attention multi-tête dans CSP-Darknet pour extraire des caractéristiques distinctives ; ensuite, un BiFPN (*Bidirectional Feature Pyramid Network*) remplace PANet pour fusionner les caractéristiques multi-niveaux ; enfin, des têtes de détection YOLO prédisent les boîtes de détection à cinq échelles. Grâce à l'attention multi-tête pour capturer le contexte global et aux techniques de fusion de caractéristiques multi-échelles, ViT-YOLO améliore la précision et la robustesse de la détection d'objet. En parallèle, YOLOS [Fang et al., 2021] est une série de modèles de détection basés sur l'architecture ViT (*Vision Transformer*), avec un minimum de modifications. Pour adapter ViT à la détection d'objets, YOLOS remplace le token de classification [CLS] par cent tokens [DET] pour la détection et utilise une perte de correspondance bipartite, permettant une prédiction d'ensemble sans conversion des séquences en cartes de caractéristiques 2D ni ajout manuel d'informations spatiales. La tête de prédiction reste aussi simple qu'une couche de classification, ce qui simplifie le modèle tout en maintenant une précision

compétitive. AdaMixer [Gao et al., 2022b] est un détecteur d'objets rapide et basé sur des requêtes, conçu pour accélérer la convergence grâce à une adaptabilité améliorée lors du décodage. AdaMixer utilise des requêtes pour échantillonner les caractéristiques de manière adaptative dans l'espace et à différentes échelles, en ajustant les décalages pour cibler efficacement les régions pertinentes des objets. Les caractéristiques échantillonnées sont ensuite décodées dynamiquement avec un MLP-Mixer adaptatif, optimisant la détection. Continual Detection Transformer (CL-DETR) [Liu et al., 2023b] aborde le défi de l'oubli catastrophique lors de l'ajout de nouvelles catégories d'objets au fil du temps. CL-DETR utilise une perte de distillation des connaissances pour préserver les prédictions informatives des versions précédentes du modèle, en filtrant les prédictions redondantes de fond et en garantissant la compatibilité avec les nouvelles étiquettes. Une stratégie de calibration assure une correspondance entre les statistiques d'entraînement et de test, améliorant ainsi la détection continue. Enfin, CAT [Ma et al., 2023] est un modèle de détection basé sur les Transformers pour la détection d'objets en monde ouvert (*Open-World Object Detection*, OWOD), visant à identifier aussi bien des objets connus qu'inconnus, et à apprendre progressivement à reconnaître les objets inconnus. CAT utilise un décodeur en cascade pour séparer les processus de localisation et d'identification et intègre un mécanisme de pseudo-étiquetage auto-adaptatif qui combine les étiquetages basés sur le modèle et l'entrée, générant des pseudo-étiquettes robustes pour mieux détecter les objets inconnus.

Les méthodes de détection d'objets basées sur les Transformers présentent plusieurs avantages : elles simplifient le pipeline de détection en éliminant des composants conçus manuellement et exploitent la capacité des Transformers à modéliser les relations spatiales globales. Cependant, elles peuvent être coûteuses en calcul et peuvent nécessiter des ajustements pour gérer efficacement les petits objets ou les objets obstrués. Ces approches constituent néanmoins une avancée importante vers une détection d'objets plus flexible et généralisée, en ouvrant la voie à des modèles de vision entièrement basés sur des Transformers.

3.3.6 Conclusion

Le développement de modèles de détection d'objets est un domaine en pleine expansion, avec des approches variées visant à répondre aux besoins spécifiques de précision, rapidité et complexité. Dans l'industrie automobile, où les contraintes de temps réel et de sécurité sont primordiales pour les applications de conduite assistée et autonome, les modèles de détection d'objets privilégiés sont ceux offrant un compromis optimal entre précision, vitesse d'inférence et efficacité en ressources. Plusieurs modèles sont couramment utilisés pour leurs caractéristiques adaptées à ces besoins. Les modèles two-stage, comme Faster R-CNN, sont appréciés pour leur précision dans des environnements complexes ou pour la détection d'objets de petite taille, bien que leur temps de calcul plus élevé limite leur utilisation. Les modèles single-stage, en particulier YOLO et SSD, sont largement populaires dans l'industrie automobile pour leur rapidité et leur efficacité en ressources. YOLOv4 et YOLOv5 sont fréquemment employés pour la détection en temps réel d'éléments critiques comme les piétons, les véhicules et les panneaux de signalisation. SSD, bien que moins performant que YOLO pour certains types de scènes, est apprécié pour sa rapidité et sa capacité à bien détecter les objets de taille moyenne. Les modèles basés sur des Transformers, tels que DETR, gagnent en popularité grâce à leur capacité à capturer des relations globales dans l'image et à éliminer des composants manuels comme les ancres. DETR et ses variantes, qui améliorent la convergence et la détection des petits objets, sont explorés pour des applications nécessitant une grande précision et une flexibilité pour détecter des objets dans divers contextes. Cependant, leur coût calculatoire et leur lenteur de convergence limitent encore leur déploiement dans des applications strictement en temps réel (ou proche).

Dans le cadre de cette étude, l'accent sera mis sur le développement d'une méthodologie permettant d'obtenir des garanties de performance. Afin de simplifier l'approche, les modèles Faster R-CNN, RetinaNet et YOLOv7 (la dernière version disponible au début de cette thèse) seront utilisés comme références dans les expérimentations et analyses à venir.

3.4 Génération d'image

La dernière partie de cet état de l'art, relative à notre application industrielle de *Field Monitoring*, concerne la génération d'images par deep learning. Pour obtenir des garanties de performances fiables sur notre système de compression-détection, l'idéal serait de tester un ensemble exhaustif d'images afin d'estimer les probabilités d'erreur et de déterminer les scènes ou situations susceptibles d'entraîner des erreurs. Cependant, une telle approche n'est pas réalisable en pratique, car nous n'avons accès qu'à un ensemble de données (images) limité. Une alternative consiste à utiliser des modèles génératifs pour acquérir des images supplémentaires. Ces modèles

permettent non seulement de générer des images similaires à celles de notre base de données d'entraînement, mais également, en exploitant leur nature du deep learning, de générer des images orientées spécifiquement vers des cas susceptibles de mettre en défaut notre système de détection, permettant ainsi une meilleure couverture des scénarios de test critiques.

Les modèles génératifs visent à capturer la distribution sous-jacente d'un ensemble de données d'entraînement, dans le but de générer de nouveaux échantillons qui imitent fidèlement cette distribution, en reproduisant autant que possible les détails fins des données d'origine. Ces modèles se divisent principalement en deux grandes catégories : les modèles basés sur la vraisemblance (*likelihood-based*) et ceux sans vraisemblance (*likelihood-free*). Les modèles basés sur la vraisemblance maximisent explicitement la probabilité des données d'entraînement, tandis que les modèles sans vraisemblance capturent la distribution par un jeu d'opposition (mini-max) entre deux réseaux. Plus précisément, les modèles génératifs basés sur la vraisemblance comprennent les *Variational AutoEncoders* (VAEs) et les *Diffusion Probabilistic Models* (DPMs), alors que les modèles sans vraisemblance regroupent principalement les *Generative Adversarial Networks* (GANs).

Dans le chapitre précédent, nous avons déjà évoqué les modèles génératifs dans le cadre de la compression d'image apprise (Chapitre 3.2.9). Dans ce chapitre, nous allons approfondir les approches de génération par VAE, GAN et diffusion, examiner l'utilisation de ces modèles dans le contexte de la génération d'images pour l'automobile, et enfin, explorer comment il est possible d'orienter la génération pour cibler spécifiquement les cas les plus critiques pour notre système de compression-détection.

3.4.1 Métriques

L'objectif de l'apprentissage génératif est que le modèle produise des données qui correspondent aux données observées. Ainsi, chaque distance entre la probabilité d'observer des données réelles $p_w(\cdot)$ et la probabilité de générer des données du modèle $p(\cdot)$ peut servir de mesure de performance pour le modèle génératif.

— **IS** (Inception Score [Salimans et al., 2016]) : corrélée avec le jugement humain, cette métrique repose sur la probabilité prédite des classes d'images à l'aide d'un modèle pré-entraîné, souvent un réseau Inception. L'Inception Score est défini comme :

$$\mathrm{IS} = \exp\left(\mathbb{E}_{x\sim p(x)}\left[D_{KL}\left(p(y|x) \parallel p(y)\right)\right]\right) \tag{3.32}$$

où $p(x)$ est la distribution des données générées, $p(y|x)$ est la probabilité prédite par le modèle Inception que l'image x appartienne à la classe y, $p(y) = \int_x p(y|x)p(x)\mathrm{dx}$ est la distribution marginale des classes sur les images générées et D_{KL} est la divergence de Kullback-Leibler (KL). Si $p(y|x)$ est fortement concentré pour chaque image, cela montre que les images générées sont nettes et bien définies. Si $p(y)$ est uniforme sur toutes les classes, cela indique que les images générées couvrent une large gamme de classes, reflétant une bonne diversité. En conséquence, un IS élevé reflète des images à la fois de haute qualité et couvrant une large gamme de classes. En termes d'ordre de grandeur, pour des datasets comme CIFAR-10 (32×32, 10 classes), un score IS autour de 5 à 9 est considéré comme élevé, indiquant que le modèle génère des images réalistes et diversifiées. Pour des datasets plus complexes comme ImageNet (224×224, $1,000$ classes), un IS supérieur à 10 est attendu pour des modèles performants, avec des modèles de pointe atteignant des scores au-delà de 100. Un inconvénient de l'Inception Score est qu'il n'utilise pas les statistiques des échantillons réels ni ne les compare à celles des échantillons synthétiques.

— **FID** (Frechet Inception Distance [Heusel et al., 2017]) : Améliore l'Inception Score. L'égalité $p(\cdot) = p_w(\cdot)$ est valable sauf pour un ensemble nom mesurable si et seulement si :

$$\int p(\cdot)f(x)\mathrm{dx} = \int p_w(\cdot)f(x)\mathrm{dx} \tag{3.33}$$

pour une base $f(\cdot)$ couvrant l'espace des fonctions où $p(\cdot)$ et $p_w(\cdot)$ résident. Ces égalités d'espérances sont utilisées pour décrire les distributions à travers des moments ou des cumulants, où $f(x)$ sont des polynômes des données x. On généralise ces polynômes en remplaçant x par la couche de codage d'un modèle Inception afin d'obtenir des caractéristiques pertinentes pour la vision. Pour des raisons pratiques, nous ne considérons que les deux premiers moments : la moyenne et la covariance. La gaussienne est la distribution à maximum d'entropie pour une moyenne et une covariance données. Nous supposons donc que les unités de codage suivent une distribution gaussienne multidimensionnelle. La différence entre deux distributions gaussiennes (images synthétiques et images réelles) est mesurée par la distance de Fréchet, également connue sous le nom de Wasserstein-2 distance. Nous appelons la distance de Fréchet $d(\cdot,\cdot)$ entre la gaussienne de moyenne (m, C) obtenue à partir de $p(\cdot)$ et la gaussienne de moyenne

(m_w, C_w) obtenue à partir de $p_w(\cdot)$, la Fréchet Inception Distance (FID), définie par :

$$\text{FID} = d^2\left((m, C), (m_w, C_w)\right) = \| m - m_w \|_2^2 + \text{Tr}\left(C + C_w - 2(CC_w)^{1/2}\right) \tag{3.34}$$

La FID est cohérente avec l'augmentation des perturbations et les jugements humains, mais est connue pour être sensible aux petits détails d'implémentation.

— **KID** (Kernel Inception Distance [Bińkowski et al., 2018]) : Métrique similaire à la FID, définie comme le carré de la Maximum Mean Discrepancy (MMD) entre les représentations extraites par le modèle Inception. On utilise un noyau polynomial défini par :

$$k(x, y) = \left(\frac{1}{d} x^T y + 1\right)^3 \tag{3.35}$$

où d est la dimension des représentations. Cela peut également être interprété comme une MMD appliquée directement sur les images d'entrée avec le noyau défini comme suit :

$$K(x, y) = k(\phi(x), \phi(y)) \tag{3.36}$$

où ϕ est la fonction qui mappe les images vers leurs représentations dans l'espace des caractéristiques du modèle Inception. Contrairement à la FID, le KID ne suppose pas une forme paramétrique pour la distribution des activations. De plus, avec le noyau cubique utilisé, le KID compare non seulement la moyenne et la variance, mais également l'asymétrie (*skewness*) des distributions.

— **Precision/Recall** [Kynkäänniemi et al., 2019] : Métrique améliorée de précision et rappel pour les distributions. l'idée principale est de former des représentations explicites non paramétriques des variétés (*manifolds*) des données réelles et générées, à partir desquelles la précision et le rappel peuvent être estimés (cf. Figure 3.55). On échantillonne des données réelles $X_r \sim P_r$ et des données générées $X_g \sim P_g$, Φ_r et Φ_g, puis nous les projetons dans un espace de caractéristiques de haute dimension à l'aide d'un classificateur pré-entraîné. Les vecteurs de caractéristiques des images réelles et générées sont notés ϕ_r et ϕ_g, respectivement, et les ensembles correspondants sont Φ_r et Φ_g. On échantillonne un nombre égal d'exemples de chaque distribution, c'est-à-dire $|\Phi_r| = |\Phi_g|$. Pour chaque ensemble de vecteurs de caractéristiques $\Phi \in \{\Phi_r, \Phi_g\}\}$, on estime la variété correspondante dans l'espace des caractéristiques. Nous obtenons cette estimation en calculant les distances euclidiennes entre tous les vecteurs de l'ensemble, puis, pour chaque vecteur, en formant une hypersphère avec un rayon égal à la distance à son k-ième plus proche voisin. Ces hypersphères définissent un volume dans l'espace des caractéristiques, qui sert d'estimation de la vraie variété. Pour déterminer si un échantillon donné ϕ se trouve dans ce volume, on définit une fonction binaire :

$$f(\phi, \Phi) = \begin{cases} 1, & \text{si } \|\phi - \phi_0\|^2 \leq \|\phi_0 - \text{NN}_k(\phi_0, \Phi)\|^2 \text{ pour au moins un } \phi_0 \in \Phi \\ 0, & \text{sinon.} \end{cases} \tag{3.37}$$

où $\text{NN}_k(\phi_0, \Phi)$ renvoie le k-ième plus proche voisin de ϕ_0 dans l'ensemble Φ. Intuitivement, $f(\phi, \Phi_r)$ permet de vérifier si une image générée ressemble à une image réelle, tandis que $f(\phi, \Phi_g)$ vérifie si une image réelle pourrait être reproduite par le générateur. La précision et le rappel sont définis comme suit :

$$\text{precision}(\Phi_r, \Phi_g) = \frac{1}{|\Phi_g|} \sum_{\phi_g \in \Phi_g} f(\phi_g, \Phi_r) \tag{3.38}$$

$$\text{recall}(\Phi_r, \Phi_g) = \frac{1}{|\Phi_r|} \sum_{\phi_r \in \Phi_r} f(\phi_r, \Phi_g) \tag{3.39}$$

La précision est calculée en vérifiant, pour chaque image générée, si elle appartient à la variété estimée des images réelles et le rappel est calculé en vérifiant, pour chaque image réelle, si elle appartient à la variété estimée des images générées. Pour une image donnée, on calcule son vecteur de caractéristiques ϕ en la passant dans un classificateur pré-entraîné, tel que VGG-16 ou Inception-v3, et en extrayant le vecteur d'activation après la deuxième couche entièrement connectée. La taille du voisinage k, est un compromis entre couvrir toute la variété (valeurs élevées de k). En général $k = 3$ est une valeur robuste pour éviter la saturation des métriques. On utilise $|\Phi| = 50,000$ conformément à la pratique standard pour la métrique FID.

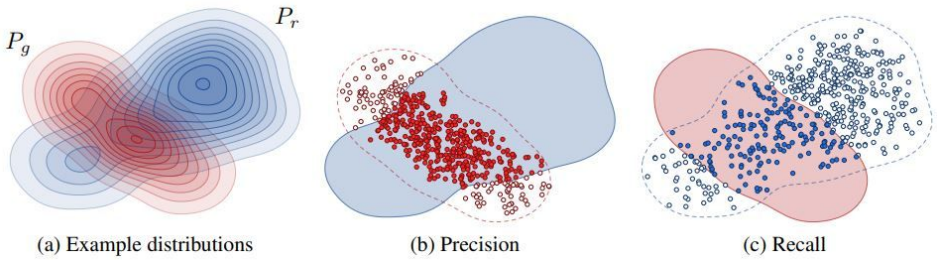


FIGURE 3.55 – Définition de la précision et du rappel pour les distributions : (b) La précision est la probabilité qu'une image aléatoire de P_g se trouve dans le support de P_r, (c) Le rappel est la probabilité qu'une image aléatoire de P_r se trouve dans le support de P_g (source [Kynkäänniemi et al., 2019]).

— **sFID** [Nash et al., 2021] : La sFID est équivalente à la FID, mais utilise des caractéristiques spatiales intermédiaires dans le réseau Inception plutôt que les caractéristiques regroupées spatialement utilisées dans la FID standard.
— **Autres métriques** : L'entraînement et l'évaluation des modèles génératifs font également appel à des métriques couramment utilisées pour la compression d'image (cf. Section 3.2.2), telles que le PSNR, le MS-SSIM ou encore le LPIPS.

3.4.2 Approches à base de VAE

Les Variational AutoEncoders (VAEs) sont des modèles génératifs qui visent à apprendre une représentation latente des données, permettant de capturer leur distribution sous-jacente et d'obtenir une compression qui préserve les caractéristiques essentielles de l'information. Plus précisément, les VAEs cherchent à établir un mapping probabiliste entre les données d'entrée et un espace latent en maximisant la vraisemblance des données tout en minimisant la divergence KL (Kullback-Leibler) entre la distribution latente apprise et une distribution a priori, généralement une distribution normale standard. L'entraînement des VAEs repose sur l'optimisation d'une borne inférieure variationnelle (ou ELBO) sur le log-vraisemblance des données, composée de deux termes : une perte de reconstruction, qui évalue la capacité du modèle à recréer les données d'entrée, et une perte de divergence KL, qui contraint la distribution latente à rester proche de la distribution a priori. Formellement, l'objectif des VAEs s'écrit comme suit :

$$L_{\text{VAE}} = -D_{\text{KL}}\left(q_\phi(z|x) \parallel p_\theta(z)\right) + \mathbb{E}_{z\sim q_\phi(z|x)}[\log\left(p_\theta\left(x|z\right)\right)], \tag{3.40}$$

où $q_\phi(z|x)$ représente une approximation de la probabilité postérieure, et $p_\theta\left(x|z\right)$ est la log-vraisemblance des données d'entraînement. L'optimisation de cette fonction de perte se fait par rapport aux paramètres des réseaux de neurones encodeur et décodeur (cf. Figure 3.56). Une fois entraîné, le décodeur des VAEs permet de générer de nouvelles images en échantillonnant depuis la distribution a priori et en décodant les variables latentes obtenues. Comparés aux GANs, les VAEs sont plus stables à entraîner grâce à leur estimation de la log-vraisemblance. Cependant, les images générées par les VAEs peuvent souvent apparaître floues ou bruitées, en raison de l'injection de bruit et des limitations dans la reconstruction au niveau des pixels. Par ailleurs, le décalage entre la distribution a priori (par exemple, une distribution gaussienne) et les données limitées d'entraînement peut rendre l'optimisation des VAEs difficile, entraînant une qualité de synthèse limitée et une instabilité de l'entraînement, notamment en cas de données limitées. Pour cette raison, les VAEs peuvent être moins adaptés à la synthèse d'images dans des scénarios avec peu de données d'entraînement.

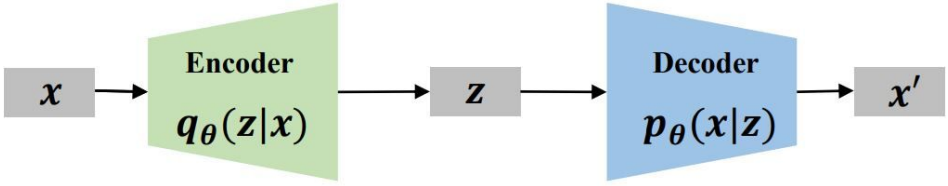


FIGURE 3.56 – Fonctionnement du VAE (source [Yang and Wang, 2023]).

Introduit en 2015, le CVAE [Sohn et al., 2015] est une extension conditionnelle du VAE classique. Ce modèle apprend une représentation structurelle conditionnée par des observations, permettant de capturer la dépen-

dance entre les entrées et les sorties. L'optimisation est réalisée via le cadre de *Stochastic Gradient Variational Bayes* (SGVB), et plusieurs stratégies, comme l'injection de bruit en entrée et des objectifs d'entraînement multi-échelle, sont introduites pour améliorer la robustesse des prédictions. Également introduit en 2015, le modèle DRAW [Gregor et al., 2015] combine les principes des VAEs avec des réseaux récurrents et un mécanisme d'attention. Contrairement aux VAEs classiques, DRAW produit des images en plusieurs étapes successives, permettant de raffiner progressivement les échantillons générés. L'encodeur et le décodeur échangent des informations via des séquences de codes latents, améliorant ainsi leur coordination. Une extension notable, alignDRAW [Mansimov et al., 2016], intègre un mécanisme d'attention conditionnelle pour générer des images à partir de descriptions textuelles, permettant une généralisation à des scènes inédites. Le VQVAE [van den Oord et al., 2017] introduit une représentation latente discrète basée sur une quantification vectorielle (cf. Figure 3.57), résolvant des problèmes tels que le *posterior collapse* souvent observé dans les VAEs. Cette approche permet une modélisation efficace des structures de grande dimension dans les données, comme les objets dans les images ou les phonèmes dans le discours, en évitant de gaspiller de la capacité sur le bruit et les détails imperceptibles. Son extension, VQVAE2 [Razavi et al., 2019], améliore les priors autorégressifs pour générer des images synthétiques de haute fidélité.

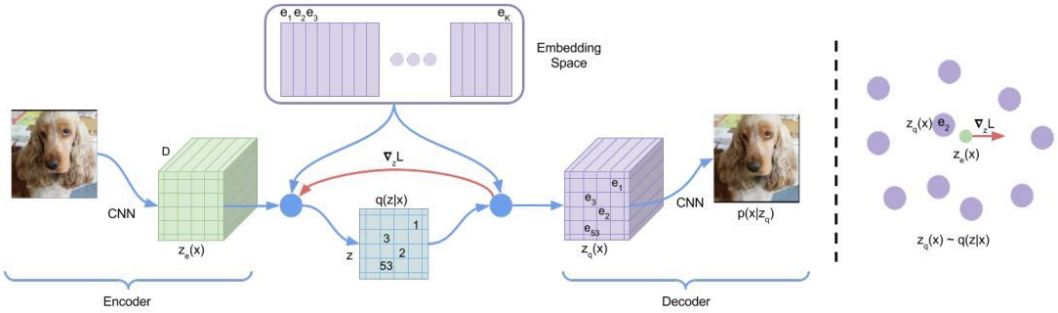


FIGURE 3.57 – À gauche : Une figure décrivant le VQ-VAE. À droite : Visualisation de l'espace d'incorporation. La sortie de l'encodeur z est mappée au point le plus proche e_2. Le gradient $\nabla_z L$ (en rouge) poussera l'encodeur à modifier sa sortie, ce qui pourrait altérer la configuration lors du passage suivant (source [van den Oord et al., 2017]).

Enfin, IntroVAE [Huang et al., 2018] combine les forces des VAEs et des GANs, en introduisant un apprentissage introspectif où le modèle estime lui-même les différences entre les données réelles et les échantillons générés. Cette méthode permet de produire des images photographiques haute résolution tout en maintenant les avantages des VAEs, comme la stabilité d'entraînement et une interprétation claire de l'espace latent. L'extension Soft-IntroVAE [Daniel and Tamar, 2021] remplace la hinge loss par une perte exponentielle lissée, améliorant la stabilité d'entraînement et permettant une analyse théorique approfondie.

Les VAEs offrent plusieurs avantages, notamment une stabilité d'entraînement, une génération d'images diversifiée et une structure latente utile pour des tâches telles que l'édition ou l'interpolation d'images. Cependant, ils ont tendance à produire des images moins réalistes que les GANs en raison de la contrainte imposée sur la distribution latente. Des efforts, comme les VQVAE et IntroVAE, visent à combler cet écart en combinant des caractéristiques des GANs et des VAEs, permettant ainsi une synthèse d'images de haute qualité tout en conservant les propriétés interprétables des VAEs.

3.4.3 Approches à base de GAN

Les Generative Adversarial Networks (GANs) [Goodfellow et al., 2014], introduits par Ian Goodfellow et ses collaborateurs en 2014, constituent l'une des approches les plus populaires et puissantes pour la génération d'images par deep learning. Ils sont généralement constitués de deux sous-réseaux : un générateur et un discriminateur. Ces deux sous-réseaux sont entraînés de manière adversariale, où le générateur tente de tromper le discriminateur en produisant des images difficiles à distinguer des images réelles, tandis que le discriminateur cherche à identifier correctement si une image est réelle ou générée (cf. Figure 3.58). Ce processus est optimisé par un jeu mini-max à deux joueurs entre le générateur et le discriminateur. Formellement, l'optimisation se fait

par :

$$L_D = -\mathbb{E}_{x\sim p_x}\left[\log\left(D(x,c)\right)\right] - \mathbb{E}_{z\sim p_z}\left[\log\left(1 - D\left(G(z,c)\right)\right)\right] \tag{3.41}$$

$$L_G = -\mathbb{E}_{z\sim p_z}\left[\log\left(D\left(G(z,c)\right)\right)\right] \tag{3.42}$$

où $D(.)$ et $G(.)$ désignent respectivement le discriminateur et le générateur, x représente les échantillons réels et $G(z)$ les échantillons générés, c est une condition supplémentaire pour la génération conditionnelle, qui est optionnelle pendant l'entraînement. En visant à atteindre un équilibre de Nash entre le générateur et le discriminateur, les GANs sont notoirement difficiles à entraîner. Cela peut entraîner divers problèmes comme la disparition des gradients, le mode collapse et la divergence de l'entraînement. Pour améliorer la stabilité de l'entraînement et les performances des GANs, de nombreuses recherches ont été menées, principalement axées sur la conception de variantes de fonctions de perte et d'architectures.

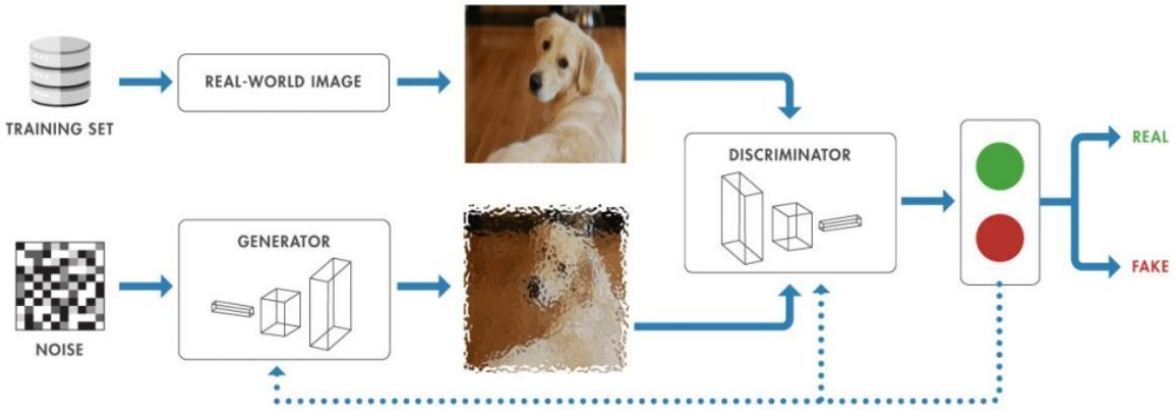


FIGURE 3.58 – Schéma de l'entraînement du GAN. Les flèches en pointillé représentent la mise à jour des poids via le gradient de l'erreur (source https://blogs.mathworks.com/deep-learning/2021/12/02/synthetic-image-generation-using-gans/).

Les GANs originaux ont posé les bases de nombreux développements, notamment les Conditional GANs (cGANs) [Mirza and Osindero, 2014] introduits en 2014, qui conditionnent la génération sur des informations supplémentaires comme des classes ou des étiquettes. Cette extension permet de contrôler le contenu des échantillons générés, ouvrant la voie à des applications comme la génération conditionnelle. En 2015, les *Laplacian Pyramid GANs* (LAPGAN) [Denton et al., 2015] ont introduit une génération hiérarchique en utilisant une pyramide laplacienne pour produire des images de manière progressive, du grossier au fin (cf. Figure 3.59). Cette approche a permis d'améliorer la qualité des échantillons en décomposant la tâche de génération en niveaux successifs.

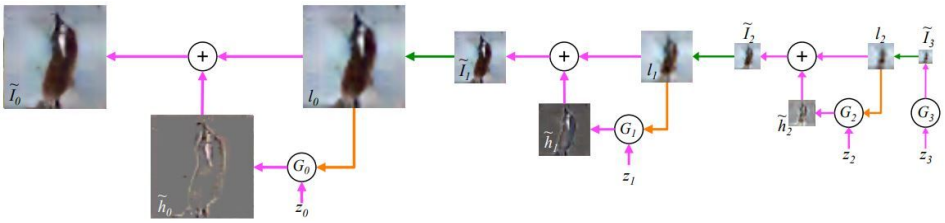


FIGURE 3.59 – Procédure d'échantillonnage du modèle LAPGAN. Le modèle commence avec un échantillon de bruit z_3 (coté droit) et utilisons un modèle génératif G_3 pour générer $\tilde{I}_3$. Celui-ci est sur-échantillonné (flèche verte) puis utilisé comme variable de conditionnement l_2 (flèche orange) pour le modèle génératif au niveau suivant, G_2. En combinaison avec un autre échantillon de bruit z_2, G_2 génère une image de différence $\tilde{h}_2$, qui est ajoutée à l_2 pour créer $\tilde{I}_2$. Ce processus se répète sur deux niveaux supplémentaires pour produire un échantillon final à pleine résolution I_0 (source [Denton et al., 2015]).

Les *Deep Convolutional GANs* (DCGANs) [Radford et al., 2016], proposées en 2016, ont introduit des architectures convolutionnelles spécifiques qui stabilisent l'entraînement des GANs et permettent un apprentissage

non supervisé efficace. DCGAN a démontré que les réseaux convolutionnels génératifs peuvent apprendre des représentations hiérarchiques, allant des parties d'objets aux scènes complètes. D'autres innovations ont suivi, comme les InfoGAN [Chen et al., 2016], qui maximisent l'information mutuelle entre les variables latentes et les observations, permettant d'apprendre des représentations interprétables et démêlées. Ces modèles ont ouvert la voie à une génération contrôlée et explicable. En 2017, des approches comme Pix2Pix [Isola et al., 2017] et CycleGAN [Zhu et al., 2017] ont exploré la traduction d'images, avec ou sans correspondances d'échantillons. Pix2Pix utilise des cGANs pour traduire des images en fonction d'une entrée donnée, tandis que CycleGAN exploite une cohérence cyclique pour traduire entre des domaines sans paires d'entraînement explicites, facilitant des tâches comme le transfert de style et la transformation d'objets (cf. Figure 3.60).

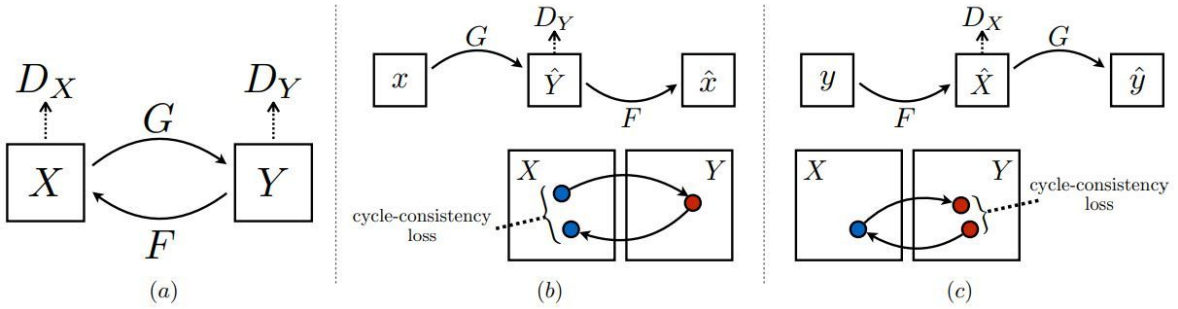


FIGURE 3.60 – (a) CycleGAN contient deux fonctions de mapping, $G : X \to Y$ et $F : Y \to X$, ainsi que des discriminateurs adversaires associés D_Y et D_X. D_Y incite G à traduire X en des sorties indiscernables du domaine Y, et inversement pour D_X et F. Pour régulariser davantage les mappings, le modèle introduit deux pertes de cohérence cyclique. (b) perte de cohérence cyclique directe : $x \to G(x) \to F(G(x)) \approx x$, et (c) perte de cohérence cyclique inverse : $y \to F(y) \to G(F(y)) \approx y$ (source [Zhu et al., 2017]).

L'année 2018 a marqué une étape importante avec les *Progressive Growing GANs* [Karras et al., 2018] (ProGANs), qui introduisent un entraînement progressif en augmentant progressivement la résolution des images générées. Cette méthode stabilise l'apprentissage et améliore la qualité et la diversité des échantillons. En 2019, les *Self-Attention GANs* (SAGANs) [Zhang et al., 2019] ont intégré un mécanisme d'attention dans les GANs pour modéliser des dépendances à longue portée dans les images, permettant de mieux coordonner les détails visuels. De même, StyleGAN a proposé une nouvelle architecture génératrice basée sur un espace latent intermédiaire, permettant de démêler les facteurs de variation et de produire des images haute fidélité (cf. Figure 3.61). Ses extensions, StyleGAN2 [Karras et al., 2020], StyleGAN3 [Karras et al., 2021] et StyleGAN-XL [Sauer et al., 2022] ont introduit des améliorations significatives en termes de qualité, stabilité et invariance des transformations spatiales.

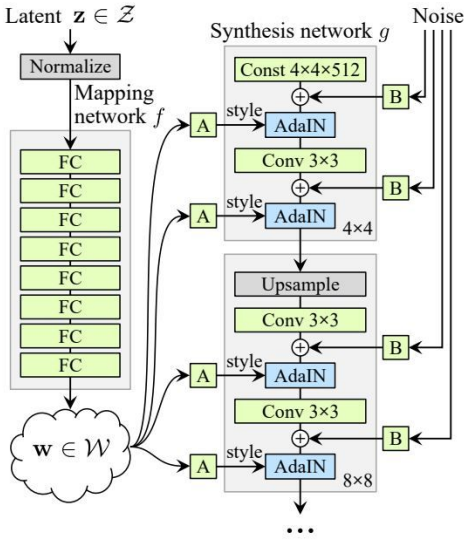


FIGURE 3.61 – Générateur dans StyleGAN. Alors qu'un générateur traditionnel transmet le code latent uniquement à travers la couche d'entrée, StyleGAN mappe d'abord l'entrée vers un espace latent intermédiaire W, qui contrôle ensuite le générateur via une normalisation adaptative par instance (AdaIN) à chaque couche de convolution. Du bruit gaussien est ajouté après chaque convolution, avant d'appliquer la non-linéarité (source [Karras et al., 2019]).

Plus récemment, des architectures hybrides combinant Transformers et GANs ont émergé. Par exemple, VQ-GAN [Esser et al., 2021] combine des encodeurs convolutionnels pour apprendre des dictionnaires visuels riches et des Transformers pour modéliser les relations à longue portée, offrant un contrôle précis sur la génération. TransGAN [Jiang et al., 2021] et ViTGAN [Lee et al., 2022c] (cf. Figure 3.62) explorent des GANs basés uniquement sur des Transformers, avec des adaptations pour stabiliser l'entraînement et générer des images haute résolution.

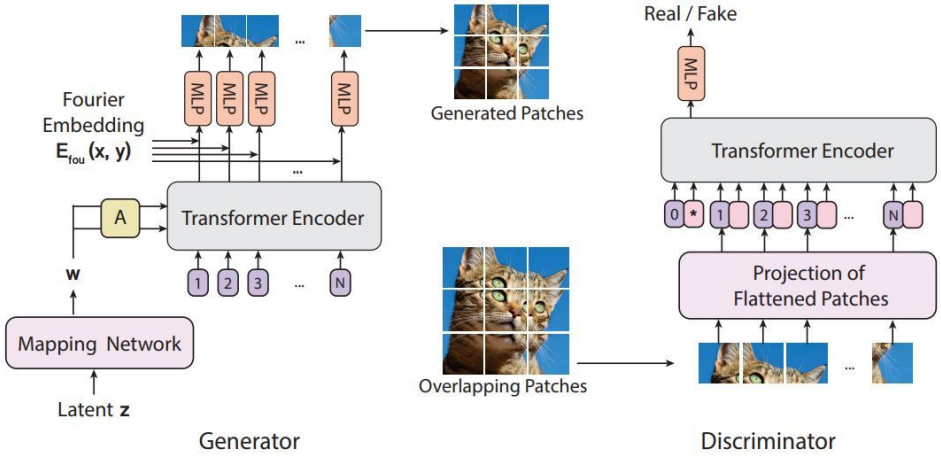


FIGURE 3.62 – Vue d'ensemble du ViTGAN (source [Lee et al., 2022c]).

Ces évolutions témoignent de la diversité et de l'intensité des recherches sur les GANs, avec des applications allant de la synthèse d'images réalistes à la traduction d'images (transfert de style) et à l'apprentissage de représentations interprétables. Les avancées récentes, notamment les combinaisons avec des Transformers, ouvrent de nouvelles perspectives pour des générations encore plus contrôlées et de haute qualité.

3.4.4 Approches à base de Diffusion

Les *Diffusion Probabilistic Models* (DPMs) [Ho et al., 2020] sont une classe de modèles génératifs qui apprennent un processus stochastique capable de transformer une distribution de bruit pur, typiquement une dis-

tribution gaussienne, en une distribution complexe qui se rapproche de celle des données d'entraînement. Le principe de base des DPMs repose sur un processus de diffusion (cf. Figure 3.63) dans lequel du bruit est progressivement ajouté à une image propre x_0 selon une séquence de paramètres de bruitage β_1, où T représente le nombre total d'étapes temporelles. Lorsque T est suffisamment grand, l'image bruitée x_T devient du bruit gaussien pur. Ce processus de bruitage est modélisé par une chaîne de Markov, où chaque étape $q(x_t|x_{t-1})$ ajoute un peu de bruit à partir de l'image de l'étape précédente. Le processus de diffusion complet est décrit par :

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta}x_{t-1}, \beta_t I) \tag{3.43}$$

$$q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1}) \tag{3.44}$$

Les DPMs sont entraînés pour inverser ce processus de diffusion, c'est-à-dire pour récupérer l'image d'origine x_0 à partir de bruit gaussien x_T, en apprenant progressivement la distribution inverse $q(x_{t-1}|x_t)$.

En termes pratiques, nous ne connaissons pas $q(x_{t-1}|x_t)$. Ce calcul est intractable, car des estimations statistiques de $q(x_{t-1}|x_t)$ nécessitent des calculs impliquant la distribution des données. Au lieu de cela, nous approchons $q(x_{t-1}|x_t)$ avec un modèle paramétré p_θ. Comme $q(x_{t-1}|x_t)$ sera également gaussienne pour un β_t suffisamment petit, nous pouvons choisir p_θ comme une gaussienne et simplement paramétrer la moyenne et la variance :

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_t heta(x_t, t)) \tag{3.45}$$

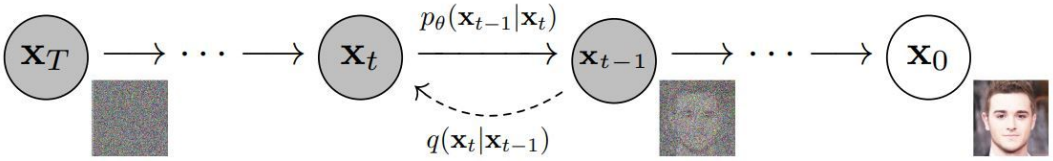
Si nous appliquons cette formule inversée à tous les pas de temps (c'est-à-dire $p_\theta(x_{0:T})$, également appelée trajectoire), nous pouvons remonter de x_T à la distribution des données :

$$p_\theta(x_{0:T}) = p_\theta(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t) \tag{3.46}$$

En conditionnant en plus le modèle sur le pas de temps t, le modèle apprendra à prédire les paramètres gaussiens (la moyenne $\mu_\theta(x_t, t)$ et la matrice de covariance $\Sigma_\theta(x_t, t)$) pour chaque pas de temps. La fonction de perte des DPMs, utilisée pour l'entraînement, est formulée comme suit :

$$L_{\text{DPM}} = \mathbb{E}_{t\sim[1,T]}\mathbb{E}_{x_0\sim p(x_0)}\mathbb{E}_{z_t\sim\mathcal{N}(0I)} \parallel z_t - z_\theta(x_t, t) \parallel^2 \tag{3.47}$$

où $z_\theta(x_t, t)$ est le réseau d'entraînement qui prédit le bruit dans x_t. Cette structure de perte relativement simple et la chaîne de Markov permettent aux DPMs de couvrir efficacement la distribution des données et de générer des échantillons de haute qualité, aux détails et textures nets. Les modèles de diffusion ont démontré des capacités impressionnantes pour générer des images réalistes, avec des applications allant de la création d'images complexes à la génération de scènes cohérentes, rendant les DPMs particulièrement prometteurs pour des applications de génération d'images où la qualité visuelle est essentielle.



(a) Processus de diffusion.

Algorithm 1 Training

1: **repeat**
2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$
3: $t \sim \text{Uniform}(\{1, \dots, T\})$
4: $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
5: Take gradient descent step on
$\nabla_\theta \left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|^2$
6: **until** converged

Algorithm 2 Sampling

1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
2: **for** $t = T, \dots, 1$ **do**
3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$
4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$
5: **end for**
6: **return** $\mathbf{x}_0$

(b) Pseudo algorithme pour l'apprentissage et l'échantillonnage.

FIGURE 3.63 – Processus probabiliste de diffusion par débruitage (source [Ho et al., 2020]).

Les *Denoising Diffusion Probabilistic Models* (DDPMs), en tant que première itération majeure, ont souffert d'une lenteur de convergence due à leur nature itérative. Pour résoudre ce problème, des variantes comme les DDIMs (*Denoising Diffusion Implicit Models*) [Song et al., 2021a] ont été introduites, remplaçant le processus markovien par un processus non-markovien. Cela permet un échantillonnage jusqu'à 50 fois plus rapide tout en maintenant une qualité d'échantillon élevée. La méthode Improved DDPMs [Nichol and Dhariwal, 2021] permet d'améliorer davantage la qualité en introduisant une variance apprenable dans le processus inverse et un bruit en cosinus dans le processus direct, réduisant considérablement le nombre de passes nécessaires pour produire des échantillons. Des efforts théoriques ont également été réalisés avec les *Variational Diffusion Models* (VDMs) [Kingma et al., 2021], qui ont simplifié les bornes variationnelles à l'aide du rapport signal-bruit, permettant une meilleure compréhension et une optimisation plus rapide grâce à des programmations invariantes par rapport au calendrier de bruit. L'application des modèles de diffusion ne se limite pas à la génération d'images ; ils sont également utilisés dans des tâches complexes de synthèse et d'édition. Par exemple, SDEdit [Meng et al., 2021a] applique des équations différentielles stochastiques (SDEs) pour guider le processus de débruitage et éditer des images de manière réaliste, tandis qu'ILVR [Choi et al., 2021] exploite des images de référence pour conditionner le processus de diffusion, garantissant des résultats avec une forte similarité sémantique. Les modèles textuels tels que DALL-E [Ramesh et al., 2021], GLIDE [Nichol et al., 2022] et Stable Diffusion [Rombach et al., 2022] (ou Latent Difusion Model, LDM) étendent ces concepts à la génération d'images guidée par du texte.

En particulier, Stable Diffusion (cf. Figure 3.64), publié par Stability AI [2], contrairement aux modèles de diffusion traditionnels qui opèrent directement dans l'espace des pixels, entraînant une grande complexité calculatoire et une lenteur de génération, les LDMs effectuent le processus de diffusion dans un espace latent via la compression des images à l'aide d'un autoencodeur, ce qui permet de conserver les caractéristiques essentielles tout en éliminant certaines informations de haute fréquence. L'autoencodeur utilisé est un VAE pré-entraîné, qui apprend un espace latent perceptuellement similaire à l'espace des pixels. Cette séparation entre l'entraînement du VAE et celui du modèle de diffusion rend la méthode adaptable à différents modèles de diffusion et permet son exécution sur des GPUs grand public. Le processus de diffusion latente est effectué dans l'espace latent, où le modèle de diffusion est entraîné à l'aide d'un réseau U-Net modifié. De cette manière, la complexité du calcul pendant la phase d'échantillonnage est fortement réduite. Pour permettre un contrôle conditionnel sur le contenu des images générées, les LDMs ajoutent un mécanisme de cross-attention au backbone U-Net, intégrant des informations de contrôle directement dans le processus de diffusion. Cette approche fait de Stable Diffusion un outil efficace, rapide et flexible pour la synthèse d'images haute résolution, adapté à des applications sur du matériel grand public.

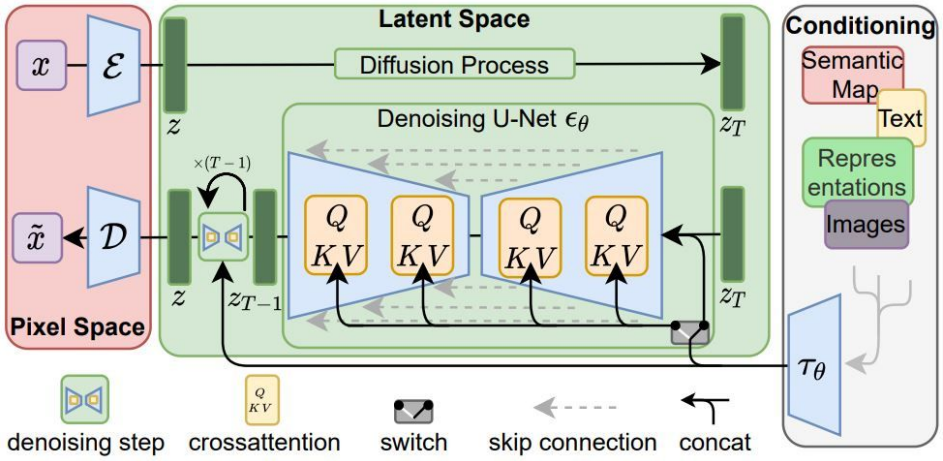


FIGURE 3.64 – Structure du LDM (source [Rombach et al., 2022]).

DALL-E2 [Ramesh et al., 2022] et Imagen [Saharia et al., 2022] vont plus loin en combinant de puissants encodeurs de texte avec des mécanismes de super-résolution, produisant des images d'une qualité et d'une fidélité textuelle remarquables. Dans un contexte d'apprentissage avec peu d'exemples, les D2C (*Diffusion-Decoding Models*) [Sinha et al., 2021] combinent les forces des VAEs et des modèles de diffusion. Ces modèles utilisent des représentations latentes apprises de manière contrastive pour générer des échantillons réalistes, même avec

2. https ://stability.ai/

peu de données annotées. La lenteur du processus d'échantillonnage a poussé à des innovations comme le DPM-Solver [Lu et al., 2022], qui accélère les modèles de diffusion probabiliste en utilisant des solveurs d'équations différentielles ordinaires (ODE), et ES-DDPM [Lyu et al., 2022], qui arrête le processus de diffusion à un stade précoce pour réduire le nombre d'étapes tout en améliorant la qualité des échantillons. Des approches telles que gDDIM [Zhang et al., 2023d] et Analytic-DPM [Bao et al., 2021] exploitent des approximations analytiques et des méthodes Monte Carlo pour accroître l'efficacité des modèles. Des méthodes comme DiffusionCLIP [Kim et al., 2022] et Blended Diffusion [Avrahami et al., 2022] exploitent la puissance des modèles de diffusion pour manipuler des attributs d'images à partir d'instructions textuelles, offrant des capacités d'édition robustes et sans nécessiter d'entraînement spécifique. DreamBooth [Ruiz et al., 2023] propose une personnalisation fine des modèles de diffusion textuel-image, permettant de contextualiser des sujets spécifiques dans divers scénarios avec un faible nombre d'images. Parti [Yu et al., 2022], quant à lui, adopte une approche différente en utilisant un modèle auto-régressif basé sur deux étapes : une tokenisation pour convertir les images en séries de tokens visuels discrets, suivie d'un entraînement séquentiel pour générer des tokens d'images à partir de tokens textuels (cf. Figure 3.65).

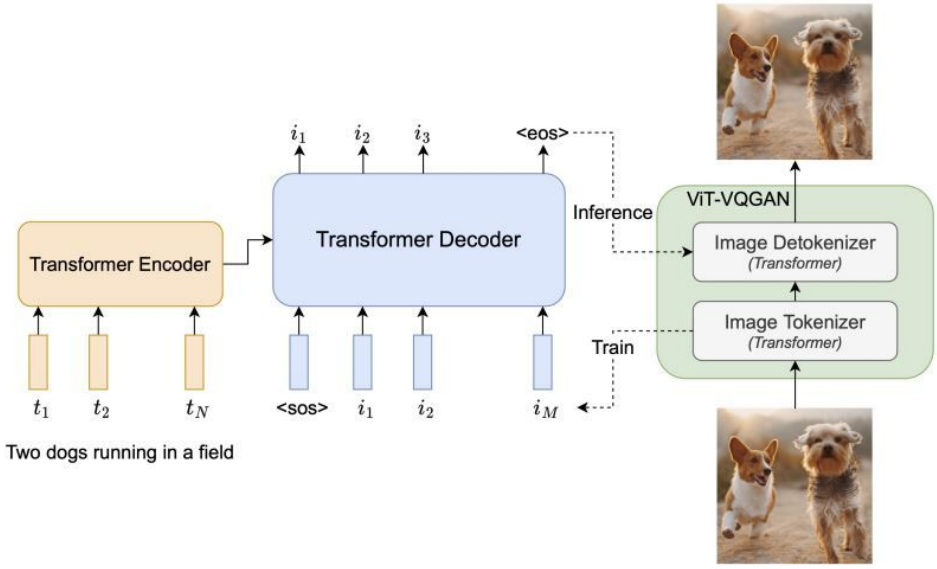


FIGURE 3.65 – Vue d'ensemble du modèle autoregressif séquence-à-séquence Parti (à gauche) pour la génération d'images à partir de texte, avec ViT-VQGAN utilisé comme tokenizer d'image (à droite) (source [Yu et al., 2022]).

La personnalisation des modèles a également été améliorée grâce à des techniques telles que *Textual Inversion* [Gal et al., 2023], où un mot pseudo-représenté est optimisé pour injecter des concepts spécifiques dans les modèles textuel-image à partir d'un petit ensemble d'images d'entraînement, tout en conservant les capacités générales du modèle. Les innovations dans les mécanismes de diffusion continuent de transformer le domaine. Cold Diffusion [Bansal et al., 2024] a démontré que les modèles de diffusion ne dépendent pas strictement de la distribution de bruit utilisée, remplaçant celle-ci par des transformations arbitraires pour diversifier les applications. ControlNet [Zhang et al., 2023c] introduit un contrôle spatial conditionnel en connectant les couches pré-entraînées d'un modèle textuel-image à un encodeur profond via des couches de convolution zéro, facilitant un réglage précis sans introduire de bruit inutile. Pour les tâches d'édition guidées, InstructPix2Pix [Brooks et al., 2023] utilise des instructions humaines pour modifier des images. En combinant GPT-3 et Stable Diffusion pour générer des ensembles d'exemples d'édition, le modèle applique rapidement des modifications en suivant les instructions sans nécessiter de fine-tuning pour chaque exemple. Cette approche est rapide et robuste pour des modifications en quelques secondes. En ce qui concerne les architectures, U-ViT [Bao et al., 2023] propose une architecture basée sur ViT qui traite toutes les entrées comme des tokens, avec des connexions longues entre couches pour la génération d'images, surpassant parfois les U-Nets traditionnels. Enfin, Stable Diffusion XL [Podell et al., 2024] et DiffFit [Xie et al., 2023] ont optimisé les schémas d'adaptation pour produire des images de grandes tailles ou pour adapter rapidement des modèles à de nouveaux domaines génératifs. Des frameworks comme DemoFusion [Du et al., 2024] et DragonDiffusion [Mou et al., 2024] introduisent des approches multi-échelles et exploitent les correspondances de caractéristiques pour maintenir la cohérence entre

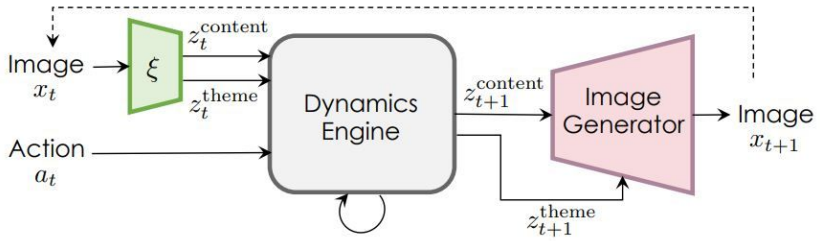
les images modifiées et originales. Ensemble, ces innovations renforcent la polyvalence des modèles de diffusion dans des applications de génération, d'édition et de personnalisation.

Les modèles de diffusion représentent une avancée majeure en génération d'images grâce à leur flexibilité et à leur capacité à produire des échantillons de haute qualité. Qu'il s'agisse de génération textuelle, d'édition ou d'accélération, ces modèles redéfinissent les standards et ouvrent la voie à des applications variées dans la vision par ordinateur, la création artistique et bien d'autres domaines. Les innovations continues dans ce domaine, telles que la personnalisation, les adaptations rapides et l'intégration de Transformers, renforcent leur position comme technologie clé pour la génération d'images par deep learning.

3.4.5 Génération dans le domaine automobile

La génération de données, notamment d'images et de vidéos, dans le domaine automobile est devenue un sujet de recherche central pour les entreprises automobile, avec des applications allant de la simulation réaliste à l'entraînement et au test de systèmes d'intelligence artificielle pour la conduite autonome. Cette approche vise à surmonter les limitations des données collectées dans le monde réel, en permettant la création de scénarios variés, contrôlables et réalistes qui peuvent être utilisés pour affiner les algorithmes et tester leur robustesse dans des conditions diverses.

Les modèles de génération pour le domaine automobile se distinguent par leur capacité à capturer et reproduire les dynamiques complexes des scènes de conduite. Par exemple, [Kim et al., 2021] propose un simulateur neuronal de haute qualité, DriveGAN, capable de simuler un environnement dynamique directement dans l'espace des pixels en apprenant à partir de séquences non annotées d'images et des actions associées (cf. Figure 3.66). Le modèle vise à atteindre un contrôle à deux niveaux : d'une part, permettre à un agent égocentrique de répondre à des actions données, et d'autre part, contrôler différents aspects de la scène, comme la météo, la couleur d'arrière-plan, ou la position des objets. DriveGAN utilise une architecture encodeur-décodeur pré-entraînée pour produire un espace latent des images, avec une architecture originale qui désentrelace les thèmes et le contenu pour améliorer la qualité de génération, en s'appuyant sur un VAE. Le moteur dynamique (*Dynamics Engine*) apprend les dynamiques de l'espace latent pour simuler les changements au fil du temps. En tant que simulateur entièrement différentiable, DriveGAN permet de rejouer des séquences vidéo enregistrées tout en modifiant les actions, offrant une grande flexibilité pour l'exploration des scènes. Entraîné sur plusieurs ensembles de données, incluant 160 heures de données de conduite réelle, DriveGAN offre un contrôle de haute qualité pour des environnements simulés, rendant possible des ajustements détaillés, tels que la modification des objets ou des conditions de la scène, sans supervision explicite. Bien que nommé GAN, le modèle repose principalement sur des principes de VAE pour sa conception.



(a) Fonctionnement de DriveGAN.



(b) DriveGAN produit une simulation de haute qualité, cohérente dans le temps, qui respecte la séquence d'actions.

FIGURE 3.66 – Fonctionnement du modèle DriveGAN et résultats (source [Kim et al., 2021]).

Pour la génération de vidéos, des approches comme VideoLDM [Blattmann et al., 2023] exploitent des

LDMs en les étendant à la synthèse vidéo. Ces modèles insèrent des couches temporelles pour aligner les images de manière cohérente, produisant ainsi des vidéos de haute résolution et consistantes sur plusieurs minutes.

De manière similaire, [Wang et al., 2024c] présente DriveDreamer, un modèle pionnier conçu pour représenter des scénarios de conduite réels de manière fidèle et détaillée. Face à la complexité des scènes de conduite, où l'espace de recherche est considérable, DriveDreamer exploite la puissance des modèles de diffusion pour construire une représentation complète et précise de l'environnement (cf. Figure 3.67). Le modèle suit un pipeline d'entraînement en deux étapes : dans la première phase, il acquiert une compréhension approfondie des contraintes structurées du trafic, tandis que la seconde phase le dote de la capacité à anticiper les états futurs. Grâce à cette approche, DriveDreamer permet la génération de vidéos précises et contrôlables, reflétant fidèlement les contraintes structurelles des scénarios de trafic réels. Une extension, DriveDreamer-2 [Zhao et al., 2024a], enrichit ce cadre en intégrant un modèle de langage de grande taille (*Large Language Model*, LLM) pour générer des vidéos de conduite personnalisées selon les requêtes des utilisateurs. Le LLM convertit les requêtes en trajectoires d'agents, puis génère une carte HD respectant les réglementations routières sur la base de ces trajectoires. Enfin, DriveDreamer-2 introduit un modèle unifié multi-vue pour améliorer la cohérence temporelle et spatiale des vidéos générées, renforçant ainsi leur réalisme et leur applicabilité.

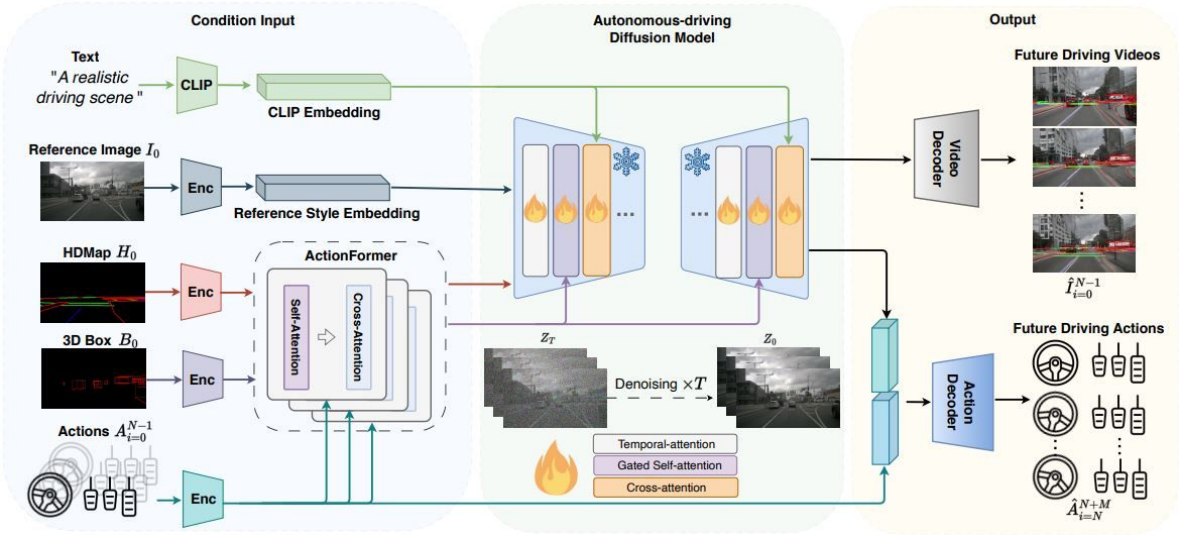


FIGURE 3.67 – Vue d'ensemble de DriveDreamer. Le cadre commence avec une image de référence I_0 et des informations structurelles de la route (c'est-à-dire la carte HD H_0 et la boîte 3D B_0). DriveDreamer utilise ActionFormer pour prédire les futures caractéristiques structurelles de la route, qui servent de conditions fournies à Auto-DM, permettant de générer des vidéos de conduite futures $\hat{I}_i^{N-1}$ pour $i = 0$. De plus, des invites textuelles permettent d'ajuster dynamiquement le style des scénarios. Le modèle intègre les actions de conduite passées et des caractéristiques multi-échelles provenant d'Auto-DM pour générer des actions de conduite futures plausibles $\hat{A}_i^{N+M}$ pour $i \geq N$ (source [Wang et al., 2024c]).

D'autres travaux, comme GAIA-1 [Hu et al., 2023], développé par l'entreprise Wayne [3], traitent la modélisation du monde comme un problème de modélisation séquentielle non supervisée, combinant vidéos, texte et actions pour générer des scénarios réalistes et dynamiques avec un contrôle granulaire sur les comportements du véhicule et les caractéristiques de la scène (cf. Figure 3.68). Parmi les propriétés émergentes de GAIA-1 figurent sa capacité à apprendre les structures de haut niveau et la dynamique des scènes, sa conscience contextuelle, sa capacité de généralisation et sa compréhension de la géométrie. Le modèle est capable de capturer des représentations qui anticipent les événements futurs et de générer des échantillons réalistes, ouvrant la voie à des simulations avancées pour la conduite autonome.

3. https://wayve.ai/thinking/introducing-gaia1/

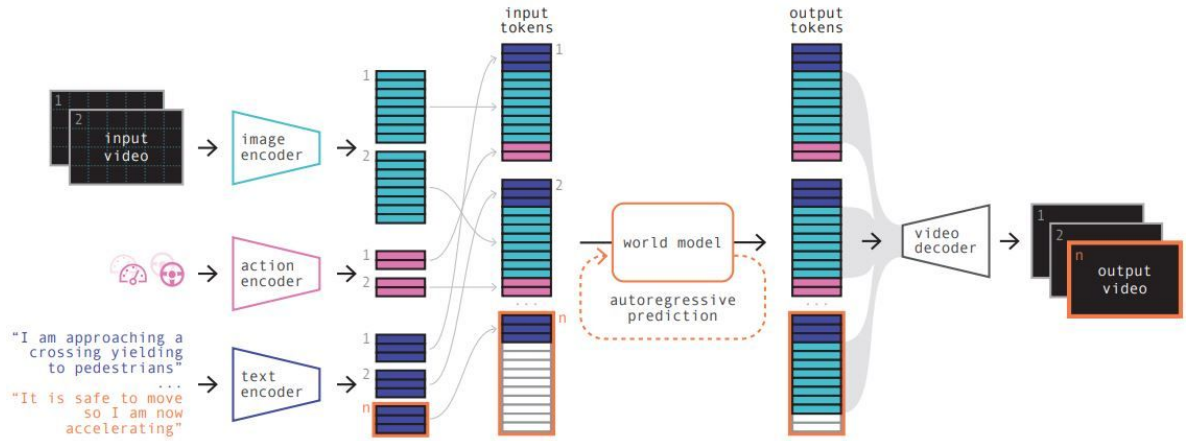


FIGURE 3.68 – Architecture de GAIA-1 (source [Hu et al., 2023]).

La génération contrôlable est également abordée par MagicDrive [Gao et al., 2024], qui introduit des contrôles géométriques 3D diversifiés. Ce modèle permet de manipuler les poses de caméra, les cartes routières, les boîtes englobantes 3D, ainsi que des descriptions textuelles, grâce à des stratégies d'encodage sur mesure. Une attention particulière est portée à la cohérence entre les vues multiples des caméras grâce à un module d'attention inter-vues (*cross-view attention module*), garantissant une représentation réaliste et uniforme des scènes générées (cf. Figure 3.69).

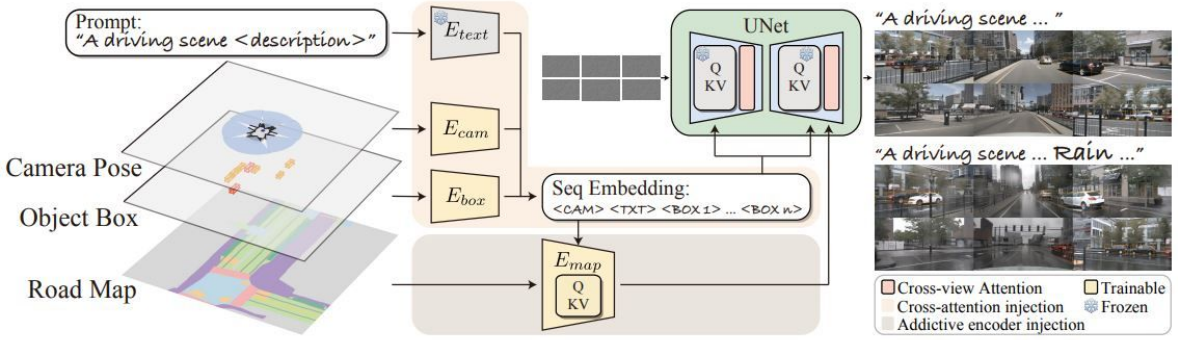


FIGURE 3.69 – Vue d'ensemble de MagicDrive (source [Gao et al., 2024]).

De même, [Su et al., 2024] propose un cadre innovant de génération contrôlable de vues de rues à partir de descriptions textuelles, nommé Text2Street. Ce modèle introduit plusieurs modules clés : un générateur de topologie routière sensible aux voies (lane-aware road topology generator), qui permet de convertir du texte en une carte précise avec des structures routières et des lignes de voie grâce à un adaptateur de comptage ; un générateur de disposition d'objets basé sur la position (position-based object layout generator), qui utilise une stratégie de diffusion à l'échelle des boîtes englobantes pour produire une disposition contrôlable des objets de trafic à partir du texte ; et un générateur d'images multi-contrôle (multiple control image generator), qui intègre la topologie routière, la disposition des objets et les descriptions météorologiques pour générer des vues de rues entièrement contrôlables.

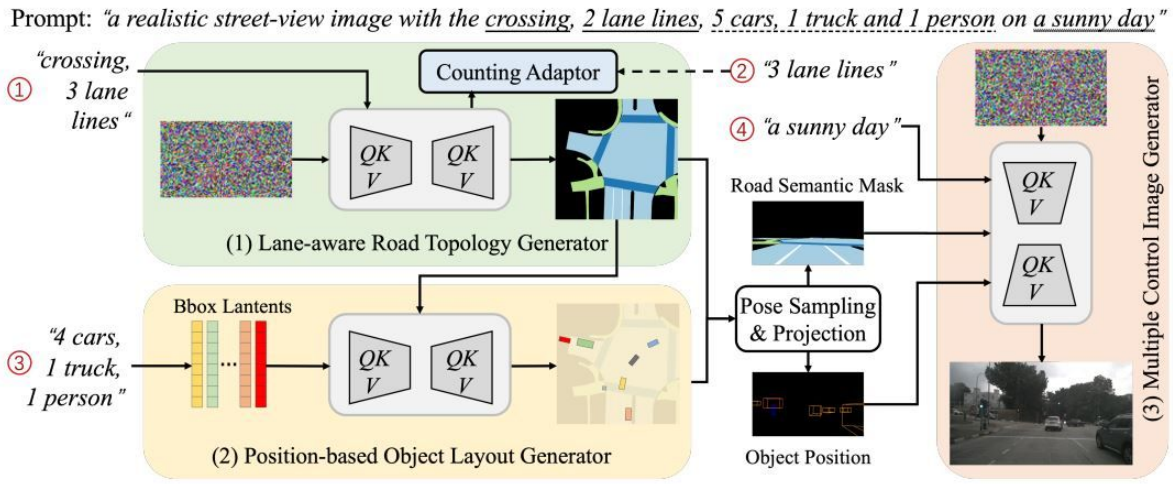


FIGURE 3.70 – Vue d'ensemble de Text2Street (source [Su et al., 2024]).

Des méthodes basées sur les modèles de diffusion, comme DrivingDiffusion [Li et al., 2024b], proposent un cadre de diffusion spatio-temporel cohérent pour générer des vidéos réalistes de scénarios de conduite multi-vues, guidées par une disposition 3D. Le processus repose sur trois étapes en cascade : la génération d'images monocadres pour plusieurs vues, la génération de vidéos monocaméra partagées par toutes les caméras, et un post-traitement pour gérer les vidéos longues. La cohérence entre les vues multiples est assurée par des échanges d'informations entre caméras adjacentes, tandis que pour le modèle temporel, les informations nécessaires à la génération des cadres suivants sont extraites des images multi-vues du premier cadre. Le modèle introduit également des local prompts pour améliorer la qualité des instances générées. En post-traitement, un algorithme de fenêtre glissante temporelle est utilisé pour renforcer la cohérence inter-vues des cadres suivants et allonger la durée des vidéos. Sans coût supplémentaire, DrivingDiffusion peut générer des vidéos réalistes et de grande échelle dans des environnements urbains complexes, soutenant ainsi diverses tâches liées à la conduite autonome.

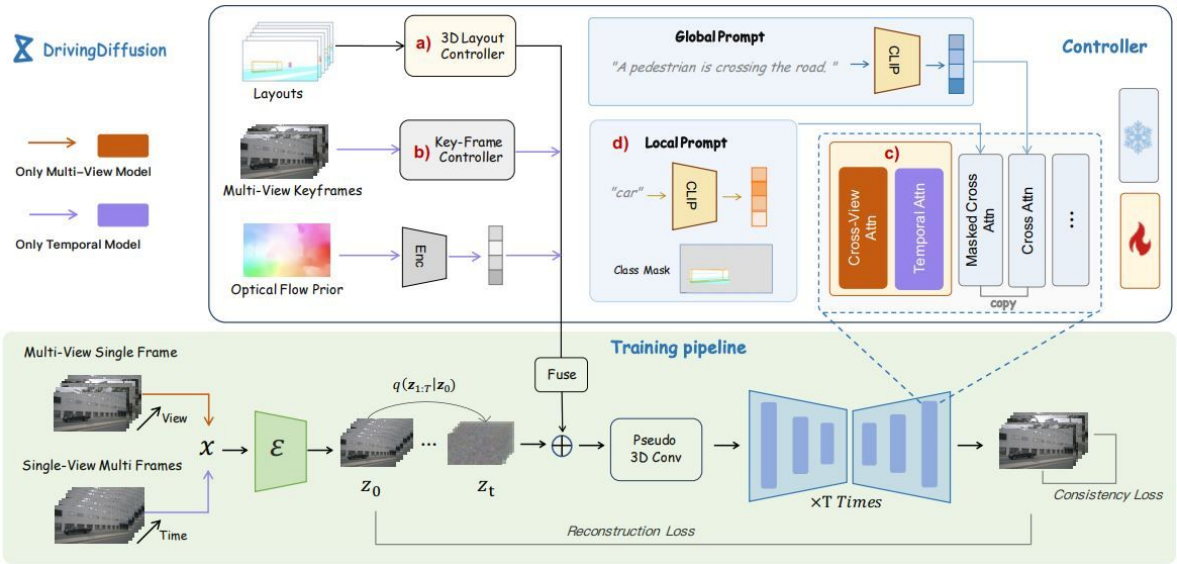


FIGURE 3.71 – Architecture de DrivingDiffusion (source [Li et al., 2024b]).

Enfin, DreamForge [Mei et al., 2024] pousse encore plus loin les capacités de génération vidéo avec des mécanismes de contrôle sophistiqués et une cohérence inter-vues améliorée. Ce dernier propose un modèle avancé de génération vidéo basé sur la diffusion et conçu pour produire des vidéos à long terme avec contrôle 3D et extensibilité (cf. Figure 3.72). DreamForge prend en charge des conditions flexibles, telles que des descriptions textuelles, des poses de caméra, des boîtes englobantes 3D et des agencements routiers, tout en intégrant des guides perspectifs pour garantir des scènes de conduite géométriquement et contextuellement précises. Le

modèle assure la cohérence entre les vues grâce à un module d'attention inter-vues (*cross-view attention*) et maintient la cohérence temporelle via une architecture autorégressive enrichie par des indices de mouvement.

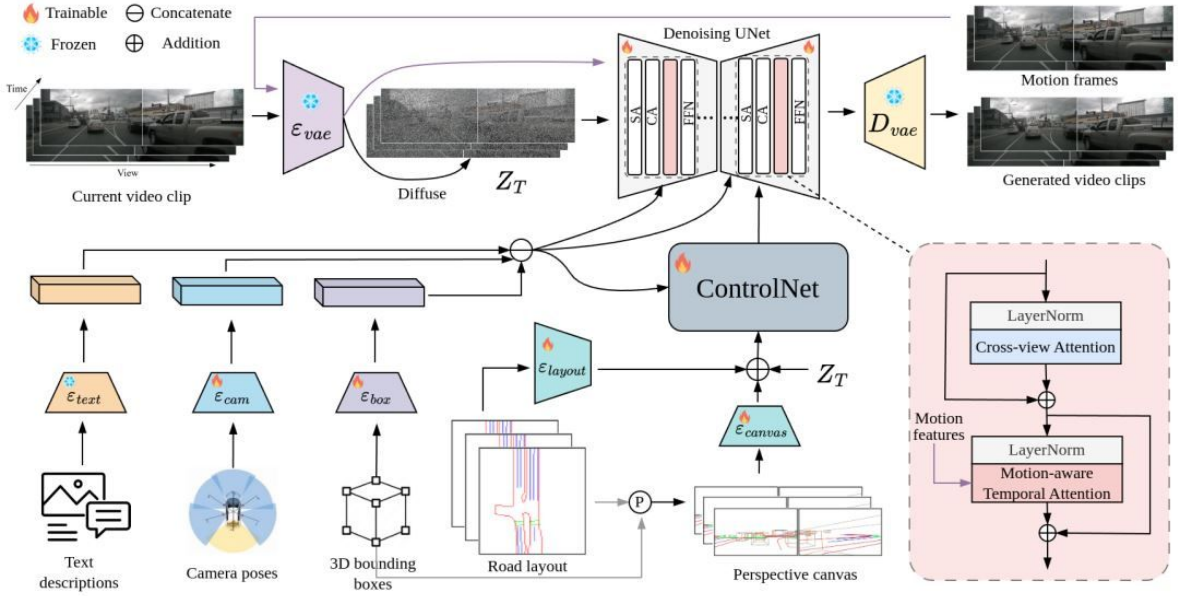


FIGURE 3.72 – Architecture de DreamForge (source [Mei et al., 2024]).

De même, DiVE [Jiang et al., 2024a] propose le premier cadre basé sur DiT conçu pour générer des vidéos cohérentes à la fois temporellement et entre différentes vues, tout en respectant précisément les agencements en vue de dessus spécifiés (cf. Figure 3.73). Le modèle utilise un mécanisme d'attention spatiale étendue sans paramètres pour garantir la cohérence inter-vues. De plus, des modules d'attention croisée et un ControlNet-Transformer sont intégrés pour améliorer davantage la précision du contrôle, permettant une génération de vidéos alignée avec des contraintes spatiales complexes.

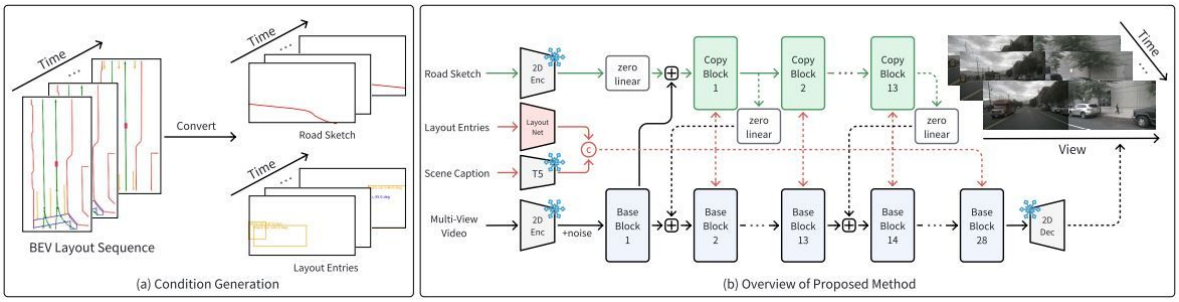


FIGURE 3.73 – Vue d'ensemble de DiVE (source [Jiang et al., 2024a]).

Ces avancées démontrent le rôle crucial des modèles génératifs dans le développement de systèmes de conduite autonome, non seulement pour améliorer la simulation et l'entraînement, mais aussi pour explorer des scénarios extrêmes ou rares, contribuant ainsi à rendre ces systèmes plus robustes et sûrs dans des environnements variés. Cependant, ces modèles présentent également des inconvénients notables : leur taille et leur complexité sont souvent très élevées, ce qui exige des ressources matérielles conséquentes pour leur entraînement et leur inférence. De plus, leur performance repose généralement sur de grandes quantités de données bien annotées, ce qui peut être difficile et coûteux à obtenir, en particulier dans le contexte de scènes de conduite variées et réalistes. Ces limitations doivent être prises en compte pour une adoption pratique et à grande échelle dans l'industrie automobile.

3.4.6 Génération d'images critiques

Dans ce chapitre, nous abordons une problématique clé liée à l'évaluation de systèmes complexes. L'objectif est de tester de manière exhaustive un tel système afin d'estimer, de façon statistique, les probabilités de bon fonctionnement et d'erreur. Cependant, cette tâche est limitée par la disponibilité restreinte des données d'images, qui ne permet pas de couvrir l'ensemble des cas potentiels. Parmi les images disponibles, certaines ne génèrent aucune erreur dans le système, tandis que d'autres provoquent des erreurs.

Dans les sections précédentes, nous avons déjà exploré comment les modèles génératifs pouvaient être utilisés pour enrichir le jeu de données en générant de nouvelles images similaires à celles issues de notre distribution initiale. Une question plus ambitieuse se pose cependant : est-il possible d'orienter un générateur d'images pour produire spécifiquement des images susceptibles de mettre en défaut notre système ? Autrement dit, peut-on cibler la génération d'images qui accentuent les faiblesses du système testé ?

Ces images, souvent appelées exemples adversariaux non-restreints (*Unrestricted Adversarial Examples*, UAE) ou exemples adversariaux naturels (*Natural Adversarial Examples*, NAE), se distinguent des exemples adversariaux traditionnels. Contrairement aux approches classiques qui ajoutent des perturbations imperceptibles à des données existantes pour induire une erreur de classification, les UAE et les NAE sont directement générés par des modèles génératifs et respectent la distribution des données d'origine. Plus précisément :

- Les UAE sont des images qui suivent la distribution des données réelles, que les humains peuvent catégoriser correctement, mais que le classificateur traite de manière erronée.
- Les NAE proviennent naturellement de l'environnement ou sont conçus pour ressembler à des données réelles, tout en étant capables de tromper les classificateurs (Figure 3.74).

Ces exemples offrent une opportunité précieuse pour explorer les limites des systèmes d'apprentissage automatique dans des scénarios réalistes. Ce chapitre examinera les concepts, méthodes et applications de la génération d'exemples adversariaux naturels pour orienter les tests vers les limites de fonctionnement du système.



FIGURE 3.74 – Exemples adversariaux naturels. Dans chaque paire, l'image d'initialisation à gauche est correctement classifiée, tandis que celle de droite (adversariale) est mal classifiée avec la prédiction incorrecte indiquée en rouge (source [Lin et al., 2024]).

Par exemple, Diff-PGD [Xue et al., 2023b] propose une approche novatrice et un cadre nommé Diffusion-Based Projected Gradient Descent pour la génération d'exemples adversariaux réalistes en exploitant des modèles de diffusion (cf. Figure 3.75). Étant donné une image propre x avec un label y et un classificateur cible f_θ paramétré par θ à attaquer, l'objectif est de générer un exemple adversarial x_{adv}, qui trompe le modèle. Les méthodes traditionnelles basées sur le gradient utilisent le gradient de la perte adversariale $g = \nabla_x l\left(f_t heta(x), y\right)$, où $f_t heta(x)$ renvoie les logits. On note également cette perte par l_{adv} pour simplifier l'optimisation de x par itérations. La mise à jour à l'étape t dans PGD (*Projected Gradient Descent* [Madry et al., 2018]) avec un pas η et n itérations est donnée par :

$$x_{t+1} = P_{B^\infty(x,\epsilon)}\left(x_t + \eta.\text{sign}\left(\nabla_{x_t} l\left(f_\theta(x_t), y\right)\right)\right) \tag{3.48}$$

où $P_{B^\infty(x,\epsilon)}(\cdot)$ est l'opérateur de projection sur la boule l^∞ de rayon ϵ. Intuitivement, le gradient "guide" l'échantillon x_t loin de la frontière de décision et le fait dévier de la distribution réelle $p(x_0)$. En guidant les gradients par un modèle de diffusion, Diff-PGD garantit que les perturbations générées restent proches de la distribution des données originales tout en maintenant leur efficacité adversariale. Cette méthode se distingue par sa capacité à séparer l'optimisation de la perte adversariale des pertes auxiliaires, telles que celles liées au contenu, au style ou au lissage, ce qui renforce la stabilité et la contrôlabilité du processus. De plus, Diff-PGD est facilement personnalisable pour des tâches spécifiques, telles que les attaques numériques, physiques ou basées sur le style, offrant ainsi une flexibilité unique par rapport aux méthodes existantes.

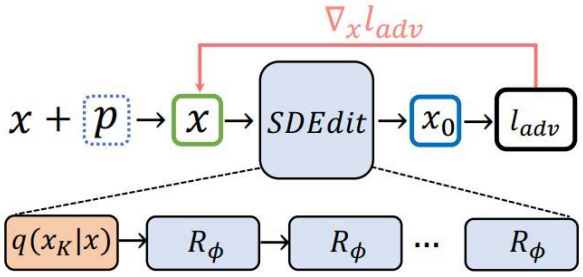


FIGURE 3.75 – Le cadre Diff-PGD : p est le prompt, q représente la diffusion directe, R_ϕ correspond au débruitage inverse, et x_0 est l'échantillon débruité (source [Xue et al., 2023b]).

L'article [Chen et al., 2023b] propose AdvDiffuser, une méthode novatrice utilisant des modèles de diffusion pour générer des exemples adversariaux naturels et non contraints (UAEs). Ces UAEs peuvent être produits soit à partir de zéro, soit conditionnellement à partir d'images de référence. Pour générer ces exemples, AdvDiffuser perturbe les images prédites en orientant leur code latent vers l'espace adversarial d'un classificateur spécifique. Une innovation clé de cette méthode est l'*adversarial inpainting*, qui perturbe sélectivement les régions non saillantes des images en se basant sur des masques dérivés de la cartographie d'activation par gradient (Grad-CAM). Cette approche ajuste la force de débruitage pixel par pixel en fonction de la saillance des objets : les régions contenant des objets importants subissent des modifications minimales, tandis que les zones moins pertinentes sont davantage altérées. Par rapport aux méthodes basées sur les GANs ou les gradients, AdvDiffuser produit des perturbations perceptuelles beaucoup plus subtiles, rendant les UAEs générés significativement plus naturels et imperceptibles. En outre, cette méthode offre la possibilité de synthétiser un nombre infini d'exemples adversariaux photoréalistes, dépassant ainsi les limitations des autres approches adversariales non contraintes tout en maintenant une grande flexibilité et efficacité. De manière similaire, AdvDiff [Dai et al., 2024b] est une méthode novatrice basée sur des modèles de diffusion conditionnels pour générer des exemples adversariaux illimités, réalistes et interprétables. Cette approche introduit deux techniques d'orientation adversariale dans le processus de génération inversé des modèles de diffusion, garantissant une grande stabilité et une haute qualité des échantillons produits (cf. Figure 3.76). La première technique intègre des connaissances adversariales a priori dans le bruit initial grâce à une guidance par échantillonnage (*adversarial guidance*). La seconde consiste à augmenter progressivement la probabilité de la classe cible pendant la génération inversée pour guider efficacement le processus (*noise sampling guidance*). En utilisant un modèle de diffusion conditionnel pré-entraîné, AdvDiff exploite également les gradients du classificateur cible pour affiner la génération, offrant ainsi une méthode stable et efficace pour produire des exemples adversariaux non contraints et interprétables.

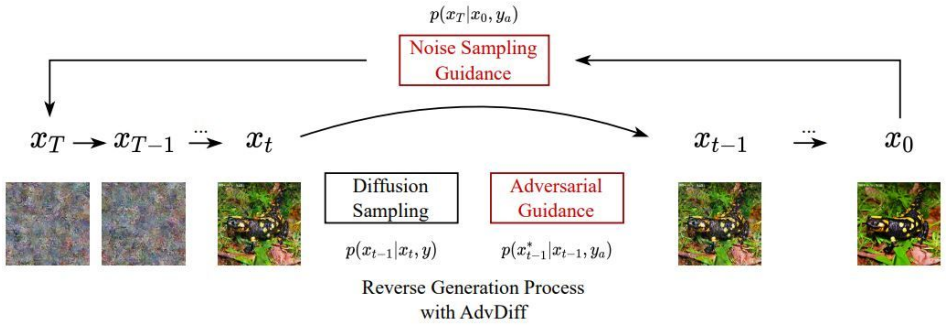


FIGURE 3.76 – Les deux techniques de guidage dans AdvDiff pour générer des exemples adversariaux non restreints. Pendant le processus de génération inverse, le guidage adversarial (*adversarial guidance*) est ajouté à l'étape x_t, ce qui injecte l'objectif adversarial y_a dans le processus de diffusion. Le guidage de l'échantillonnage du bruit (*noise sampling guidance*) modifie le bruit initial en augmentant la probabilité conditionnelle de y_a (source [Dai et al., 2024b]).

SD-NAE [Lin et al., 2024] introduit une méthode novatrice pour générer des exemples adversariaux naturels (Natural Adversarial Examples, NAEs) en exploitant le modèle Stable Diffusion. SD-NAE synthétise activement ces exemples via une optimisation contrôlée dans l'espace latent des embeddings conditionnels (cf. Figure 3.77). En manipulant les tokens de classe guidés par les gradients de la perte de classification d'un classifieur cible, la

méthode génère des exemples visuellement réalistes qui induisent des erreurs de classification. Une extension notable, SD-MIAE [Haque et al., 2024], améliore la génération de NAEs en termes de fidélité visuelle et de taux de mauvaise classification. SD-MIAE utilise un cadre en deux phases : une phase initiale d'optimisation des embeddings pour produire des images mal classifiées mais réalistes, et une phase basée sur un mécanisme de momentum pour stabiliser les perturbations adversariales au fil des itérations. Cette stabilisation renforce à la fois la qualité visuelle et l'efficacité des exemples adversariaux générés, ouvrant ainsi la voie à des améliorations dans l'utilisation des capacités texte-vers-image de Stable Diffusion.

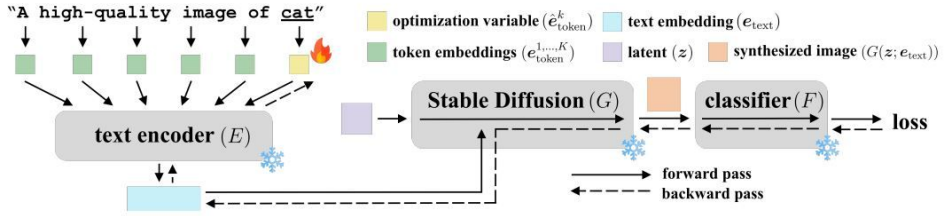


FIGURE 3.77 – Guidé par le gradient de la perte rétropropagé depuis le classificateur, SD-NAE génère des NAEs en optimisant uniquement l'embedding des tokens liés à la classe, tout en gardant tous les modèles figés (source [Lin et al., 2024]).

Enfin, EvoSeed [Kotyan et al., 2025] est un cadre algorithmique novateur utilisant une stratégie évolutive pour générer des échantillons adversariaux photoréalistes en environnement black-box. En combinant des modèles de diffusion conditionnels et des classifieurs auxiliaires, EvoSeed optimise un vecteur initial à l'aide de l'algorithme CMA-ES (*Covariance Matrix Adaptation Evolution Strategy*). Ce vecteur, une fois traité par le modèle de diffusion conditionnel, produit un échantillon adversarial naturel qui est mal classifié par le modèle cible, et ce, même en l'absence d'accès direct à ce dernier (cf. Figure 3.78).

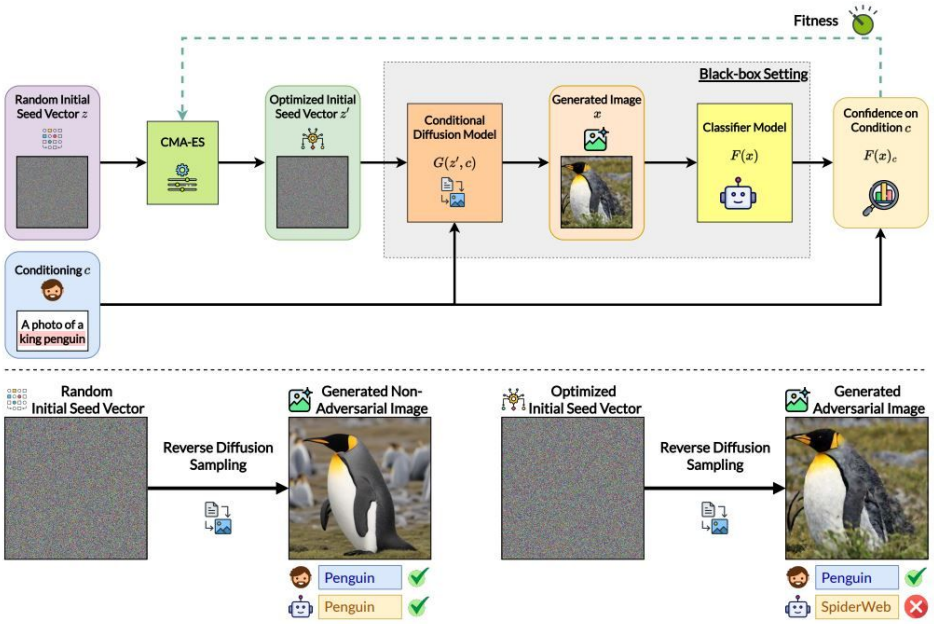


FIGURE 3.78 – Illustration du cadre EvoSeed pour optimiser le vecteur de graine initial z afin de générer un exemple adversarial naturel. La stratégie d'évolution par adaptation de la matrice de covariance (CMA-ES) affine itérativement le vecteur de graine initial z et trouve un vecteur de graine adversarial z'. Ce vecteur de graine adversarial z' peut ensuite être utilisé par le modèle de diffusion conditionnel G pour générer un exemple adversarial naturel x, capable de tromper le modèle de classification F (source [Kotyan et al., 2025]).

Ces méthodes présentent un potentiel remarquable pour améliorer l'évaluation de la robustesse des modèles en générant des perturbations réalistes et imperceptibles. Les modèles basés sur la diffusion, comme Diff-PGD et AdvDiffuser, offrent une meilleure stabilité et un contrôle accru par rapport aux approches classiques basées sur

les gradients ou les GANs. Cependant, leur complexité calculatoire et leur dépendance à des ressources importantes comme de grandes bases de données annotées, restent des défis pour leur déploiement à grande échelle. En conclusion, les méthodes de génération d'exemples adversariaux naturels enrichissent considérablement les outils d'évaluation de la robustesse des modèles d'apprentissage automatique, tout en ouvrant la voie à des recherches futures sur l'amélioration de leur efficacité et de leur accessibilité pour des applications critiques. En effet, dans le cadre de la diffusion, calculer le gradient de la sortie par rapport à l'entrée, quand le processus intermédiaire nécessite plusieurs *pass-forward* du modèle de débruitage, utilise énormément de mémoire.

3.4.7 Conclusion

Les modèles de diffusion représentent une avancée majeure en génération d'images grâce à leur flexibilité et à leur capacité à produire des échantillons de haute qualité. Ils sont souvent jugés supérieurs aux modèles VAE et GAN en raison de leur capacité à capturer de manière plus précise la distribution des données et à générer des échantillons avec moins d'artefacts visuels ou de problèmes de mode collapse (quand un générateur se "fixe" sur un ou quelques modes spécifiques de la distribution d'entraînement). Les innovations continues dans ce domaine, telles que la personnalisation, les adaptations rapides et l'intégration de Transformers, renforcent leur position comme technologie clé pour la génération d'images par deep learning.

Cependant, les modèles de diffusion présentent également des inconvénients. Leur coût calculatoire élevé, en particulier pendant la phase d'échantillonnage, peut limiter leur utilisation dans des environnements nécessitant une faible latence. Bien que des avancées comme la diffusion latente (LDMs) aient réduit cette charge, les modèles de diffusion restent globalement plus lents à générer des échantillons que les GAN. De plus, leur lourdeur et leur besoin en mémoire en font des modèles exigeants, nécessitant souvent des GPU de grande capacité pour l'entraînement et l'inférence, ce qui peut compliquer leur déploiement dans des environnements à ressources limitées. Leur convergence peut également être longue, nécessitant des ressources importantes en termes de puissance de calcul et de données d'entraînement. Enfin, leur capacité à modéliser des détails très fins dépend fortement de la qualité des données d'entraînement et de l'optimisation des hyperparamètres, ce qui peut constituer un défi dans des scénarios où les données sont limitées ou imparfaites.

3.5 Conclusion

Les méthodes de compression d'images, de détection d'objets et de génération d'images sont essentielles dans les systèmes de vision par ordinateur, notamment pour des applications critiques comme la conduite autonome. Bien qu'elles aient connu des avancées significatives, chaque catégorie de modèles présente des avantages et des limitations influençant leur adoption.

Les modèles de compression d'images se déclinent en plusieurs approches. Les méthodes à taux fixe offrent une implémentation simple, mais nécessitent un réentraînement pour modifier le taux de compression, ce qui limite leur praticité. En revanche, les méthodes à taux variable permettent un ajustement dynamique du débit, répondant ainsi aux besoins fluctuants en termes de qualité visuelle ou de contraintes de stockage. Cependant, elles sont généralement moins performantes en termes de compromis taux-distorsion. Par ailleurs, les compresseurs orientés tâches, conçus pour prioriser des régions ou des caractéristiques spécifiques (comme les objets d'intérêt pour la détection), maximisent la précision des tâches en aval tout en réduisant les besoins en transmission. Toutefois, leur complexité accrue augmente la consommation en calcul et en mémoire, rendant leur intégration plus exigeante.

Pour la détection d'objets, les modèles two-stage (comme Faster R-CNN) offrent une haute précision mais souffrent d'une latence élevée, tandis que les modèles single-stage (comme YOLO) privilégient la vitesse au détriment d'une légère baisse de précision. Les modèles basés sur les Transformers (comme DETR) introduisent une approche prometteuse en capturant des relations globales, mais leur coût calculatoire reste élevé.

Enfin, les modèles génératifs, tels que les GANs et les modèles de diffusion, ont transformé la génération de données synthétiques en permettant la simulation d'environnements variés et contrôlés. Les modèles de diffusion se distinguent par leur flexibilité et leur capacité à produire des images de haute qualité, bien qu'ils soient complexes à entraîner et gourmands en ressources. Leur performance dépend fortement de la diversité des données d'entraînement.

À l'issue de cet état de l'art exhaustif, nous avons sélectionné plusieurs modèles adaptés aux besoins spécifiques de notre étude, en tenant compte des contraintes de performances et des exigences de notre application.

Cette sélection s'est effectuée avec l'objectif de garantir que nos analyses et les garanties de performances établies ne soient pas biaisées par le choix d'un modèle particulier, en considérant plusieurs alternatives pour chaque composant de la chaîne de traitement. Pour la compression d'images, le compromis entre débit, distorsion et vitesse d'inférence a été au cœur de notre choix. Les modèles STF (basé sur les Transformers) et QRes-VAE (cf. Figures 3.16 et 3.13) se sont distingués par leurs performances équilibrées et leur rapidité. En effet, sur un GPU NVIDIA RTX A5500, ces modèles permettent un encodage et un décodage en moins d'une seconde pour des images de dimensions $3 \times 640 \times 1280$, répondant ainsi aux contraintes quasi temps réel de notre application. En ce qui concerne la détection d'objets, nous avons retenu trois modèles largement adoptés dans l'industrie automobile : YOLOv7, RetinaNet (cf. Section 3.3.4), et Faster R-CNN (cf. Figure 3.42). Ces détecteurs présentent des architectures et des priorités différentes : YOLOv7 privilégie la vitesse d'inférence, essentielle pour des systèmes en temps réel, tandis que RetinaNet et Faster R-CNN offrent une précision accrue au prix d'une latence plus élevée. Cette diversité dans les approches garantit une analyse robuste et sans biais de l'impact des distorsions introduites par la compression. Enfin, pour la génération d'images, nous avons priorisé les modèles capables de produire des images de haute qualité et résolution. Les modèles de diffusion, tels que Stable Diffusion (cf. Figure 3.64), ont montré des performances remarquables en termes de réalisme et de simplicité d'utilisation, surpassant les GANs et les VAEs dans notre contexte d'application. Les détails spécifiques de l'entraînement, des configurations et des performances obtenues avec chacun de ces modèles seront présentés dans les chapitres 6.2.1, 6.2.2 et 6.2.3.

3.6 Récapitulatif de l'état de l'art pour l'application industrielle

TABLE 3.1 – Tableau récapitulatif de l'état de l'art des méthodes de Compression d'images apprise.

Méthodes	**Caractéristiques**	**Articles**
Auto-encodeurs	Produit représentations compressées plus compactes et meilleure qualité visuelle que JPEG2000	[Ballé et al., 2017, Cheng et al., 2018, Zhou et al., 2018, Ballé et al., 2018, Fu et al., 2023, Brummer and De Vleeschouwer, 2021]
Modèles de contexte	amelioration du compromis compression et distortion, augmentation temps de calcul (non temps réel)	[Lee et al., 2018, Minnen et al., 2018b, Minnen and Singh, 2020, Ma et al., 2021, He et al., 2021, Guo et al., 2022, Ma et al., 2022a, Shizhan, 2023, Wang et al., 2023b, Ruan et al., 2023b, Jeon et al., 2023, Sui et al., 2024]
Multi-échelles	Exploitation des détails à différentes échelles pour une reconstruction fidèle, grande flexibilité, plus rapide, robuste à la perte de données, meilleur ratio taux-distortion	[Rippel and Bourdev, 2017, Hu et al., 2020, Ma et al., 2020, Ma et al., 2022b, Xue et al., 2023a, Gao et al., 2021, Liu et al., 2022b, Duan et al., 2023]
Attention et contenu	Allocation adaptative des bits, précision visuelle améliorée, s'adapte dynamiquement au contenu de l'image, coût computationnel élevé, latence accrue ($\sim 1-2$ s par image à haute résolution), complexité architecturale	[Mentzer et al., 2018, Li et al., 2018, Cheng et al., 2020b, Chen et al., 2021, Wang et al., 2022b, Wang and Wang, 2022, Tang et al., 2023]
Transformers	capture des dépendances globales, précision contextuelle accrue, scalabilité à haute résolution, performant pour faibles taux de compression, complexité computationnelle élevée, besoins massifs en données d'entraînement, latence accrue	[Zhu et al., 2022, Zou et al., 2022, Liu et al., 2023a, Kao et al., 2023, Shen and Liu, 2023, Ruan et al., 2023a, Wang et al., 2024d]
Modèles génératifs	reconstruction de haute qualité, capacité d'apprentissage des structures globales, utilisation efficace de la prior de données, complexité calculatoire importante, temps d'entraînement long, latence élevée ($\sim 1-3$ s par image, selon la résolution), dépendance au biais du modèle, instabilité lors de l'entraînement	[Santurkar et al., 2018, Tschannen et al., 2018, Mentzer et al., 2020, Mentzer et al., 2020, Jiang et al., 2023, Yang and Mandt, 2023, Careil et al., 2024]
Taux compression variables	permettent de choisir dynamiquement le débit binaire, une seule architecture est nécessaire pour plusieurs niveaux de compression, complexité du modèle, pour débits spécifiques les performances peuvent être légèrement inférieures à un modèle optimisé uniquement pour ce taux, coût d'entraînement élevé	[Jia et al., 2019, Cai et al., 2019, Yang et al., 2021, Tao et al., 2023, Zhang et al., 2023f, Choi et al., 2019, Song et al., 2021b, Cai et al., 2022a, Shi et al., 2023, Tong et al., 2023, Liang et al., 2023, Guo et al., 2024, Lu et al., 2021, Sun et al., 2021, Cui et al., 2021, Lee et al., 2022a, Lee et al., 2022b, Yin et al., 2022, Guo et al., 2023, Qin et al., 2024]
ROI et orientés tâches	priorisation des zones d'intérêt, amélioration des performances pour les tâches ($\sim +5-10$ % AP@0.5 pour des taux de compression équivalents), réduction de la distorsion critique, qualité dégradée hors ROI	[Ballé et al., 2017, Akutsu and Naruko, 2020, Gao et al., 2023, Gao et al., 2022a, Liu et al., 2022a, Lei et al., 2023, Cao et al., 2023, Chen et al., 2023c, Liebender et al., 2024, Shindo et al., 2024a, Shindo et al., 2024b, Dai et al., 2024a, Li et al., 2024a, Yang et al., 2024, Liu et al., 2024]

TABLE 3.2 – Tableau récapitulatif de l'état de l'art des méthodes de Détection d'objets.

Méthodes	**Caractéristiques**	**Articles**
Two-stage	Performances généralement supérieures pour la localisation et la classification que single stage, meilleure gestion des objets petite taille et scènes encombrées, flexibilité, raffinement des prédictions, gestion efficace des déséquilibres foreground/background, temps inférence plus élevé par rapport aux modèles single-stage (Faster R-CNN : $\sim$ 7 FPS), moins adapté au temps réel, architecture plus complexe, coût computationnel élevé	[Girshick et al., 2014, He et al., 2014, Girshick, 2015, Ren et al., 2016, Dai et al., 2016, Lin et al., 2017a, He et al., 2017, Cai and Vasconcelos, 2018, Qin et al., 2019, Qiao et al., 2021]
Single-stage	Architecture plus simple que les modèles two-stage, Inférence en temps réel (YOLOv7 $\sim$ 30 FPS), Facilité à s'adapter à des résolutions élevées, apprentissage conjoint de la localisation et classification, moins précis sur objets petite taille ou scénarios complexes par rapport aux modèles two-stage, sensible au déséquilibre foreground/background durant entrainement, sensibilité accrue aux erreurs de localisation	[Redmon et al., 2016, Redmon and Farhadi, 2017, Farhadi and Redmon, 2018, Bochkovskiy et al., 2020, Wang et al., 2021a, Wang et al., 2021b, Ge et al., 2021, Li et al., 2022, Wang et al., 2023a, Xu et al., 2022, Jocher et al., 2023, yol, , Wang et al., 2024b, Wang et al., 2024a, Khanam and Hussain, 2024, Liu et al., 2016, Lin et al., 2017b, Tian et al., 2019, Zhou et al., 2019, Law and Deng, 2020, Tan et al., 2020]
Transformers	Mécanismes d'attention permettent de capturer dépendances longue portée entre objets dans une image, meilleure capacité à traiter des données haute résolution, flexibilité des données d'entrée, souvent en tête des benchmarks, adaptabilité aux grands ensembles de données, coût computationnel élevé, besoins en données massifs, moins efficace pour les petites échelles, plus lent ($\sim 5 - 10$ FPS pour DETR sur GPU haute performance), complexité d'entraînement	[Carion et al., 2020, Zhu et al., 2021, Dai et al., 2021, Meng et al., 2021b, Zang et al., 2022, Wang et al., 2022c, Zhang et al., 2022a, Dai et al., 2022, Cai et al., 2022b, Gupta et al., 2022, Ouyang, 2022, Ouyang, 2023a, Ouyang, 2023b, Xu et al., 2023a, Zhang et al., 2023a, Chen et al., 2023a, Ye et al., 2023, Pu et al., 2024, Zhang et al., 2023b, Zeng et al., 2024, Zhao et al., 2024b, Wang et al., 2021d, Xu et al., 2021, Zhang et al., 2023e, Zhang et al., 2021b, Fang et al., 2021, Gao et al., 2022b, Liu et al., 2023b, Ma et al., 2023]

TABLE 3.3 – Tableau récapitulatif de l'état de l'art des méthodes de Génération d'images.

Méthodes	**Caractéristiques**	**Articles**
VAE	facilité d'entraînement, images générées de moyenne qualité, difficulté à capturer les détails complexes, mode collapse possible même si rare	[Sohn et al., 2015, Gregor et al., 2015, Mansimov et al., 2016, van den Oord et al., 2017, Razavi et al., 2019, Huang et al., 2018, Daniel and Tamar, 2021]
GAN	Qualité visuelle meilleur que pour VAE, modélisation fine des textures, applications variées (conditionnelle, transfert style, inpainting), capacité de transfert, scalabilité, entraînement instable (mode collapse), difficulté à modéliser des distributions complexes, besoin de grandes bases de données	[Goodfellow et al., 2014, Mirza and Osindero, 2014, Denton et al., 2015, Radford et al., 2016, Chen et al., 2016, Isola et al., 2017, Zhu et al., 2017, Karras et al., 2018, Zhang et al., 2019, Karras et al., 2020, Karras et al., 2021, Sauer et al., 2022, Esser et al., 2021, Jiang et al., 2021, Lee et al., 2022c, Kim et al., 2021]
Diffusion	Qualité exceptionnelle des images, entrainement plus stable, modélisation probabiliste explicite, performances sur des résolutions élevées, applications variées, coût calculatoire élevé, complexité d'échantillonnage (itératif), besoin en mémoire GPU, modèles souvent volumineux	[Ho et al., 2020, Song et al., 2021a, Nichol and Dhariwal, 2021, Kingma et al., 2021, Meng et al., 2021a, Choi et al., 2021, Ramesh et al., 2021, Nichol et al., 2022, Rombach et al., 2022, Ramesh et al., 2022, Saharia et al., 2022, Sinha et al., 2021, Lu et al., 2022, Lyu et al., 2022, Zhang et al., 2023d, Bao et al., 2021, Kim et al., 2022, Avrahami et al., 2022, Ruiz et al., 2023, Yu et al., 2022, Gal et al., 2023, Bansal et al., 2024, Zhang et al., 2023c, Brooks et al., 2023, Bao et al., 2023, Podell et al., 2024, Xie et al., 2023, Du et al., 2024, Mou et al., 2024, Blattmann et al., 2023, Wang et al., 2024c, Zhao et al., 2024a, Hu et al., 2023, Gao et al., 2024, Su et al., 2024, Li et al., 2024b, Mei et al., 2024, Jiang et al., 2024a, Xue et al., 2023b, Chen et al., 2023b, Dai et al., 2024b, Lin et al., 2024, Haque et al., 2024, Kotyan et al., 2025]

Chapitre 4

Stratégies d'obtention de garanties

Contents

4.1 Introduction 118
4.2 Méthodes d'estimation statistique 120
4.2.1 Estimation par Bootstrap 120
Bootstrap déterministe 120
Boostrap epistémique 121
4.2.2 Notions de probabilité 122
4.3 Méthodologie pour l'obtention de garanties de performances 123
4.3.1 Identification de l'ODD et acquisition de données 124
4.3.2 Estimation Statistique et Probabiliste du Risque d'Erreur 127
Cadre générique 127
Cadre avec classes 131
4.3.3 Génération de données critiques aux modules d'IA 133
4.3.4 Interprétation de la méthode 136
4.3.5 Comment rajouter des données ? 138
4.4 Conclusion et perspectives 138

4.1 Introduction

L'obtention de garanties de performances pour les systèmes d'intelligence artificielle (IA) demeure un défi majeur, comme exploré dans le chapitre 2.

La vérification formelle des réseaux de neurones vise à garantir que ces modèles respectent certaines propriétés critiques, en particulier dans des applications où la fiabilité et la sécurité sont essentielles. Cette approche consiste à vérifier si la sortie d'un réseau de neurones f satisfait des contraintes définies sur ses plages d'entrée. Les méthodes de vérification formelle, telles que les solveurs SMT, la programmation mixte en nombres entiers (MILP), l'interprétation abstraite ou les techniques basées sur les constantes de Lipschitz, permettent de certifier des propriétés comme la robustesse aux perturbations ou la stabilité des prédictions. Elles offrent un haut niveau de confiance dans des scénarios spécifiques et se révèlent utiles pour des cas bien définis, comme notre cadre d'étude sur la chaîne compression-détection. Cependant, ces méthodes souffrent de limitations significatives :

- Elles sont souvent limitées à des réseaux de petite ou moyenne taille ou à des architectures simples.
- Les garanties fournies sont locales ou spécifiques à des perturbations bien caractérisées (par exemple, des bruits gaussiens ou adversariaux).
- Elles peinent à traiter des situations plus complexes ou déterministes, comme les distorsions causées par la compression d'images.
- Elles sont coûteuses en termes de calcul, rendant leur application impraticable pour des modèles modernes complexes ou des données haute dimension.

Dans le cadre spécifique de notre chaîne compression-détection, l'objectif est de garantir que les distorsions causées par la compression restent sous un seuil prédéfini, afin de ne pas compromettre significativement les performances du module de détection d'objets. Ces distorsions, assimilables à des attaques adversariales, posent

des défis de formalisation et de vérification en raison de la complexité des modèles impliqués et de la nature déterministe des perturbations. Dans ce contexte, les méthodes de vérification formelle se révèlent inadaptées, notamment parce qu'elles ne généralisent pas bien à l'ensemble des données possibles et qu'elles sont mal équipées pour traiter des modèles complexes avec des données de haute dimension.

D'autre part, les méthodes de certification statistiques offrent une alternative prometteuse pour garantir les performances des systèmes d'IA. Contrairement à la vérification formelle, qui repose sur des démonstrations exhaustives et déterministes, ces approches s'appuient sur des analyses empiriques et des hypothèses probabilistes. Elles permettent d'estimer, avec un certain niveau de confiance, la probabilité qu'un système respecte une propriété donnée ou fonctionne correctement dans un ensemble de conditions spécifiées. Ces méthodes, telles que le *Randomized Smoothing*, la PAC-Bayes, ou la *Conformal Prediction*, se révèlent particulièrement adaptées à des contextes où une évaluation exhaustive est impraticable, en raison de la complexité ou de la taille des modèles et des données. Cependant, elles présentent également des limites :

- Les garanties probabilistes dépendent fortement de la représentativité des données utilisées pour l'évaluation et la calibration.
- Ces approches nécessitent souvent de grandes quantités de données pour produire des estimations fiables.
- Elles peuvent s'avérer coûteuses en calcul pour des modèles complexes ou des espaces d'entrée de haute dimension.

Malgré leurs limites, ces méthodes constituent une base précieuse dont on va s'inspirer pour développer une méthodologie de garanties adaptée aux systèmes intégrant des modules d'IA.

Il n'existe pas encore de solution universelle permettant de démontrer systématiquement que les systèmes respectent certaines propriétés spécifiques. Ces limitations sont exacerbées par la croissance de la complexité des modèles, la diversité des garanties recherchées, et la variabilité des scénarios testés. La malédiction des dimensions aggrave ces défis, rendant les méthodes traditionnelles coûteuses et difficilement applicables à grande échelle. Cela souligne le besoin de solutions innovantes pour analyser des modèles d'IA sophistiqués et garantir leur fiabilité. Dans ce contexte, la validation empirique reste la méthode dominante, en particulier dans les applications industrielles. Elle consiste à évaluer les performances des modèles sur des ensembles de données représentatifs et des scénarios critiques ou extrêmes. Les simulateurs tels que CARLA [1] ou Gazebo [2] permettent de compléter les données réelles en explorant des configurations difficiles à reproduire physiquement. Cependant, cette validation ne fournit pas de garanties strictes : elle repose sur des observations limitées et ne couvre qu'une fraction de l'espace des données possibles.

L'objectif de cette thèse est de développer une méthodologie permettant d'obtenir des garanties de performances pour des modules d'IA. Ces garanties sont définies comme des probabilités d'erreur estimées statistiquement, conditionnées par des hypothèses explicites (par exemple, la représentativité des datasets ou la validité des modèles statistiques). Cette méthodologie vise à fournir une évaluation probabiliste robuste, tout en identifiant les vulnérabilités potentielles des systèmes et en quantifiant les risques d'erreur dans des scénarios variés. Dans ce contexte, une garantie de performance correspond à une probabilité d'erreur associée à des intervalles de confiance et des hypothèses clairement définies. L'objectif est d'offrir une base solide pour optimiser et renforcer les performances des systèmes étudiés.

Dans notre approche, nous nous inspirerons des méthodes PAC-Bayes et Conformal Risk Control (CRC), qui exploitent respectivement une distribution de probabilité sur les paramètres du réseau ou une distribution sur les données d'entrée pour fournir des garanties de performance. PAC-Bayes encadre l'erreur de généralisation en intégrant des distributions a priori et a posteriori sur les paramètres des modèles, tandis que CRC se concentre sur le contrôle des risques agrégés tels que le taux d'erreur ou le risque attendu en s'appuyant sur des données calibrées. Nous adopterons des méthodes statistiques et probabilistes pour pallier les insuffisances des approches traditionnelles. Ces techniques permettront d'estimer les probabilités d'erreur à partir d'échantillons représentatifs tout en quantifiant l'incertitude associée via des intervalles de confiance ou des barres d'erreur. En combinant ces inspirations, notre méthode intégrera une dimension probabiliste solide, non seulement pour évaluer les performances globales des systèmes, mais aussi pour identifier leurs vulnérabilités dans des scénarios variés. Ce cadre méthodologique vise à offrir une évaluation fiable des performances en tenant compte de la variabilité des données et des conditions opérationnelles. Il permettra également de mieux anticiper les comportements des modèles face à des perturbations ou des configurations rares, tout en garantissant des prédictions robustes et bien calibrées dans des contextes complexes.

1. https://carla.org/
2. https://gazebosim.org/home

4.2 Méthodes d'estimation statistique

Dans cette section, nous nous concentrerons sur les garanties statistiques associées à l'évaluation des performances du système. Ces garanties, loin d'être absolues, reposent sur des estimations statistiques basées sur les données disponibles. Nous explorerons notamment l'estimation des incertitudes, qui permet de quantifier les marges d'erreur associées à une mesure ou une prédiction. À cet effet, des méthodes comme le bootstrap seront présentées. Cette technique, qui repose sur des rééchantillonnages des données, permet d'estimer empiriquement les incertitudes sans présupposer une distribution probabiliste sous-jacente, offrant ainsi une flexibilité adaptée aux contextes où les distributions sont mal connues.

Par ailleurs, nous introduirons certains principes fondamentaux de la probabilité, tels que le théorème des espérances totales, pour mieux appréhender les disparités et les inégalités présentes dans les jeux de données. Ces outils théoriques seront essentiels pour intégrer des variations inhérentes aux données et garantir une approche rigoureuse dans nos analyses. Ensemble, ces méthodes offriront un cadre robuste pour évaluer les performances et les limites du système en tenant compte des incertitudes et des hétérogénéités dans les données.

4.2.1 Estimation par Bootstrap

La méthode Bootstrap est une technique statistique non paramétrique introduite par Bradley Efron en 1979 [Efron, 1979]. Elle est utilisée pour estimer la distribution d'une statistique (comme la moyenne, la médiane ou l'écart-type) à partir d'un échantillon de données. Contrairement aux méthodes traditionnelles d'inférence statistique, qui reposent souvent sur des hypothèses strictes concernant la distribution sous-jacente des données (comme par exemple la loi normale), le bootstrap offre une approche plus flexible et robuste, particulièrement utile lorsque ces hypothèses ne sont pas vérifiées ou que la taille de l'échantillon est petite. Les méthodes traditionnelles, telles que l'estimation d'intervalles de confiance basés sur des approximations normales ou l'utilisation de tests statistiques paramétriques, nécessitent généralement des hypothèses spécifiques sur la distribution des données. Par exemple, il est souvent présumé que les données suivent une distribution normale ou que les échantillons sont suffisamment grands pour que la loi des grands nombres puisse être invoquée. Le bootstrap, en revanche, ne fait pas de telles hypothèses. Il repose sur la méthode de rééchantillonnage avec remise, ce qui permet d'estimer la distribution de la statistique d'intérêt directement à partir de l'échantillon de données observé. En générant un grand nombre d'échantillons bootstrap, on peut alors construire une distribution empirique de la statistique, à partir de laquelle des intervalles de confiance et d'autres mesures d'incertitude peuvent être dérivés. Cependant, bien que le bootstrap soit généralement robuste, il peut introduire un biais dans certaines situations, particulièrement lorsque l'échantillon de données est très petit ou que les données contiennent des valeurs extrêmes. Le bootstrap suppose que l'échantillon observé est représentatif de la population sous-jacente, ce qui peut ne pas être le cas. Le succès du bootstrap dépend de l'hypothèse que l'échantillon observé est un bon représentant de la population. Si cette hypothèse est violée, les estimations obtenues peuvent être inexactes.

Bootstrap déterministe

La méthode s'applique de la manière suivante (cf. Algorithme 1) : supposons qu'on dispose d'un échantillon de données $X = \{x_1, x_2, ..., x_n\}$ de taille n, indépendants et identiquement distribuées (iid) selon une loi de probabilité inconnue, et on souhaite estimer la statistique θ, comme par exemple l'espérance de la loi, sa médiane ou encore sa variance. Pour cela, on crée un grand nombre B (en général $B \geq 1000$), appelés "échantillons bootstrap", chacun de taille n en échantillonnant aléatoirement avec remise à partir de l'échantillon original X. Cela signifie que chaque observation du jeu de données initial peut apparaître plusieurs fois ou ne pas apparaître du tout dans un échantillon bootstrap. Ensuite, pour chaque échantillon bootstrap, on calcul l'estimateur d'intérêt θ (par exemple, la moyenne, la médiane, la variance, etc.). Cela donne une distribution d'estimations appelée distribution bootstrap $\hat{\theta} = \{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_B\}$. Enfin, l'estimation de la statistique θ est simplement la moyenne des valeurs de la distribution bootstrap : $\bar{\hat{\theta}} = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}_b$. Pour calculer un intervalle de confiance de niveau $1-\alpha$, il existe plusieurs méthodes mais nous utiliserons la méthode des percentiles qui consiste à trier les valeurs $\hat{\theta}_b$ de la distribution bootstrap par ordre croisant, puis à déterminer empiriquement les quantiles $\hat{q}_{\frac{\alpha}{2}}$ et $\hat{q}_{1-\frac{\alpha}{2}}$ (cf. Figures 4.1) : $\text{IC}_{1-\alpha} = \left[\hat{q}_{\frac{\alpha}{2}}, \hat{q}_{1-\frac{\alpha}{2}}\right]$.

Il est important de noter que la précision sur l'intervalle de confiance estimé $\text{IC}_{1-\alpha}$ dépend de n selon $P(\bar{\hat{\theta}} \in \text{IC}_{1-\alpha}) = 1-\alpha+O(n^{-1})$, plus n est grand plus l'estimation de la statistique et de l'intervalle de confiance associé est fiable.

Algorithm 1 Méthode Bootstrap

Entrées : Échantillon original $X = \{x_1, x_2, \dots, x_n\}$, Nombre d'échantillons bootstrap B
Sortie : Distribution bootstrap de la statistique $\hat{\theta}$
Pour $b = 1$ à B **faire**
    $X_b \leftarrow$ EchantillonBootstrap(X) ▷ Tirage aléatoire **avec remise** à partir de X
    $\hat{\theta}_b \leftarrow$ CalculerStatistique(X_b) ▷ Calculer la statistique $\hat{\theta}$ sur l'échantillon bootstrap X_b
Fin Pour
Retourner $\{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_B\}$ ▷ Retourner la distribution bootstrap de la statistique

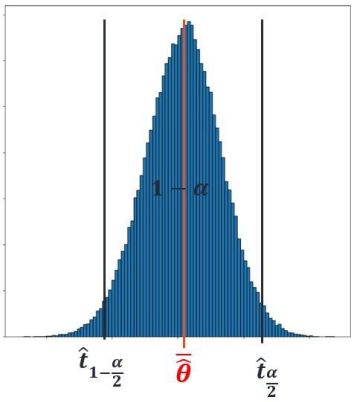


FIGURE 4.1 – Distribution boostrap de l'estimateur θ.

Boostrap epistémique

Dans le cas où les données observées sont incertaines, une variante de la méthode d'estimation par bootstrap, appelée bootstrap épistémique, peut être utilisée.

Selon [Grzegorzewski and Romaniuk, 2022], traditionnellement, la plupart des résultats expérimentaux sont des données à valeurs réelles, et les méthodes statistiques se réfèrent généralement à de telles données. Cependant, les mesures dans la vie réelle sont souvent imprécises. De plus, dans de nombreuses situations où les résultats sont à valeurs réelles, nous sommes en réalité confrontés à leurs perceptions qui ne sont pas nécessairement précises mais plutôt vagues. On fait souvent appelle a la théorie des ensembles flous qui est un outil efficace pour modéliser et analyser l'imprécision. Dans notre cas, supposons que notre échantillon $X_1, ..., X_n$ se compose de n variables aléatoires indépendantes et identiquement distribuées provenant d'une distribution inconnue. Cependant, au lieu d'une réalisation à valeur réelle $(x_1, ..., x_n)$ de cet échantillon, nous n'observons que sa perception imprécise, modélisée par un échantillon flou $(\tilde{x}_1, ..., \tilde{x}_n)$ tel que chaque $\tilde{x}_i$ représente un couple $(x_i, [x_i^-, x_i^+]_{1-\beta})$ où x_i est la réalisation la plus probable et $[x_i^-, x_i^+]_{1-\beta}$ est un intervalle de confiance de niveau $1-\beta$ autour de x_i.

La méhtode boostrap épistémique (voir Algorithme 2) est similaire à la méthode bootstrap déterministe (cf. section 4.2.1) avec une étape d'échantillonnage supplémentaire : on crée un grand nombre B d'échantillons bootstrap, chacun de taille n en échantillonnant aléatoirement avec remise à partir de l'échantillon flou original $\tilde{X}$, puis pour chaque échantillon $\tilde{x}_i$ on ré-échantillonne à nouveau selon une loi normale centré en x_i et d'écart-type $\sigma_{1-\beta} = \frac{x_i^+ - x_i^-}{2.q_{1-\beta/2}}$ de manière que selon cette loi normale l'intervalle $[x_i^-, x_i^+]$ soit de probabilité $1-\beta$ (cf. Figure 4.2). Ensuite, pour chaque échantillon bootstrap, on calcul l'estimateur d'intérêt θ, ce qui nous donne une distribution bootstrap $\hat{\theta} = \{\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_B\}$. L'estimation de la statistique θ et d'un intervalle de confiance se fait de la même manière que pour la méthode déterministe.

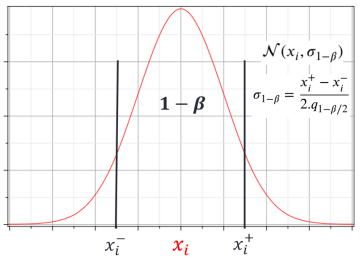


FIGURE 4.2 – Loi normale centré en x_i et d'écart-type $\sigma_{1-\beta}$

Algorithm 2 Méthode Bootstrap Épistemique

Entrées : Échantillon flou original $\tilde{X} = \{\tilde{x}_i = (x_i, [x_i^-, x_i^+]_{1-\beta})\}_{i=1}^n$, Nombre d'échantillons bootstrap B
Sortie : Distribution bootstrap de la statistique $\hat{\theta}$
Pour $b = 1$ à B **faire**
    $\tilde{X}_b \leftarrow$ EchantillonBootstrap($\tilde{X}$) ▷ Tirage aléatoire **avec remise** à partir de $\tilde{X}$
    Pour $i = 1$ à n **faire**
        $X_b[i] \leftarrow$ Echantillon($\tilde{X}_b[i] = \tilde{x}_i \sim \mathcal{N}(x_i, \sigma_{1-\beta})$) ▷ Échantillonnage selon $\mathcal{N}(x_i, \sigma_{1-\beta} = \frac{x_i^+ - x_i^-}{2.q_{1-\beta/2}})$
    Fin Pour
    $\hat{\theta}_b \leftarrow$ CalculerStatistique(X_b) ▷ Calculer la statistique $\hat{\theta}$ sur l'échantillon bootstrap X_b
Fin Pour
Retourner $\{\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_B\}$ ▷ Retourner la distribution bootstrap de la statistique

Dans la suite, nous introduirons quelques notions de probabilités nécessaires, notamment celles liées à l'estimation de l'espérance mathématique et au théorème des espérances totales.

4.2.2 Notions de probabilité

Considérons l'espace de probabilité $(\Omega, \mathcal{F}, \mathbb{P})$, où Ω, aussi appelé "univers", est l'ensemble des résultats possibles d'une expérience aléatoire, $\mathcal{F}$ est une tribu (ensemble d'événements mesurables), et $\mathbb{P}$ est une mesure de probabilité définissant la probabilité de chaque événement dans $\mathcal{F}$. Soit $X : \Omega \to \mathbb{R}^d$ une variable aléatoire qui associe à chaque événement $\omega \in \Omega$ une valeur déterministe $x \in \mathbb{R}^d$, représentant une observation ou une réalisation de la variable aléatoire dans l'espace des états $\mathbb{R}^d$.

Pour estimer l'espérance mathématique $\mathbb{E}[X]$ d'une variable aléatoire X à partir d'un échantillon de données, la méthode la plus courante est d'utiliser la moyenne empirique de l'échantillon. Supposons que nous disposions d'un échantillon $X_1, X_2, \ldots, X_n$ de n observations indépendantes identiquement distribuées (i.i.d.) de X. L'estimateur de l'espérance $\mathbb{E}[X]$ est alors donné par la moyenne empirique :

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i \tag{4.1}$$

La justification théorique de cet estimateur repose sur le théorème de la loi des grands nombres (LGN), qui assure que sous certaines conditions (comme l'indépendance et l'identique distribution des X_i ainsi que la finitude de $\mathbb{E}[X]$), la moyenne empirique $\hat{\mu}$ converge presque sûrement vers l'espérance véritable $\mathbb{E}[X]$ lorsque la taille de l'échantillon n tend vers l'infini. Cela signifie que plus l'échantillon est grand, plus l'estimation de l'espérance mathématique par la moyenne empirique est précise.

Le théorème de l'espérance totale est un autre outil fondamental en théorie des probabilités, utilisé pour calculer l'espérance d'une variable aléatoire en décomposant le problème en cas plus simples. Supposons que X soit une variable aléatoire et que $\{A_1, A_2, \ldots, A_n\}$ soit une partition de l'espace probabilisé Ω, c'est-à-dire que les événements A_i sont mutuellement exclusifs et que leur union couvre tout l'espace Ω :

$$\Omega = \bigcup_{i=1}^{n} A_i \quad \text{et} \quad A_i \bigcap A_j = \emptyset \quad \forall i \neq j \tag{4.2}$$

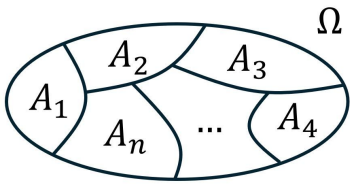


FIGURE 4.3 – Découpage de Ω en n partition disjointes A_i.

alors le théorème de l'espérance totale affirme que l'espérance de X peut être exprimée comme une somme pondérée des espérances conditionnelles par rapport à chaque événement A_i :

$$\mathbb{E}[X] = \sum_{i=1}^{n} \mathbb{E}[X \mid A_i].\mathbb{P}(A_i) \tag{4.3}$$

Cela signifie que pour obtenir l'espérance totale de X, il suffit de prendre l'espérance de X conditionnée sur chaque A_i (c'est-à-dire l'espérance de X sachant que l'événement A_i s'est produit) et de pondérer chaque espérance conditionnelle par la probabilité que A_i se produise. Ce théorème est particulièrement utile dans les situations où il est plus facile de calculer les espérances conditionnelles que l'espérance globale directement.

4.3 Méthodologie pour l'obtention de garanties de performances

Dans cette partie, nous allons décrire les différentes étapes permettant d'obtenir des garanties de performances pour un système composé de modules d'IA. L'objectif est de s'appuyer sur des outils statistiques et probabilistes pour formaliser et quantifier les performances de tels systèmes, en tenant compte des incertitudes et des variations dans les données.

Considérons un système S, composé de modules d'IA, représenté par une fonction F qui prend en entrée X, provenant d'un espace $\mathcal{X}$, et produit une prédiction Y appartenant à l'espace $\mathcal{Y}$. Notre objectif est de garantir que les performances de ce système respectent un certain niveau de tolérance, défini en termes d'une fonction d'erreur.

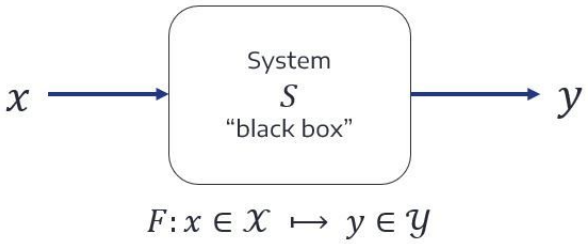


FIGURE 4.4 – Système composé de modules d'IA.

Soit $e : \mathcal{X} \to \mathbb{R}^+$ une fonction d'erreur qui associe à chaque entrée $x \in \mathcal{X}$ une valeur numérique positive ou nulle. Contrairement à une fonction de perte utilisée dans le cadre de l'apprentissage des modèles, cette fonction d'erreur sert à mesurer la qualité des prédictions du système sur une nouvelle donnée x et elle soit être interprétable par l'utilisateur (ce qui n'est pas toujours le cas avec une fonction de coût pour l'entrainement).

Nous cherchons à évaluer la probabilité que l'erreur dépasse un seuil de tolérance donné $\epsilon > 0$. Cette probabilité s'écrit sous la forme suivante :

$$\mathbb{P}(e(x) \geq \epsilon) \tag{4.4}$$

où x est une donnée issue de l'espace d'entrée $\mathcal{X}$, suivant une distribution qui peut refléter les conditions d'utilisation réelle du système. Dans ce cadre, les garanties de performance que nous souhaitons établir consistent à fournir des estimations précises de cette probabilité, en tenant compte des propriétés du système, de ses modules d'IA, et des données utilisées pour son évaluation.

4.3.1 Identification de l'ODD et acquisition de données

La première étape dans la conception d'un système autonome consiste à définir l'*Operational Design Domain* (ODD), ou Domaine Opérationnel de Conception, qui constitue le cadre de référence pour l'acquisition des données nécessaires à sa conception et à sa validation.

Il existe de nombreuses recherches, notamment industrielles, pour développer des méthodes permettant de définir l'ODD de manière structurée et exhaustive. L'article [Mehlhorn et al., 2023] explore les avancées récentes dans la modélisation, la standardisation et l'évaluation des ODD pour les systèmes autonomes. Il souligne des défis cruciaux, tels que l'absence d'une terminologie standardisée, la difficulté à définir des scénarios exhaustifs, et les obstacles à l'intégration des cas limites, également appelés *Edge Cases*. Ces défis mettent en lumière la nécessité d'une méthodologie rigoureuse pour garantir que l'ODD couvre l'ensemble des situations pertinentes.

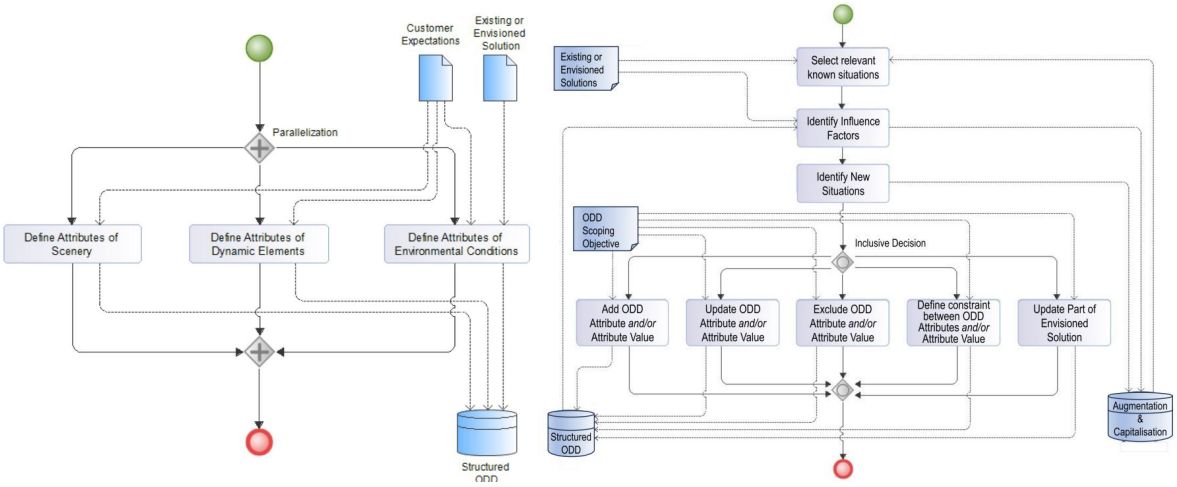
L'aspect fondamental de l'ODD réside dans la définition et la structuration de ses attributs, qui représentent les dimensions critiques où le système doit opérer en toute sécurité. Ces attributs incluent [ODD, 2021] (liste non exhaustive) :

- **Conditions environnementales** : Ces paramètres incluent la météo (pluie, neige, brouillard), la luminosité (jour, nuit), et les particules (poussière, fumée). Ils influencent directement la performance des capteurs et des mécanismes de décision.
- **Infrastructures physiques** : Le type d'infrastructures (sites industriels, bâtiments, tunnels), les surfaces de travail (sols, routes, plateformes) et les marquages ou limites physiques définissent les zones opérationnelles.
- **Zones** : Certaines zones spécifiques, comme des environnements urbains ou des zones de travail contrôlées, imposent des restrictions supplémentaires sur le fonctionnement du système.
- **Conditions dynamiques** : Cela inclut la densité des éléments présents dans l'environnement (machines, piétons, véhicules) et les événements imprévus tels que des obstacles.
- **Classes d'objets** : Le système doit détecter et interagir avec divers objets, qu'il s'agisse de machines, d'humains ou d'obstacles inattendus, en adaptant son comportement en conséquence.
- **Contraintes opérationnelles** : Elles englobent des éléments tels que les limites de vitesse, les caractéristiques du trafic, les zones de construction, et les règles spécifiques à certaines zones (par exemple, réduction de vitesse dans une zone scolaire). Ces contraintes doivent être prises en compte pour garantir une adaptation appropriée du système.

Cependant, comme le souligne l'étude [Heyn et al., 2022], dériver un ODD à partir de données réelles reste un défi considérable. Il est souvent difficile de décrire des paramètres non numériques avec précision, et il n'existe pas de cadre standardisé permettant de garantir que tous les scénarios pertinents ont été identifiés. Cette incertitude est particulièrement problématique pour les scénarios rares, qui exigent des hypothèses souvent mal documentées. Des initiatives comme l'ISO 34503 [iso, 2023] proposent une taxonomie hiérarchique pour structurer les attributs de l'ODD, facilitant ainsi la catégorisation des conditions de fonctionnement, des objets dynamiques, et des types d'infrastructures. Cette structuration aide à garantir une couverture des scénarios de test plus complète et systématique.

Les approches basées sur les ontologies jouent un rôle clé dans l'organisation des composants de l'ODD. Une ontologie fournit une structure formelle pour représenter les connaissances d'un domaine, en définissant des concepts, leurs relations, et leurs interdépendances. Par exemple, une méthode basée sur les ontologies [Chen and Kloul, 2018] utilise des structures hiérarchiques pour générer des cas d'utilisation, en commençant par des couches statiques (infrastructures fixes), des couches d'interaction (relations entre objets dynamiques et statiques), et des couches de génération (création de scénarios). Ces approches permettent d'aligner les composants de l'ODD avec des tests scénarisés, assurant ainsi une évaluation robuste. D'autres travaux, comme [Jatzkowski et al., 2021], automatisent la création de graphes de compétences pour modéliser les capacités du système et leurs dépendances dans un ODD donné. Ces graphes facilitent le développement et la validation des systèmes autonomes, tout en réduisant la dépendance aux processus manuels.

Récemment, l'étude [Adedjouma et al., 2024] propose une méthodologie en trois étapes. Tout d'abord, l'initialisation de l'ODD (cf. Figure4.5a) consiste à capturer des attributs tels que le paysage, les éléments dynamiques et les conditions environnementales en fonction des attentes des utilisateurs. Ensuite, le raffinement de l'ODD (cf. Figure 4.5b) s'appuie sur des approches analytiques et basées sur les risques, en exploitant les retours d'expérience issus de systèmes similaires pour affiner les descriptions initiales. Enfin, la consolidation de l'ODD vise à identifier les éléments redondants et à garantir la conformité avec les exigences sectorielles et réglementaires. Cette approche structurée assure que les définitions des ODD restent complètes, cohérentes et adaptables aux besoins en constante évolution.



(a) Processus d'initialisation de l'ODD.

(b) Processus de raffinement de l'ODD.

FIGURE 4.5 – Illustration de la méthode proposée par [Adedjouma et al., 2024].

La complexité d'un Operational Design Domain (ODD) est directement liée au nombre et à la variabilité de ses attributs. Plus les attributs sont nombreux et variés, plus l'ODD sera complexe, reflétant une plus grande diversité dans les données et les situations que le système doit gérer. Cette complexité dépend fortement du système autonome étudié. Pour des systèmes simples, tels qu'un système de reconnaissance de défauts sur une pièce industrielle dans une chaîne de production en entrepôt, l'ODD est relativement limité. Dans ce cas, les conditions environnementales sont contrôlées et constantes, la lumière est stable, et la pièce analysée est toujours identique. Les attributs tels que les infrastructures physiques, les zones géographiques et les conditions dynamiques sont donc fixes et peu variés. La tâche se résume essentiellement à collecter suffisamment d'images de la pièce, avec et sans défaut, pour entraîner et valider le système. En revanche, dans des contextes plus complexes comme la conduite autonome, l'ODD d'un système, par exemple une caméra frontale embarquée pour la détection d'objets d'intérêt, est nettement plus varié. Les conditions environnementales, telles que la météo ou l'illumination, les infrastructures physiques, comme le type de route ou la qualité des marquages, ainsi que les conditions dynamiques, incluant la densité du trafic et les interactions avec des objets mobiles, rendent l'ODD beaucoup plus riche et complexe.

L'ODD est défini au début du processus de conception du système, reflétant les capacités opérationnelles initiales. Une fois l'ODD établi, il devient crucial de concevoir un ensemble représentatif de scènes et de scénarios couvrant l'ensemble de ses dimensions. Ces scènes doivent inclure non seulement les cas fréquents, mais aussi des scénarios rares ou critiques, généralement qualifiés *Edge and Corner*, susceptibles de mettre en difficulté le système. Les données nécessaires peuvent être collectées à partir d'observations réelles, impliquant des campagnes de mesure dans des environnements variés, ou générées à l'aide de simulations contrôlées. Les données réelles, obtenues via des capteurs, caméras ou appareils de mesure, permettent de capturer fidèlement les attributs de l'ODD, tels que les conditions météorologiques, les caractéristiques des infrastructures et les objets présents dans la scène. Ces données doivent ensuite être organisées en fonction des attributs définis dans l'ODD pour garantir une analyse et une validation structurées (cf. Figure 4.6). Une attention particulière doit être portée à l'acquisition des données dites Edge and Corner Cases – des situations limites ou rares qui, bien que peu fréquentes, peuvent être critiques pour évaluer la robustesse du système. Ces cas extrêmes permettent de tester les limites du système et de renforcer sa capacité à gérer des scénarios imprévus. Les simulations, en particulier, se révèlent précieuses pour générer ces situations difficiles à capturer dans des environnements réels.

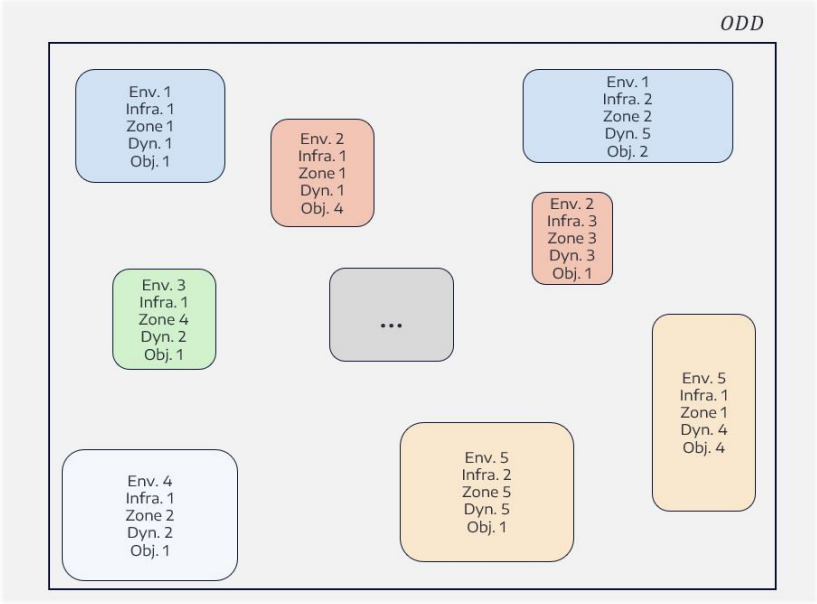


FIGURE 4.6 – Illustration d'un ensemble de jeux de données regroupées par attributs de l'ODD.

Ensuite, si nécessaire, ces données doivent être annotées, une tâche souvent coûteuse et laborieuse, mais essentielle pour l'apprentissage et l'évaluation de modèles d'IA. D'autre part, les environnements de simulation permettent de générer des scénarios synthétiques en ajustant précisément les paramètres de l'ODD. Ces simulations offrent une flexibilité considérable pour explorer des situations extrêmes, simuler des défaillances ou tester des configurations impossibles à reproduire dans la réalité. L'utilisation conjointe des données réelles et simulées est cruciale pour couvrir l'ensemble du spectre des scénarios de test. Les données réelles renforcent la validation dans des environnements authentiques, tandis que les simulations permettent de compléter ces tests en créant des cas rares, critiques ou complexes. Ce processus, structuré autour de l'ODD, permet une évaluation robuste et exhaustive des performances du système autonome dans ses limites opérationnelles. Dans le cadre de la validation et de la vérification, l'ODD fournit également un cadre pour structurer les tests et évaluer la sécurité des systèmes dans des conditions variées. La génération de scénarios à partir de l'ODD, bien qu'elle reste un défi, permet d'identifier les cas critiques et d'optimiser les procédures de test pour garantir la robustesse du système.

Bien évidemment, il n'est pas envisageable d'imaginer et de dresser la liste de toutes les scènes et de tous les scénarios possibles. Certains resteront inconnus. D'ailleurs, l'ISO 21448 [iso, 2022] distingue clairement entre les scénarios connus, sur lesquels le système peut être testé, et les scénarios inconnus (cf. Figure 1.2b), ce qui constitue justement tout l'intérêt du système de *Field Monitoring*. Pour gérer ces scénarios inconnus, diverses méthodes proposées par [iso, 2022] permettent d'évaluer les risques résiduels liés à des comportements dangereux potentiels du système. Ces approches, telles que les simulations aléatoires, les retours d'expérience et les comparaisons avec des systèmes existants, visent à mieux anticiper les défaillances potentielles. L'objectif est de démontrer que la probabilité de rencontrer un scénario inconnu entraînant un comportement dangereux est suffisamment faible. Mais ces approches restent tout de même limitées.

En conclusion, l'ODD joue un rôle central dans la conception, le développement et la validation des systèmes autonomes. Il permet de définir les limites d'opération et de structurer les scénarios nécessaires pour garantir la sécurité et l'efficacité du système. Cependant, l'ODD n'est pas une entité figée tout au long de ce processus. Bien au contraire, il s'agit d'un cadre évolutif qui peut être ajusté pour refléter plus fidèlement les limites réelles de fonctionnement du système. Lors des premières phases de conception, une définition a priori de l'ODD est établie, basée sur les capacités prévues du système et les attentes des utilisateurs. Cette définition initiale sert de point de départ pour guider la collecte de données, la conception des scénarios de test, et le développement des modules du système. Toutefois, au fil du développement et des phases de test, le système peut rencontrer des difficultés à satisfaire toutes les conditions ou exigences de l'ODD initial. Dans ce cas, une redéfinition a posteriori de l'ODD devient nécessaire. Ce processus consiste à ajuster les limites de l'ODD pour exclure certains scénarios ou conditions dans lesquels le système ne peut fonctionner de manière fiable ou sécurisée, ou

pour intégrer de nouvelles découvertes issues des tests. Cette approche itérative garantit que l'ODD reste réaliste et pertinent, en alignant les attentes opérationnelles sur les capacités réelles du système. De plus, l'ODD doit évoluer en parallèle des avancées technologiques et des changements dans les environnements opérationnels. Des outils modernes, tels que les modèles de langage (LLMs) entraînés sur de grandes quantités de données, offrent aujourd'hui une opportunité unique d'identifier de manière exhaustive les paramètres critiques de l'ODD. En combinant des définitions itératives avec ces nouvelles technologies, il est possible de renforcer la fiabilité et la sécurité des systèmes autonomes tout au long de leur cycle de vie, tout en maintenant un cadre clair pour évaluer leurs performances et leur robustesse.

4.3.2 Estimation Statistique et Probabiliste du Risque d'Erreur

Comme décrit précédemment, nous considérons un système S composé d'un ou plusieurs modules d'IA. Ce système prend en entrée une donnée x appartenant à un espace $\mathcal{X}$ et produit en sortie une donnée y dans un espace $\mathcal{Y}$, via une fonction $F : \mathcal{X} \to \mathcal{Y}$ (cf. Figure 4.4).

Pour évaluer les performances du système, nous définissons une fonction d'erreur $e : \mathcal{X} \to \mathbb{R}^+$, qui associe à chaque donnée d'entrée x une mesure quantifiant l'erreur commise par le système. Contrairement aux fonctions de coût (*loss*) utilisées pour l'entraînement des modèles, la fonction d'erreur e doit être interprétable et compréhensible pour les utilisateurs. De plus, la fonction e doit croître lorsque l'erreur augmente, reflétant ainsi de manière monotone la gravité de l'écart par rapport à une certaine référence. Elle fournit une indication claire sur les performances du système dans des conditions réelles, jouant ainsi un rôle essentiel dans l'évaluation de la qualité des prédictions réalisées. Nous introduisons également un seuil d'erreur acceptable $\epsilon > 0$, représentant le niveau maximal d'erreur tolérable. L'objectif est alors de fournir une estimation probabiliste sous la forme $\mathbb{P}(e(x) \geq \epsilon)$, qui exprime la probabilité qu'une donnée x d'entrée entraîne une erreur $e(x)$ supérieure ou égale au seuil ϵ lorsque le système S est utilisé. Cette probabilité peut être interprétée comme une mesure de la probabilité de défaillance du système. Elle constitue une garantie de performance, reflétant la robustesse et la fiabilité du système dans un cadre générique.

Une fois l'ODD défini et les données nécessaires acquises, nous supposons disposer d'un ensemble de jeux de données $\mathcal{D}$, constitué de d sous-ensembles $\mathcal{D} = D_1, \ldots, D_d$, chacun structuré en fonction des attributs définis par l'ODD (cf. Figure 4.6). Ces sous-ensembles sont supposés disjoints, c'est-à-dire qu'aucune donnée ne peut appartenir simultanément à plusieurs D_i.

Dans un premier temps, nous établirons une méthodologie générale pour l'estimation des garanties de performances, applicable à tout système S, quel que soit son objectif ou la nature des modules d'IA qui le composent. Cette approche générique servira de base pour une analyse rigoureuse et indépendante des spécificités contextuelles du système. Par la suite, ce cadre sera adapté à des cas particuliers, notamment lorsque l'évaluation des performances dépend d'un ensemble de classes, comme dans les tâches de classification, de détection d'objets ou de segmentation sémantique. Cette distinction permettra de garantir une flexibilité et une pertinence accrues dans l'analyse des performances pour diverses applications.

Cadre générique

Soit X une variable aléatoire définie sur l'espace probabilisé $(\Omega, \mathcal{F}, P)$, où Ω représente l'ensemble des résultats possibles (l'univers des événements), $\mathcal{F}$ est une tribu (ensemble des événements mesurables), et P est une mesure de probabilité qui attribue une valeur entre 0 et 1 à chaque événement $A \in \mathcal{F}$.

Nous cherchons à estimer une probabilité de la forme $\mathbb{P}(e(X) \geq \epsilon)$, qui représente la probabilité que l'erreur $e(X)$ dépasse un seuil de tolérance ϵ. Pour cela, nous définissons une variable aléatoire Z sur Ω, telle que :

$$Z(X) = \begin{cases} 0 \text{ si } e(X) < \epsilon \\ 1 \text{ si } e(X) \geq \epsilon \end{cases} = \mathbb{1}_{\{e(X) \geq \epsilon\}} \tag{4.5}$$

où $\mathbb{1}_{e(X) \geq \epsilon}$ est la fonction indicatrice de l'événement $e(X) \geq \epsilon$. Cette fonction prend la valeur 1 lorsque l'erreur $e(X)$ dépasse ou atteint le seuil ϵ, et la valeur 0 sinon.

En calculant l'espérance de la variable aléatoire Z, nous obtenons :

$$\mathbb{E}[Z] = 1 \cdot \mathbb{P}(e(X) \geq \epsilon) + 0 \cdot \mathbb{P}(e(X) < \epsilon) = \mathbb{P}(e(X) \geq \epsilon) \tag{4.6}$$

Ainsi, la probabilité $\mathbb{P}(e(X) \geq \epsilon)$ peut être interprétée comme l'espérance mathématique de la variable aléatoire Z, soit :

$$\mathbb{P}(e(X) \geq \epsilon) = \mathbb{E}[Z(X)] \tag{4.7}$$

Cette formulation offre une simplification conceptuelle importante en ramenant l'estimation de la probabilité recherchée au calcul d'une espérance. Grâce à cette définition, nous pouvons appliquer des outils statistiques et probabilistes tels que le théorème de l'espérance totale ou des techniques d'estimation pour décomposer et calculer $\mathbb{E}[Z(X)]$ dans divers contextes. Cela permet notamment de prendre en compte les particularités des données disponibles, des sous-populations étudiées, ou des distributions conditionnelles, tout en maintenant une approche rigoureuse et systématique.

Cette esperance n'est pas reellement calculable car cela engendrerai une intégrale sur Ω qui est souvent tres grand et de plus nous ne connaissons pas la distribution de probabilité sous-jacente à X et Z :

$$\mathbb{E}[Z(X)] = \int_{\Omega} Z(X(\omega))\mathrm{d}\mathbb{P}(\omega) = \int_{\Omega} \mathbb{1}_{\{e(X(\omega))\geq\epsilon\}}\mathrm{d}\mathbb{P}(\omega) \tag{4.8}$$

Donc il nous faut l'estimer via des méthodes statistiques.

Nous pourrions tout simplement estimer cette espérance via une moyenne empirique (comme vu dans la section 4.2.2) sur l'ensemble de nos datasets, comme suit :

$$\mathbb{E}[Z(X)] \approx \frac{1}{\sum_{i=1}^{d} |D_i|} \sum_{i=1}^{d} \sum_{x\in D_i} \mathbb{1}_{\{e(x)\geq\epsilon\}} \tag{4.9}$$

où $|D_i|$ représente le cardinal de D_i. Cependant, cette estimation peut être fortement biaisée si les disparités entre les jeux de données ne sont pas prises en compte. En effet, les conditions de collecte, les emplacements géographiques, et les types de scénarios diffèrent souvent de manière significative d'un dataset à l'autre. Ces variations, influencées par des facteurs tels que les objectifs des applications, les méthodologies de collecte des données, et les contraintes de ressources, peuvent biaiser les résultats et limiter la représentativité des conclusions par rapport à la réalité modélisée. Par exemple, des tâches plus complexes nécessitent généralement un plus grand volume de données pour capturer la diversité des variations possibles. La méthodologie de collecte joue également un rôle crucial : certains jeux de données sont constitués sur une longue période ou couvrent plusieurs lieux géographiques, tandis que d'autres sont assemblés rapidement avec un objectif spécifique et restreint. De plus, la qualité des annotations et les conditions dans lesquelles elles sont produites influencent directement la taille et la représentativité des données. Des annotations détaillées, bien que précieuses, exigent davantage de ressources, ce qui limite souvent le volume de données annotées. Les disparités peuvent aussi provenir de la disponibilité des données sources, des contraintes légales et éthiques, ou encore des ressources financières et humaines mobilisées. Pour les données simulées, celles-ci pouvant théoriquement être générées en quantités illimitées, elles peuvent exacerber les disparités entre les datasets en raison de leur volume potentiellement disproportionné par rapport aux données réelles. Cela peut même conduire les données simulées à dominer l'estimation, risquant d'éclipser l'influence des données réelles et de fausser les conclusions. Si ces disparités ne sont pas prises en compte lors de l'estimation des garanties de performances, elles risquent d'introduire des biais significatifs dans les résultats. Des différences de taille, de diversité ou de conditions de collecte entre les datasets peuvent fausser l'évaluation en conduisant à une surévaluation ou une sous-évaluation des performances. Par exemple, un dataset plus volumineux ou plus diversifié pourrait dominer l'estimation, rendant les conclusions non représentatives des réalités opérationnelles que le système doit affronter. Il est donc essentiel de tenir compte de ces facteurs pour garantir une estimation équilibrée et pertinente des performances.

Pour tenir compte des disparités entre les jeux de données lors de l'estimation des garanties, nous utiliserons le théorème de l'espérance totale introduit dans la section 4.2.2, qui permet de calculer l'espérance d'une variable aléatoire en décomposant le problème en cas plus simples. Soit $A_1, \cdots, A_p$ une partition de l'espace probabiliste Ω, ce qui signifie que les événements A_i sont mutuellement exclusifs et que leur union couvre tout l'espace Ω. La loi de l'espérance totale indique que l'espérance d'une variable aléatoire W peut s'exprimer comme une somme pondérée des espérances conditionnelles par rapport à chaque événement A_i :

$$\mathbb{E}[W] = \sum_{i=1}^{p} \mathbb{E}[W|A_i] \cdot P(A_i) \tag{4.10}$$

Ce théorème nous permet de décomposer l'espérance des variables aléatoires, ce qui est particulièrement utile lorsque les données proviennent de sources hétérogènes.

Appliquons le théorème de l'espérance totale à notre problème d'estimation. Nous divisons l'espace Ω en p partitions disjointes, en veillant à ce que chaque jeu de données D_j soit contenu dans l'une de ces partitions. Ce partitionnement est conceptuel, car dans la pratique, les données sont souvent représentées dans des espaces de haute dimension, rendant une partition visuelle directe difficile. Nous définissons l'ensemble $\{A_i\}_{i=1}^{p}$ comme une partition disjointe de l'espace Ω, avec $p \geq d$, où p est le nombre total de partitions et d le nombre de jeux de données disponibles (cf. Figure 4.7).

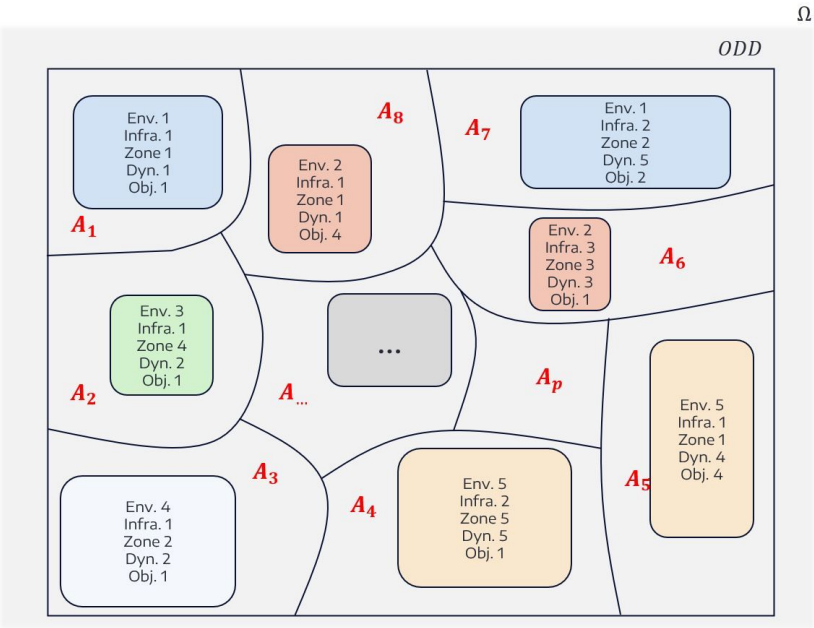


FIGURE 4.7 – Illustration d'un partitionnement des jeux de données.

En appliquant le théorème de l'espérance totale, nous obtenons :

$$\mathbb{E}[Z] = \sum_{i=1}^{p} \mathbb{E}[Z|A_i] \cdot P(A_i) \tag{4.11}$$

Cependant, comme $p \geq d$, seules certaines des partitions A_i contiennent effectivement des données issues d'un jeu D_j. Nous définissons le sous-ensemble $I \subseteq 1, \cdots, p$ tel que, pour tout $i \in I$, $D_i \subset A_i$. Le complément I^c représente les partitions vides. Par conséquent, nous ne pouvons estimer que les espérances conditionnelles $\mathbb{E}[Z|A_i]$ pour les A_i associés à un dataset D_i. Ainsi, l'espérance $\mathbb{E}[Z]$ peut s'écrire :

$$\mathbb{E}[Z] = \sum_{i \in I} \mathbb{E}[Z|A_i] \cdot P(A_i) + \gamma, \tag{4.12}$$

où γ représente l'erreur d'approximation due aux partitions non incluses dans I, définie par :

$$\gamma = \sum_{i \in I^c} \mathbb{E}[Z|A_i] \cdot P(A_i) \tag{4.13}$$

Si nous supposons que la probabilité des événements non inclus dans I est négligeable, c'est-à-dire :

$$\sum_{i \in I^c} P(A_i) \approx 0, \tag{4.14}$$

alors l'erreur d'approximation γ peut être considérée comme faible. Nous obtenons ainsi l'approximation suivante :

$$\mathbb{E}[Z] \approx \sum_{i \in I} \mathbb{E}[Z|A_i] \cdot P(A_i), \tag{4.15}$$

Il est cependant important de noter que γ dépend de la quantité de données disponibles. Plus nous collectons de jeux de données et augmentons la taille des datasets, plus γ diminue, améliorant ainsi la précision de l'estimation. Si nous choisissons $n = d$, c'est-à-dire un nombre de partitions égal au nombre de datasets, cette simplification revient à supposer que chaque dataset correspond exactement à une partition de Ω. Cependant, cela signifie que nous ignorons les contributions des partitions vides (non associées à un dataset), ce qui peut introduire un biais. Si γ (l'erreur d'approximation) est non négligeable, une telle simplification pourrait conduire à une sous-estimation ou une surestimation de $\mathbb{E}[Z]$, compromettant ainsi la fiabilité des garanties de performance. Il est donc crucial de considérer ces limitations et de chercher à compléter les données disponibles par l'acquisition de nouveaux datasets.

Ensuite, en posant $A_i = \{X \in D_i\}$, nous obtenons finalement :

$$\mathbb{E}[Z] \approx \sum_{i \in I} \mathbb{E}[Z | X \in D_i] \cdot P(X \in D_i) \tag{4.16}$$

Enfin, en appliquant l'estimation par moyenne empirique, la méthodologie pour obtenir des garanties de performance se résume en deux étapes d'estimation :

1. **Estimation intra-jeu de données** (*intra-dataset*) : Pour chaque jeu de données D_j, nous estimons $\mathbb{E}[Z | X \in D_j]$ à l'aide de la méthode bootstrap (cf. Algorithme 1) avec un intervalle de confiance de niveau $1 - \beta$:

$$\mathbb{E}[Z | X \in D_j] \approx \hat{\theta}_j = \frac{1}{|D_j|} \sum_{x \in D_j} z(x) = \frac{1}{|D_j|} \sum_{x \in D_j} \mathbb{1}_{\{e(x) \geq \epsilon\}} \tag{4.17}$$

   avec l'intervalle de confiance $\mathrm{IC}_\beta^j = [\hat{\theta}_j^-, \hat{\theta}_j^+]$.

2. **Estimation inter-jeu de données** (*inter-dataset*) : Nous combinons les estimations des différents datasets D_j à l'aide de la méthode bootstrap épistémique (cf. Algorithme 2) avec un intervalle de confiance de niveau $1 - \alpha$:

$$\mathbb{E}[Z] \approx \hat{\theta} = \sum_{j=1}^{d} \hat{\theta}_j \cdot P(X \in D_j) = \sum_{j=1}^{d} \hat{\theta}_j \cdot p_j \tag{4.18}$$

   avec l'intervalle de confiance $\mathrm{IC}_\alpha = \left[\hat{\theta}^-, \hat{\theta}^+\right]$.

Les probabilités $\left\{p_j = P\left(X \in D_j\right)\right\}_{j=1}^d$ représentent la probabilité qu'une donnée x, choisie aléatoirement, appartienne à l'un des ensembles de données $\{D_j\}_{j=1}^d$. En d'autres termes, ces probabilités quantifient la répartition relative des données entre les différents jeux de données. Cependant, ces probabilités sont en pratique inconnues, car elles dépendent de la distribution réelle des données, qui est souvent inaccessible. Afin de contourner cette limitation, ces probabilités peuvent être estimées expérimentalement ou définies en fonction de connaissances a priori. Dans ce cadre, nous considérons $\{p_j\}_{j=1}^d$ comme des hyperparamètres, paramétrable par un expert. Ces hyperparamètres, bien qu'idéalisés, doivent vérifier la propriété $\sum_{j=1}^d p_j = 1$. Pour simplifier le processus de définition des $\{p_j\}_{j=1}^d$, nous utilisons un système de pondération basé sur des poids relatifs $\{w_j\}_{j=1}^d$. Les poids $\{w_j\}_{j=1}^d$ reflètent la probabilité relative de chaque jeu de données par rapport aux autres, c'est-à-dire l'importance ou la fréquence relative de chaque dataset. Une fois ces poids définis, les probabilités associées sont calculées à l'aide de la formule suivante :

$$p_j = \frac{w_j}{\sum_{k=1}^d w_k}, \quad \forall j \in \{1, ..., d\} \tag{4.19}$$

Ce système de pondération permet une approche flexible pour l'estimation des probabilités $\{p_j\}_{j=1}^d$. Les poids $\{w_j\}_{j=1}^d$ peuvent être définis en s'appuyant sur des informations a priori ou des connaissances fournies par des experts du domaine, tels que des spécialistes de la sécurité ou des ingénieurs en modélisation de scénarios. Ces experts peuvent évaluer, sur la base d'expériences ou de données empiriques, la probabilité relative que chaque jeu de données corresponde à des cas réels spécifiques. Ainsi, ce système combine la simplicité d'un modèle de pondération avec la flexibilité nécessaire pour intégrer des connaissances spécifiques au domaine, permettant d'estimer de manière pratique et cohérente les probabilités $\{p_j\}_{j=1}^d$ dans des contextes complexes.

La méthodologie proposée repose sur deux étapes d'estimation, chacune ayant un impact potentiel sur l'estimation finale. Dans la première étape, si la quantité de données dans l'un des jeux de données est insuffisante, l'estimation intra-dataset peut manquer de précision. Cela se reflète directement dans les intervalles de confiance associés, de niveau $1-\beta$, notés $\text{IC}_\beta = [\hat{\theta}^-, \hat{\theta}^+]$. Une imprécision similaire peut également affecter la deuxième étape d'estimation inter-dataset, amplifiant ainsi les incertitudes globales.

Lorsqu'un nouveau jeu de données est ajouté, la méthodologie reste flexible et extensible. Il suffit d'appliquer la première étape d'estimation à ce nouveau dataset pour calculer sa valeur $\hat{\theta}_{d+1}$ et l'intervalle de confiance correspondant. Ensuite, la deuxième étape d'estimation est mise à jour en prenant en compte l'ensemble élargi $\left\{\hat{\theta}_1, \cdots, \hat{\theta}_{d+1}\right\}$, avec les intervalles de confiance associés.

Enfin, si pour le système S et son application, il est plus pertinent d'évaluer une probabilité de bon fonctionnement plutôt qu'une probabilité de défaillance, il suffit de considérer la quantité complémentaire $1-\mathbb{P}(e(x) \geq \epsilon)$. Cela revient à estimer $1 - \mathbb{E}[Z]$, ce qui peut être interprété comme la probabilité que l'erreur du système reste inférieure au seuil ϵ. Cette adaptation conserve la cohérence de la méthodologie tout en répondant aux besoins spécifiques de l'analyse.

Cadre avec classes

Dans le cas où la fonction F d'un système S traite des classes (par exemple, dans des tâches comme la classification, la détection d'objets ou la segmentation sémantique), il est particulièrement pertinent d'évaluer les performances du système à la fois globalement et individuellement pour chaque classe. Cette analyse détaillée permet non seulement d'obtenir une vision d'ensemble des performances, mais également de mieux identifier les classes pour lesquelles les résultats sont insuffisants. Une telle granularité offre une compréhension approfondie des forces et des faiblesses du système, permettant de guider efficacement les améliorations nécessaires.

Considérons une variable aléatoire X, définie sur un espace probabilisé $(\Omega, \mathcal{F}, P)$, représentant une donnée, ainsi qu'un seuil $\epsilon > 0$ représentant une limite d'erreur tolérable. Nous introduisons un ensemble de l classes $C = \{c_1, \cdots, c_l\}$ et définissons sur Ω l'évènement $\{C \in X\}$, correspondant à l'ensemble des données X dans lesquelles les classes de C interviennent dans les sorties du système S. Dans ce cadre, la fonction d'erreur $e(x)$ est spécifiquement définie pour évaluer les erreurs relatives uniquement aux classes de C. Autrement dit, $e(x)$ ne considère que les contributions des classes de C dans les performances du système. Cela implique que sur l'événement $\{C \notin X\}$, la fonction d'erreur est nulle ($e(x) = 0$). Par exemple, dans une application de type classification, cet évènement $\{C \in X\}$ inclut les données X pour lesquelles :

- une classe de C est correctement prédite (*True Positive*),
- une classe de C est faussement prédite (*False Positive*),
- ou une classe de C aurait dû être prédite mais ne l'est pas (*False Negative*).

Autrement dit, $\{C \in X\}$ regroupe toutes les données X pour lesquelles les classes de C sont impliquées, qu'elles soient bien ou mal traitées par le système S.

Pour quantifier les performances globales et spécifiques à chaque classe, nous définissons les variables aléatoires suivantes sur Ω :

$$Z(X) = \begin{cases} 0 \text{ si } e(X) < \epsilon \\ 1 \text{ si } e(X) \geq \epsilon \end{cases} = \mathbb{1}_{\{e(X) \geq \epsilon\}} \tag{4.20}$$

$$V(X) = \begin{cases} 0 \text{ si } C \notin X \\ 1 \text{ si } C \in X \end{cases} = \mathbb{1}_{\{C \in X\}} \tag{4.21}$$

où $Z(X)$ indique si l'erreur $e(X)$ dépasse le seuil ϵ, et $V(X)$ signale si au moins une classe de C intervient dans la données X via le système S. Les espérances de ces variables aléatoires permettent de calculer les probabilités :

$$\mathbb{E}[Z] = P(e(X) \geq \epsilon) \tag{4.22}$$

$$\mathbb{E}[V] = P(C \in X) \tag{4.23}$$

Nous cherchons à décomposer la probabilité $P(e(X) \geq \epsilon)$ pour mieux comprendre les erreurs spécifiques aux classes C. En utilisant la propriété que $\{e(X) \geq \epsilon\} \subset \{C \in X\}$ et que $\{e(X) \geq \epsilon\} \bigcap \{C \notin X\} = \emptyset$, on peut écrire :

$$P(e(X) \geq \epsilon) = P\left(\{e(X) \geq \epsilon\} \bigcap \{C \in X\}\right) \tag{4.24}$$

ce qui nous conduit à la relation :

$$P(e(X) \geq \epsilon) = P(e(X) \geq \epsilon | C \in X) \cdot P(C \in X) \tag{4.25}$$

Comparé au cadre générique, où l'on estime la probabilité globale que l'erreur $e(X)$ dépasse un certain seuil ϵ, soit $P(e(X) \geq \epsilon)$, ici nous introduisons deux grandeurs supplémentaires qui permettent une analyse plus fine. La probabilité conditionnelle $P(e(X) \geq \epsilon | C \in X)$ exprime la probabilité que l'erreur dépasse le seuil ϵ, sur l'ensemble des données X où les classes C interviennent. Cela reflète spécifiquement les performances du système sur les classes de C, indépendamment des autres classes potentielles ou des données globales. Une valeur élevée pour cette probabilité indique que le système a des difficultés particulières à traiter ces classes. La probabilité $P(C \in X)$, quant à elle, mesure la fréquence d'apparition des classes C dans la population globale. Une classe fréquente aura un impact plus significatif sur la probabilité globale d'erreur, tandis qu'une classe rare pourrait avoir une contribution plus limitée, même si son taux d'erreur est élevé.

En combinant cette relation avec les définitions des variables aléatoires Z et V, nous établissons une expression reliant l'espérance de Z aux probabilités conditionnelles et marginales du système étudié :

$$\mathbb{E}[Z] = \mathbb{E}[Z | C \in X] \cdot \mathbb{E}[V] \tag{4.26}$$

La suite du processus s'aligne sur le cadre générique présenté précédemment. En introduisant une partition $\left\{A_i = \{X \in D_i\}\right\}_{i=1}^{p}$ de l'espace Ω, et en appliquant le théorème de l'espérance totale, on obtient une approximation de chacune des espérances mentionnées ci-dessus (similaire à celle de l'équation 4.16) :

$$\mathbb{E}[Z] \approx \sum_{j=1}^{d} \mathbb{E}[Z | X \in D_j] \cdot P(X \in D_j) \tag{4.27}$$

$$\mathbb{E}[Z | C \in X] \approx \sum_{j=1}^{d} \mathbb{E}[Z | C \in X, X \in D_j] \cdot P(X \in D_j) \tag{4.28}$$

$$\mathbb{E}[V] \approx \sum_{j=1}^{d} \mathbb{E}[V | X \in D_j] \cdot P(X \in D_j) \tag{4.29}$$

Enfin, comme dans le cadre générique, nous estimons à nouveau ces trois grandeurs en appliquant, pour chacune d'elles, les étapes d'estimation intra-jeu et inter-jeux de données :

1. **Estimation intra-jeu de données** (*intra-dataset*) par bootstrap (cf. Algorithme 1) avec un intervalle de confiance de niveau $1 - \beta$:

$$\mathbb{E}[Z | X \in D_j] \approx \hat{\theta}_j = \frac{1}{|D_j|} \sum_{x \in D_j} \mathbb{1}_{\{e(x) \geq \epsilon\}} \tag{4.30}$$

$$\mathbb{E}[Z | C \in X, X \in D_j] \approx \hat{\theta}_j^{e,C} = \frac{1}{|D_j|} \sum_{x \in D_j} \mathbb{1}_{\{C \in X\} \bigcap \{e(x) \geq \epsilon\}} \tag{4.31}$$

$$\mathbb{E}[V | X \in D_j] \approx \hat{\theta}_j^{C} = \frac{1}{|D_j|} \sum_{x \in D_j} \mathbb{1}_{\{C \in X\}} \tag{4.32}$$

2. **Estimation inter-jeu de données** (*inter-dataset*) par bootstrap épistémique (cf. Algorithme 2) avec un intervalle de confiance de niveau $1 - \alpha$:

$$\mathbb{E}[Z(X)] \approx \hat{\theta} = \sum_{j=1}^{d} \hat{\theta}_j \cdot p_j \tag{4.33}$$

$$\mathbb{E}[Z(X) | C \in X] \approx \hat{\theta}^{e,C} = \sum_{j=1}^{d} \hat{\theta}_j^{e,C} \cdot p_j \tag{4.34}$$

$$\mathbb{E}[V(X)] \approx \hat{\theta}^{C} = \sum_{j=1}^{d} \hat{\theta}_j^{C} \cdot p_j \tag{4.35}$$

Par la suite, nous définissons les hyperparamètres $\{p_j\}_{j=1}^{d}$ qui sont partagés entre les trois grandeurs mentionnées ci-dessus de la même manière que pour le cadre générique (cf. Equation 4.19).

Dans le chapitre 6, nous appliquerons cette méthode à l'exemple du Field Monitoring, un système comprenant un module de compression d'image et un autre de détection d'objets. Ensuite, nous examinerons comment généraliser cette méthode à d'autres types de systèmes et à d'autres applications dans le chapitre 7.

4.3.3 Génération de données critiques aux modules d'IA

Nous avons déjà souligné l'importance de générer des données synthétiques dans la section 4.3.1, notamment pour combler les lacunes de nos ensembles de données par rapport à l'ODD. Ces lacunes concernent principalement des scénarios rares, difficiles ou coûteux à capturer via des capteurs. Par exemple, dans le cadre des véhicules autonomes équipés de caméras embarquées, ces scénarios incluent des scènes d'accidents de la route, des conditions météorologiques extrêmes comme des tempêtes de neige ou de fortes pluies, ou encore des interactions atypiques entre usagers de la route. Pour pallier ces limitations, il est essentiel de synthétiser ces données en utilisant des simulateurs capables de produire des représentations fidèles et réalistes des scénarios cibles. Cependant, avant d'intégrer ces données dans nos ensembles d'apprentissage ou de validation, il est crucial de vérifier deux aspects : d'une part, que le système étudié interprète correctement ces données simulées, et d'autre part, que ces données soient perçues comme suffisamment réalistes par les modules d'IA du système.

En complément, il est également nécessaire d'acquérir des données spécifiquement liées aux modules d'IA du système, c'est-à-dire des données susceptibles de mettre en évidence leurs faiblesses ou leurs limites. Une question clé consiste alors à identifier les images ou scénarios qui posent problème à ces modules. Contrairement aux attaques adversariales qui cherchent délibérément à exploiter des vulnérabilités spécifiques et mathématiques, l'objectif ici est d'explorer l'espace des données possibles pour détecter des cas critiques susceptibles de provoquer des erreurs significatives. Ces cas critiques, qu'ils soient liés à des conditions extrêmes (comme des occlusions ou des angles de vue atypiques) ou à des configurations particulières des objets dans l'environnement, peuvent ensuite être utilisés pour affiner le système étudié, soit par réentraînement des modèles pour mieux les gérer, soit par l'ajout de mécanismes de sécurité pour atténuer leurs impacts. Une méthode prometteuse pour générer ces données critiques consiste à utiliser des modèles de deep learning génératifs. Ces modèles permettent de produire des données synthétiques issues de la même distribution que les données d'entraînement, mais peuvent être orientés pour révéler des scénarios problématiques. Il existe une grande variété de modèles génératifs adaptés à différents types de données et applications, chacun optimisé pour des objectifs spécifiques.

Pour générer différents types de données, plusieurs modèles sont adaptés selon les besoins. Pour le texte, les modèles autoregressifs comme GPT [Brown et al., 2020], basés sur des Transformers, excellent en prédisant séquentiellement les mots et sont utilisés pour la génération de texte, la traduction ou les systèmes de dialogue. Pour l'audio, WaveNet [van den Oord et al., 2016] ou DiffWave [Kong et al., 2020] permettent une synthèse vocale ou musicale réaliste, tandis que MelGAN [Kumar et al., 2019] ou WaveGAN [Donahue et al., 2018] génèrent des spectrogrammes ou signaux bruts. Pour les graphes, les Graph Neural Networks (GNNs) ou Graph Autoencoders [Hasanzadeh et al., 2019] produisent des structures adaptées à des applications comme la modélisation de réseaux sociaux ou la découverte de molécules. Pour les vidéos, des modèles tels que les 3D-CNNs, TGAN2 [Saito et al., 2020], ou MoCoGAN [Tulyakov et al., 2018] génèrent des séquences cohérentes, utiles pour la génération vidéo ou l'interpolation. Enfin, pour les données tabulaires ou temporelles, des modèles comme TabTransformer [Huang et al., 2020] ou TimeGPT [Garza and Mergenthaler-Canseco, 2023] s'appliquent à des domaines variés comme la finance ou l'IoT.

Pour simplifier, concentrons-nous sur le cas de la génération d'images. Les données visuelles, comme les images, ne sont pas uniformément réparties dans l'espace multidimensionnel où elles résident. Elles occupent une petite sous-région, souvent modélisée comme une variété : un sous-espace de faible dimension plongé dans un espace à haute dimension. Cette hypothèse, introduite pour expliquer la structure des données naturelles, repose sur l'idée que les données réelles sont soumises à des contraintes intrinsèques, imposées par des lois physiques, biologiques ou esthétiques. Par exemple, une image de visage humain respecte des relations géométriques et texturales précises qui la distinguent des configurations aléatoires de pixels. La variété des données visuelles capture donc ces régularités et contraintes, formant une sous-structure dans l'espace des images. Cette hypothèse guide la plupart des modèles génératifs en supposant que les données réelles sont concentrées autour de cette

variété de faible dimension. Dans ce cadre, la tâche de génération d'images peut être vue comme l'apprentissage et l'échantillonnage d'une distribution complexe $P_{\text{data}}(x)$, où x représente une image. Dans la section 3.4, nous avons exploré plusieurs méthodes de génération d'images. Voici un aperçu des principales approches, chacune exploitant différemment l'hypothèse de la variété.

Par exemple, les VAEs sont des modèles probabilistes qui apprennent une représentation latente z dans un espace de faible dimension tout en maximisant une borne inférieure de la vraisemblance des données. Ils consistent en deux parties : un encodeur projette une image x dans un espace latent $z \sim P(z|x)$ et un décodeur reconstruit x à partir de z. Pendant l'entraînement, une régularisation impose à z de suivre une distribution simple, comme une gaussienne. Cette contrainte permet au modèle d'apprendre une représentation latente qui capture la variété des données réelles, tout en facilitant l'échantillonnage.

De même, les GANs, reposent sur une compétition entre deux réseaux neuronaux : Un générateur G transforme un vecteur latent $z \sim P_z$ (tiré d'une distribution simple, comme une gaussienne) en une image $x = G(z)$, puis Un discriminateur D distingue les images réelles $x \sim P_{\text{data}}$ des images synthétiques $x \sim P_G$. Les GANs apprennent implicitement à représenter la variété des données réelles en poussant G à produire des images indistinguables des données réelles. Cependant, ils peuvent souffrir de problèmes comme le mode collapse, où certaines régions de la variété ne sont pas bien représentées.

Enfin, les modèles de diffusion offrent une approche plus récente. Ils modélisent un processus en deux étapes : une image x_0 est progressivement bruitée jusqu'à une distribution simple $x_T \sim \mathcal{N}(0, I)$. Pendant la génération, un processus inverse débruite x_T pour revenir à x_0. Ces modèles capturent la distribution des données en suivant une trajectoire qui traverse la variété des données réelles, offrant une couverture robuste et évitant le mode collapse.

Dans le contexte de l'évaluation et de la garantie de performances du système S, la génération de données synthétiques vise à créer des scénarios susceptibles de mettre en défaut le système. Contrairement aux attaques adversariales, qui modifient directement des données existantes, cette approche oriente le processus de génération lui-même. Le processus de génération repose sur des modèles génératifs comme les GANs, les VAEs ou les modèles de diffusion. Ces modèles prennent en entrée un ensemble d'informations pouvant inclure du bruit aléatoire, des conditions spécifiques (texte, annotations, etc.) ou des contraintes supplémentaires pour s'assurer que les données générées appartiennent à la distribution réelle. À partir de ces entrées, le générateur produit une donnée synthétique $\tilde{x}$, qui est ensuite passée dans le système S. Le système calcule une erreur via une fonction d'erreur L. Similaire aux méthodes décrit dans la section 3.4.6, l'idée clé est d'exploiter le gradient de cette erreur pour orienter le processus de génération. Cela permet d'augmenter la probabilité que les nouvelles données synthétiques induisent des erreurs dans le système S, tout en restant proches de la distribution des données réelles (cf. Figures 4.8 et 4.9).

Cependant, il y a des conditions pour la fonction d'erreur L afin de permettre une génération orientée par le gradient. Tout d'abord, L doit être différentiable par rapport aux entrées du modèle génératif, afin de calculer le gradient nécessaire pour guider la génération. Cela implique que la fonction L doit être continue et avoir des dérivées partielles définies dans le domaine des données considérées. De plus, L doit être conçue de manière à refléter l'objectif spécifique de la génération. Par exemple, si l'objectif est de maximiser l'erreur du système étudié, L doit être proportionnelle à une métrique d'évaluation pertinente (par exemple, une perte de détection ou une erreur de classification). Enfin, il est essentiel que L soit cohérente avec les contraintes opérationnelles et les objectifs du système, de manière à générer des données représentatives et exploitables pour améliorer les performances ou tester les limites du système.

Cette approche présente plusieurs avantages significatifs. Tout d'abord, elle respecte la distribution réelle des données : les échantillons générés restent proches de la diversité observée dans les données réelles, garantissant ainsi leur réalisme. Ensuite, grâce à l'orientation par le gradient, cette méthode permet une exploration ciblée de l'espace des données, concentrant les efforts sur les régions susceptibles de révéler des faiblesses du système. Enfin, elle se distingue par son adaptabilité, puisqu'elle peut être appliquée à une grande variété de types de données et de modèles, ce qui en fait une solution polyvalente pour évaluer les performances des systèmes d'IA. condition

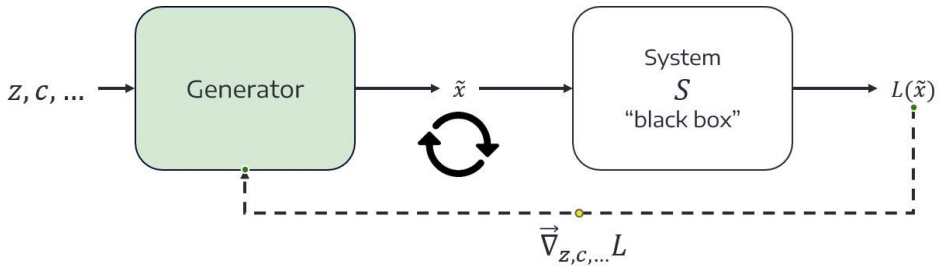


FIGURE 4.8 – Schéma du processus itératif de génération de données critiques en orientant celles-ci selon la direction du gradient.

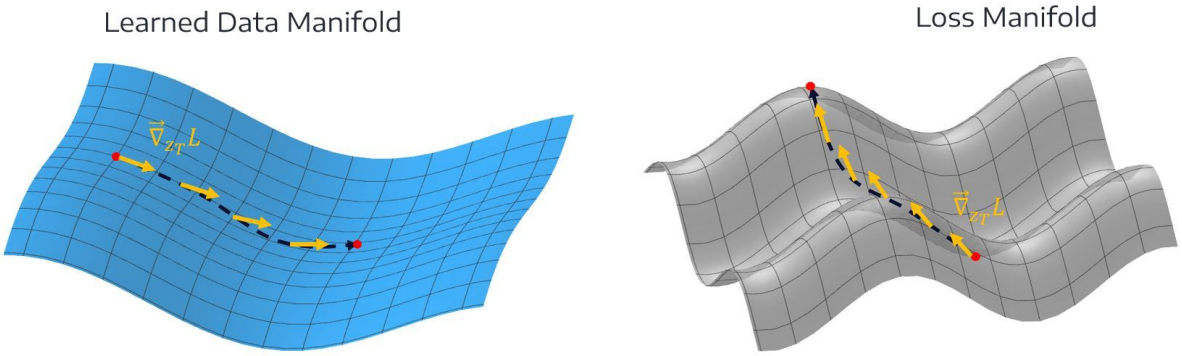


FIGURE 4.9 – Illustration du processus de génération : contraint à la variété des données, la génération produit des images maximisant une fonction d'erreur $\mathcal{L}$ grâce à l'orientation par gradient.

En pratique, les données réelles ne forment pas toujours une variété parfaitement lisse. Elles peuvent inclure des discontinuités ou des singularités, par exemple lorsque des transitions abruptes apparaissent entre différentes catégories de données. De plus, certaines régions de l'espace des données peuvent être peu densément peuplées, rendant leur représentation plus incertaine. Ces irrégularités peuvent rendre difficile l'approximation de la distribution réelle et limiter la capacité des modèles génératifs à capturer fidèlement ces subtilités. Un autre défi majeur réside dans la couverture limitée des données d'entraînement. Les échantillons collectés pour entraîner un modèle sont souvent incomplets et ne couvrent qu'une petite portion de la variété réelle. Par conséquent, les modèles génératifs apprennent une représentation biaisée ou partielle, ce qui peut nuire à leur capacité à généraliser sur des scénarios ou des conditions moins représentés. Un autre défi majeur réside dans la couverture limitée des données d'entraînement. Les échantillons collectés pour entraîner un modèle sont souvent incomplets et ne couvrent qu'une petite portion de la variété réelle. Par conséquent, les modèles génératifs apprennent une représentation biaisée ou partielle, ce qui peut nuire à leur capacité à généraliser sur des scénarios ou des conditions moins représentés. Par ailleurs, pour certains types de données, l'idée même de variété peut être inappropriée. Par exemple, dans des domaines où les données présentent une forte complexité structurelle ou une très grande variabilité (comme les graphes complexes ou les signaux audio non stationnaires), l'hypothèse de variété lisse peut ne pas convenir. Malgré ces limitations, cette démarche demeure une étape cruciale dans la méthodologie d'obtention de garanties de performances pour des systèmes complexes intégrant des modules d'intelligence artificielle. En offrant un cadre pour modéliser et échantillonner les données, elle permet d'orienter le développement et l'évaluation des modèles. Cependant, il est essentiel de considérer ses limites et de compléter cette approche par des stratégies adaptées, telles que l'incorporation de méthodes capables de gérer les discontinuités ou la génération de données synthétiques pour explorer les régions sous-représentées de l'espace des données.

Dans la section 6.6, nous mettrons en œuvre cette méthode dans le cadre du *Field Monitoring*, un système intégrant à la fois un module de compression d'image et un module de détection d'objets. Nous exploiterons un modèle de diffusion pour générer des images synthétiques, en orientant le processus de génération dans la direction du gradient de l'erreur. Ce cas d'application illustrera concrètement comment cette méthodologie peut être utilisée pour évaluer des systèmes complexes intégrant des modules d'intelligence artificielle.

4.3.4 Interprétation de la méthode

Avant d'appliquer la méthode à notre chaîne compression-détection dans le chapitre suivant, il est essentiel de présenter les fondements de l'approche et de souligner ses objectifs ainsi que ses limites.

Déterminer la probabilité d'erreur d'un système $P(e(X) \geq \epsilon)$ nécessite de connaître la distribution réelle des données $P(X)$. Cette distribution représente l'ensemble des situations possibles que le système peut rencontrer dans son domaine opérationnel (ODD). Cependant, dans la pratique, cette distribution réelle est généralement inconnue, car elle dépend de nombreux facteurs complexes et souvent imprévisibles, tels que la variabilité des conditions environnementales, la diversité des scénarios, ou encore la rareté de certains cas critiques (cf. Figure 4.10).

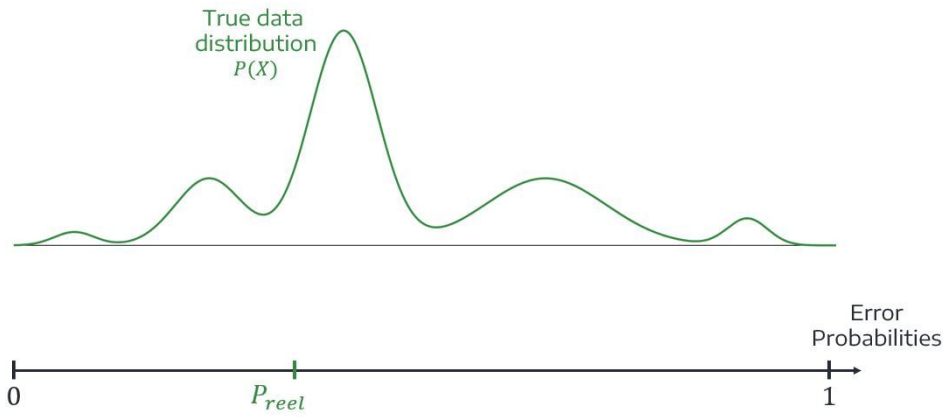


FIGURE 4.10 – Illustration de la distribution réelle inconnue des données X et de la vraie probabilité d'erreur nommée ici P_{reel}.

Pour pallier cette limitation, nous collectons des échantillons de données provenant de différentes sources (datasets réels, simulations, campagnes de mesure, etc.). Ces échantillons ne sont pas nécessairement représentatifs de la distribution réelle $P(X)$. En effet, ils peuvent refléter des biais introduits par les méthodes de collecte, les emplacements géographiques, ou les conditions spécifiques dans lesquelles les données ont été acquises. Chaque jeu de données D_j capture ainsi une vision locale et partielle de la distribution réelle. S'appuyer uniquement sur un jeu de données unique pour estimer $P(e(X) \geq \epsilon)$ pourrait conduire à une sur-approximation ou une sous-approximation de cette probabilité, rendant l'évaluation biaisée (cf. Figure 4.11).

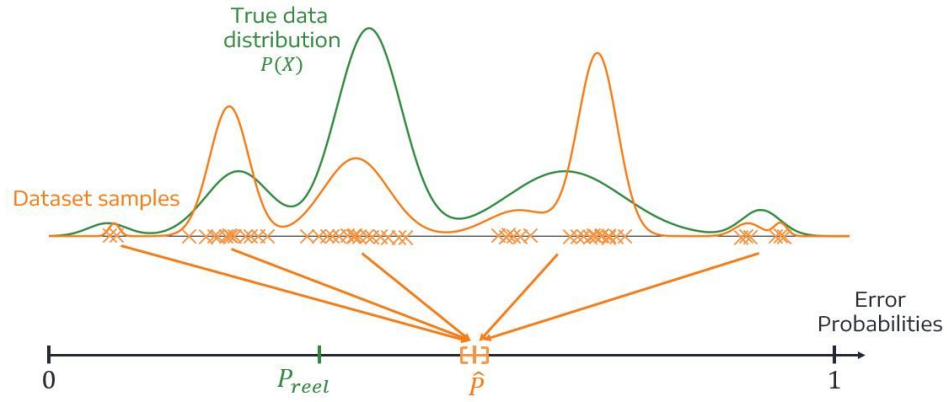


FIGURE 4.11 – Illustration de l'estimation biaisée de la probabilité d'erreur $\hat{P}$ à partir de la distribution de données échantillonnées.

La méthode proposée repose sur une approche structurée pour approximer la distribution réelle $P(X)$ et, par conséquent, la probabilité d'erreur réelle $P(e(X) \geq \epsilon)$. Elle s'appuie sur deux principes clés (cf. Figure 4.12) :

1. Estimation locale sur chaque jeu de données : Chaque jeu de données D_j est traité comme une sous-partie de l'espace global Ω. La probabilité conditionnelle $P(e(X) \geq \epsilon | X \in D_j)$ est estimée localement

sur chaque dataset à l'aide de la moyenne empirique. Cette étape permet de capturer les particularités de chaque sous-distribution représentée par D_j, tout en prenant en compte les éventuelles variations spécifiques aux conditions de collecte.

2. Pondération des estimations locales : Comme jeux de données ne sont pas toujours représentatifs à parts égales de la distribution réelle $P(X)$, une pondération basée sur des informations a priori est appliquée. Ces poids relatifs, p_j, reflètent l'importance ou la fréquence relative de chaque jeu de données dans l'ensemble des situations modélisées par la distribution réelle $P(X)$. Les poids sont définis à l'aide d'une expertise ou d'une analyse des scénarios représentés dans chaque dataset..

Grâce à cette combinaison d'estimations locales et de pondération globale, nous obtenons une approximation de la probabilité d'erreur réelle sous la forme :

$$P(e(X) \geq \epsilon) \approx \sum_{j=1}^{d} P(e(X) \geq \epsilon | X \in D_j) \cdot p_j \tag{4.36}$$

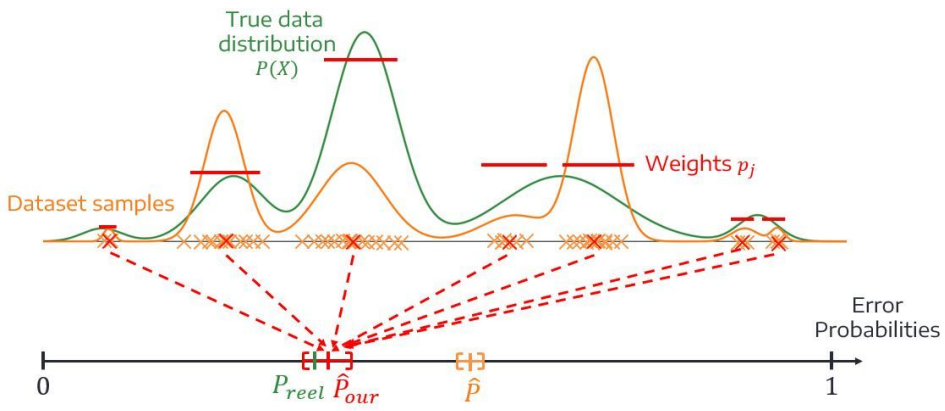


FIGURE 4.12 – Illustration de l'estimation de la probabilité d'erreur corrigée $\hat{P}_{our}$ par un système de pondération.

Cette méthode ne cherche pas à reconstituer exactement la distribution réelle $P(X)$, mais plutôt à s'en approcher de manière pragmatique et rigoureuse. Elle tire parti des informations disponibles tout en prenant en compte les disparités entre les datasets. Ainsi, elle permet :

- Une meilleure représentation de la diversité des cas potentiels, en particulier lorsque les données réelles sont rares ou difficilement accessibles.
- Une gestion flexible des biais liés à la collecte des données, grâce à la pondération p_j qui équilibre l'influence des jeux de données en fonction de leur pertinence vis-à-vis de l'application.

De plus, l'utilisation de méthodes statistiques telles que le bootstrap permet de fournir des intervalles de confiance, reflétant ainsi le niveau de précision des estimations tout en intégrant les incertitudes inhérentes aux données.

Bien que puissante, cette méthode reste une approximation. Des biais peuvent persister en raison de la qualité et de la quantité des données collectées, de la non-représentativité de certains datasets ou encore d'hypothèses a priori incorrectes dans la définition des poids p_j. Cependant, la méthodologie est évolutive : en intégrant de nouveaux jeux de données ou en révisant les pondérations, il est possible de réduire les incertitudes et de se rapprocher progressivement de la distribution réelle. En particulier, l'ajout de données critiques ou rares (générées par des simulations ou observées dans des environnements spécifiques) permet de couvrir des cas limites et d'améliorer la robustesse des estimations.

En conclusion, bien que la distribution réelle $P(X)$ reste inconnue, cette méthode offre une stratégie structurée et pragmatique pour l'approximer. En combinant des observations locales avec des connaissances a priori, elle permet d'obtenir une estimation robuste de la probabilité d'erreur réelle tout en tenant compte des disparités entre datasets. Sa flexibilité et son évolutivité en font une solution adaptable à divers systèmes, tels que la classification, la détection d'objets ou encore la segmentation. Cette approche permet également de gérer les cas critiques et les scénarios rares, grâce à l'intégration conjointe de données mesurées et simulées. Enfin, l'ajout de nouveaux datasets ou la révision continue des poids w_j (c'est-à-dire des probabilités relatives p_j) offre la possibilité d'affiner en permanence les estimations et de garantir des performances fiables dans des conditions

variées. Cette méthode, bien que générique, sera appliquée spécifiquement à la chaîne compression-détection dans le chapitre suivant pour illustrer sa pertinence et son efficacité.

4.3.5 Comment rajouter des données ?

Supposons que nous fassions l'acquisition de nouvelles données. Une question cruciale se pose : doit-on ajouter ces données à l'un des datasets existants ? Ou créer un nouveau dataset ? Ce choix est déterminant pour garantir la validité et la précision des estimations obtenues dans le cadre de la méthodologie proposée. La décision entre enrichir un dataset existant D_d ou en créer un nouveau D_{d+1} repose sur plusieurs facteurs que nous examinerons rigoureusement :

1. Cohérence des attributs des données :
   - Si les nouvelles données partagent les mêmes attributs que D_j (par exemple, collectées dans des conditions similaires telles que le même environnement, type de scénarios ou contraintes expérimentales), elles devraient être ajoutées à D_j. Cela enrichit l'échantillon, augmentant ainsi la précision de l'estimation intra-dataset $\hat{\theta}_j \approx \mathbb{E}[Z|Z \in D_j]$ tout en réduisant l'incertitude liée aux intervalles de confiance IC_β^j.
   - Si les nouvelles données diffèrent significativement, c'est-à-dire qu'elles présentent des attributs distincts (nouveaux environnements, conditions, ou scénarios), elles ne doivent pas être ajoutées à D_j pour préserver l'homogénéité et la spécificité des données dans D_j. Dans ce cas, il est préférable de constituer un nouveau dataset D_{d+1}, garantissant une séparation claire et une estimation plus précise pour chaque partition.
2. Diversité des scénarios couverts : L'ajout d'un nouveau dataset $(d+1)$ permet d'améliorer la couverture des partitions dans l'espace Ω, ce qui réduit l'erreur d'approximation γ, liée aux partitions non couvertes (I^c). Si les nouvelles données introduisent des scénarios ou des distributions absents des datasets actuels, un nouveau dataset doit être créé pour assurer une meilleure couverture de l'espace des scénarios possibles. Cette approche permet aussi de mieux répondre aux exigences d'exhaustivité des tests et d'assurer une évaluation plus complète des performances du système.
3. Impact sur les pondérations $\{p_j\}$:
   - Ajouter des données à un dataset existant augmente sa représentativité dans la population globale. La pondération associée $p_j = P(X \in D_j)$ devra alors être réajustée pour refléter ce changement, garantissant une pondération équilibrée entre les datasets.
   - Créer un nouveau dataset introduit une nouvelle probabilité p_{d+1}, qui doit être définie. Cela peut être plus complexe si les informations a priori sur la fréquence des nouveaux scénarios sont incertaines. Toutefois, cette approche offre une meilleure représentation des scénarios variés, surtout si les nouvelles données diffèrent significativement des précédentes.

En conclusion, il est recommandé d'ajouter les nouvelles données aux datasets existants lorsque leurs attributs sont cohérents avec ceux des données déjà présentes dans D_j, car cela améliore directement les estimations intra-dataset. En revanche, si les nouvelles données présentent des attributs ou scénarios distincts, un nouveau dataset doit être créé, car cela garantit la préservation de l'homogénéité des datasets et une couverture plus diversifiée des scénarios. Si les nouvelles données sont ambiguës, partageant certains attributs tout en introduisant de nouvelles dimensions, il est préférable de consulter des experts du domaine (par exemple, en simulation, collecte de données, ou sécurité) pour déterminer l'approche optimale. En effet, la décision doit toujours être guidée par l'objectif de maintenir l'homogénéité au sein des datasets tout en maximisant la diversité des scénarios couverts dans l'espace Ω. Cette démarche impacte directement la précision des estimations intra-dataset et inter-dataset, ainsi que l'interprétabilité et la robustesse des résultats obtenus.

4.4 Conclusion et perspectives

La méthodologie présentée dans cette section propose une approche rigoureuse et structurée pour estimer les garanties de performance d'un système, en particulier lorsque la distribution réelle des données $P(X)$ est inconnue. En s'appuyant sur une décomposition en estimations locales, calculées sur des jeux de données spécifiques, et sur une pondération globale basée sur des informations a priori fournies par une expertise métier, cette méthode permet d'obtenir une approximation robuste et pertinente de la probabilité d'erreur $P(e(X) \geq \epsilon)$. Elle offre plusieurs avantages clés :

- Flexibilité : La méthode est générique et adaptable à une large gamme de systèmes, qu'il s'agisse de tâches de classification, de détection ou de segmentation.
- Gestion des cas critiques : L'intégration de données réelles et simulées permet de couvrir les cas limites et rares de l'ODD, essentiels pour évaluer les performances d'un système.
- Évolutivité : La méthodologie peut être continuellement enrichie par l'ajout de nouveaux datasets ou la révision des pondérations p_j, améliorant ainsi progressivement la précision des estimations. Concernant l'ajout de nouvelles données, pour de futures travaux, on pourrait envisager une approche en clustering les données avant de les assigner à un dataset. Cela permettrait de grouper les données par similarité statistique ou selon des attributs prédéfinis, garantissant une meilleure cohérence intra-dataset.

En complément, la génération de données synthétiques par modèles génératifs dans le sens du gradient constitue un outil additionnel. Cette approche permet de cibler explicitement les scénarios les plus susceptibles de mettre en défaut le système, en générant des exemples synthétiques orientés pour maximiser une fonction d'erreur L. En opérant dans un espace contraint, supposé correspondre à une variété des données réelles, cette méthode garantit que les données générées restent plausibles tout en explorant des zones critiques du domaine opérationnel. Cette stratégie, inspirée des techniques de génération adversariale, enrichit la diversité des cas testés et contribue à renforcer les garanties de performance du système.

Toutefois, cette approche reste une approximation, et sa qualité dépend fortement de la représentativité et de la diversité des données disponibles. Les biais liés à la collecte des données ou à la définition des pondérations doivent être surveillés et corrigés pour garantir la fiabilité des conclusions. Cette méthodologie constitue une base solide pour l'évaluation des systèmes autonomes dans leurs domaines opérationnels, offrant des garanties quantitatives sur leurs performances tout en prenant en compte les disparités inhérentes aux données disponibles.

Dans le chapitre suivant, nous appliquerons cette méthodologie à un système composé de deux d'IA : un module de compression d'image, permettant de réduire la taille des données transmises ou stockées, et un module de détection d'objets basé sur l'intelligence artificielle, chargé d'identifier et de localiser des objets d'intérêt dans les images compressées. Cette chaîne compression-détection représente une problématique clé dans de nombreuses applications, comme la surveillance, les véhicules autonomes ou les systèmes embarqués. Nous commencerons par évaluer la probabilité d'erreur globale du système en appliquant la méthode présentée, en tenant compte des spécificités des modules individuels et de leur interaction. Par la suite, nous explorerons différentes stratégies pour optimiser cette chaîne, notamment en ajustant les paramètres de compression pour améliorer les performances de détection tout en maintenant une efficacité optimale. Ce chapitre nous permettra de démontrer la pertinence et la flexibilité de la méthodologie dans un contexte pratique, tout en identifiant des pistes concrètes d'amélioration pour des systèmes complexes et modulaires.

Chapitre 5

Application industrielle : définition du système, conception et analyse des données multi-sources

Contents

5.1 Introduction . . . 140
5.2 Définition du système et de l'ODD . . . 141
5.3 Acquisition et analyse de données . . . 145
5.3.1 Données naturelles . . . 145
Waymo . . . 146
BDD100K . . . 149
Stellantis . . . 152
Boxy . . . 155
Audi . . . 156
CarCrash . . . 157
DashCam . . . 158
NuImages . . . 159
Waterloo . . . 162
PeSOTIF . . . 163
Edge & Corner (E&C) . . . 165
5.3.2 Données Simulées . . . 168
5.3.3 Étude sur l'utilisation de données simulées . . . 174
5.3.4 Récapitulatif de l'ensemble des jeux de données . . . 179
5.3.5 Conclusion et perspectives . . . 179

5.1 Introduction

Dans cette section, nous appliquons la méthodologie d'obtention de garanties de performances, définie précédemment, à une application de *Field Monitoring*, composée d'une chaîne de traitement d'images intégrant un module de compression d'images et un détecteur d'objets, tous deux basés sur des modèles de deep learning.

Pour rappel, comme indiqué dans la section 1, le *Field Monitoring* est un système de traitement d'images, dédiée à la sécurité dans le contexte des véhicules autonomes, déclenché par des événements spécifiques, appelés déclencheurs ou *triggers*, signalant des situations critiques ou anormales du système de conduite autonome (cf. Fig 1.3). Ces *triggers* incluent, par exemple, une demande de reprise de contrôle par le conducteur (avec ou sans défaillance du système), le déclenchement de l'ABS (système d'assistance au freinage), une incohérence dans la détection de trajectoire, un freinage brusque, un faible temps avant collision (proche du déclenchement du freinage automatique d'urgence, AEB), la détection de choc (déploiement de l'airbag), ou une crevaison.

Lorsqu'un tel événement est détecté, le système enregistre une fenêtre temporelle de 10 secondes, centrée autour du moment déclencheur ([-5s avant, +5s après]), à une fréquence minimale de 5 images par seconde. Ces images sont ensuite compressées sous forme de fichiers binaires pour réduire leur taille et simplifier leur transfert vers un centre de données Stellantis, où elles sont stockées. À une étape ultérieure, les fichiers binaires sont décompressés pour restaurer les images, lesquelles sont ensuite analysées automatiquement à l'aide de détecteurs d'objets ou d'algorithmes de reconnaissance de scènes. Ces analyses permettent d'identifier les éléments pertinents de l'environnement et d'évaluer les causes ayant conduit à l'événement critique.

L'objectif de ce chapitre est d'appliquer les deux premières étapes de la méthodologie détaillée dans le chapitre 4, en vue d'obtenir des garanties de performances pour la chaîne compression-détection. Plus précisément, nous allons dans un premier temps définir rigoureusement le système étudié et établir son ODD, afin de circonscrire les conditions opérationnelles dans lesquelles les garanties de performances seront évaluées. Ensuite, nous procéderons à la conception et à l'analyse de bases de données multi-sources, en combinant des données issues de sources naturelles et synthétiques, pour assurer une couverture exhaustive des scénarios représentatifs.

Dans un premier temps, dans la section 5.2, nous définirons les spécifications du système *Field Monitoring*, en identifiant précisément ses composants et les interactions entre le module de compression d'images et le détecteur d'objets. Nous y introduirons également une métrique d'erreur adaptée, servant de référence pour évaluer les performances globales du système. La définition de l'ODD, quant à elle, permettra de structurer et d'orienter l'acquisition des données en identifiant les environnements, conditions et situations critiques à prendre en compte pour garantir une évaluation cohérente et rigoureuse. Dans un second temps, dans la section 5.3, nous concevrons et analyserons des bases de données diversifiées provenant de sources multiples. Ces sources incluront des données naturelles, collectées à partir de caméras embarquées dans des environnements réels, ainsi que des données synthétiques, générées par un simulateur.

5.2 Définition du système et de l'ODD

Dans notre étude de cas, l'application se concentre sur une caméra embarquée située à l'avant d'un véhicule autonome (caméra frontale) qui perçoit des image en couleur de dimension $3 \times 640 \times 1280$ avec 3, 640 et 1280 représentant respectivement les 3 canaux RGB, la hauteur et la largeur de l'image. Par mesure de simplification, la dimension des images sera abrégée au format 640×1280.

Lorsqu'une fonction de conduite autonome présente un dysfonctionnement, un trigger (déclencheur) est envoyé au système Field Monitoring. Ce système s'active pour récupérer une séquence d'images associée à l'événement critique. Ces images, qui capturent les instants avant et après l'événement, sont ensuite compressées par un module de deep learning, converties en fichiers binaires pour faciliter leur transfert, puis envoyées à un centre de données. Les fichiers compressés sont récupérés ultérieurement, décompressés pour restaurer les images d'origine, et analysés par un module de détection d'objets automatique. L'objectif principal de cette étude est d'évaluer et de garantir que les distorsions induites par la compression ont un impact minimal, voire inexistant, sur la précision et la fiabilité de la détection d'objets. Ici, nous nous préoccupons uniquement de l'impact de la compression sur la détection et non sur la performance intrinsèque du détecteur. En d'autres termes, l'objectif est de garantir la robustesse de la détection vis-à-vis de la compression, et non d'évaluer directement les performances mêmes de la détection. Cela dit, il est possible qu'il existe un lien entre ces deux aspects, ce que nous analyserons plus en détail dans la suite de l'étude. Pour cela, nous définissons un système S composé de deux modules principaux : un module de compression d'images, qui réduit la taille des images tout en préservant leur qualité autant que possible, et un module de détection d'objets, qui identifie les objets d'intérêt (véhicules, piétons, cyclistes) dans les images. La Figure 4.4 illustre ce système, que nous décrivons ci-après.

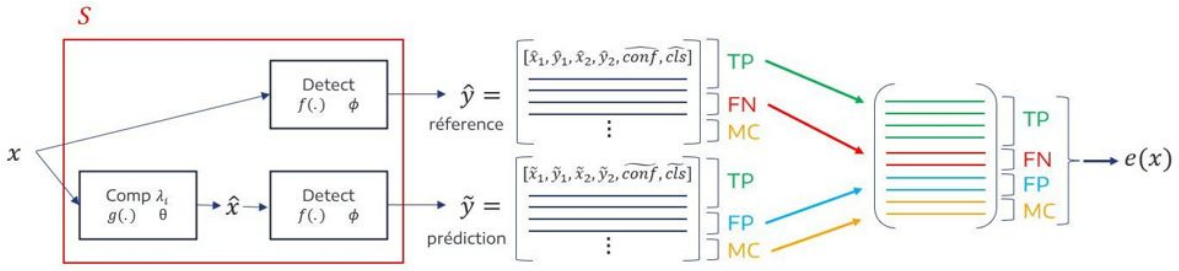


FIGURE 5.1 – Illustration du système S (encadré en rouge) et du processus de calcul de la fonction d'erreur $e(x)$.

Le système S est composé de deux modules basés sur l'intelligence artificielle :

- Module de compression d'images $g(.)$: Le module de compression est un modèle de deep learning paramétré par θ, dont les paramètres sont optimisés pendant un processus d'entraînement. Ce module applique une compression à une image d'entrée x, pour produire une image compressée puis décompressée $\hat{x}$. La compression est ajustable via un hyperparamètre λ_i, qui détermine le compromis entre débit et distorsion pendant l'entraînement. L'image d'entrée x et l'image reconstruite $\hat{x}$ appartiennent toutes deux à l'espace $[0,1]^{640\times1280}$. La fonction $g(.)$ est définie comme suit :

$$\begin{array}{rccc} g\ :\ [0,1]^{640\times1280} & \longrightarrow & [0,1]^{640\times1280} \\ x & \longmapsto & \hat{x} \end{array} \tag{5.1}$$

- Module de détection d'objets $f(.)$: Le module de détection d'objets est également un modèle de deep learning, paramétré par ϕ, appris au cours de son entraînement. Ce module est appliqué deux fois : une première fois directement à l'image d'entrée x pour produire la détection de référence $\hat{y}$, et une seconde fois à l'image compressée puis décompressée $\hat{x}$, pour produire la prédiction $\tilde{y}$. Les sorties $\hat{y}$ et $\tilde{y}$ sont des listes de boîtes 2D représentant les objets détectés. Chaque boîte est un vecteur de la forme $[x_1, x_2, y_1, y_2, \text{conf}, \text{cls}]$, où (x_1, y_1) est le point supérieur gauche, (x_2, y_2) est le point inférieur droit (cf. Figure 5.2), conf $\in [0,1]$ représente le niveau de confiance (probabilité qu'un objet soit présent) et cls indique la classe prédite (0 pour les véhicules, 1 pour les piétons et 2 pour les cyclistes). La fonction $f(.)$ est définie comme suit :

$$\begin{array}{rccc} f\ :\ [0,1]^{640\times1280} & \longrightarrow & \mathbb{R}^{n\times6} \\ x & \longmapsto & \hat{y} \end{array} \tag{5.2}$$

  où n est le nombre de boîtes détectées.

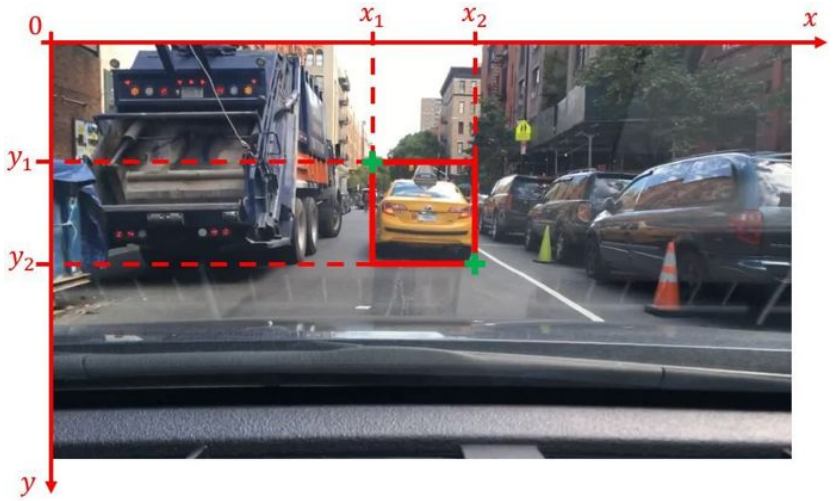


FIGURE 5.2 – Illustration du système de coordonnées pour les boîtes 2D.

On souhaite comparer et évaluer les détections effectuées sur les images compressées $\tilde{y}$ par rapport aux détections de référence $\hat{y}$, obtenues à partir des images non compressées. Comme vue dans la section 3.3.2, cette comparaison permet de catégoriser les détections en quatre classes principales :

- True Positives (TP) : Détections correctes, où une boîte détectée dans $\tilde{y}$ correspond exactement à une boîte dans $\hat{y}$, sans être affectée par la compression.
- False Negatives (FN) : Objets manqués, présents dans la référence $\hat{y}$ mais absents dans $\tilde{y}$, indiquant que la compression a entraîné la perte d'une détection.
- False Positives (FP) : Objets détectés par erreur, présents dans $\tilde{y}$ mais absents de $\tilde{y}$, signalant une détection introduite à tort par la compression.
- Misclassifications (MC) : Objets mal classifiés, où la classe d'une détection dans $\tilde{y}$ diffère de la référence $\hat{y}$.

Ces classifications (TP, FN, FP, MC) sont ensuite synthétisées dans une liste ou un vecteur final, qui sert de base pour évaluer les performances du système. Dans la suite des etapes de la methodologie, il faut définir une

fonction d'erreur qui quantifier la performances du système de maniere interpretable pour les utilisateurs. Nous définissons donc la fonction d'erreur $e(x)$ qui quantifie la quantité d'erreur causées par la compression. Elle est donnée par :

$$e(x) = \#FN + \#FP + \#MC \tag{5.3}$$

où # représente la quantité respective de FN, FP et MC. Autrement dit, la fonction $e(x)$ évalue l'impact global de la compression sur la qualité des détections effectuées par le système.

Maintenant que nous avons bien défini le système dont on souhaite garantir les performances ainsi que son domaine d'application, l'étape suivante de la méthodologie consiste à définir son Operational Design Domain (ODD). Cette étape permet de déterminer les cadres pertinents pour les données d'entrée (les images) que le système pourrait rencontrer dans son fonctionnement réel.

En effet, l'espace mathématique des entrées de notre système est défini par $[0, 1]^{H \times W}$, correspondant à des images normalisées avec les trois canaux RGB. Cependant, toutes les images de cet espace ne sont pas représentatives des situations que notre système rencontrera. Par exemple, il n'est pas nécessaire de garantir les performances du système sur des données totalement hors contexte ou irréalistes (par exemple, des images sans rapport avec la conduite). C'est pourquoi il est crucial de restreindre cet espace théorique des entrées à un sous-ensemble défini par l'ODD, lequel spécifie les conditions opérationnelles dans lesquelles le système doit fonctionner en toute fiabilité. Pour ce faire, il est nécessaire de définir les attributs pertinents de l'ODD. Ces attributs permettent de caractériser l'environnement, les contraintes et les scénarios dans lesquels le système sera exposé. Dans notre cas d'application, les catégories principales incluent :

- **Conditions environnementales** : elles englobent des paramètres tels que l'éclairage (jour, nuit, lever/coucher du soleil), les conditions météorologiques (ciel clair, pluie, neige, vent) et l'état du pare-brise (salissures, marques d'essuie-glace, fissures). Ces conditions influencent directement la visibilité et les performances des capteurs.
- **Infrastructures physiques** : elles comprennent les types de routes (autoroute, intersection, zone de stationnement), les surfaces de la chaussée (asphalte, béton) ainsi que les marquages au sol (lignes simples, multiples, passages piétons). Ces éléments définissent les caractéristiques physiques de l'environnement que le système doit interpréter.
- **Zones géographiques** : elles spécifient les régions où le système est destiné à opérer (par exemple, Europe, USA, Chine) ainsi que les zones d'interférences comme les tunnels, ponts, ou chantiers. D'autres zones délimitées géographiquement, telles que les centres-villes ou les campus scolaires, imposent également des contraintes particulières.
- **Conditions dynamiques** : elles concernent des aspects comme la densité de trafic (faible, moyenne, élevée) ou la présence d'accidents (voitures en feu, interventions d'urgence). Ces conditions peuvent altérer le comportement du système.
- **Objets** : cette catégorie inclut les différents types d'objets que le système doit identifier et traiter, comme les véhicules (voitures, camions, motos), les piétons (debout, marchant, tombant) et les cyclistes, ainsi que des objets ignorés tels que des panneaux, animaux ou débris.

Ces attributs permettent de structurer et de limiter efficacement l'espace des entrées, en excluant les cas irréalistes ou hors du domaine d'application attendu. Cette définition de l'ODD est essentielle pour garantir que le système est testé et validé dans des conditions représentatives de son utilisation réelle. Une illustration partielle de l'ODD est présentée dans la Figure 5.3 suivante, qui détaille les principales catégories d'attributs et leurs sous-éléments spécifiques.

Maintenant que le système et son ODD sont clairement définis, nous disposons d'un cadre précis pour restreindre et caractériser les données pertinentes. Cette étape nous permet de nous concentrer exclusivement sur les scénarios et conditions opérationnelles spécifiés, en excluant les cas hors du domaine d'application. Nous pouvons ainsi procéder à l'acquisition des données adaptées à notre application, en veillant autant que possible à ce qu'elles couvrent de manière représentative les attributs définis dans l'ODD, tels que les conditions environnementales, les infrastructures physiques, les zones géographiques, les conditions dynamiques et les objets détectables.

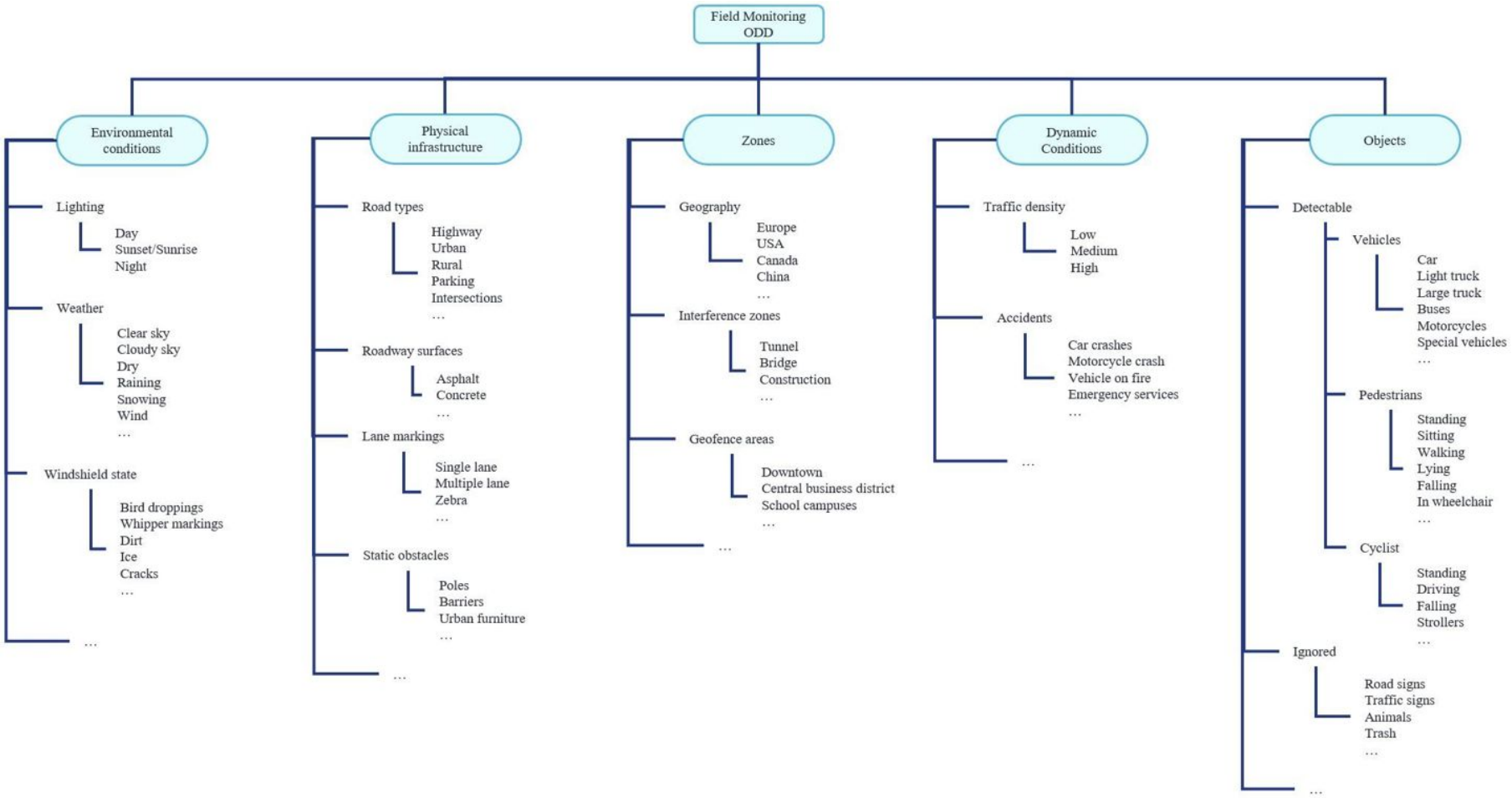


FIGURE 5.3 – ODD partiel du système Field Monitoring.

5.3 Acquisition et analyse de données

Afin d'entraîner les modules d'IA, d'évaluer correctement les performances de notre système et d'estimer des garanties de performances, nous aurons besoin de bases de données variées couvrant autant que possible l'ensemble des situations définies par l'ODD précédemment défini. Ces données pourront être de deux types : naturelles, c'est-à-dire obtenues via une acquisition réelle par une caméra embarquée à l'avant d'un véhicule, ou synthétiques, générées par des simulateurs ou des outils de modélisation numérique. Les données synthétiques permettent de compléter les scénarios difficiles à capturer dans le monde réel et de tester le système dans des situations contrôlées ou extrêmes.

Pour entraîner nos modules d'IA, notamment celui de détection d'objets, nous aurons besoin d'annotations précises. Certaines bases de données seront ainsi annotées à l'aide de boîtes englobantes 2D, définies par les coordonnées $[x_1, y_1, x_2, y_2]$ pour indiquer la position de l'objet dans l'image, et une classe associée cls, où 0 correspond aux véhicules, 1 aux piétons, et et 2 aux cyclistes. Étant donné que les données peuvent provenir de différentes sources, il sera nécessaire de les uniformiser afin de respecter un format homogène. Par exemple, toutes les images seront redimensionnées en 640×1280 pour garantir une compatibilité entre les ensembles de données. En cas de rognage, nous privilégierons la suppression des parties hautes de l'image, souvent occupées par le ciel (zone non pertinente pour l'application), ainsi que des côtés droit et gauche de manière équitable pour préserver les informations principales.

Un point particulier concernant notre système mérite d'être souligné : pour évaluer ses performances et obtenir des garanties, nous n'avons pas besoin de données annotées. En effet, nous comparons une prédiction réalisée sur une image compressée à une autre prédiction réalisée sur la version non compressée de la même image. Cela signifie que seules les images brutes sont nécessaires, ce qui simplifie considérablement la tâche. Cependant, comme nous le verrons dans le chapitre 7, d'autres types d'applications et systèmes nécessiteront impérativement des annotations, notamment pour valider leurs performances sur des ensembles de test réels.

Pour garantir la cohérence des annotations lors de l'intégration de multiples jeux de données et éviter les biais introduits par des objets trop petits, nous appliquerons des seuils de taille minimale. Les boîtes englobant les véhicules devront contenir au moins 1200 pixels, tandis que celles des piétons et des cyclistes devront contenir au moins 500 pixels. Cela permettra de garantir une qualité minimale des données annotées et d'améliorer la robustesse du modèle face aux objets mal représentés.

Enfin, toutes les images acquises, qu'elles soient naturelles ou synthétiques, seront compressées et stockées au format JPEG avec un taux de compression de 95, afin de minimiser la consommation de stockage tout en préservant la qualité visuelle essentielle au fonctionnement des modules d'IA.

5.3.1 Données naturelles

Dans cette section, nous dresserons la liste des différentes bases de données d'images naturelles qui ont été acquises ou construites au cours de notre étude. Un aspect important de ces données, susceptible d'influencer notre analyse, est que la majorité des sources sont en libre accès, mais elles sont généralement soumises aux régulations européennes de protection des données (GDPR). Cela impose certaines restrictions, notamment le floutage des visages et des plaques d'immatriculation, rendant les images récupérées rarement brutes ou totalement exploitables dans leur forme originale. Ces limitations peuvent introduire des biais visuels ou des artefacts qui doivent être pris en compte lors de l'entraînement ou de l'évaluation des modèles. Les images varient également en fonction du positionnement de la caméra frontale lors de l'acquisition. Lorsqu'elle est placée à l'intérieur du véhicule, la présence du pare-brise peut introduire des perturbations visuelles, comme des reflets, des salissures ou des marques d'essuie-glace, susceptibles d'affecter les performances des modèles de détection. En revanche, lorsque la caméra est positionnée à l'extérieur du véhicule, ces effets sont minimisés, mais l'exposition directe aux éléments extérieurs (pluie, poussière) peut avoir un autre type d'impact sur la qualité des données. Les bases de données se distinguent également par leur origine. Les acquisitions dites amateurs, telles que celles issues de vidéos YouTube ou d'enregistrements personnels, présentent souvent des variations significatives dans la qualité des images et le positionnement des caméras. La caméra est généralement placée à l'intérieur du véhicule, ce qui accroît la variabilité des conditions et des perturbations observées. À l'inverse, les acquisitions dites professionnelles, souvent réalisées par des entreprises spécialisées, bénéficient d'un matériel sophistiqué et consistent généralement en des caméras montées à l'extérieur des véhicules. Ces données offrent une meilleure qualité globale et une plus grande homogénéité, ce qui peut faciliter leur exploitation. Un autre aspect crucial à prendre en compte est la licence associée à chaque base de données. Dans le cadre de cette recherche académique, les licences des bases utilisées permettent généralement une exploitation à des fins non commerciales, ce qui inclut

l'entraînement des modèles et l'évaluation méthodologique. Cependant, lorsque cette méthodologie sera appliquée dans un cadre industriel, il sera impératif de vérifier avec précision les termes des licences d'utilisation. Cela inclut les restrictions éventuelles sur l'usage commercial des données, les obligations de mention des auteurs ou des organismes créateurs, ainsi que les conditions liées à la modification, redistribution ou intégration des données dans des produits finaux. Par exemple, certaines bases de données, bien qu'en accès libre, imposent des limites strictes pour l'usage dans des contextes industriels. Il sera alors nécessaire de prévoir un budget pour l'obtention de licences commerciales ou pour la création de jeux de données propriétaires. Ces considérations sont d'autant plus importantes que le non-respect des licences pourrait entraîner des litiges juridiques, impactant la mise en œuvre industrielle des systèmes. Enfin, il est important de noter que les métadonnées associées à ces bases de données (comme les conditions environnementales, les informations temporelles ou les caractéristiques des scènes) peuvent grandement enrichir leur utilité pour notre étude. Ces informations, lorsqu'elles sont disponibles, permettent de mieux analyser l'impact des différents attributs de l'ODD sur les performances du système. Cependant, toutes les bases de données ne fournissent pas ces métadonnées de manière uniforme, ce qui introduit une autre source de variabilité à considérer dans notre approche.

Waymo

Waymo Open Dataset [1] [Sun et al., 2020] est une base de données créée par Waymo LLC en 2020, anciennement connu sous le nom de Google Self-Driving Car Project, qui est une entreprise américaine de technologie de conduite autonome dont le siège est à Mountain View, en Californie. Elle est une filiale d'Alphabet Inc, la société mère de Google. Parmi les deux ensembles de données que propose Waymo, celui que nous exploiterons se nomme "Perception". Ses données comprennent des séquences d'images RGB haute résolution (1280×1920 et 1040×1920) capturées par plusieurs caméras (latéral gauche, avant gauche, avant, avant droit et latéral droit) positionnées sur le toit du véhicule, des nuages de points LIDAR en 3D et des annotations pour trois types d'objets : véhicule, piéton et cycliste. Les données ont été acquises majoritairement de jour aux États-Unis, avec la caméra située à l'extérieur sur le toit du véhicule, à Phoenix sur une zone de $40km^2$ et à San Francisco et Mountain View sur une zone cumulée de $36km^2$ (cf. Figure 5.4 et Tableau 5.1). Dans [Sun et al., 2020], Mountain View et Phoenix sont décrits comme des environnements de type suburban, caractérisés par des zones résidentielles plus étendues et moins denses, ce qui implique que les images capturées dans ces zones contiennent davantage de routes larges, d'espaces ouverts et de véhicules à distance. En revanche, San Francisco, étant un environnement urbain dense, présente des scènes plus encombrées avec des piétons, des cyclistes et des véhicules proches les uns des autres.



FIGURE 5.4 – Lieux géographiques d'acquisition des données Waymo : San Francisco à gauche, Mountain View au centre et Phoenix à droite.

1. http ://www.waymo.com/open

TABLE 5.1 – Informations sur les scénarios de Waymo.

boites 2D	**Véhicule**	**Piéton**	**Cycliste**	**Total**
	9M	2.7M	81k	12M
# Scènes	**Phoenix**	**Mountain View**	**San Francisco**	**Total**
	379	124	497	1000
	Jour	**Nuit**	**Aube**	**Total**
	806	102	92	1000

Dans notre cas, nous créons une base de données nommée également Waymo, contenant exclusivement les images issues de la caméra frontale, donnant un total de $195,823$ images, de dimension 1280×1920, qu'on redimensionne en 640×1280 (quelques exemples sont montrés dans la Figure 5.5). Pour chaque image, on récupère également leurs annotations pour les classes "véhicule", "piéton" et "cycliste", qui n'ont pas besoin d'être modifiées, car conformes à nos besoins. Cependant, nous rappelons que des limitations sont appliquées sur la taille des boîtes englobantes (au moins 1200 pixels pour les véhicules et 500 pixels pour les piétons et cyclistes).

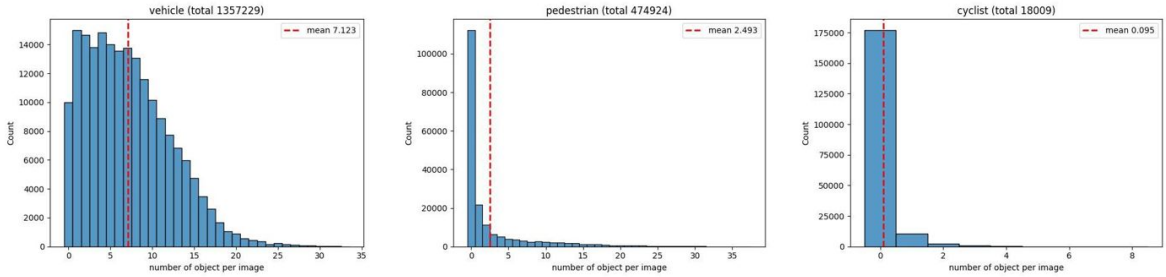


(a) Centre-ville. (b) Banlieue

(c) Ville. (d) Nuit et Pluie.

FIGURE 5.5 – Exemples d'images Waymo.

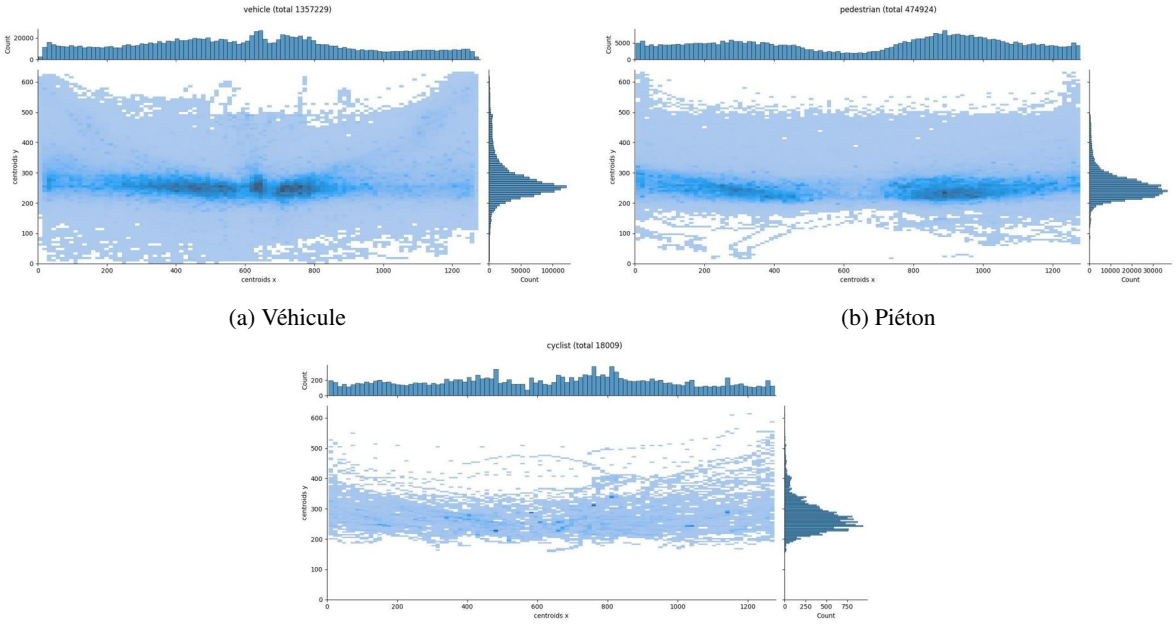
Une analyse statistique de la répartition des objets, de leurs positions et de leurs tailles dans l'ensemble des images est présentée dans les Figures 5.6, 5.7 et 5.8. On peut tout d'abord voir dans la Figure 5.6 que, dans l'ensemble, les images du jeu de données Waymo contiennent beaucoup plus de véhicules (en moyenne 7.1 par image) que de piétons (en moyenne 2.5 par image) et de cyclistes (en moyenne 0.1 par image). Cela est dû à la capture d'images en milieu urbain et sur route, où les véhicules dominent.



(a) Véhicule (b) Piéton (c) Cycliste

FIGURE 5.6 – Histogramme de la quantité d'objets par image de Waymo pour les classes véhicule, piéton et cycliste.

De plus, dans la Figure 5.7, on observe que pour les trois classes, les objets sont répartis de manière plutôt homogène sur l'axe des x, mais se concentrent autour du centre de l'image sur l'axe des y, ce qui est dû à la perspective de la caméra embarquée à l'avant du véhicule capturant principalement la route au centre de l'image.

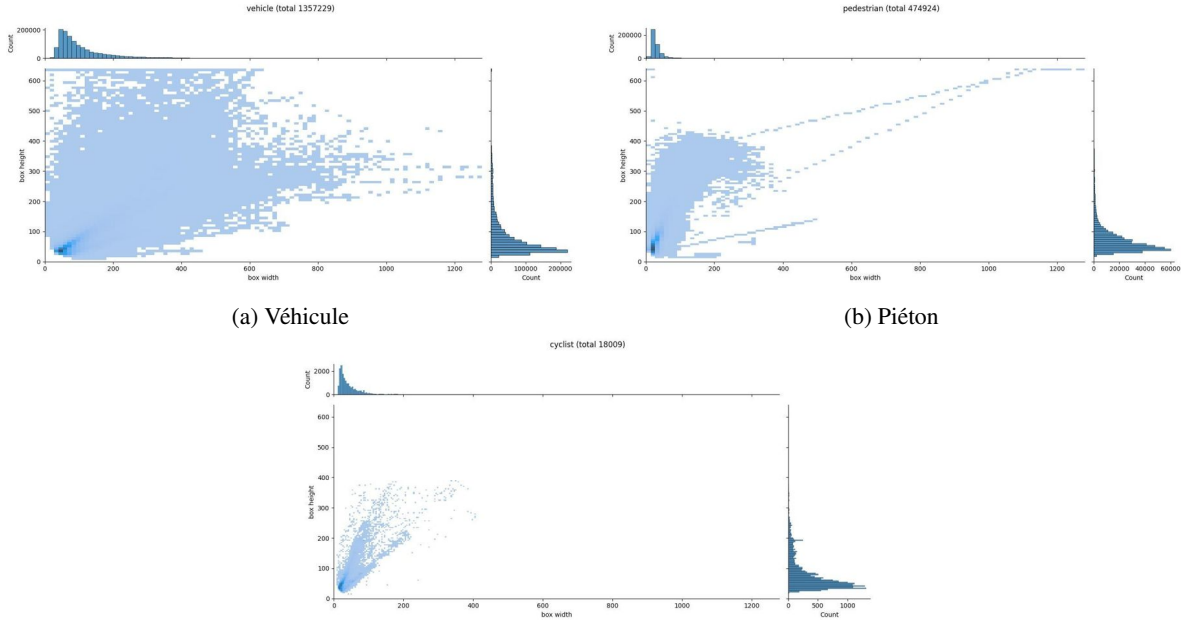


(a) Véhicule (b) Piéton

(c) Cycliste

FIGURE 5.7 – Histogramme 2D de la distribution des centroïdes des boîtes dans les images de Waymo pour les classes véhicule, piéton, cycliste.

Enfin, dans la Figure 5.8, on remarque que les boîtes englobantes des véhicules présentent une distribution étendue, variant de très petites à assez grandes tailles, mais sont majoritairement inférieures à 200×200. Cela reflète les différentes distances des véhicules par rapport à la caméra. En revanche, pour les piétons et les cyclistes, les tailles des boîtes sont beaucoup plus concentrées, avec la majorité étant inférieure à 100×100. Les boîtes plus grandes, dépassant 500×500, observées chez les piétons, semblent être des erreurs d'annotation.



(a) Véhicule

(b) Piéton

(c) Cycliste

FIGURE 5.8 – Histogramme 2D de la distribution de la largeur et hauteur des boites dans les images de Waymo pour les classes véhicule, piéton et cycliste.

BDD100K

Le BDD100K [2] (Berkeley DeepDrive 100K)[Yu et al., 2020] est une base de données développée en 2020 par l'équipe du Berkeley AI Research (BAIR). Il contient environ 80, 000 images annotées provenant de vidéos enregistrées sur des routes, couvrant un large éventail de scénarios de conduite dans diverses conditions environnementales et géographiques. Les données incluent principalement des images et vidéos capturées à partir de caméras montées à l'avant de véhicules, avec une résolution de $3 \times 720 \times 1280$ (720p). Chaque vidéo dure environ 40 secondes et est enregistrée à 30 images par seconde. Les images sont annotées pour plusieurs tâches, telles que la détection d'objets via des boîtes englobantes 2D (véhicules, piétons, cyclistes, feux de circulation) et la segmentation sémantique (routes, trottoirs, bâtiments, etc.). Un aspect clé de BDD100K est sa couverture d'une grande variété de conditions environnementales, incluant différentes conditions météorologiques (ciel dégagé, pluie, neige, nuit) et des moments variés de la journée (jour, crépuscule, nuit). Le dataset est conçu pour être utilisé dans une large gamme d'environnements, des zones urbaines denses aux banlieues plus étendues et aux zones rurales. Il couvre également des données collectées dans diverses villes et régions des États-Unis (cf. Figure 5.9).

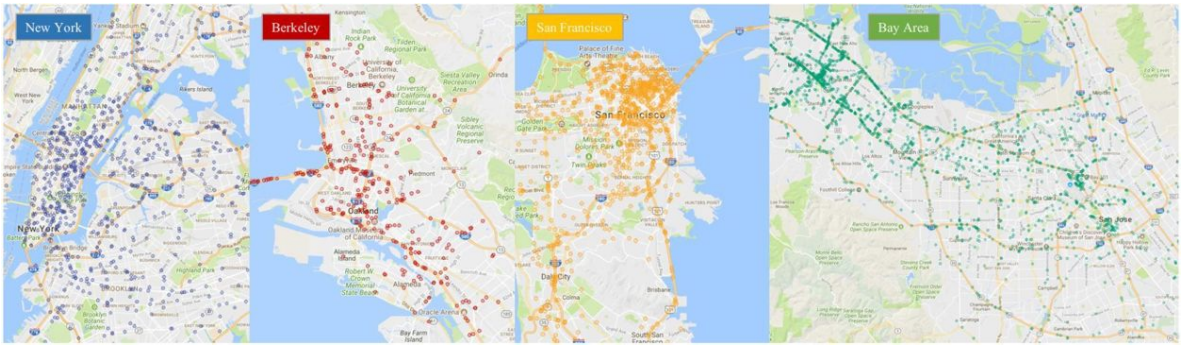


FIGURE 5.9 – Lieux géographique d'acquisition des données BDD100K.

Dans notre cas, nous créons une base de données nommée également BDD100K, contenant l'ensemble des

2. bdd-data.berkeley.edu

80, 000 images (quelques exemples sont présentés dans la Figure 5.10), de dimension 740×1280, qu'on redimensionne en 640×1280. De plus, nous ajusterons les annotations afin qu'elles soient conformes à nos besoins. Pour ce faire, nous regroupons et modifions les classes de BDD100K en "véhicule", "piéton" et "cycliste".



(a) (b)

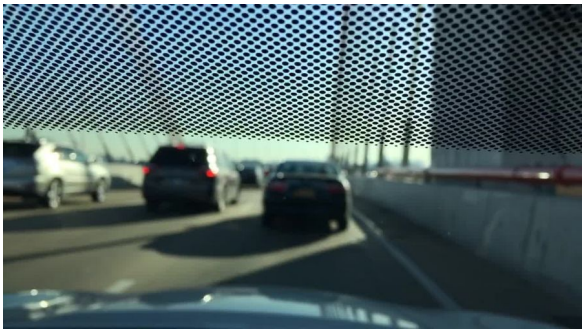
(c) (d)

FIGURE 5.10 – Exemples d'images BDD100K.

Une autre particularité de la base de données BDD100K est que la caméra frontale est placée à l'intérieur du véhicule, ce qui implique que le pare-brise peut générer des reflets indésirables, perturbant ainsi la qualité des images capturées. Ces reflets peuvent interférer avec la détection d'objets et la segmentation sémantique en faussant la perception visuelle des algorithmes de vision par ordinateur. De plus, si la caméra est mal positionnée, cela peut entraîner une mauvaise couverture de la scène routière, réduisant ainsi la précision dans la capture des éléments critiques tels que les piétons ou les véhicules (cf. Figure 5.11).



(a) Mauvaise mise au point de la caméra.



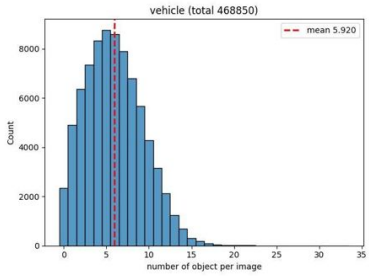
(b) Caméra mal positionnée.



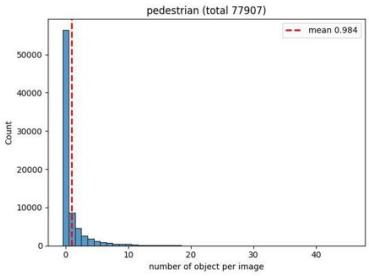
(c) Reflet sur le par-brise.

FIGURE 5.11 – Exemples d'images BDD100K où le pare-brise impacte les images.

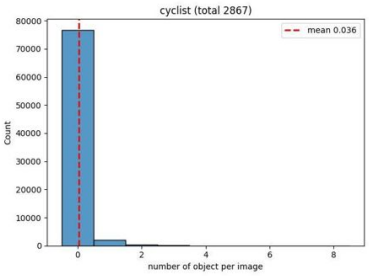
L'analyse statistique des objets dans la base de données BDD100K montre que les images contiennent en moyenne beaucoup plus de véhicules (environ 6 par image) que de piétons (environ 1 par image) et de cyclistes (environ 0.036 par image), en raison de la nature urbaine et routière des scènes capturées (cf. Figure 5.12).



(a) Véhicule.



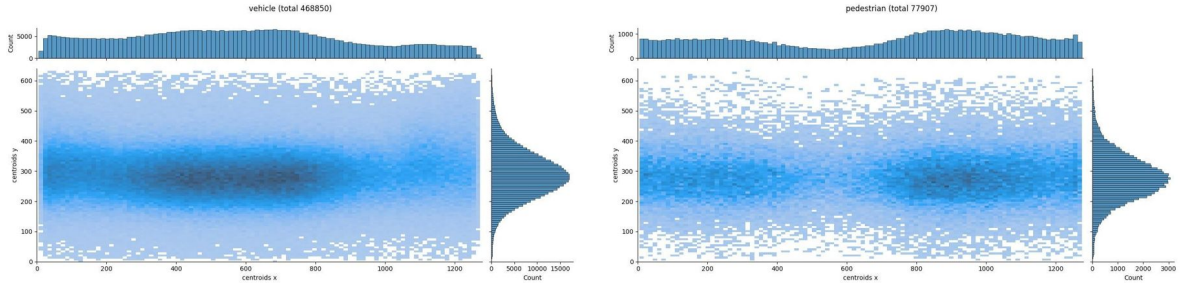
(b) Piéton.



(c) Cycliste.

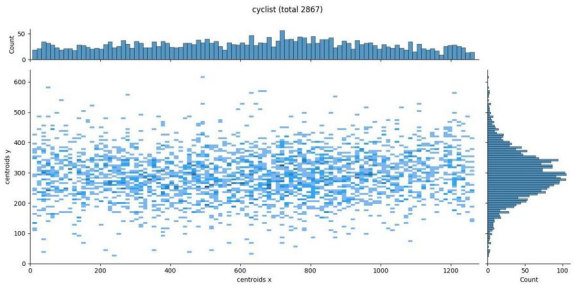
FIGURE 5.12 – Histogramme de la quantité d'objets par image sur BDD100K pour les classes véhicule, piéton et cycliste.

De plus, les objets sont répartis de manière homogène sur l'axe des x, mais se concentrent au centre de l'image sur l'axe des y, mais plus dispersé ou étendu comparé à la base Waymo (cf. Figure 5.13).



(a) Véhicule.

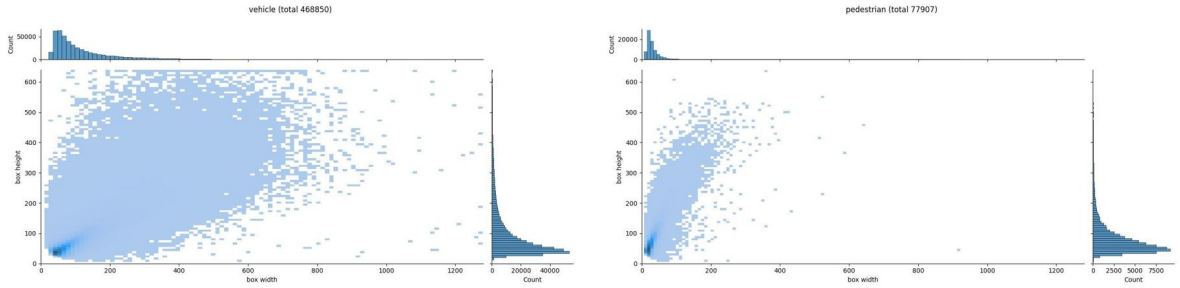
(b) Piéton.



(c) Cycliste.

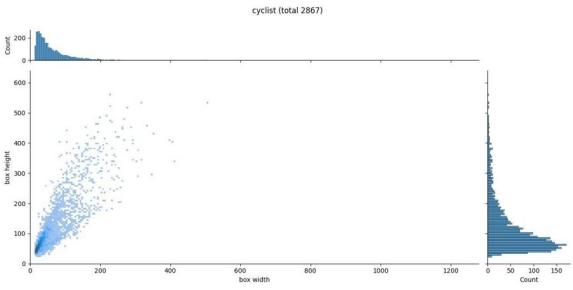
FIGURE 5.13 – Histogramme 2D de la distribution des centroïdes des boîtes dans les images sur BDD100K pour les classes véhicule, piéton, cycliste.

Enfin, les boîtes englobantes des véhicules varient en taille, la plupart étant inférieures à 200×200, tandis que celles des piétons et des cyclistes sont majoritairement inférieures à 100×200 (cf. Figure 5.14).



(a) Véhicule.

(b) Piéton.



(c) Cycliste.

FIGURE 5.14 – Histogramme 2D de la distribution de la largeur et hauteur des boites dans les images sur BDD100K pour les classes véhicule, piéton et cycliste.

Stellantis

Cette base de données privée, fournie par l'entreprise Stellantis, comprend 20, 139 images de dimensions 720×1280, capturées par une caméra frontale située à l'intérieur du véhicule. Les images couvrent une grande

diversité de situations et des lieux géographiques multiples en France, allant des zones rurales aux centres urbains denses. Les annotations associées incluent des boîtes englobantes 2D pour plusieurs classes d'objets, telles que "voiture", "camion", "bus", "deux roues" (avec ou sans conducteur), "signalisation", ainsi que divers autres éléments d'infrastructure.

Dans notre cas, nous utilisons cette base pour constituer un ensemble de données nommé Stellantis, contenant l'intégralité des 20, 139 images (quelques exemples sont présentés dans la Figure 5.15). Pour harmoniser cette base avec nos besoins, nous redimensionnons toutes les images de leur format original 3×720×1280 à 640×1280. De plus, nous regroupons et modifions les annotations pour les adapter à nos besoins, en consolidant les classes de Stellantis en "véhicule", "piéton" et "cycliste".

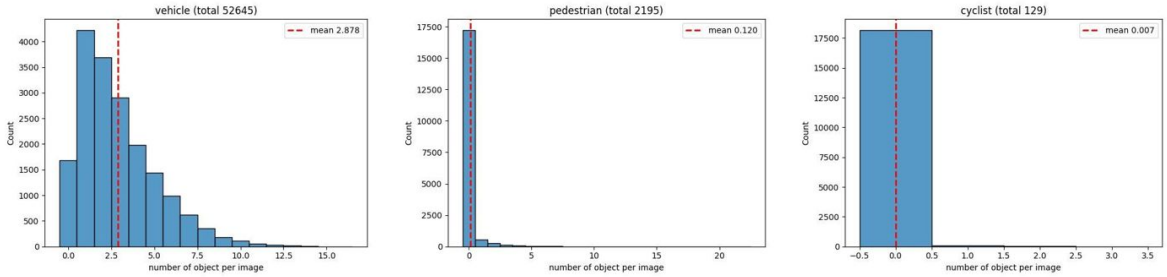


(a) Centre-ville. (b) Campagne.

(c) Nuit. (d) Autoroute.

FIGURE 5.15 – Exemples d'images Stellantis.

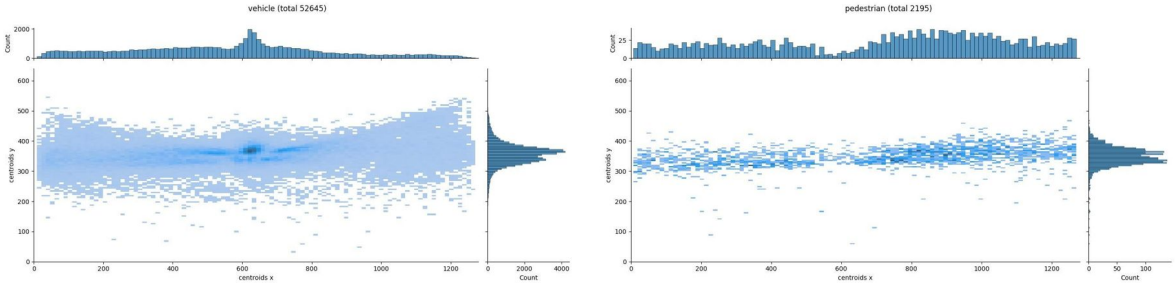
L'analyse statistique des objets dans la base de données Stellantis révèle une répartition déséquilibrée des classes d'objets, avec une nette prédominance des véhicules par rapport aux piétons et cyclistes. En moyenne, chaque image contient environ 3 véhicules, tandis que le nombre moyen de piétons est beaucoup plus faible (environ 0,1 par image) et celui des cyclistes est négligeable (environ 0,007 par image). Cette distribution reflète probablement une forte proportion de scènes capturées dans des environnements ruraux et sur autoroutes, où les cyclistes et piétons sont naturellement moins fréquents, voire absents (cf. Figure 5.16). Ces observations soulignent un biais inhérent à la base de données, qui devra être pris en compte dans l'évaluation des performances du système.



(a) Véhicule. (b) Piéton. (c) Cycliste.

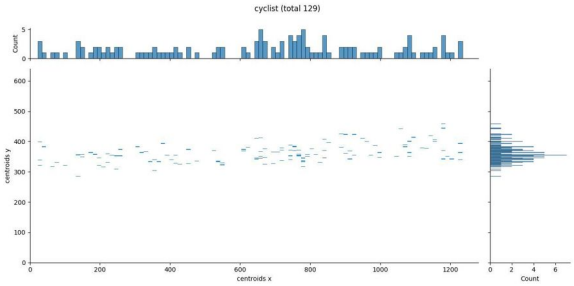
FIGURE 5.16 – Histogramme de la quantité d'objets par image sur Stellantis pour les classes véhicule, piéton et cycliste.

De plus, l'analyse de la répartition spatiale des objets dans les images montre que les centroïdes des objets sont uniformément distribués sur l'axe des x, mais se concentrent majoritairement au centre de l'image sur l'axe des y. Ce phénomène est dû à la perspective offerte par la caméra embarquée, où les objets situés à l'horizon tendent à se regrouper dans la partie centrale verticale de l'image (cf. Figure 5.17).



(a) Véhicule.

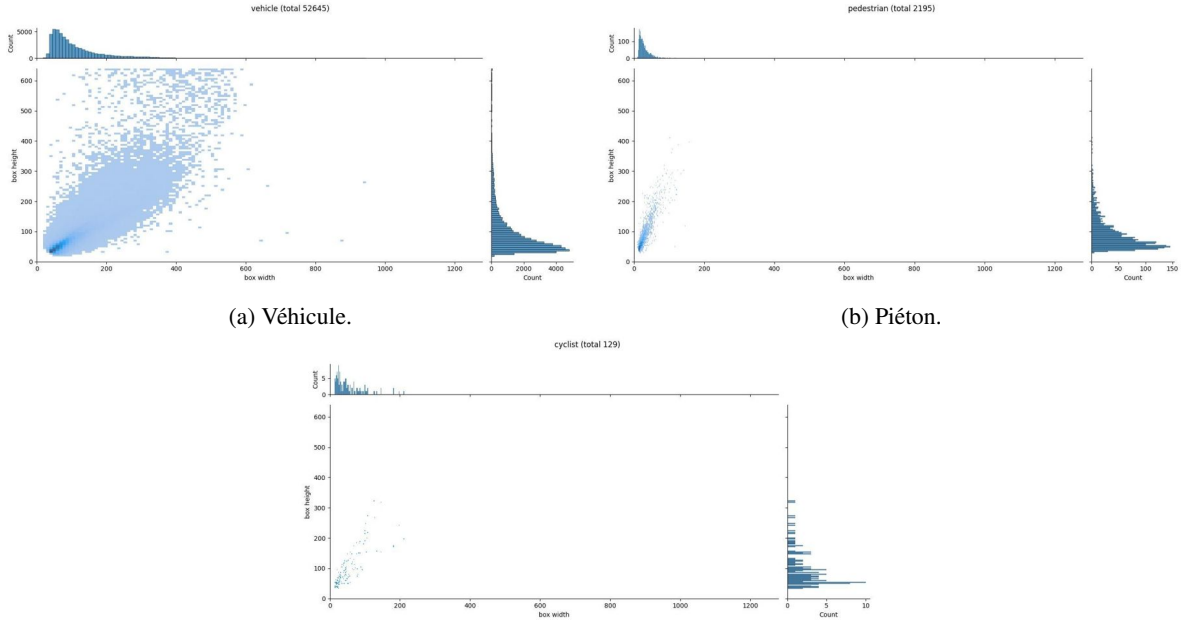
(b) Piéton.



(c) Cycliste.

FIGURE 5.17 – Histogramme 2D de la distribution des centroïdes des boîtes dans les images sur Stellantis pour les classes véhicule, piéton, cycliste.

Enfin, en ce qui concerne la taille des boîtes englobantes, les véhicules présentent une plus grande variabilité, bien que la majorité des boîtes soient inférieures à 200×200 pixels. Les boîtes correspondant aux piétons et aux cyclistes sont globalement plus petites, la plupart ayant une taille inférieure à 100×100 pixels (cf. Figure 5.18).



(a) Véhicule.

(b) Piéton.

(c) Cycliste.

FIGURE 5.18 – Histogramme 2D de la distribution de la largeur et hauteur des boites dans les images sur Stellantis pour les classes véhicule, piéton et cycliste.

Boxy

La base de données Boxy [Behrendt, 2019] contient un total de 178, 525 images capturées par une caméra frontale située à l'extérieur du véhicule, avec une résolution native de 2464×2056. Ces images sont annotées avec des boîtes englobantes en 2D et 3D. Tous les véhicules se déplaçant dans la même direction que la caméra, qu'ils soient sur la voie principale, les rampes d'accès et de sortie, ou sur des routes parallèles, sont regroupés sous une seule classe intitulée "véhicule". Cette classe inclut une large variété de types de véhicules : voitures particulières, camions, camping-cars, bateaux, transporteurs de voitures, équipements de construction et motos. Cependant, certaines limitations ont été relevées dans les annotations : tous les véhicules présents dans les images ne sont pas systématiquement annotés, ce qui rend ces données partiellement exploitables dans le cadre de notre étude. En tout, la base contient 1, 990, 806 véhicules annotés dans des scènes exclusivement capturées sur les autoroutes de la région de la baie de San Francisco. Les données couvrent principalement des segments de routes telles que les routes 85 et 92 ainsi que les autoroutes 101 et 280 en Californie. Les enregistrements incluent une large gamme de conditions météorologiques, allant de journées ensoleillées à des périodes de ciel couvert, de fortes pluies, de crépuscule et même de conduite nocturne. Ces images capturent également une variété de conditions de circulation, depuis des routes presque désertes jusqu'à des embouteillages importants, offrant ainsi une vue représentative de la conduite typique sur autoroute.

Dans notre cas, nous utilisons cette base de données pour créer un ensemble nommé également Boxy, qui contient l'intégralité des 178, 525 images disponibles (quelques exemples sont présentés dans la Figure 5.19). Pour rendre ces images compatibles avec nos besoins et nos autres ensembles de données, nous redimensionnons les images de leur format original 2464×2056 à une résolution standardisée de 640×1280.



(a) (b)

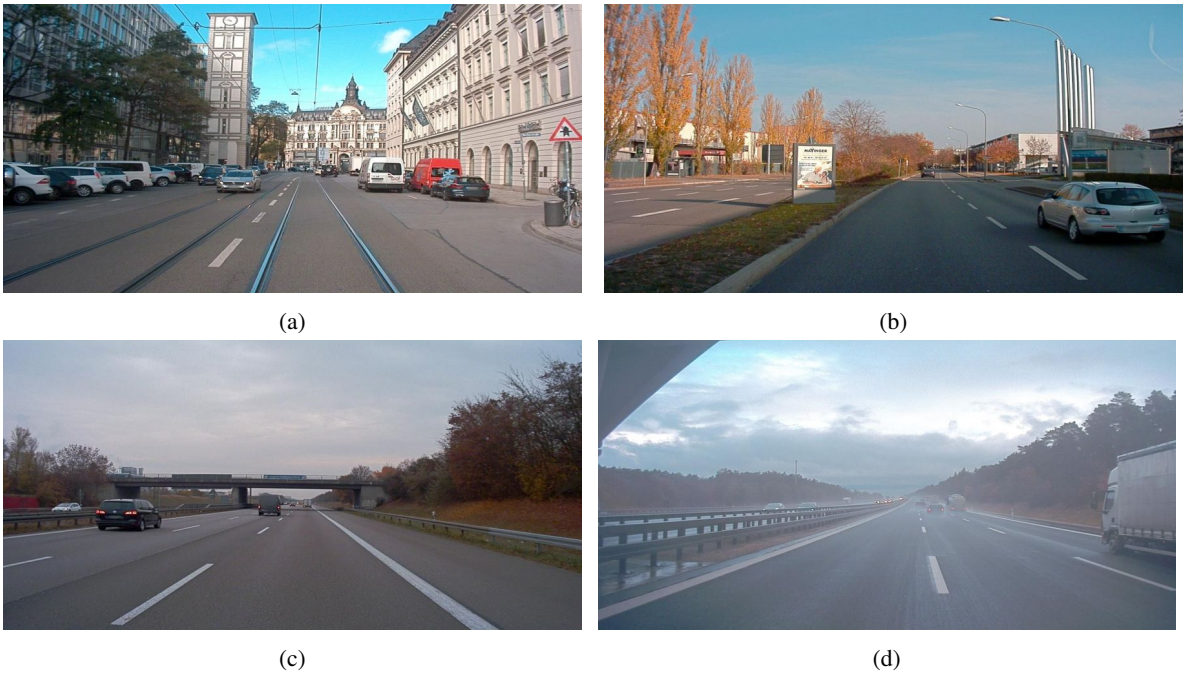
(c) (d)

FIGURE 5.19 – Exemples d'images Boxy.

Audi

La base de données Audi Autonomous Driving Dataset [Geyer et al., 2020] (A2D2) est un ensemble de données développé par Audi pour soutenir la recherche en conduite autonome. Ce dataset contient des données multimodales collectées en Allemagne par un véhicule équipé de plusieurs capteurs, notamment des caméras, des LIDARs et des RADARs. Ces données sont utilisées pour des tâches variées, telles que la détection d'objets, la segmentation sémantique, la segmentation d'instances, et d'autres processus essentiels pour le développement et l'évaluation des systèmes de conduite autonome. L'ensemble de données comprend des images synchronisées avec des nuages de points 3D, accompagnées d'annotations riches telles que des boîtes englobantes 3D et des labels de segmentation sémantique. La suite de capteurs utilisée offre une couverture complète à 360° autour du véhicule grâce à six caméras positionnées à l'avant, sur les côtés, et à l'arrière (quelques exemples sont montrés dans la Figure 5.20), ainsi qu'à cinq unités LiDAR. Ces capteurs permettent un enregistrement temporellement synchronisé et spatialement aligné, garantissant une corrélation précise entre les données collectées. En termes d'annotations, l'A2D2 fournit des labels pour des images non séquentielles, incluant un total de $41,277$ images de dimensions 1208×1920, avec des annotations de segmentation sémantique. Parmi celles-ci, $12,497$ images sont également dotées de boîtes englobantes 3D pour les objets situés dans le champ de vision de la caméra frontale. Les annotations couvrent une large variété d'objets, tels que les véhicules, piétons, panneaux de signalisation, cyclistes, et autres éléments pertinents pour la conduite autonome. Les données ont été collectées dans une grande variété d'environnements, comprenant des zones urbaines densément peuplées, des zones rurales, et des autoroutes. Ces environnements reflètent également une diversité de conditions météorologiques et temporelles, incluant des scènes diurnes et nocturnes, ainsi que des conditions pluvieuses.

Dans notre cas, nous créons une base de données nommée Audi, contenant l'ensemble des $12,497$ images issues de la caméra frontale (quelques exemples sont présentés dans la Figure 5.20). Ces images sont redimensionnées de leurs dimensions originales 1208×1920 à 640×1280 pour s'harmoniser avec nos autres ensembles de données.



(a) (b)

(c) (d)

FIGURE 5.20 – Exemples d'images Audi

CarCrash

Le Car Crash Dataset[Bao et al., 2020] (CCD) est une base de données conçue pour l'anticipation des accidents de la route, offrant des annotations détaillées sur les attributs environnementaux et les causes des collisions. Ce dataset contient 1,500 vidéos de collisions extraites de YouTube, chacune d'une durée de 5 secondes, capturées à une fréquence de 10 images par seconde et avec des dimensions de 720×1280. Les annotateurs ont ajouté des annotations temporelles précises pour marquer les moments clés des accidents, tels que l'instant exact de l'impact. En complément, divers attributs environnementaux ont été étiquetés, comme les conditions de luminosité (jour/nuit) et les conditions météorologiques (neige, pluie, temps clair), ce qui rend ce dataset particulièrement adapté à l'analyse contextuelle des accidents (cf. Tableau5.2). Cependant, le dataset présente certaines limitations. Tout d'abord, il ne propose pas d'annotations au niveau des objets individuels (par exemple, voitures, piétons ou cyclistes). De plus, la fréquence d'images relativement basse (10 images par seconde) peut entraîner une qualité d'image insuffisante pour des analyses très détaillées ou des algorithmes nécessitant une résolution temporelle élevée. Les vidéos sont également sujettes à divers artefacts visuels, tels que des bandes noires sur les côtés ou du texte incrusté (comme des dates ou des horodatages), qui peuvent perturber l'entraînement ou l'évaluation des modèles (cf. Figure 5.21).

Dans notre cas, après extraction des vidéos, nous créons une base de données nommée également CarCrash, composée d'un total de 75,000 images, obtenues en échantillonnant les vidéos à leur fréquence native. Pour assurer l'uniformité avec nos autres ensembles de données, les images sont redimensionnées de leurs dimensions originales 720×1280 à 640×1280.

TABLE 5.2 – Informations sur les vidéos de CarCrash.

Luminosité	**Météo**	**Ego impliqué ?**
Jour : 1,325 Nuit : 175	Normale : 1,141 Pluie : 124 Neige : 235	Oui : 801 Non : 699



(a) (b)

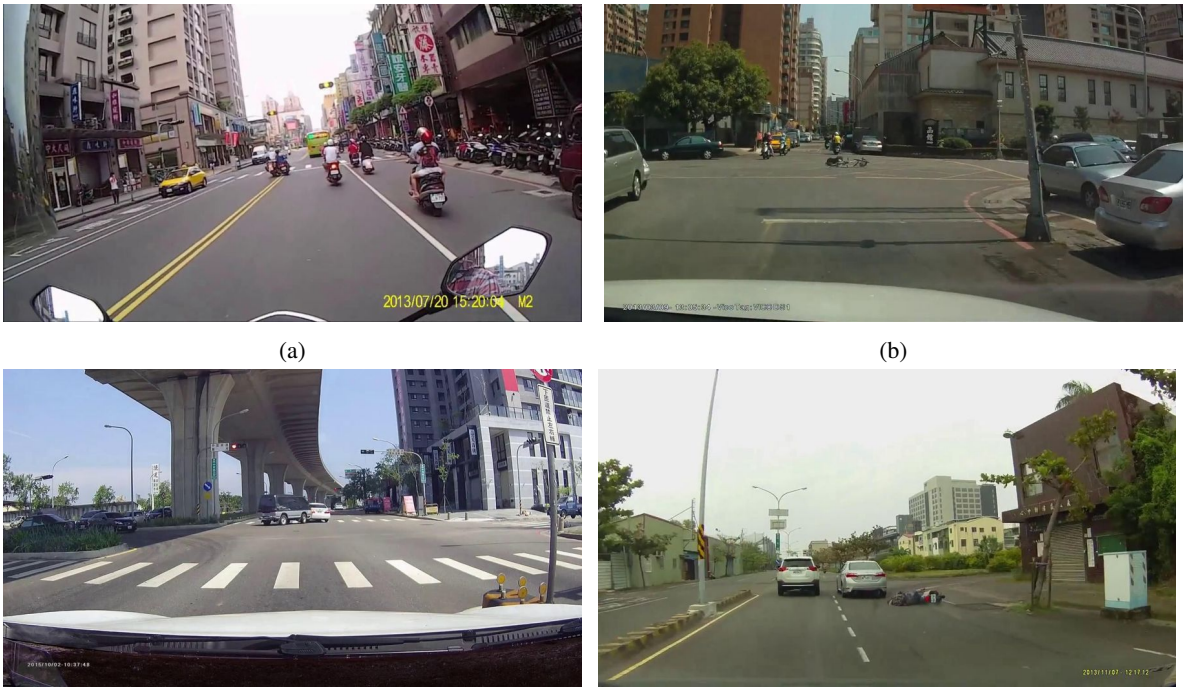
(c) (d)

FIGURE 5.21 – Exemples d'images CarCrash.

DashCam

D²-City [Che et al., 2019] est une vaste base de données de vidéos de conduite qui se concentre sur l'analyse des accidents de la route et des interactions routières. Elle comprend plus de 10,000 vidéos enregistrées à l'aide de caméras de tableau de bord, avec des dimensions d'image de 720×1280, des durées variant entre 3 et 6 secondes, et une fréquence d'image de 25 images par seconde. Parmi ces vidéos, environ 1,000 sont annotées avec des détections précises et des suivis d'objets routiers pour chaque image. Ces annotations incluent des boîtes englobantes ainsi que des identifiants de suivi pour une grande variété d'objets, tels que des voitures, camionnettes, bus, camions, piétons, motos, et vélos. Cette granularité permet des analyses détaillées des interactions entre différents types d'usagers de la route. Ce dataset se distingue par sa grande diversité. Les vidéos ont été collectées dans plusieurs grandes villes de Chine, offrant une variété impressionnante de conditions environnementales et routières. Ces données couvrent des environnements urbains et périurbains, des autoroutes, ainsi que des scénarios spécifiques comme des intersections complexes ou des zones de construction. En outre, les conditions météorologiques documentées incluent le ciel clair, la pluie, la neige, et des conditions nocturnes. La diversité de la densité de trafic est également notable, allant des routes vides aux embouteillages importants (cf. Figure 5.22).

Dans notre cas, pour des raisons de contraintes de temps et de mémoire, nous avons sélectionné uniquement 620 vidéos parmi l'ensemble disponible. Après extraction des images à partir des vidéos, nous avons créé une base de données nommée DashCam, contenant un total de 62,000 images. Ces images sont redimensionnées de leurs dimensions originales 720×1280 à 640×1280 pour assurer une cohérence avec nos autres ensembles de données.



(a) (b)

(c) (d)

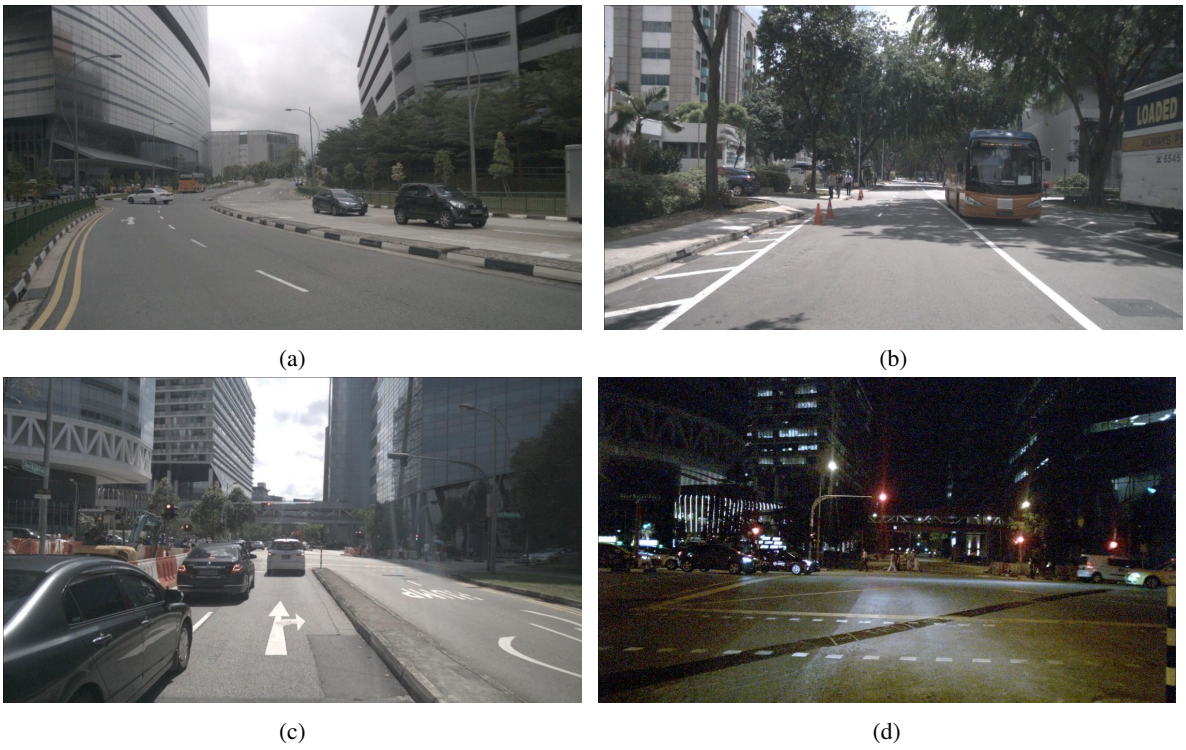
FIGURE 5.22 – Exemples d'images DashCam.

NuImages

La base de données NuImages est un sous-ensemble de la base de données nuScenes [Caesar et al., 2020], conçue pour les systèmes de perception des véhicules autonomes. Elle comprend 1,098,246 images capturées à partir de six caméras embarquées sur des véhicules opérant dans des environnements urbains variés, dont 214,519 proviennent de la caméra frontale, avec une résolution de 900×1600 (quelques exemples sont présentés dans la Figure 5.23). Les données ont été collectées dans deux villes emblématiques : Boston (Seaport et South Boston) et Singapour (One North, Holland Village et Queenstown), deux zones connues pour leur trafic dense et les situations de conduite hautement complexes qu'elles présentent. Parmi l'ensemble des images, 93,526, dont 18,376 issues de la caméra frontale, sont annotées en 2D avec des boîtes englobantes, des masques de segmentation pour divers objets, et des annotations pour la segmentation de scène. Ces annotations, extraites de 500 enregistrements de conduite, sont soigneusement sélectionnées : 75% des images ont été choisies via des techniques d'apprentissage actif pour capturer des classes rares comme les vélos, tandis que les 25% restantes ont été échantillonnées de manière uniforme afin de garantir une diversité représentative des conditions routières. La base couvre une large gamme de conditions météorologiques et d'éclairage, comprenant des scènes de pluie, de neige et de nuit. Elle se distingue également par la richesse et la précision de ses annotations, adaptées à des applications variées telles que la détection d'objets et la segmentation d'instances en milieu urbain. Les annotations incluent une grande variété d'objets liés à l'environnement routier, tels que les véhicules (voitures, camions, bus, motos), les cyclistes, les piétons, ainsi que des éléments d'infrastructure comme les panneaux de signalisation, feux de circulation, cônes de circulation, et autres mobiliers urbains. Chaque objet annoté est également enrichi d'attributs spécifiques, comme l'orientation et l'état dynamique (en mouvement ou à l'arrêt), offrant des informations supplémentaires utiles pour la compréhension des scènes routières.

Dans notre cas, nous avons créé une base de données nommée NuImages, composée de l'ensemble des images capturées, incluant celles provenant de la caméra frontale (214,519) et celles issues de toutes les caméras (1,098,246). Ensuite, nous avons constitué une autre base de données, nommée NuImagesLabels, qui contient toutes les images annotées. Cependant, 602 de ces images ont été exclues, car leurs annotations ne répondent pas à nos besoins, notamment en raison de la classe "remorque"/"trailer", qui est difficilement intégrable dans la classe "véhicule" (voir Figure 5.24). Après cette exclusion, NuImagesLabels comprend 17,774 images issues de la caméra frontale et 12,238 images annotées provenant de l'ensemble des caméras. Pour garantir une uniformité avec nos autres ensembles de données, nous redimensionnons les images de leur résolution originale 900×1600 à 640×1280. Par ailleurs, pour simplifier les annotations et les aligner avec notre problématique, nous avons

regroupé les classes de NuImagesLabels en trois grandes catégories : "véhicule", "piéton", et "cycliste".



(a) (b)

(c) (d)

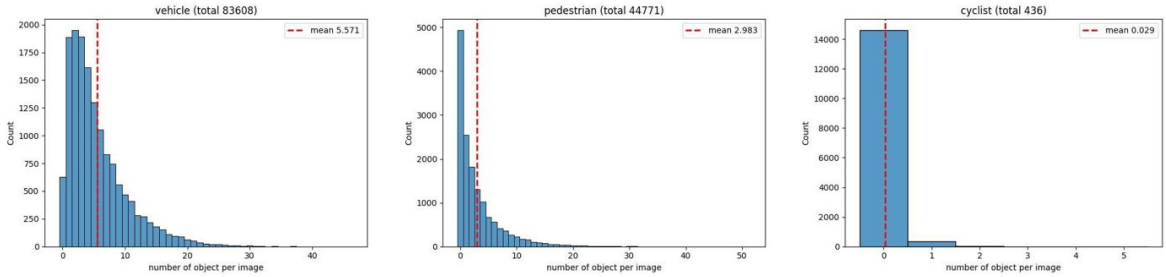
FIGURE 5.23 – Exemples d'images NuImages.



(a) Camion et remorque en deux classes distinctes. (b) Une remorque rattachée à aucun véhicule.

FIGURE 5.24 – Exemples de remorques dans NuImagesLabels.

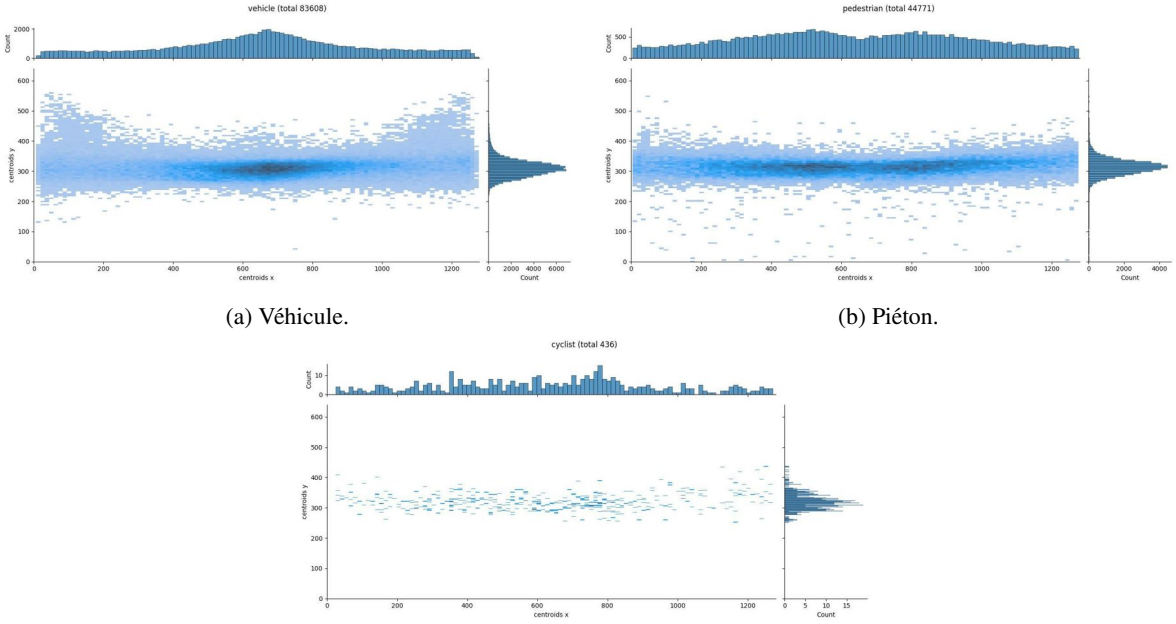
L'analyse statistique des objets dans la base de données NuImagesLabels issues de la caméra frontale révèle une répartition distincte des classes d'objets. Les images contiennent en moyenne environ 5.5 véhicules par image, ce qui en fait la classe la plus représentée, suivie des piétons avec environ 3 par image, tandis que les cyclistes sont rares, avec une occurrence moyenne de seulement 0.03 par image. Ces observations sont illustrées dans la Figure 5.25 et reflètent une tendance typique des environnements urbains où les véhicules et piétons sont majoritaires, mais où les cyclistes restent une classe rare.



(a) Véhicule. (b) Piéton. (c) Cycliste.

FIGURE 5.25 – Histogramme de la quantité d'objets par image sur NuImagesLabels(Front) pour les classes véhicule, piéton et cycliste.

En termes de distribution spatiale, les centroïdes des objets sont répartis de manière relativement homogène sur l'axe des x, traduisant une bonne couverture latérale des scènes. Cependant, sur l'axe des y, les objets se concentrent davantage au centre de l'image. Cette distribution est cohérente avec l'angle de vision et la perspective de la caméra frontale, qui favorisent une représentation plus dense des objets dans la zone centrale du champ de vision (cf. Figure 5.26).

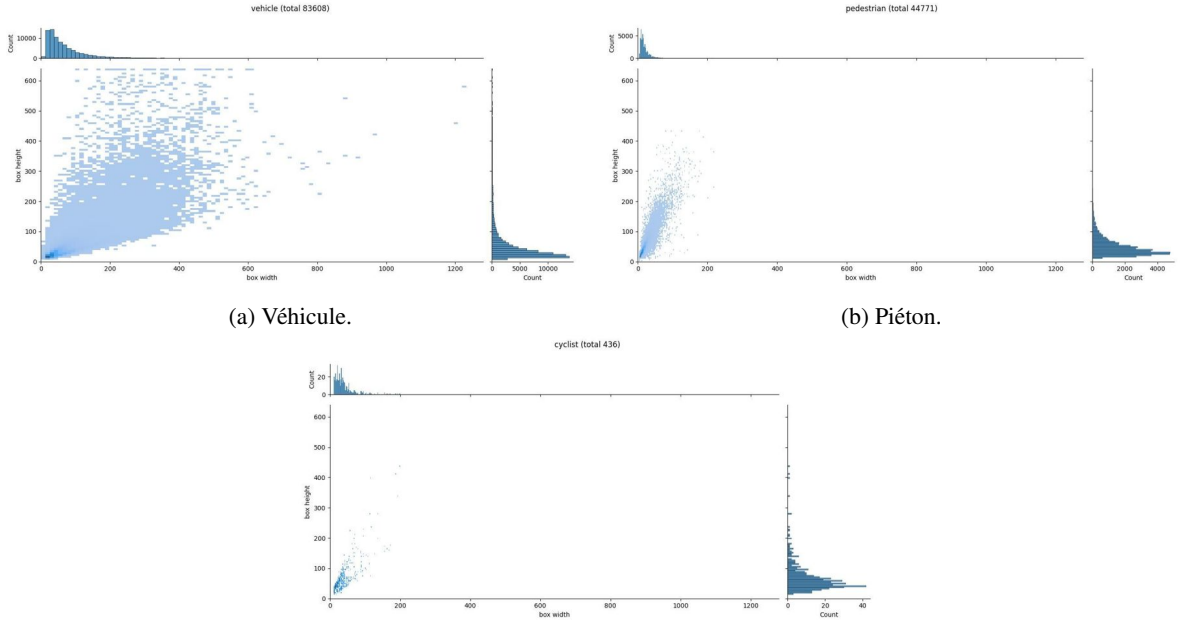


(a) Véhicule. (b) Piéton.

(c) Cycliste.

FIGURE 5.26 – Histogramme 2D de la distribution des centroïdes des boîtes dans les images sur NuImagesLabels(Front) pour les classes véhicule, piéton, cycliste.

Concernant la taille des boîtes englobantes, celles des véhicules montrent une plus grande variabilité, mais la majorité d'entre elles sont inférieures à 200×200 pixels. En revanche, les boîtes des piétons et des cyclistes sont nettement plus petites, avec la plupart des annotations inférieures à 100×100 pixels (cf. Figure 5.27).



(a) Véhicule.

(b) Piéton.

(c) Cycliste.

FIGURE 5.27 – Histogramme 2D de la distribution de la largeur et hauteur des boites dans les images sur NuImagesLabels(Front) pour les classes véhicule, piéton et cycliste.

Waterloo

Le jeu de données Canadian Adverse Driving Conditions (CADC) [Pitropov et al., 2021] a été collecté dans la région de Waterloo, au Canada (cf. Figure 5.28), et se concentre spécifiquement sur les conditions de conduite hivernales difficiles. Il a été capturé à l'aide de la plateforme de véhicule autonome Autonomoose, un projet commun entre le Toronto Robotics and AI Lab de l'Université de Toronto et le Waterloo Intelligent Systems Engineering Lab de l'Université de Waterloo, basé sur une Lincoln MKZ modifiée. Les données ont été enregistrées sur trois jours en mars 2018 et février 2019, couvrant 75 séquences de conduite réparties sur plus de 20 km. Ces séquences incluent divers niveaux de circulation et d'intensité de chutes de neige, offrant une large diversité de scénarios de conduite hivernale. En tout, le jeu de données contient 6,917 images annotées, issues de la caméra frontale, capturées dans une variété de conditions météorologiques hivernales telles que les routes enneigées, les brouillards de neige, ou les chaussées dégagées. Les données ont été acquises à l'aide de 8 caméras Ximea MQ013CG-E2 placées autour du véhicule, opérant à une fréquence de 10 Hz avec une résolution native de 1024×1280. Ces caméras ont été complétées par un capteur LiDAR (Velodyne VLP-32C) et un système de positionnement intégré GNSS+INS (Novatel OEM638), garantissant une synchronisation précise entre les données visuelles et les mesures LiDAR. Les annotations, fournies par Scale AI, servent de vérité terrain pour la détection et le suivi 3D d'objets tels que les voitures, piétons, camions, bus, cyclistes, et même des animaux comme les chevaux. Les objets annotés incluent des attributs détaillés tels que leur position, leur orientation, et leur vitesse relative. Pour un sous-ensemble des données, les étiquettes 3D sont disponibles et ont été échantillonnées toutes les 300 ms, représentant entre 50 et 100 images par trajet, soit environ 15 à 30 secondes de données par séquence. Ces annotations permettent des analyses détaillées et des tests rigoureux pour des algorithmes de détection et de suivi en 3D, dans des scénarios de conduite réalistes et contraignants.

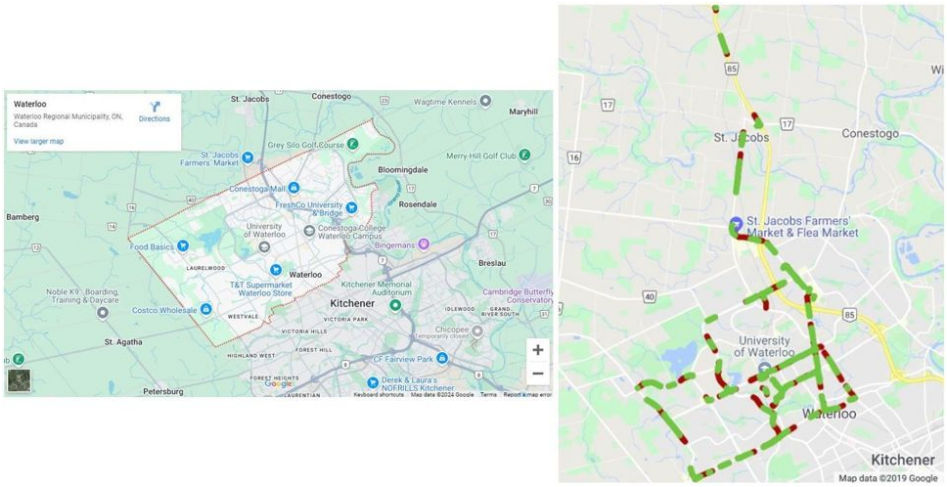
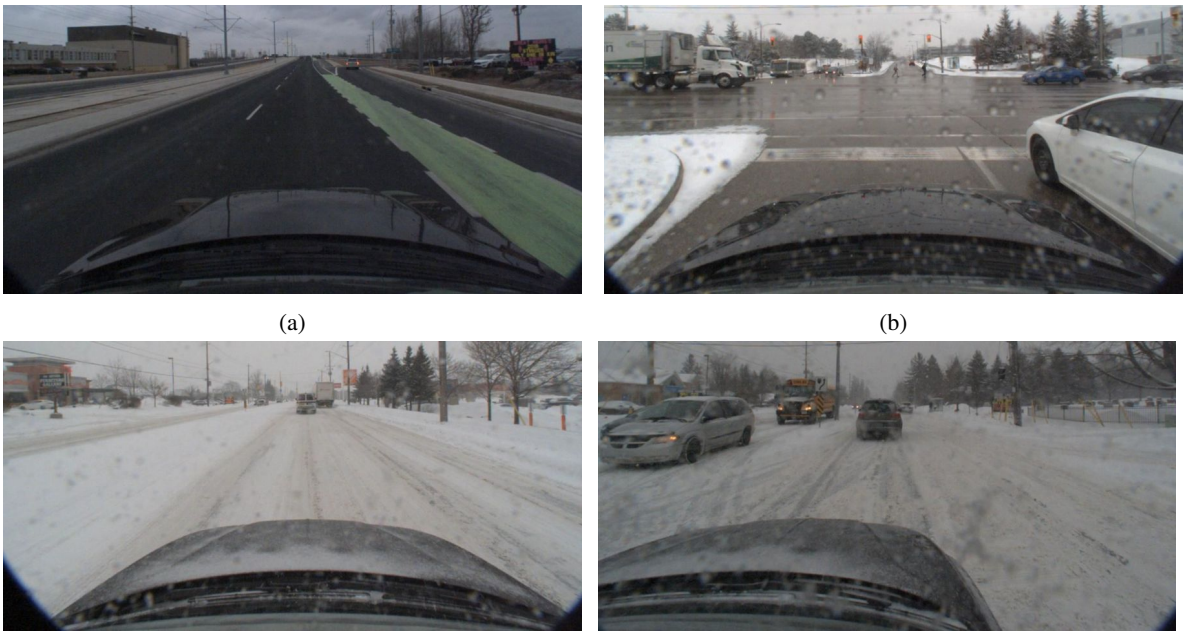


FIGURE 5.28 – Lieux géographique d'acquisition des données Waterloo.

Dans notre cas, nous avons créé une base de données nommée Waterloo, contenant l'ensemble des 6, 917 images issues de la caméra frontale (quelques exemples sont présentés dans la Figure 5.29) et celles issues de toutes les caméras (108, 775). Afin d'assurer une cohérence avec nos autres ensembles de données, nous avons redimensionné les images de leur résolution native 1024×1280 à 640×1280.



(a) (b)

(c) (d)

FIGURE 5.29 – Exemples d'images Waterloo.

PeSOTIF

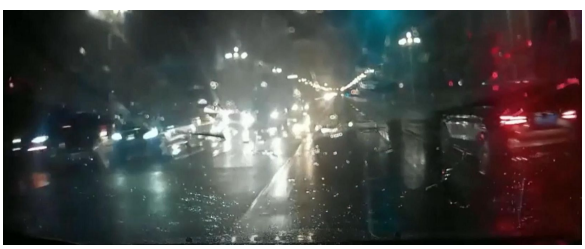
Le jeu de données PeSOTIF [Peng et al., 2023] (Perception Safety of the Intended Functionality) a été conçu pour relever les défis rencontrés par les algorithmes de perception dans les systèmes de conduite autonome, en particulier dans des scénarios rares et complexes. Ces situations, souvent qualifiées de scénarios "long-tail" ou, dans l'industrie automobile, de "cas limites" ("*edge and corner*"), représentent des événements peu fréquents mais critiques, où les systèmes autonomes sont susceptibles de rencontrer des limites fonctionnelles. Ces scénarios peuvent compromettre la sécurité de la fonctionnalité prévue (SOTIF), en raison des insuffisances des algorithmes, de l'incomplétude des données d'apprentissage, ou de la nature dynamique et imprévisible de l'environnement. Le jeu de données repose sur une analyse détaillée des conditions déclenchant des erreurs de

perception et est organisé en deux sous-ensembles principaux (cf. Tableau 5.3). Le premier, intitulé "environnement", regroupe des scénarios où des conditions environnementales altèrent les capacités de perception. Il est subdivisé en catégories telles que "pluie", "neige", "particules" et "illumination", avec une distinction entre des facteurs naturels (par exemple, la lumière directe du soleil) et artificiels (comme l'éblouissement provoqué par un phare en plein faisceau dans la catégorie "illumination"). Le second sous-ensemble, intitulé "objet", inclut des scénarios affectant la capacité de cognition. Les images sont réparties entre des situations dites "communes" (par exemple, des usagers de la route adoptant des postures ou ayant des apparences inhabituelles) et "peu communes" (comme des obstacles temporaires ou des objets perturbateurs tels que des débris ou des cônes de signalisation renversés). Ces catégories permettent de couvrir un large éventail de cas critiques tout en mettant en évidence des aspects rarement représentés dans d'autres bases de données. En moyenne, chaque image contient 2.27 objets clés et 2.47 mobjets normaux, mettant en avant une densité significative d'éléments d'intérêt pour les algorithmes de perception. Il est également important de noter que ces images ne proviennent pas exclusivement de caméras embarquées à l'avant d'un véhicule. Certaines ont été capturées à l'aide d'autres dispositifs, tels que des appareils photo ou des téléphones portables. Cette diversité des sources entraîne des variations significatives dans les résolutions des images et les perspectives sur les scènes et les objets. Cela pose un défi particulier pour l'intégration de ces images dans un ODD cohérent, en raison des différences de points de vue et des contextes souvent éloignés de la conduite autonome. Ce problème de compatibilité avec l'ODD est d'autant plus récurrent que ces types d'images sont rares, mais leur inclusion est cruciale pour tester la robustesse des algorithmes dans des cas limites.

TABLE 5.3 – Conditions de déclenchement partielles des capteurs visuels provenant de [Peng et al., 2023].

Libellé Principal	**Libellé Secondaire**	**Libellé Tertiaire**	**Sources de Déclenchement**
Environnement	Pluie	Naturelle	taches, traînées, brume, éclaboussures, flaques d'eau, etc.
		Fait à la main	bouches d'incendie, arroseurs, gouttes de pluie synthétiques, etc.
	Neige	Naturelle	taches, flocons, brume, couverture de neige, etc.
		Fait à la main	chasse-neige, flocons de neige synthétiques, etc.
	Particules	Naturelle	diffusion, brume, effet Tyndall, etc.
		Fait à la main	émissions de véhicules, sites d'accidents, poussière de chantier, etc.
	Illumination	Naturelle	lumière directe du soleil, reflets, ombres, etc.
		Fait à la main	phares en plein faisceau, températures de couleur inhabituelles, etc.
Objet	Commun	Apparence	chevauchements, occlusions, graffiti, voitures personnalisées, etc.
		Posture	voitures renversées, personnes allongées au sol, cascades, etc.
	Peu Commun	-	animaux, glissements de terrain, objets tombants, etc.

Dans le cadre de cette thèse, nous avons créé une base de données nommée également PeSOTIF, contenant l'intégralité des 1, 126 images disponibles (quelques exemples sont présentés dans la Figure 5.30). Pour assurer une cohérence avec nos autres ensembles de données, toutes les images ont été redimensionnées à une résolution standardisée de 640×1280. Ce redimensionnement garantit une intégration fluide dans nos pipelines de traitement tout en maintenant une qualité suffisante pour les analyses et tests ultérieurs.



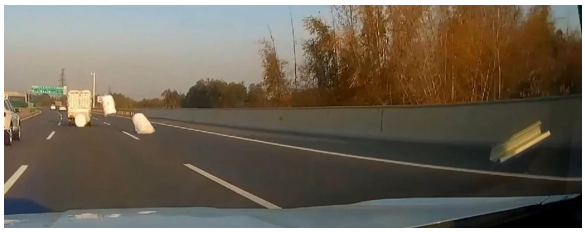
(a) Illumination.



(b) Illumination.



(c) Objet avec posture inhabituelle.



(d) Objet avec posture inhabituelle.

FIGURE 5.30 – Exemples d'image PeSOTIF

Edge & Corner (E&C)

Nous avons constitué une base de données à partir de 28 vidéos issues de YouTube, illustrant des situations routières extrêmes. Ces vidéos couvrent une variété de scénarios rares et critiques, tels que des piétons et cyclistes chutant, des collisions entre véhicules à quatre et deux roues, des véhicules renversés ou en feu, des personnes en fauteuil roulant, ainsi que des conditions météorologiques intenses, comme de fortes pluies ou des chutes de neige. Ces situations, souvent qualifiées de "cas limites" ou de "scénarios long-tail", offrent une représentation précieuse de cas rarement capturés dans des bases de données classiques. Après extraction des vidéos, nous avons créé une base de données nommée Edge & Corner, que nous abrégerons E&C, en sélectionnant uniquement des images correspondant aux scénarios extrêmes mentionnés ci-dessus. Cette sélection a permis de constituer un ensemble de $1,365$ images, initialement de dimensions 720×1280, que nous avons redimensionnées à 640×1280 pour garantir une cohérence avec nos autres ensembles de données. Les images ont ensuite été annotées manuellement avec les classes "véhicule", "piéton" et "cycliste", à l'aide du logiciel open source Label Studio [3]. Le processus d'annotation a présenté des défis significatifs, en particulier en raison de la visibilité limitée des objets dans certains cas extrêmes, comme les véhicules enfumés ou les piétons partiellement obscurcis. Par ailleurs, certaines situations ont soulevé des ambiguïtés sémantiques, notamment dans le cas des cyclistes chutant. Par définition, un cycliste est considéré comme un vélo accompagné de son conducteur. Cependant, lorsque le conducteur tombe et que les deux objets se séparent, le conducteur est alors classifié comme un piéton, tandis que le vélo devient un objet inanimé et n'est plus inclus dans les classes principales. Cette même logique a été appliquée aux motos dans des situations similaires. Ces ajustements conceptuels soulignent l'importance d'une définition claire et cohérente des classes dans le cadre d'annotations pour des cas extrêmes. Similaire à la base PeSOTIF, les images de E&C ne proviennent pas exclusivement de caméras embarquées à l'avant d'un véhicule. Certaines ont été capturées avec d'autres dispositifs, tels que des appareils photo ou des téléphones portables. Bien que la résolution des images soit constante après redimensionnement, les points de vue varient considérablement, offrant une diversité de perspectives sur les scènes et les objets. Cette diversité sera explorée dans l'analyse statistique ci-dessous et soulève des questions quant à l'appartenance de ces images à l'ODD défini. Ces considérations seront discutées en détail pour évaluer la pertinence des images dans le cadre de notre étude.

3. https://labelstud.io/

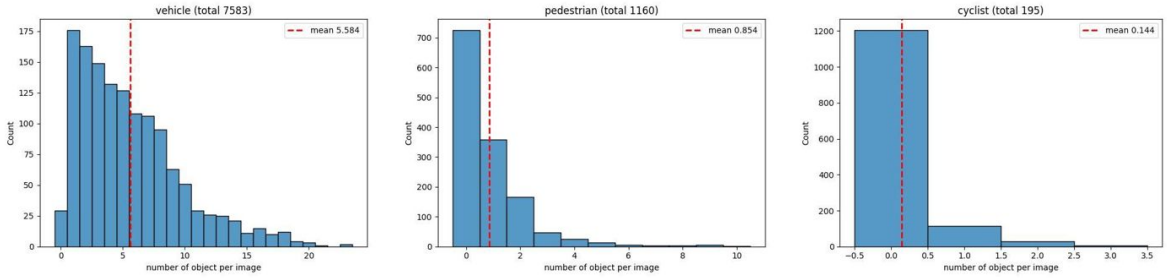


(a) (b)

(c) (d)

FIGURE 5.31 – Exemples d'image E&C

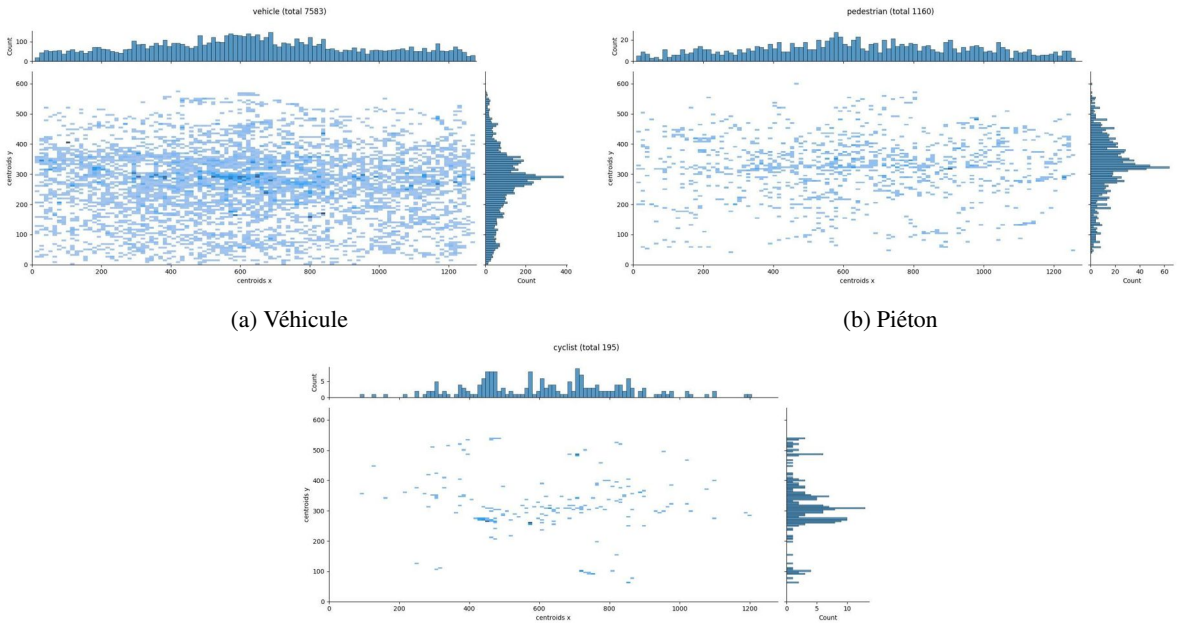
Comme pour les bases de données précédentes, l'analyse statistique des objets dans la base de données E&C révèle que les images contiennent en moyenne davantage de véhicules (5.6 par image) que de piétons (0.8 par image) et très peu de cyclistes (0.1 par image), comme illustré dans la Figure 5.32. Cette distribution reflète la nature des scènes capturées, qui incluent majoritairement des environnements où les véhicules sont prédominants, mais avec des occurrences occasionnelles de piétons et cyclistes dans des situations extrêmes.



(a) Véhicule (b) Piéton (c) Cycliste

FIGURE 5.32 – Histogramme de la quantité d'objets par image sur E&C pour les classes véhicule, piéton et cycliste.

Une particularité de cette base de données réside dans la dispersion des objets d'intérêt sur l'axe vertical, comme le montre la Figure 5.33. Contrairement aux bases précédentes, où les objets sont souvent concentrés au centre de l'image, les objets dans E&C sont plus variés en termes de position verticale, témoignant d'une plus grande diversité de perspectives et de cadrages. Cette caractéristique est attribuable à l'utilisation de multiples types de caméras, comme des appareils photo et téléphones portables, qui introduisent des angles de vue variés, souvent décalés par rapport à une caméra frontale standard.



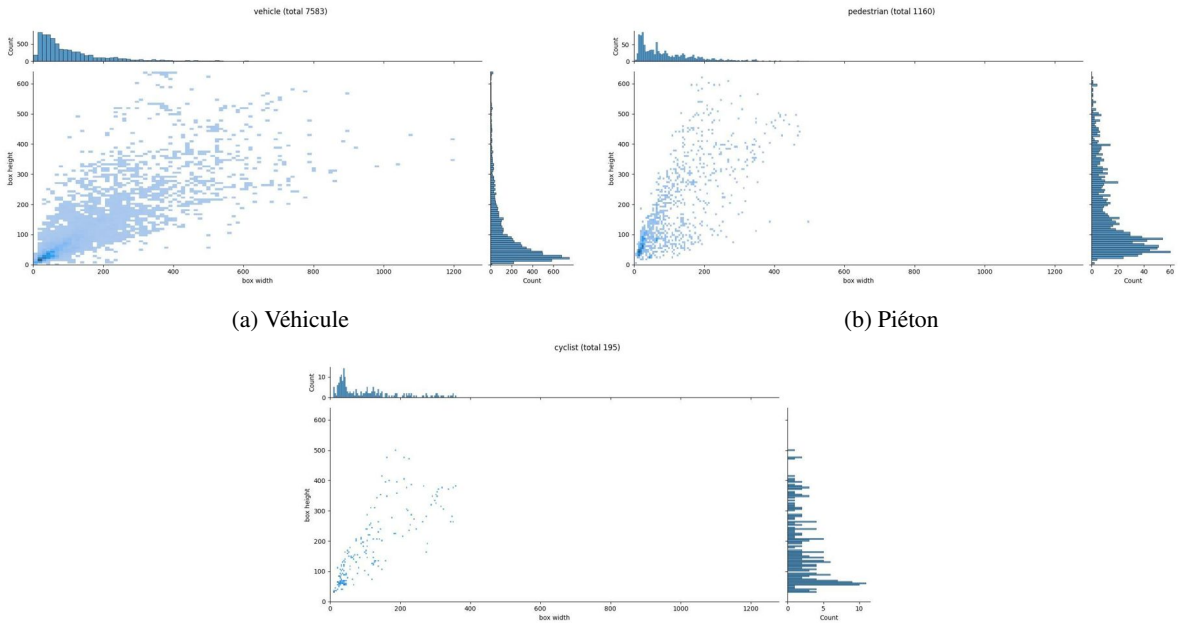
(a) Véhicule

(b) Piéton

(c) Cycliste

FIGURE 5.33 – Histogramme 2D de la distribution des centroïdes des boîtes dans les images sur E&C pour les classes véhicule, piéton, cycliste.

Enfin, la taille des boîtes englobantes des objets montre également une plus grande variabilité par rapport aux autres bases de données (cf. Figure 5.34). Les véhicules mesurent généralement moins de 200×200 pixels, bien qu'une fraction atteigne des dimensions plus importantes dans certaines scènes. Les piétons présentent une gamme de tailles plus étendue, allant jusqu'à 500×200 pixels, tandis que les cyclistes, bien que rares, peuvent mesurer jusqu'à 400×200. Cette variabilité accrue reflète des distances variables entre les objets et la caméra, ainsi que des différences dans les perspectives. Notamment, la taille plus importante des boîtes englobantes pour les piétons et les cyclistes indique que ces derniers sont souvent plus proches de la caméra, ce qui correspond aux scénarios de collisions ou de chutes capturés dans les vidéos sources.



(a) Véhicule

(b) Piéton

(c) Cycliste

FIGURE 5.34 – Histogramme 2D de la distribution de la largeur et hauteur des boites dans les images sur E&C pour les classes véhicule, piéton et cycliste.

En conclusion, pour entraîner, évaluer et estimer des garanties de performance sur notre chaîne de compression-détection, nous avons constitué et exploité plusieurs jeux de données naturelles. Ces bases de données offrent une diversité précieuse en termes de conditions environnementales, de perspectives et de classes d'objets annotés. Cependant, elles restent insuffisantes pour couvrir l'ensemble des scènes et scénarios définis par notre ODD. Certaines situations critiques ou spécifiques, notamment les scénarios dits *Edge and Corner*, sont peu ou pas représentées dans ces données, limitant ainsi leur pertinence pour des tests exhaustifs. De plus, les jeux de données existants pour les scénarios critiques présentent plusieurs lacunes. D'une part, leur quantité est souvent trop faible pour permettre un entraînement ou une évaluation statistiquement significative. D'autre part, ils ne respectent pas toujours strictement les conditions définies par notre ODD, introduisant ainsi des biais potentiels. Enfin, les annotations associées à ces données posent des défis importants, notamment en raison de la faible visibilité des objets dans les scènes extrêmes et de l'incertitude concernant leur présence ou leur position exacte. Ces limitations rendent l'acquisition de données simulées incontournable. La simulation permet de compléter ces jeux de données en générant des scénarios spécifiques et contrôlés, correspondant aux besoins définis par notre ODD.

Dans la suite de cette étude, nous explorerons en détail comment tirer parti de la simulation pour enrichir nos bases de données, combler ces lacunes, et garantir une couverture des conditions nécessaires à l'évaluation de notre système.

5.3.2 Données Simulées

L'utilisation de données synthétiques est devenue une pratique essentielle pour le développement et l'évaluation des modèles d'intelligence artificielle (IA), notamment dans des contextes où les données réelles sont insuffisantes, coûteuses ou difficiles à obtenir. Ces données permettent de couvrir des scénarios rares ou critiques, et d'enrichir les ensembles d'entraînement et de validation pour améliorer les performances des modèles. Nous distinguons deux types principaux d'images synthétiques :

— Image semi-synthétique : construite à partir d'une image réelle puis modifiée par un ordinateur à l'aide de diverses méthodes telles que l'Inpainting [Leclerc et al., 2022, Zhang et al., 2020b] ou la méthode de transfert de domaine [Richter et al., 2023].

— Image complètement synthétique : construite à partir d'un monde virtuel [Strohmayer and Kampel, 2021, Fabbri et al., 2021, Erfanian Ebadi et al., 2022, Ebadi et al., 2022] ou par un modèle de deep learning dit génératifs comme les modèles GAN [Goodfellow et al., 2014, Lee et al.,] ou encore les modèles de Diffusion [Gao et al., 2024, Wang et al., 2024c, Hu et al., 2023]

Pour des tâches telles que la détection d'objets ou la segmentation sémantique, les données synthétiques servent à deux fins principales : améliorer les performances de modèles et évaluer leurs faiblesses. L'intégration de données synthétiques avec des données réelles dans l'ensemble d'entraînement [Strohmayer and Kampel, 2021, Yilmaz et al., 2023, Natarajan and Madden, 2023, Park et al., 2020, Erfanian Ebadi et al., 2022, Ebadi et al., 2022, Alvey et al., 2021, Prakash et al., 2019, Borkman et al., 2021] peut augmenter de manière significative la précision moyenne de modèles jusqu'à 32%, comme le montre [Westerski and Teck, 2023]. Une technique efficace est la randomisation de domaine, dans laquelle les paramètres du simulateur tels que l'éclairage, la pose ou la texture des objets sont variés de manière irréaliste. Avec un affinage supplémentaire sur des données réelles, cette approche dépasse souvent l'utilisation de données réelles seules [Westerski and Teck, 2023, Tremblay et al., 2018]. Cependant, il reste difficile de se fier uniquement aux données synthétiques pour l'entraînement en raison de l'écart de domaine entre les données réelles et synthétiques, fortement influencé par la qualité des images générées. Dans le cadre d'un processus de validation, on peut également générer de nouvelles images synthétiques spécifiques pour tester davantage un modèle et trouver ses défauts, comme dans [Leclerc et al., 2022, Hu et al., 2024].

Pour répondre au problème du manque de données, en particulier pour les scénarios rares et critiques, une solution consiste à générer des données synthétiques via des simulations. Ces données permettent de compléter les jeux de données réels en créant des scénarios ciblés qui respectent les conditions définies par l'ODD. Dans notre cas, les données à générer sont des images RGB de scènes de conduite, notamment des scénarios *Edge and Corner*, capturées par une caméra embarquée à l'avant d'un véhicule autonome. Ces données doivent représenter des événements variés tels que des collisions, des chutes de piétons ou de cyclistes, ou encore des conditions météorologiques extrêmes. Pour cela, il s'agira de créer une simulation d'un environnement 3D comprenant un véhicule équipé d'une caméra frontale, naviguant dans un environnement routier modélisé où différents événements peuvent être simulés. Ces environnements peuvent être construits sur mesure, en intégrant des éléments tels que des routes, des infrastructures urbaines, des véhicules, et des usagers de la route, ainsi que des conditions météorologiques spécifiques (neige, pluie, faible éclairage). Cette flexibilité permet de générer des scènes variées qui seraient difficiles, voire impossibles, à capturer dans des conditions réelles.

En collaboration avec l'entreprise Unity Technologies [4], nous avons développé une application permettant de simuler un véhicule équipé d'une caméra embarquée. Cette application génère des images RGB accompagnées de fichiers d'annotations contenant les boîtes englobantes 2D des objets d'intérêt, tout en reproduisant des parcours dans un environnement urbain. Elle permet également de simuler des événements rares, spécialement conçus pour répondre à des besoins spécifiques, tels que des situations difficiles ou coûteuses à recréer dans des conditions réelles. L'application offre une solution complète pour enrichir les ensembles de données en générant des scènes variées et contrôlées. Une fois lancée, elle débute par une étape de configuration où les paramètres contextuels de la simulation sont définis (cf. Figure 5.35). Ces paramètres incluent la luminosité, les conditions météorologiques (comme la pluie ou le brouillard), ainsi que l'état du pare-brise. Cette flexibilité permet de moduler l'environnement pour couvrir une large gamme de scénarios, y compris ceux correspondant à des cas limites ou des situations critiques rarement observées dans les données réelles.

4. Unity Software Inc. est une société de développement de logiciels de jeux vidéo basée à San Francisco, https ://unity.com/fr



FIGURE 5.35 – Écran de démarrage et paramétrisation de la simulation.

Une fois les paramètres de contexte définis et la simulation démarrée, l'application génère des scénarios d'une durée de 11 minutes à une fréquence de 24 images par seconde. Ces scénarios se déroulent dans un environnement urbain et incluent une série d'événements, programmés pour se produire dans l'ordre suivant :

- Événement 1 (cf. Figure 5.36a) : Le véhicule s'arrête à un feu rouge, tandis que des piétons traversent aléatoirement sur le passage pour piétons. L'un d'entre eux tombe, se relève, et termine sa traversée.
- Événement 2 (cf. Figure 5.36b) : Le véhicule s'arrête à un feu rouge. Des piétons, dont certains en fauteuil roulant, traversent le passage pour piétons.
- Événement 3 (cf. Figure 5.36c) : Le véhicule s'arrête (hors passage pour piétons) pour laisser passer des animaux (chiens, chats, oiseaux) et des piétons, dont certains en fauteuil roulant. Un piéton tombe, se relève, et termine sa traversée.
- Événement 4 (cf. Figure 5.36d) : À une intersection avec feu rouge, le véhicule s'arrête. Plusieurs véhicules et cyclistes traversent, et la route devant est fermée en raison de travaux.
- Événement 5 (cf. Figure 5.36e) : Le véhicule s'arrête à une intersection avec feu rouge. Des véhicules et cyclistes traversent, tandis que la route devant est bloquée par un accident impliquant une voiture retournée.
- Événement 6 (cf. Figure 5.36f) : Le véhicule s'arrête à une intersection avec feu rouge. Un cycliste est heurté par un véhicule et tombe. Une ambulance arrive pour secourir le cycliste.
- Événement 7 (cf. Figure 5.36g) : Le véhicule s'arrête à une intersection pour laisser traverser des animaux (chiens, chats, oiseaux) et des piétons.
- Événement 8 (cf. Figure 5.36h) : Sur une rue à trois voies sans feu, le véhicule s'arrête lorsque deux véhicules circulant en sens opposés entrent en collision, l'un d'eux se renversant sur le côté.
- Événement 9 (cf. Figure 5.36i) : À une intersection avec feu rouge, le véhicule s'arrête. Des véhicules et cyclistes traversent, tandis qu'une voiture en feu bloque la route devant.



(a) Évènement 1 (day, clear sky, dry, no wipping, pigeon droppings).



(b) Évènement 2 (day, cloudy sky, raining, wipping, wiper markings).



(c) Évènement 3 (day, cloudy sky, snowing, wipping, wiper markings).



(d) Évènement 4 (day, cloudy sky, fog, no wipping, wiper markings).



(e) Évènement 5 (sunset/sunrise, clear sky, dry, no wipping, pigeon droppings).



(f) Évènement 6 (sunset/sunrise, cloudy sky, raining, wipping, wiper markings).



(g) Évènement 7 (sunset/sunrise, cloudy sky, fog, no wipping, pigeon droppings).



(h) Évènement 8 (night, clear sky, dry, no wipping, wiper markings).



(i) Évènement 9 (night, clear sky, dry, no wipping, pigeon droppings).

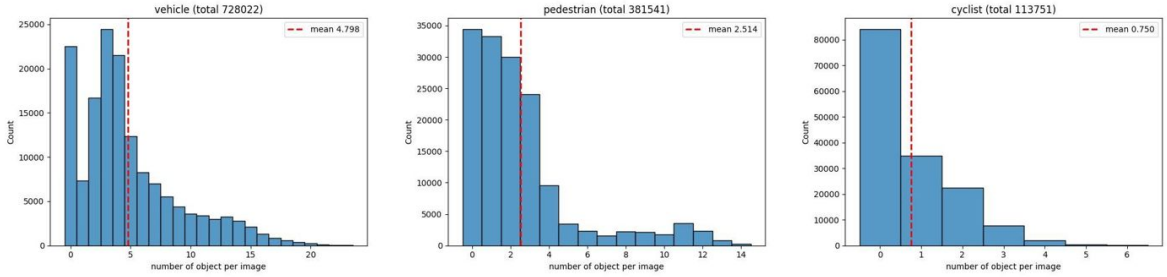
FIGURE 5.36 – Visualisation des évènements avec des variations des paramètres de contexte (cf. Figure 5.35).

Enfin, grâce à notre simulateur, nous avons généré un total de 24 scénarios correspondant à 24 combinaisons distinctes de paramètres de contexte (cf. Tableau 5.4). Chaque scénario a donné lieu à une vidéo d'environ 11 minutes. En échantillonnant à une fréquence de 12 images par seconde, chaque vidéo produit 6, 511 images, soit un total de 156, 264 images pour l'ensemble des scénarios. Cette base de données sera désormais désignée sous le nom Unity.

TABLE 5.4 – Combinaisons de paramètres de contexte par scénarios (cf. Figure 5.35).

n°	Time of Day	Weather		Windshield	
		Sky	State	Wipping	State
1	day	clear	dry	no	wiper markings
2	day	clear	dry	no	pigeon droppings
3	day	cloudy	raining	yes	wiper markings
4	day	cloudy	raining	yes	pigeon droppings
5	day	cloudy	snowing	yes	wiper markings
6	day	cloudy	snowing	yes	pigeon droppings
7	day	cloudy	fog	no	wiper markings
8	day	cloudy	fog	no	pigeon droppings
9	sunset/sunrise	clear	dry	no	wiper markings
10	sunset/sunrise	clear	dry	no	pigeon droppings
11	sunset/sunrise	cloudy	raining	yes	wiper markings
12	sunset/sunrise	cloudy	raining	yes	pigeon droppings
13	sunset/sunrise	cloudy	snowing	yes	wiper markings
14	sunset/sunrise	cloudy	snowing	yes	pigeon droppings
15	sunset/sunrise	cloudy	fog	no	wiper markings
16	sunset/sunrise	cloudy	fog	no	pigeon droppings
17	night	clear	dry	no	wiper markings
18	night	clear	dry	no	pigeon droppings
19	night	cloudy	raining	yes	wiper markings
20	night	cloudy	raining	yes	pigeon droppings
21	night	cloudy	snowing	yes	wiper markings
22	night	cloudy	snowing	yes	pigeon droppings
23	night	cloudy	fog	no	wiper markings
24	night	cloudy	fog	no	pigeon droppings

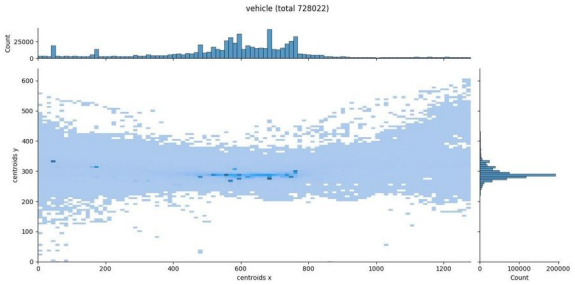
L'analyse statistique des objets des données synthétiques révèle des tendances distinctes dans la répartition et les caractéristiques des annotations selon les classes. En moyenne, les images contiennent environ 4.8 véhicules par image, ce qui en fait la classe la plus représentée, suivie des piétons avec 2.5 objets par image. Les cyclistes, bien que présents, sont significativement moins fréquents, avec une occurrence moyenne de 0.75 par image (cf. Figure 5.37). Cette distribution reflète une dynamique typique des environnements urbains simulés, où les véhicules dominent les scènes, tandis que les cyclistes restent une classe plus rare.



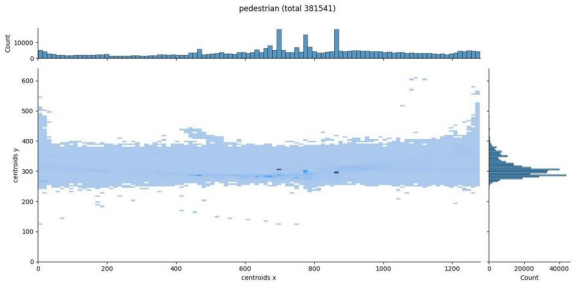
(a) Véhicule. (b) Piéton. (c) Cycliste.

FIGURE 5.37 – Histogramme de la quantité d'objets par image des données synthétiques pour les classes véhicule, piéton et cycliste.

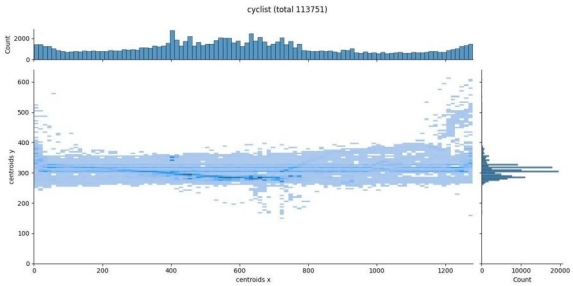
En ce qui concerne la répartition spatiale, les centroïdes des objets montrent une distribution relativement homogène le long de l'axe des x, traduisant une bonne couverture horizontale des scènes. Sur l'axe des y, cependant, les objets se concentrent davantage dans la zone centrale de l'image, ce qui correspond à l'orientation de la caméra frontale et son angle de vision focalisé (cf. Figure 5.38).



(a) Véhicule.



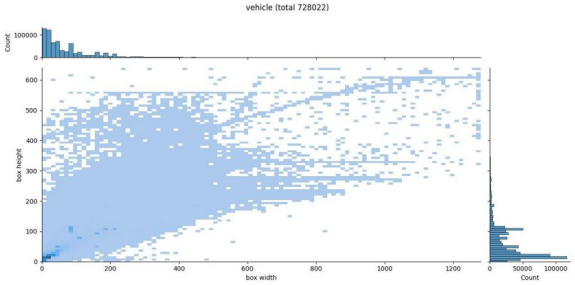
(b) Piéton.



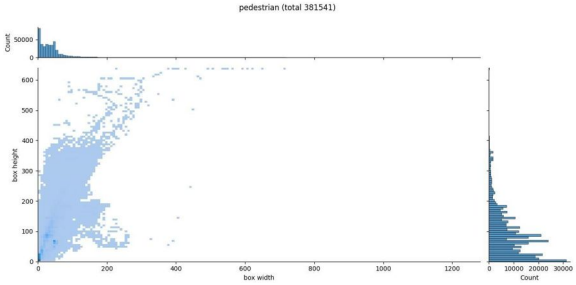
(c) Cycliste.

FIGURE 5.38 – Histogramme 2D de la distribution des centroïdes des boîtes dans les données synthétiques pour les classes véhicule, piéton, cycliste.

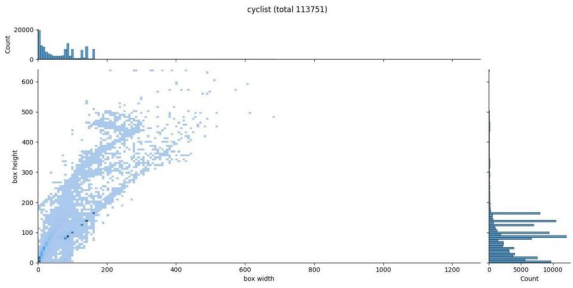
Enfin, pour les tailles des boîtes englobantes, les annotations des véhicules affichent une variabilité plus importante, bien que la majorité des boîtes soient inférieures à 200×200 pixels. Les annotations des piétons et des cyclistes, en revanche, sont généralement plus petites, avec la plupart des boîtes englobantes mesurant moins de 100×100 pixels (cf. Figure 5.39).



(a) Véhicule.



(b) Piéton.



(c) Cycliste.

FIGURE 5.39 – Histogramme 2D de la distribution de la largeur et hauteur des boites dans les données synthétiques pour les classes véhicule, piéton et cycliste.

5.3.3 Étude sur l'utilisation de données simulées

À partir du simulateur présenté précédemment, nous avons pu générer une grande quantité de données, incluant des situations rares et critiques. Cependant, une question essentielle se pose : ces données synthétiques peuvent-elles remplacer les données naturelles ou réelles ? En effet, les données réelles et synthétiques appartiennent à des "domaines" différents, une différence souvent désignée par le terme "domain gap" en anglais. Cette distinction soulève plusieurs interrogations : peut-on utiliser les données synthétiques de la même manière que les données réelles ? Ces données générées sont-elles suffisamment précises et réalistes pour servir de substitut aux données réelles ? Dans cette section, nous examinerons ces questions en montrant que, bien que les données synthétiques ne soient pas suffisantes seules pour remplacer les données réelles, elles constituent un complément précieux pour des tâches d'entraînement et d'évaluation.

Pour étudier les performances des systèmes de détection d'objets en explorant l'écart entre les données réelles et synthétiques, nous avons constitué deux ensembles distincts de datasets. Le premier ensemble, nommé D, est construit à partir des données naturelles issues des jeux de données Waymo (5.3.1) et BDD100K (5.3.1). Cet ensemble est partitionné en trois sous-ensembles : D_{train}, D_{valid} et D_{test}, dont les volumes respectifs sont présentés dans le Tableau 5.5. De manière similaire, le second ensemble, nommé U, repose sur les données synthétiques issues de la base Unity. Celui-ci est également partitionné en trois sous-ensembles : U_{train}, U_{valid} et U_{test}, avec des volumes détaillés dans le même tableau. Afin d'explorer l'écart entre les domaines des données réelles et synthétiques, deux détecteurs d'objets YOLOv7 [Wang et al., 2023a] ont été entraînés. Le premier détecteur a été exclusivement entraîné sur les données naturelles (D), tandis que le second a été formé uniquement sur les données synthétiques (U). Cette démarche permet non seulement d'évaluer les performances respectives de chaque type de données, mais aussi d'analyser la complémentarité et les limitations des données synthétiques dans des scénarios pratiques.

TABLE 5.5 – Datasets naturelles et synthétiques simulés.

Nom	**Partition**	**Bases de données**	**Volumétrie (#images)**	
			Par base	Total
D	*train*	Waymo	124, 668	180, 558
		BDD100K	55, 890	
	valid	Waymo	31, 168	45, 141
		BDD100K	13, 973	
	test	Waymo	39, 987	49, 987
		BDD100K	10, 000	
U	*train*	Unity	116, 264	116, 264
	valid	Unity	30, 000	30, 000
	test	Unity	10, 000	10, 000

Les résultats présentés dans le Tableau 5.6 montrent que, lorsque YOLOv7 est entraîné sur des données synthétiques U_{train}, il atteint une performance quasi parfaite sur le jeu de données synthétique U_{test} (mAP@.5= 0.98). Cependant, ses performances chutent significativement lorsqu'il est testé sur des données réelles D_{test} (mAP@.5= 0.58). Ces observations mettent en évidence que YOLOv7 apprend des caractéristiques spécifiques aux ensembles de données sur lesquels il est entraîné, et que ces caractéristiques diffèrent en raison de l'écart de domaine entre les données synthétiques et réelles. Ce phénomène est cohérent avec l'état de l'art, où un tel écart est fréquemment observé (cf. [Westerski and Teck, 2023], Tableau 4, colonne SYNTH DIFF), montrant une perte de performance notable lorsqu'un modèle est entraîné exclusivement sur des données synthétiques. Des travaux comme [Fabbri et al., 2021] suggèrent que la clé pour réduire cet écart repose sur la diversité des données, tandis que d'autres, comme [Prakash et al., 2019], montrent des approches permettant de combler l'écart dans certains cas spécifiques (BDD100K [Yu et al., 2020]), mais pas toujours, notamment sur des jeux de données comme KITTI [Geiger et al., 2012]. Dans notre cas, l'écart de domaine observé confirme que les données synthétiques, bien qu'utiles, ne peuvent pas entièrement remplacer les données réelles. En revanche, lorsque YOLOv7 est entraîné sur des données réelles D_{train}, il atteint des performances similaires sur les ensembles de test réels D_{test} (mAP@.5= 0.788) et synthétiques U_{test} (mAP@.5= 0.763). Ces résultats suggèrent que si le modèle parvient à détecter un certain type d'objet dans les images réelles, il est également capable de détecter ce même type d'objet dans les images synthétiques. Cela indique que les données synthétiques peuvent être utilisées efficacement pour des tâches de vérification et pour identifier les lacunes éventuelles du modèle, tout en nécessitant un entraînement

initial sur des données réelles pour garantir une généralisation adéquate.

TABLE 5.6 – Estimation par bootstrap, avec $B = 100,000$ et avec un intervalle de confiance à 95%, du mAP@.5 de YOLOv7 entrainé et testé sur les datasets réel D et synthétique U.

Train \ Test	**Dataset réel D_{est}**	**Dataset synthétique U_{test}**
Dataset réel D_{train}	0.7883 ± 0.0046	0.7630 ± 0.0044
Dataset synthétique U_{train}	0.5805 ± 0.0081	0.9871 ± 0.0022

Dans la Figure 5.40, une analyse macro est réalisée pour évaluer l'impact des différents paramètres de contexte sur les performances de détection, en testant le modèle YOLOv7, entraîné sur le jeu de données réelles D (courbes en bleu), sur l'ensemble des 24 scénarios synthétiques décrits dans le Tableau 5.4. En comparant les conditions d'éclairage (day, sunset/sunrise et night), le mAP@.5 suit un schéma similaire, mais sunset/sunrise semble être la condition qui affecte le plus négativement les performances moyennes de détection (indiquées par les lignes horizontales violettes). Ensuite, en différenciant les scénarios pairs et impairs, il est évident que les wiper markings (présents dans les scénarios impairs) réduisent davantage le mAP@.5 que les pigeon droppings (scénarios pairs). Enfin, en examinant l'impact des conditions météorologiques, les performances sont maximales sous un clear sky et dry conditions, mais diminuent légèrement avec snowing. En revanche, raining et fog combinés avec wiper markings entraînent une dégradation significative des performances du détecteur. Ces observations indiquent que pour améliorer YOLOv7, il serait pertinent d'élargir l'entraînement avec des images supplémentaires capturées dans des conditions pluvieuses et brumeuses, où les performances sont les plus impactées.

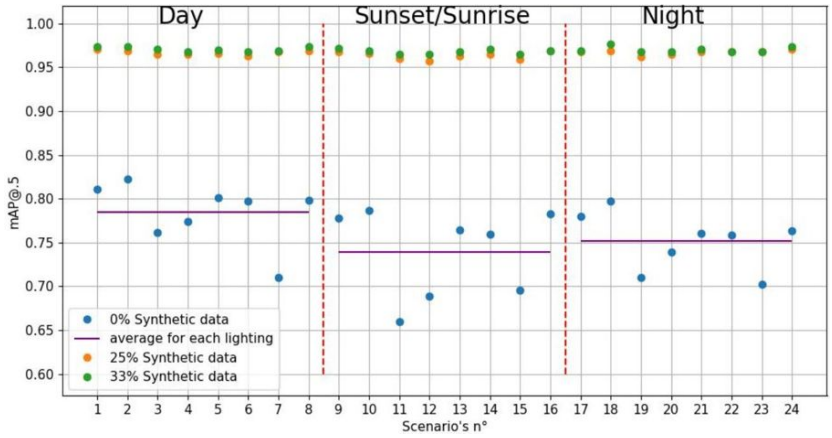


FIGURE 5.40 – Analyse macro : mAP@.5 sur tous les scénarios à 24 (cf. Table 5.4).

Passant à une analyse micro, en comptabilisant les erreurs de détection (faux négatifs, faux positifs et mauvaises classifications) par image pour chaque scénario, nous identifions les événements spécifiques où le modèle montre des faiblesses. Les Figures 5.41 et 5.42 mettent en évidence les lacunes du détecteur dans des situations complexes. Par exemple, certains objets comme les piétons tombant (événements 1 et 3, cf. Figure 5.36a), les cyclistes renversés (événement 6, cf. Figure 5.36f), les véhicules et travailleurs de chantier (événement 4, cf. Figure 5.36d), ou encore les véhicules retournés (événements 5 et 8, cf. Figure 5.36h) ne sont presque jamais détectés. De plus, des objets atypiques comme les piétons en fauteuil roulant (événements 2 et 3, cf. Figure 5.36b) sont souvent pas détectés ou mal classifiés, confondus avec des véhicules ou des cyclistes. Ces observations montrent clairement qu'un ré-entraînement du modèle, en intégrant ces types de scénarios et objets spécifiques, est essentiel pour améliorer les performances, la robustesse et la sécurité de la détection. De manière intéressante, certains objets rares comme les véhicules en feu (événement 9, cf. Figure 5.36i) sont bien détectés, suggérant que les données synthétiques simulées peuvent être utilisées efficacement pour évaluer et identifier les lacunes des modèles de détection. En conclusion, cette approche souligne le potentiel des données synthétiques pour renforcer les performances des systèmes de détection dans des scénarios rares et critiques.

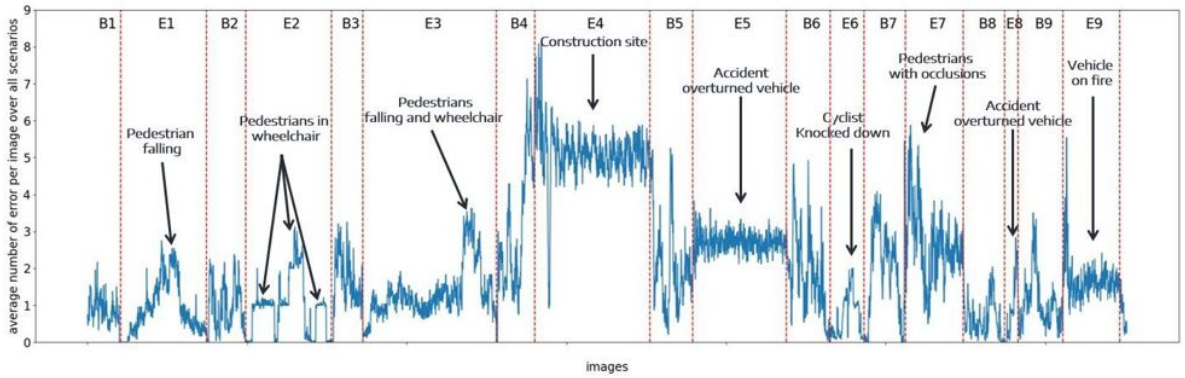
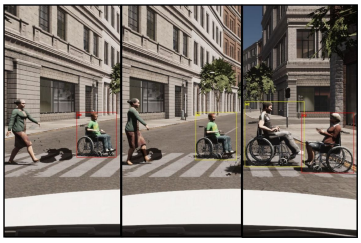


FIGURE 5.41 – Nombre moyen d'erreurs par image sur l'ensemble des 24 scénarios (cf. Tableau 5.4). E_i représente l'événement numéro i et B_i les phases intermédiaire sans évènement



(a) Événement 1 : le piéton qui tombe n'est systématiquement pas détecté (FN).



(b) Événement 2 : les piétons en fauteuil roulant ne sont souvent pas détectés (FN) ou sont mal classés (MC).



(c) Événement 4 : les véhicules et travailleurs de chantier ne sont pas détectés (FN).



(d) Événement 6 : le cycliste renversé n'est pas détecté (FN).



(e) Événement 8 : le véhicule renversé n'est pas détecté (FN).



(f) Événement 9 : la voiture en feu est globalement bien détectée.

FIGURE 5.42 – Certains des événements du scénario n°2 (cf. Table 5.4) avec les erreurs rencontrées à différents moments (de gauche à droite) : Faux Négatif (FN) en rouge et Mauvaise Classification (MC) en jaune.

De plus, nous démontrons que l'intégration de données synthétiques dans les jeux de données d'entraînement peut significativement améliorer les performances des modèles de détection. Pour cela, nous avons entraîné trois variantes du modèle YOLOv7 sur des jeux de données augmentés, composés initialement du jeu de données D_{train} (données naturelles). À ce dernier, nous avons successivement ajouté 25%, 33% et 50% de données synthétiques issues du jeu de données U_{train}. Cette approche, détaillée dans la Figure 5.43 et le Tableau 5.7, permet d'évaluer l'impact progressif de l'enrichissement par des données simulées sur les capacités du modèle. Ces expérimentations visent à explorer la complémentarité entre les données réelles et synthétiques et à identifier le meilleur équilibre entre les deux pour optimiser les performances globales du détecteur.



FIGURE 5.43 – Description des jeux de données d'entraînement augmentés avec des données synthétiques.

TABLE 5.7 – Datasets augmentés avec des données synthétiques.

Nom	**Partition**	**Bases de données**	**Volumétrie (#images)**	
			Par base	Total
*DU*25	*train*	Waymo	124, 668	240, 744
		BDD100K	55, 890	
		Unity	60, 186	
*DU*33	*train*	Waymo	124, 668	269, 489
		BDD100K	55, 890	
		Unity	88, 931	
*DU*50	*train*	Waymo	124, 668	361, 116
		BDD100K	55, 890	
		Unity	180, 558	

Nous avons évalué ces modèles sur trois ensembles distincts : les ensembles de test réels D_{test} et synthétiques U_{test}, ainsi que le jeu de données E&C (5.3.1), qui contient des images représentant des situations critiques similaires à celles simulées (par exemple, voiture en feu ou renversée, piétons qui tombent, etc.). Les résultats sont résumés dans le Tableau 5.8. Nous observons que les modèles entraînés avec l'ajout de 25% et 33% de données synthétiques montrent une amélioration notable des performances sur les ensembles réels et synthétiques. Ces résultats sont cohérents avec ceux rapportés dans [Westerski and Teck, 2023], Tableau 4, colonne REAL+SYNTH DIFF, où l'intégration de données synthétiques dans l'entraînement a entraîné un gain de mAP@.5 allant de 1% à 32%. La Figure 5.40 confirme cette tendance, avec les nouveaux modèles (courbes jaune et verte) atteignant des mAP significativement plus élevés sur l'ensemble des scénarios.

En ce qui concerne le jeu de données E&C, l'ajout de données synthétiques au jeu de données d'entraînement a permis une amélioration notable tant au niveau quantitatif que qualitatif. Par exemple, le modèle entraîné avec 33% de données synthétiques montre une augmentation de mAP@.5 de près de 2% sur ce jeu de données (cf. Tableau 5.8). Par ailleurs, la Figure 5.44 illustre que certains objets auparavant non détectés ou mal classés, comme les piétons qui tombent ou les véhicules renversés, sont désormais correctement détectés après cet enrichissement des données d'entraînement.

Cependant, il est crucial de souligner que l'ajout d'une quantité excessive de données synthétiques (plus de 50%) peut entraîner une dégradation significative des performances globales. Cette baisse est attribuable à l'écart de domaine (*domain gap*) entre les données synthétiques et naturelles, qui peut introduire un biais si les données synthétiques dominent le processus d'entraînement.

TABLE 5.8 – Estimation par bootstrap, avec $B = 100,000$ et avec un intervalle de confiance à 95%, du mAP@.5 de YOLOv7 entrainé sur le dataset de données réelles augmenté avec des données synthétiques.

Train / **Test**	D_{train}	$DU25_{train}$	$DU33_{train}$	$DU50_{train}$	U_{train}
mAP@.5 sur D_{test}	0.7883 ± 0.0046	**0.7973** ± 0.0035	**0.7921** ± 0.0042	**0.7967** ± 0.0044	0.5805 ± 0.0081
mAP@.5 sur U_{test}	0.7630 ± 0.0044	**0.975** ± 0.0027	**0.9774** ± 0.0035	**0.9814** ± 0.0027	**0.9871** ± 0.0022
mAP@.5 sur E&C	0.7386 ± 0.0274	0.7341 ± 0.0255	**0.7518** ± 0.0248	**0.7441** ± 0.0241	0.5046 ± 0.0314



(a) Entrainé sur D_{train} (à gauche), un véhicule renversé n'est pas détecté (FN). Entrainé sur $DU33_{train}$ (à droite), il est détecté.



(b) Entrainé sur D_{train} (à gauche), un piéton se relevant n'était pas détecté (FN). Entrainé sur $DU33_{train}$ (à droite), il est détecté.



(c) Entrainé sur D_{train} (à gauche), un piéton en fauteuil roulant était mal classé en tant que véhicule (MC). Entrainé sur $DU33_{train}$ (à droite), il est correctement classé en tant que piéton.



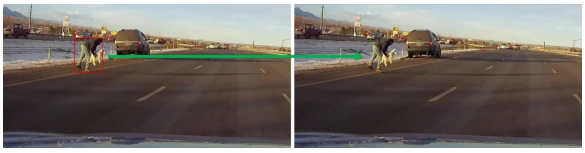
(d) Entrainé sur D_{train} (à gauche), une voiture en feu n'était pas détectée (FN). Entrainé sur $DU33_{train}$ (à droite), elle est détectée.



(e) Entrainé sur D_{train} (à gauche), un piéton au milieu de la route n'était pas détecté (FN). Entrainé sur $DU33_{train}$ (à droite), il est détecté.



(f) Entrainé sur D_{train} (à gauche), un piéton tombé de sa moto n'était pas détecté (FN). Entrainé sur $DU33_{train}$ (à droite), il est détecté.



(g) Entrainé sur D_{train} (à gauche), un piéton ramassant son chien sur l'autoroute n'était pas détecté (FN). Entrainé sur $DU33_{train}$ (à droite), il est détecté.



(h) Entrainé sur D_{train} (à gauche), un véhicule renversé n'était pas détecté (FN). Entrainé sur $DU33_{train}$ (à droite), il est détecté.

FIGURE 5.44 – Exemples d'images du jeu de données E&C avec des erreurs de détection avec YOLOv7 entraîné sur D_{train} sans données synthétiques (à gauche) et YOLOv7 entraîné sur $DU33_{train}$ avec 33% de données synthétiques (à droite). Code couleur : Faux Négatif (FN) en rouge et Mauvaise Classification (MC) en jaune.

En conclusion, nous avons démontré le potentiel considérable de la simulation pour améliorer les performances des modèles de détection d'objets. À l'aide du modèle YOLOv7, nous avons mis en évidence l'efficacité des images générées par simulation numérique, en particulier pour tester le modèle sur des scénarios critiques ou extrêmes (*edge and corner*). Ces scénarios, souvent impossibles ou coûteux à capturer dans des situations réelles, permettent d'identifier les lacunes du modèle, de détecter ses défauts potentiels et, in fine, d'affiner ses performances. L'un des avantages majeurs de la simulation réside dans la précision de ses annotations. Contrairement aux annotations manuelles, qui peuvent être influencées par les limites des capacités visuelles et la subjectivité des annotateurs, notamment pour les cas limites, les annotations issues de la simulation sont entièrement précises et exemptes d'erreurs humaines. Cependant, pour maximiser les bénéfices de cette approche, il est impératif que la simulation soit suffisamment réaliste. Plus le réalisme est élevé, plus les avantages en termes de performance et de généralisabilité sont significatifs. Dans sa forme actuelle, notre simulateur, conçu comme une preuve de concept, reste basique. Les perspectives d'amélioration incluent l'intégration de scénarios *edge and corner* plus variés, la génération d'environnements supplémentaires tels que des paysages ruraux ou autoroutiers, ainsi qu'une augmentation de la randomisation des caractéristiques des objets (apparence, position et dynamique). Comme le souligne [Fabbri et al., 2021], une plus grande diversité dans les données synthétiques générées pourrait réduire encore davantage l'écart entre les domaines synthétiques et naturels. Par ailleurs, des approches complémentaires mériteraient d'être explorées, telles que le pré-entraînement des modèles avec des données synthétiques avant leur fine-tuning sur des données réelles. Plusieurs études, comme [Erfanian Ebadi et al., 2022, Prakash et al., 2019], ont démontré que cette stratégie peut améliorer significative-

ment les performances de détection sur des images réelles. En somme, bien que notre simulateur actuel fournisse une base prometteuse, il existe de nombreuses opportunités d'amélioration. En renforçant le réalisme et la diversité des données synthétiques, nous pourrons tirer davantage parti de la simulation pour développer des modèles de détection d'objets plus robustes, précis et adaptés à des situations complexes.

Ces résultats ont conduit à la publication de l'article [Jaspar et al., 2024].

5.3.4 Récapitulatif de l'ensemble des jeux de données

Le tableau suivant récapitule l'ensemble des bases de données mentionnées précédemment :

TABLE 5.9 – Récapitulatif de l'ensemble des jeux de données.

Nom	Type	Caméra	Volumétrie	Localisation	Annotées	Caractéristiques
Waymo 5.3.1	Naturelles	Frontale	195, 823	USA	✓	Ville et Autoroute
BDD100K 5.3.1		Frontale	79, 863	USA		Ville et Autoroute
Stellantis 5.3.1		Frontale	20, 139	France		Ville et Autoroute
Boxy 5.3.1		Frontale	178, 525	USA	✗	Autoroute
Audi 5.3.1		Frontale	12, 497	Allemagne	✗	Ville et Autoroute
CarCrash 5.3.1		Frontale	75, 000	N/A	✗	Accidents
DashCam 5.3.1		Frontale	62, 000	Chine	✗	Accidents
NuImages 5.3.1		Frontale	214, 519	USA Pennsylvanie Malaisie	✗	Ville et Autoroute
		Multiple	1, 098, 246		✗	
NuImagesLabels 5.3.1		Frontale	17, 774		✓	
		Multiple	91, 238			
Waterloo 5.3.1		Frontale	6, 917	Canada	✗	Neige
PeSOTIF 5.3.1		N/A	1, 126	N/A	✗	edge and corner
Edge&Corner 5.3.1		N/A	1, 365	N/A	✓	edge and corner
Unity 5.3.2	Synthétiques	Frontale	156, 274	N/A	✓	edge and corner

5.3.5 Conclusion et perspectives

En conclusion, nous avons collecté un peu plus de 1, 000, 000 d'images, principalement issues d'enregistrements réels réalisés à l'aide de caméras embarquées, complétées par des données générées de manière synthétique. Ces données ont été soigneusement sélectionnées et préparées pour servir deux objectifs principaux : l'entraînement des modèles de compression, de détection et de diffusion, et l'évaluation de l'impact de la compression sur la détection d'objets, afin d'estimer des garanties de performances. Cependant, il est important de noter que ces données restent insuffisantes pour couvrir l'ensemble des scénarios définis dans l'ODD. De nombreux cas critiques ou extrêmes, indispensables pour une évaluation complète, ne sont pas encore représentés. Cela limite la portée des garanties de performance actuelles et souligne la nécessité d'acquérir davantage de données variées, aussi bien réelles que synthétiques, pour améliorer ces garanties. Concernant la simulation,

bien qu'elle constitue une base prometteuse dans ce projet, elle demeure incomplète. Le simulateur manque de diversité, notamment en ce qui concerne les scénarios critiques. Par exemple, un seul scénario impliquant un cycliste renversé est présent, et des variations avec différents points de vue seraient nécessaires pour une analyse plus approfondie. De plus, la simulation se limite actuellement à des environnements urbains. Des extensions vers des environnements ruraux et autoroutiers, accompagnées d'événements adaptés à ces contextes, seraient cruciales pour une meilleure représentativité des données et une réduction du domain gap observé entre les données réelles et synthétiques. Dans ce contexte, Stellantis pourra mobiliser ses ressources internes pour combler ces lacunes, notamment en diversifiant les scénarios et en augmentant la randomisation des paramètres dans les simulations. Par ailleurs, il existe de nombreux autres jeux de données disponibles en open source ou en interne chez Stellantis, qui pourraient enrichir les futures expérimentations.

Dans la suite de cette thèse, nous décrirons les modèles de compression, de détection et de diffusion utilisés, ainsi que leur processus d'apprentissage, pour analyser et améliorer les performances globales du système.

Chapitre 6

Application industrielle : estimation de garanties

Contents

6.1 Introduction . . . 181
6.2 Apprentissage des modèles . . . 182
6.2.1 Entrainement de la compression . . . 182
Compression Agnostique . . . 184
Compression ROI . . . 188
6.2.2 Entrainement de la Détection . . . 190
6.2.3 Entrainement de la Diffusion . . . 191
6.2.4 Conclusion . . . 195
6.3 Analyse de l'effet de la compression sur la détection . . . 195
6.4 Estimation des probabilités d'erreur . . . 200
6.4.1 Estimation intra-datasets . . . 202
6.4.2 Estimation inter-datasets . . . 206
Distribution uniforme . . . 206
Distribution pour le Field Monitoring . . . 209
6.4.3 Conclusion . . . 214
6.5 Optimisation des performances . . . 215
6.5.1 Réentrainer la détection . . . 216
Ajout d'images naturelles . . . 216
Ajout d'images synthétiques . . . 218
Ajout d'images comprimées . . . 218
6.5.2 Ré-entrainer la compression . . . 220
6.6 Génération d'images par diffusion . . . 222
6.6.1 Maximiser l'erreur liée à la compression . . . 224
6.6.2 Maximiser l'erreur de détection induite par la compression . . . 227
6.7 Conclusion et perspectives . . . 237

6.1 Introduction

Ce chapitre constitue la suite du précédent, qui a établi les fondations du système *Field Monitoring* et procédé à la conception de multiples bases de données provenant de sources diverses. Grâce à cette structuration préalable, nous disposons désormais d'un ensemble de données représentatif des scénarios critiques et des conditions opérationnelles définis dans l'ODD. Ces données serviront à appliquer les étapes suivantes de la méthodologie d'obtention de garanties de performances.

L'objectif principal de ce chapitre est d'évaluer l'impact de la compression d'images sur les performances du détecteur d'objets et de garantir que les dégradations induites par cette compression restent inférieures à un

seuil tolérable, assurant ainsi la fiabilité et la robustesse des résultats du système dans des conditions variées. Cette étape est essentielle pour quantifier les erreurs, identifier les vulnérabilités des modules, et proposer des solutions pour limiter les défaillances.

Dans la section 6.2, nous débuterons par sélectionner les modèles de compression et de détection adaptés à notre étude, puis procéderons à leur apprentissage sur les données collectées. Cette étape permettra d'obtenir des modules opérationnels aux performances optimales pour l'évaluation. Ensuite, dans la section 6.3, nous analyserons l'effet de la compression sur la détection d'objets. Cette analyse vise à comprendre les interactions entre les deux modules, en évaluant comment les distorsions induites par la compression affectent la capacité du détecteur à identifier correctement les objets dans les images. Nous poursuivrons dans la section 6.4 avec l'estimation des probabilités d'erreur. En appliquant les étapes méthodologiques définies précédemment, nous évaluerons ces erreurs de manière rigoureuse en tenant compte des variabilités intra-dataset et inter-dataset. Cela permettra de fournir des estimations probabilistes robustes et des garanties de performances adaptées aux exigences du système étudié. Dans la section 6.5, nous identifierons des pistes concrètes d'optimisation pour améliorer les résultats obtenus et réduire les probabilités d'erreur estimées. Enfin, dans les sections 6.6, nous exploiterons des modèles de génération d'images par diffusion pour explorer les sources d'erreur potentielles. Cette étape permettra de générer des images critiques orientées pour maximiser les erreurs de détection et d'identifier les scénarios les plus vulnérables. Sur cette base, nous proposerons des approches d'optimisation visant à renforcer la robustesse du système et à réduire les risques de défaillance.

6.2 Apprentissage des modèles

Dans cette section, nous présentons les modèles de compression, de détection et de diffusion qui ont été employés dans le cadre de notre étude. Une description détaillée de chaque modèle sera donnée, accompagnée d'une explication des processus d'entraînement ainsi que des jeux de données utilisés pour leur développement. Il est important de souligner que, dans le cas des modèles de compression et de détection, l'objectif principal n'est pas de maximiser leurs performances individuelles. En effet, notre étude se concentre sur l'analyse de l'impact de la compression sur la détection d'objets ainsi que sur l'estimation des garanties de performances globales du système. Par conséquent, nous nous limiterons à décrire les configurations de base de ces modèles, sans chercher à optimiser leurs performances intrinsèques.

6.2.1 Entrainement de la compression

À travers l'état de l'art présenté dans la section 3.2, nous avons sélectionné deux modèles qui se distinguent par leurs caractéristiques et répondent efficacement aux exigences de notre application. Ces modèles offrent un bon équilibre entre qualité de compression et vitesse d'exécution, éléments essentiels dans le cadre de notre étude.

Le premier modèle, nommé STF (*Symmetrical TransFormer*) [Zou et al., 2022] (cf. Figure 3.16), est une architecture basée sur les Transformers. STF propose une approche novatrice de compression d'images en intégrant des blocs de Transformers symétriques dans un encodeur de sous-échantillonnage et un décodeur de sur-échantillonnage. En combinant un mécanisme d'attention locale, basé sur des fenêtres adaptatives, avec un apprentissage de caractéristiques globales, ce modèle est capable de capturer des informations clés à différents niveaux, local et global. Cette capacité lui permet de répondre aux besoins spécifiques de la compression d'images tout en maintenant une haute efficacité.

Le second modèle, nommé QRes-VAE [Duan et al., 2023] (cf. Figure 3.13), est une architecture multi-échelles qui réinvente les Variational Autoencoders (VAE) basés sur ResNet pour la compression avec pertes. QRes-VAE se distingue par son utilisation de variables latentes adaptées via une quantification et un codage entropique efficaces. Cette approche adopte un schéma de compression dit "coarse-to-fine" (du plus grossier au plus détaillé), permettant un encodage et un décodage parallèles. Cette conception accélère considérablement les processus sur les GPU, rendant le modèle bien adapté à des applications à grande échelle. En outre, QRes-VAE dépasse les performances des méthodes antérieures en termes de compression d'images naturelles avec pertes, tout en maintenant une excellente fidélité de reconstruction.

Ces modèles ont été sélectionnés pour leurs excellentes performances en termes de compromis entre taux de compression et distorsion, ainsi que pour leur vitesse d'encodage-décodage particulièrement efficace. En effet, ils permettent de traiter une image de dimensions 640×1280 en moins de 0.5 secondes, ce qui est un atout

majeur pour des applications nécessitant des traitements en temps quasi-réel ou sur des ensembles de données volumineux (cf. Tableau 6.1).

TABLE 6.1 – Vitesse d'encodage et de décodage pour chaque compresseur pour une image de dimension 640×1280, mesurée sur un GPU NVIDIA RTX A5500. Les performances de nos modèles sont comparées à celles de deux modèles plus anciens, mais couramment mentionnés dans la littérature : mbt2018 [Minnen et al., 2018b] et cheng2020_attn [Cheng et al., 2020b].

	Encodage (en seconde)	**Décodage (en seconde)**
STF	0.21 ± 0.02	0.13 ± 0.02
QRes-VAE	0.25 ± 0.009	0.15 ± 0.007
mbt2018	4.78 ± 0.21	10.83 ± 0.38
cheng2020_attn	3.78 ± 0.09	8.87 ± 0.37

Pour entraîner ces modèles, nous utilisons le dataset D, déjà présenté dans la section 5.3.3. Ce dataset est composé exclusivement de données naturelles issues des bases Waymo (5.3.1) et BDD100K (5.3.1), et partitionné en trois sous-ensembles : D_{train}, D_{valid} et D_{test} (cf. Tableau 6.2).

TABLE 6.2 – Datasets initiaux pour l'apprentissage de la Compression et de la Détection.

Nom	**Partition**	**Bases de données**	**Volumétrie (#images)**	
			Par base	Total
D	*train*	Waymo	124,668	180,558
		BDD100K	55,890	
	valid	Waymo	31,168	45,141
		BDD100K	13,973	
	test	Waymo	39,987	49,987
		BDD100K	10,000	

Nous considérons deux approches distinctes pour la compression des images :

— **Compression Agnostique** : Ce type de compression ne tient pas compte de l'information sémantique contenue dans l'image. Autrement dit, elle ne fait pas de distinction entre les différents objets présents dans l'image, tels que les voitures, piétons ou cyclistes, mais se concentre uniquement sur des aspects géométriques tels que les contours, les zones uniformes en couleur, ou les textures. Cette méthode vise à minimiser une fonction de coût définie par :

$$L = R + \lambda \cdot D(x, \hat{x}) \tag{6.1}$$

où R représente le débit binaire (taux de compression), $D(x, \hat{x})$ est une mesure de la distorsion entre l'image d'origine x et l'image reconstruite $\hat{x}$, généralement calculée à l'aide du MSE (Mean Squared Error, ou erreur quadratique moyenne) ou du MS-SSIM (Indice de Similarité Structurelle Multi-échelle), et λ est un hyperparamètre réglant le compromis entre le débit binaire et la distorsion. Cette méthode est simple et efficace pour des applications où l'information sémantique n'est pas primordiale, mais elle peut entraîner une dégradation significative des objets importants dans certaines situations.

— **Compression ROI** (Region Of Interest) : Contrairement à la compression agnostique, la compression ROI adapte le niveau de compression en fonction de l'importance sémantique des différentes régions de l'image. Par exemple, les zones contenant des objets d'intérêt (véhicules, piétons, cyclistes) sont compressées avec une précision plus élevée, tandis que les zones moins critiques (route, ciel, arrière-plan) peuvent être davantage compressées sans compromettre les performances de détection des objets. Inspirés par [Song et al., 2021c], nous avons reformulé la fonction de coût pour inclure une pondération spatiale :

$$L = R + \Lambda^T D \tag{6.2}$$

Ici, $\Lambda = [\lambda_i]_{i=1}$ est un vecteur de multiplicateurs de Lagrange (de taille N, correspondant au nombre total de pixels dans l'image). Chaque multiplicateur λ_i est attribué à un pixel x_i de l'image, avec une valeur élevée pour les pixels appartenant à des objets d'intérêt, et une valeur faible sinon. Si la distorsion

est mesurée par le MSE, la perte devient :

$$L = R + \sum_{i}^{N} \lambda_i \frac{(x_i - \hat{x}_i)^2}{N} \tag{6.3}$$

Dans cette approche, le vecteur Λ peut être vu comme une carte de "qualité" à la même résolution que l'image. Pour simplifier, nous supposons que λ_i prend uniquement deux valeurs : $\lambda_i \in \lambda^+, \lambda^-$ avec $\lambda^+ > \lambda^-$. Cela permet de prioriser les régions importantes tout en compressant davantage les zones moins critiques. Une extension naturelle de cette méthode consisterait à introduire plusieurs niveaux de λ pour des classes d'objets spécifiques, par exemple :

$$\lambda_i \in \lambda_{\text{voiture}}, \lambda_{\text{piéton}}, \lambda_{\text{arrière-plan}} \quad \text{avec} \quad \lambda_{\text{piéton}} > \lambda_{\text{voiture}} > \lambda_{\text{arrière-plan}} \tag{6.4}$$

Cette hiérarchie pourrait être particulièrement utile si un détecteur est plus performant pour certains objets que pour d'autres, permettant ainsi d'optimiser davantage les régions moins sensibles. Toutefois, l'introduction de plusieurs valeurs pour λ augmente la complexité et pourrait rencontrer des limites pratiques, un aspect qui ne sera pas approfondi dans cette étude. Cette méthode innovante, qui permet de convertir un modèle de compression générique, quelle que soit son architecture, en un modèle de compression ROI, a conduit au dépôt du brevet intitulé « Procédé et dispositif de compression avec pertes et relatif aux zones d'intérêts d'images numériques utilisant des paramètres déterminés à partir d'un réseau de neurones », 2023, N° FR2304591, actuellement en instance.

Compression Agnostique

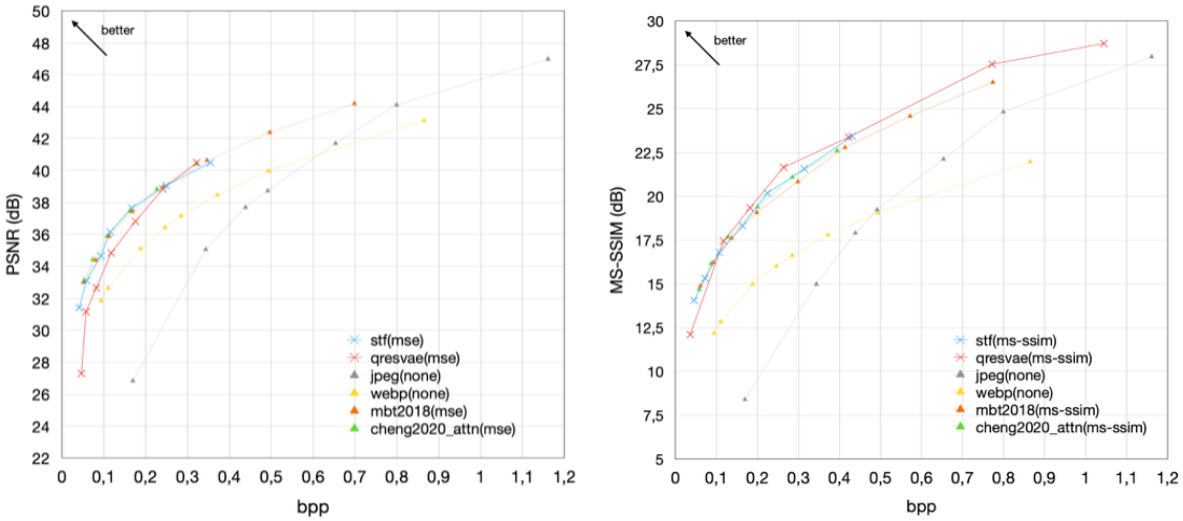
Dans le cadre de la compression agnostique, nous avons entraîné nos deux modèles en utilisant les datasets D_{train} et D_{valid}, puis évalué leurs performances sur le dataset D_{test} (cf. Tableau 6.3). L'entraînement a été effectué en suivant la fonction de coût définie par l'équation (6.1), avec deux métriques de distorsion, MSE et MS-SSIM. Les hyperparamètres λ, repris des articles [Zou et al., 2022] et [Duan et al., 2023], explorés au cours de cette étape sont les suivants :

TABLE 6.3 – Modèles de compression agnostique entraînés avec différents hyperparamètres λ.

Architecture	Datasets d'entrainement	Nom	Hyperparamètres λ
STF	D_{train} D_{valid}	stf(mse)	$\lambda_1 = 0.0009$, $\lambda_2 = 0.0018$, $\lambda_3 = 0.0035$, $\lambda_4 = 0.0067$, $\lambda_5 = 0.013$, $\lambda_6 = 0.025$, $\lambda_7 = 0.0483$
		stf(ms-ssim)	$\lambda_1 = 1.2$, $\lambda_2 = 2.4$, $\lambda_3 = 4.58$, $\lambda_4 = 8.73$, $\lambda_5 = 16.64$, $\lambda_6 = 31.73$, $\lambda_7 = 60.5$
QRes-VAE		qresvae(mse)	$\lambda_1 = 4$, $\lambda_2 = 8$, $\lambda_3 = 16$, $\lambda_4 = 32$, $\lambda_5 = 64$, $\lambda_6 = 128$, $\lambda_7 = 256$
		qresvae(ms-ssim)	$\lambda_1 = 0.1$, $\lambda_2 = 1$, $\lambda_3 = 2$, $\lambda_4 = 4$, $\lambda_5 = 8$, $\lambda_6 = 32$, $\lambda_7 = 64$

Dans la suite, nous désignerons les modèles de compression selon la convention suivante : stf(mse)$_{\lambda_i}$ fait référence au compresseur STF entraîné avec la métrique MSE et l'hyperparamètre λ_i. La même notation s'applique au compresseur QRes-VAE ainsi qu'à l'utilisation de la métrique MS-SSIM. Pour évaluer les compresseurs sur plusieurs images, une moyenne des performances est nécessaire. La Figure 6.1 présente les résultats de notre entraînement : nous y montrons le PSNR moyen (en dB) pour les modèles entraînés avec la métrique MSE, et le MS-SSIM moyen (en dB) pour ceux entraînés avec MS-SSIM, en fonction du débit binaire moyen (*bpp*) pour chaque valeur de λ. Ces performances sont comparées à celles de méthodes standard comme JPEG [Wallace, 1992] et WebP [web, 2010], ainsi qu'à deux modèles plus anciens de compression par apprentissage profond : mbt2018 [Minnen et al., 2018b] et cheng2020_attn [Cheng et al., 2020b], disponibles dans la bibliothèque PyTorch CompressAI [Bégaint et al., 2020]. Ces derniers ont été entraînés sur le dataset Vimeo90K [Xue et al., 2019], composé de 90k clips vidéo couvrant une large variété de scènes et d'actions. Les résultats montrent clairement que les compresseurs par apprentissage profond surpassent largement les méthodes standard comme JPEG et WebP, en particulier à faibles débits (inférieurs à 0.4 bpp). Bien que mbt2018 et cheng2020_attn offrent des performances comparables à celles de STF et QRes-VAE en termes de qualité,

ils présentent une vitesse d'encodage et de décodage nettement inférieure. Par exemple, ils nécessitent environ 36 secondes par image, soit 9 fois plus de temps que STF et QRes-VAE, qui offrent des performances bien plus rapides (cf. Tableau 6.1).



(a) Évaluation moyenne des compresseurs entraînés avec la métrique MSE.

(b) Évaluation moyenne des compresseurs entraînés avec la métrique MS-SSIM.

FIGURE 6.1 – Analyse quantitative des performances des modèles de compression sur D_{test}.

Il est crucial de souligner que la Figure 6.1 masque un aspect fondamental de la compression d'image par deep learning : l'adaptation dynamique du taux de compression en fonction du contenu géométrique de chaque image, comme les contours ou les zones de couleurs uniformes. En examinant la Figure 6.2, on observe que, pour chaque valeur de λ, les images compressées génèrent une distribution $\{(\text{PSNR}_i, \text{bpp}_i)\}_{i=1:N}$. Cela révèle que certaines images peuvent être fortement compressées tout en conservant un excellent niveau de qualité PSNR, tandis que d'autres, bien que moins compressées, présentent un niveau de PSNR relativement faible. Cette variabilité est également observable pour la métrique MS-SSIM, mettant en évidence l'impact du contenu spécifique de chaque image sur les performances de compression.

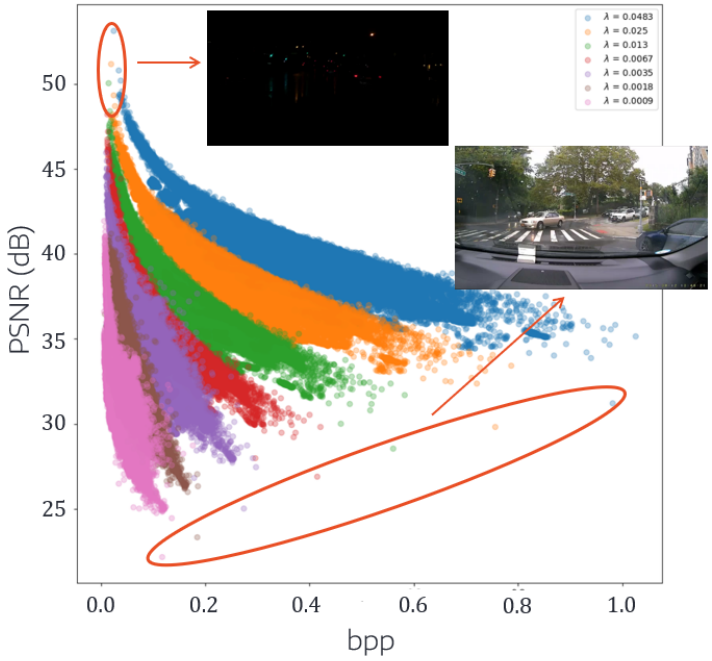


FIGURE 6.2 – Relation entre le PSNR et le bpp pour chaque image de D_{test}, obtenue avec le modèle stf(mse) pour différentes valeurs de l'hyperparamètre λ.

En analysant les Figures 6.3 et 6.4, on remarque que les compresseurs basés sur le deep learning engendrent généralement des distorsions visuelles globales, souvent caractérisées par un flou perceptible. Les modèles entraînés avec la métrique MSE privilégient les régions contenant des contours prononcés ou des transitions marquées, leur allouant une proportion plus importante de bits. En revanche, les modèles entraînés avec la métrique MS-SSIM adoptent une distribution des bits plus uniforme à travers l'image, ce qui favorise la préservation de la structure globale et des détails visuels.



FIGURE 6.3 – Comparaison qualitative d'une image compressée à l'aide de différentes méthodes de compression et à divers taux de compression (les annotations indiquent les valeurs de bpp/PSNR/MS-SSIM).

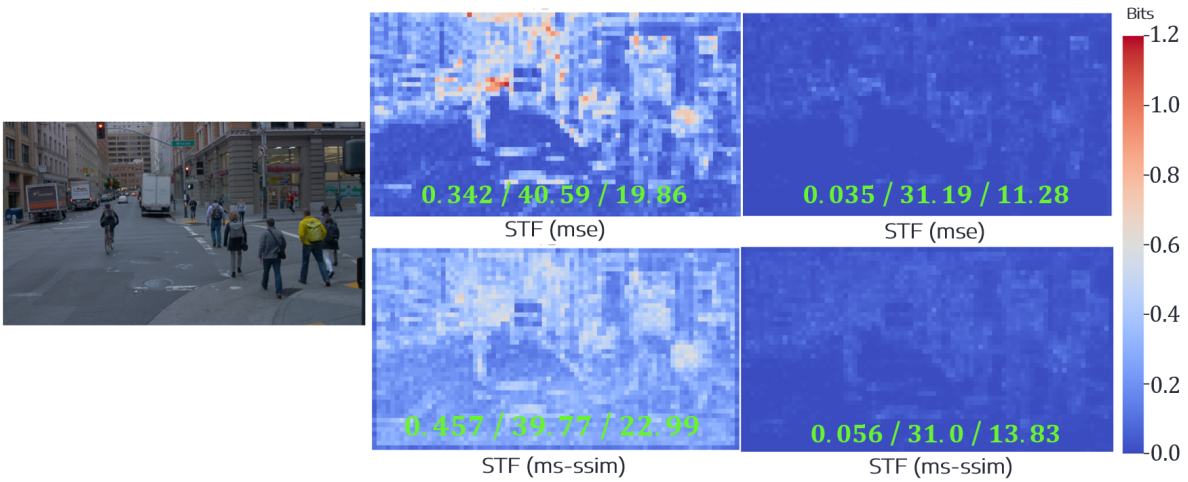


FIGURE 6.4 – Répartition des bits entre des taux de compression élevés et faibles avec le compresseur STF (résultats similaires observés avec QRes-VAE), accompagnée des valeurs correspondantes de bpp, PSNR et MS-SSIM annotées en vert.

Cependant, à des niveaux de compression très élevés, ces distorsions globales peuvent être aggravées par l'apparition d'artefacts visibles, comme le montrent les exemples illustrés dans la Figure 6.5.



(a) Image d'origine non comprimée.



(b) Image compressée avec artefacts.

FIGURE 6.5 – Observation d'artefacts sur une image Waymo fortement compressées à l'aide du modèle $\text{stf(mse)}_{\lambda_1}$ ($bpp = 0.036$).

Compression ROI

Si nous pouvions fortement compresser toutes les zones d'une image non pertinentes pour notre application (par exemple le ciel, les immeubles, les arbres, etc.) tout en maintenant une haute qualité d'image pour les objets d'intérêt (véhicules, piétons et cyclistes), il serait possible, comparé à la compression agnostique, de réduire davantage la taille des images sans compromettre la performance de détection qui en découle. Pour atteindre cet objectif, nous réentraînons nos modèles de compression en adoptant une approche ROI (*Region of Interest*), en utilisant la fonction de coût définie par l'équation (6.3). Les cartes de "qualité" nécessaires à cet entraînement sont construites à partir des annotations associées à chaque image, les boîtes 2D définissant précisément les positions des objets d'intérêt (cf. Figure 6.6).

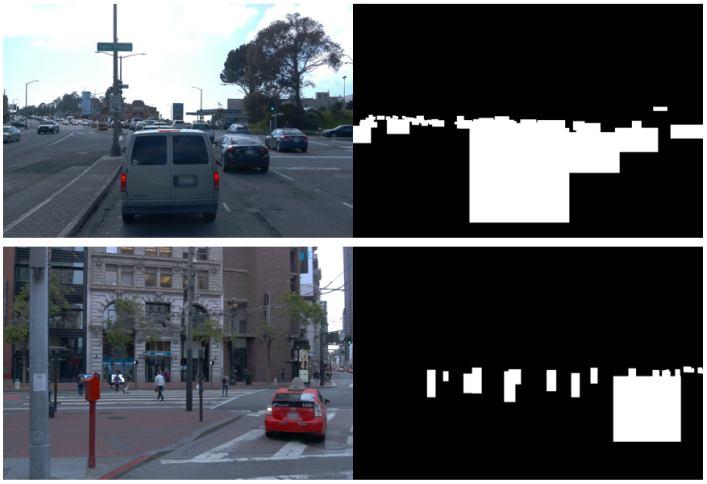


FIGURE 6.6 – Exemples d'images issues du dataset Waymo (cf. Section 5.3.1) et de leurs cartes de qualité associées. Les zones correspondant aux véhicules, où une compression minimale est souhaitée, sont mises en évidence en blanc (λ^+), tandis que l'arrière-plan, où une compression plus forte est tolérée, est représenté en noir (λ^-).

L'entraînement est réalisé à nouveau sur les datasets D_{train} et D_{valid} (cf. Tableau 6.2), en explorant les hyperparamètres λ suivants :

TABLE 6.4 – Modèles de compression appris ROI.

Architecture	Datasets d'entrainement	Nom	Hyperparamètres λ
STF	D_{train} D_{valid}	stf(mse)	$\lambda_8 = \{\lambda^- = 1.25 \cdot 10^{-5}, \lambda^+ = 0.08\}$ $\lambda_9 = \{\lambda^- = 10^{-3}, \lambda^+ = 0.08\}$
QRes-VAE		qresvae(mse)	$\lambda_8 = \{\lambda^- = 2, \lambda^+ = 128\}$

La Figure 6.7 illustre les résultats qualitatifs de nos nouveaux modèles stf(mse) basés sur les régions d'intérêt (Figures 6.7 (b) et (c)), comparés aux modèles STF agnostiques précédemment entraînés (Figure 6.7 (a)). Ces résultats mettent en évidence une répartition intelligente des bits, avec une allocation majoritaire aux pixels correspondant aux objets d'intérêt. Cela se traduit par une délimitation plus nette des objets par rapport au reste de l'image, qui apparaît davantage flouté. En d'autres termes, nos compresseurs ROI ont démontré leur capacité à prioriser les régions pertinentes, montrant ainsi une forme d'apprentissage des zones d'intérêt dans les images.

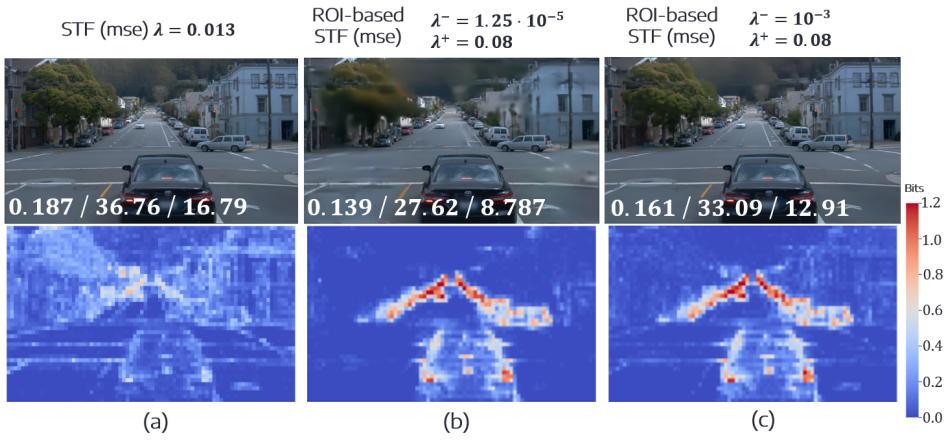


FIGURE 6.7 – Image compressée et allocation de bits respective pour le modèle STF agnostique (a) et les modèles STF ROI avec deux ensembles d'hyperparamètres, λ_8 (b) et λ_9 (c). Les annotations en blanc indiquent respectivement les valeurs de bpp/PSNR/MS-SSIM pour chaque configuration.

En entraînant les compresseurs avec la fonction de coût définie par l'équation (6.3), ces derniers acquièrent la capacité de différencier certaines zones des images en fonction de leur importance. Autrement dit, ces compresseurs développent une capacité implicite de détection des objets, bien que leur architecture ne soit pas initialement conçue dans ce but. Cependant, ce type de compression introduit un risque supplémentaire. En effet, elle repose sur la capacité du modèle à détecter correctement les objets. Si un objet n'est pas détecté, il pourrait être fortement compressé, au point de devenir méconnaissable. Cela pose un problème majeur dans le contexte de cette thèse, où une telle dégradation des objets d'intérêt est incompatible avec l'objectif de garantir de la chaîne compression-détection.

6.2.2 Entrainement de la Détection

Dans cette étude, nous avons retenu trois modèles de détection d'objets issus de l'état de l'art présenté dans la section 3.3 : YOLOv7 [Wang et al., 2023a], Faster R-CNN [1] [Ren et al., 2016] et RetinaNet [2] [Lin et al., 2017b].

YOLOv7, dernière version publiée au démarrage de cette thèse, est un modèle *single-stage* qui marque des avancées significatives en matière de détection en temps réel. Grâce à des innovations architecturales telles que le réseau E-ELAN, il offre une précision accrue tout en réduisant considérablement le nombre de paramètres et la charge de calcul par rapport à ses prédécesseurs comme YOLOv4 et YOLOR. Ces caractéristiques en font un choix optimal pour des applications nécessitant à la fois rapidité et performance. Faster R-CNN, en revanche, est une architecture *two-stage* classique qui repose sur un Réseau de Propositions de Régions (RPN) pour générer des boîtes candidates avant leur classification. Bien qu'il soit moins rapide que YOLOv7, il demeure une référence en matière de précision, notamment pour les tâches où une détection fine et robuste est requise. Enfin, RetinaNet est un modèle *single-stage* qui se distingue par l'introduction de la Focal Loss, une fonction de perte conçue pour gérer le déséquilibre entre les classes avant-plan et arrière-plan. Associé à un backbone ResNet et à un Feature Pyramid Network (FPN), il combine simplicité, rapidité et précision, offrant un excellent compromis dans des environnements avec des déséquilibres importants.

Comme pour les compresseurs, ces trois modèles ont été entraînés sur le dataset D, constitué de données issues des bases Waymo (5.3.1) et BDD100K (5.3.1), selon les partitions décrites dans le Tableau 6.5.

TABLE 6.5 – Datasets initiaux pour l'apprentissage de la Compression et de la Détection.

Nom	**Partition**	**Bases de données**	**Volumétrie (#images)**	
			Par base	Total
D	*train*	Waymo	124, 668	180, 558
		BDD100K	55, 890	
	valid	Waymo	31, 168	45, 141
		BDD100K	13, 973	
	test	Waymo	39, 987	49, 987
		BDD100K	10, 000	

Les résultats des performances des détecteurs sur le dataset D_{test} sont présentés dans le tableau 6.6 suivant :

TABLE 6.6 – Average Precision (AP@.5) des détecteurs sur le dataset D_{test} avec intervalles de confiance à 95% estimés par méthode bootstrap ($B = 100,000$).

AP@.5 / **Modèles**	**Véhicule**	**Piéton**	**Cycliste**	**mAP**
FasterRCNN	0.9434 ± 0.0003	0.847 ± 0.004	0.61 ± 0.01	0.802 ± 0.004
RetinaNet	0.936 ± 0.004	0.878 ± 0.001	0.65 ± 0.013	0.821 ± 0.0047
YOLOv7	0.9298 ± 0.0001	0.834 ± 0.007	0.60 ± 0.01	0.788 ± 0.004

Faster R-CNN et RetinaNet surpassent légèrement YOLOv7 en termes de précision moyenne (mAP), avec des scores particulièrement solides sur les classes de piétons et de cyclistes. Cependant, YOLOv7 se distingue par sa

1. FasterRCNN_ResNet50_FPN_v2 from PyTorch
2. RetinaNet_ResNet50_FPN_v2 from PyTorch

rapidité d'inférence et une faible utilisation de mémoire, ce qui en fait une solution adaptée pour des applications en temps réel ou des systèmes embarqués. Tous les détecteurs affichent des performances élevées sur la classe des véhicules ($AP \approx 0.93$), reflet de leur forte prépondérance dans les datasets Waymo et BDD100K (cf. Figures 5.6 et 5.12). En revanche, les performances sur les classes de piétons et de cyclistes sont inférieures, particulièrement pour les cyclistes ($AP \leq 0.65$), en raison de leur faible fréquence dans les données d'entraînement. Faster R-CNN, bien qu'étant plus lent, excelle dans la précision grâce à son approche *two-stage*. RetinaNet, avec son utilisation de la Focal Loss, compense le déséquilibre entre les classes avant-plan et arrière-plan, offrant un bon compromis entre simplicité, rapidité et précision. YOLOv7, bien qu'affichant un mAP légèrement inférieur, combine vitesse et efficacité, le rendant idéal pour des contraintes de ressources. Ces résultats montrent que le choix du détecteur doit être guidé par les priorités spécifiques de l'application, qu'il s'agisse de maximiser la précision ou de minimiser les délais d'inférence.

6.2.3 Entrainement de la Diffusion

Notre objectif est de construire un générateur d'images capable de produire des images de haute qualité visuelle tout en offrant la possibilité d'orienter leur contenu en fonction d'un certain niveau d'erreur. Plus précisément, nous cherchons à générer de nouvelles images, soit à partir d'un bruit initial, soit à partir d'une image existante, tout en intégrant une certaine flexibilité sur les caractéristiques visuelles souhaitées.

Pour ce faire, nous nous appuyons sur le modèle Stable Diffusion (LDM) décrit dans [Rombach et al., 2022]. Nous utilisons une version de l'architecture et des méthodes d'apprentissage open-source, incluant les fonctions de coût et les hyperparamètres, disponibles sur `https://github.com/hkproj/pytorch-stable-diffusion` et `https://github.com/CompVis/stable-diffusion`. Ce modèle repose sur deux composants principaux (cf. Figure 6.8) :

- Un autoencodeur : Cet autoencodeur encode une image dans un espace latent, réduisant ainsi sa dimensionnalité, et la décode ensuite pour reconstruire l'image originale. Cela permet de travailler directement dans un espace latent, bien plus compact, tout en conservant les détails essentiels de l'image.
- Un modèle de diffusion basé sur une architecture UNet : Ce modèle effectue le processus de diffusion dans l'espace latent, c'est-à-dire qu'il réalise un débruitage progressif pour générer des images de haute qualité à partir de bruit aléatoire ou d'autres latents.

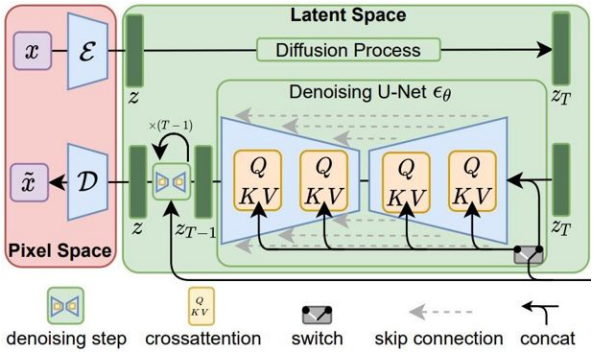


FIGURE 6.8 – Illustration du modèle de diffusion sans le conditionnement (source [Rombach et al., 2022]).

L'entraînement de notre modèle de diffusion sera effectué en deux phases. La première est l'entrainement de l'autoencodeur. Cette phase permettra de maîtriser la compression et la reconstruction des images dans l'espace latent. Un autoencodeur bien entraîné garantit une bonne qualité de reconstruction après la diffusion. La seconde est l'entrainement du UNet pour la diffusion. Le UNet sera entraîné à réaliser le processus de débruitage dans l'espace latent, une étape cruciale pour générer des images réalistes. Nous avons construit deux datasets adaptés à ces phases d'entraînement, décrits dans le tableau 6.7.

TABLE 6.7 – Datasets pour l'apprentissage du modèle de diffusion.

Nom	Partition	Bases de données	Volumétrie (#images)	
			Par base	Total
Diffusion_Front	*train*	Waymo	185, 823	660, 483
		BDD100K	67, 863	
		Stellantis	17, 139	
		Boxy	168, 525	
		Audi	10, 687	
		NuImages (Front)	204, 519	
		Waterloo (Front)	5, 917	
	valid	Waymo	10, 000	47, 800
		BDD100K	12, 000	
		Stellantis	3, 000	
		Boxy	10, 000	
		Audi	1, 800	
		NuImages (Front)	10, 000	
		Waterloo (Front)	1, 000	
Diffusion_All	*train*	Waymo	189, 847	1, 626, 989
		BDD100K	77, 441	
		Stellantis	19, 573	
		Boxy	345, 936	
		Audi	12, 120	
		NuImages (Toutes)	1, 064, 803	
		Waterloo (Toutes)	107, 116	
	valid	Waymo	5, 976	50, 000
		BDD100K	2, 422	
		Stellantis	566	
		Boxy	5, 557	
		Audi	377	
		NuImages (Toutes)	33, 443	
		Waterloo (Toutes)	1, 659	

Les dimensions des images ont été réduites à 320×640 (au lieu de 640×1280) pour des raisons de mémoire et de faisabilité dans le cadre de l'entraînement.

Contrairement à l'architecture standard de Stable Diffusion, nous avons décidé de retirer le module CLIP, qui est généralement utilisé pour conditionner le contenu des images générées à partir de textes ou d'autres types d'entrées. Cette simplification est justifiée par notre besoin spécifique de travailler sur des images en fonction de niveaux d'erreur visuels, sans inclure de conditionnement textuel. Cela permet également de réduire la complexité du modèle et de concentrer nos efforts sur les aspects essentiels liés à la diffusion et à la qualité visuelle des images.

L'autoencodeur utilisé dans notre modèle repose sur une architecture de Variational Autoencoder (VAE) composée de multiples blocs ResNet et d'une couche d'Attention. Ce dernier encode une image d'entrée de taille 320×640 dans un espace latent réduit de dimension 40×80. Plus précisément, pour une image $x \in \mathbb{R}^{3 \times H \times W}$ dans l'espace RGB, l'encodeur E encode l'image en une représentation latente $z = E(x)$, tandis que le décodeur D reconstruit l'image originale à partir du latent, donnant $\hat{x} = D(z) = D(E(x))$. La représentation latente z se trouve dans un espace de dimension réduite $z \in \mathbb{R}^{c \times h \times w}$, où $f = H/h = W/w$ représente le facteur de réduction. Dans notre cas, nous utilisons $f = 8$, ce qui permet une compression significative des dimensions tout en préservant les informations essentielles. Pour optimiser la reconstruction des images, l'autoencodeur est entraîné en intégrant deux composants principaux dans sa fonction de coût :

- Une fonction de coût perceptuelle (LPIPS, cf. Section 3.2.2), qui mesure la différence perceptuelle entre l'image originale x et l'image reconstruite $\hat{x}$, en se basant sur les caractéristiques extraites par un réseau pré-entraîné.
- Un discriminateur de type PatchGAN, comme proposé dans [Isola et al., 2017], basé sur un classificateur convolutionnel, qui évalue si l'image reconstruite est "réelle" ou "fausse". Ce discriminateur, en ne

pénalisant que la structure à l'échelle de petits blocs (patches) d'image, améliore la qualité des détails dans la reconstruction.

Pour l'entraînement, nous utilisons les datasets Diffusion_Front$_{train}$ et Diffusion_Front$_{valid}$ (cf. Tableau 6.7).

Le UNet est responsable de la mise en œuvre du processus de diffusion, qui constitue le cœur de la génération d'images dans notre modèle. Les modèles de diffusion sont des modèles probabilistes conçus pour apprendre une distribution de données $p(x)$ en débruitant progressivement une variable aléatoire initialement distribuée selon une loi normale. Ce processus peut être vu comme l'apprentissage du processus inverse d'une chaîne de Markov fixe de longueur T, où une séquence d'images bruitées $x_1, x_2, ..., x_T$ est progressivement reconvertie en une image nette x_0. Pour la synthèse d'images, les modèles de diffusion les plus performants s'appuient sur une version ajustée de la borne inférieure variationnelle (Evidence Lower Bound, ELBO) sur $p(x)$, qui guide l'apprentissage des scores de débruitage. En d'autres termes, ces modèles peuvent être interprétés comme une séquence d'autoencodeurs de débruitage $\epsilon_\theta(x_t, t)$, pour $t = 1, \dots, T$, où chaque autoencodeur est entraîné à prédire une version débruitée de son entrée x_t, une version bruitée de l'image initiale x_0. L'objectif d'entraînement du modèle de diffusion peut être simplifié comme suit :

$$L = \mathbb{E}_{x,\epsilon\sim\mathcal{N}(0,1),t}\left[\| \epsilon - \epsilon_\theta(x_t, t) \|_2^2\right] \tag{6.5}$$

où t est échantillonné uniformément dans l'ensemble $1, ..., T$, $\epsilon \sim \mathcal{N}(0, 1)$ est un bruit gaussien ajouté, et $\epsilon_\theta(x_t, t)$ est la prédiction du modèle pour le bruit. Cet objectif vise à minimiser l'écart entre le bruit réellement ajouté et celui prédit par le modèle, permettant ainsi au réseau d'apprendre à débruiter les images à chaque étape t de la diffusion inverse.

Nous entraînons deux modèles UNet sur les datasets Diffusion_Front (comprenant uniquement des images provenant des caméras frontales) et Diffusion_All (incluant des images issues de toutes les caméras) qu'on nomme respectivement diff_front et diff_all. Ces deux ensembles de données permettent de comparer l'impact de la diversité des angles de vue sur la qualité des images générées par les modèles de diffusion. Pour évaluer les performances des modèles entraînés, nous utilisons deux métriques standard en génération d'images (cf. Section 3.4.1 : Inception Score (IS) et Frechet Inception Distance (FID). L'Inception Score mesure la diversité et la qualité visuelle des images générées, tandis que le FID évalue la proximité statistique entre les images générées et les données réelles. Ces métriques offrent une vision complémentaire de la qualité des modèles de diffusion.

Les résultats obtenus pour les deux configurations de modèles sont récapitulés dans le Tableau 6.8. Le modèle diff_all se distingue par des résultats supérieurs, avec un IS plus élevé, indiquant une meilleure diversité et qualité des images générées, et un FID inférieur, reflétant une plus grande proximité avec la distribution des données réelles. Ces observations soulignent que l'entraînement sur un ensemble de données plus diversifié, incluant plusieurs points de vue caméra, a permis au modèle diff_all de capturer une plus grande variété de scénarios.

TABLE 6.8 – Performances des modules de diffusion évaluées sur 300 échantillons et comparées aux données du dataset Diffusion_Front$_{valid}$ et Diffusion_All$_{valid}$.

Modèles	**IS↑**	**FID↓**
diff_front	3.515	3.32
diff_all	4.82	2.012

Nous générons un ensemble d'images de manière non conditionnée à l'aide des deux modèles diff_front et diff_all. Les résultats obtenus sont présentés dans les Figures 6.9 et 6.10. Ces figures illustrent des exemples d'images synthétisées par chaque modèle. Les performances des deux modèles semblent globalement similaires en termes de qualité visuelle. Cependant, le modèle diff_all se distingue par une plus grande diversité dans les générations, notamment en ce qui concerne les points de vue des caméras (cf. Figure 6.10). Cela s'explique par la richesse des données d'entraînement issues de plusieurs perspectives (frontales, latérales, arrière), ce qui enrichit les capacités génératives du modèle. Toutefois, toutes les images générées ne sont pas exploitables, c'est-à-dire qu'elles ne sont pas toujours suffisamment réalistes pour une utilisation pratique. Certaines images présentent des artefacts ou des incohérences, ce qui limite leur qualité perçue. De plus, on observe une nette disparité entre les classes d'objets générés : les véhicules sont généralement de bonne qualité visuelle, tandis que les piétons sont représentés de manière beaucoup moins détaillée, avec une qualité nettement inférieure. Les cyclistes, quant à eux, sont extrêmement rares dans les générations, voire absents dans de nombreux cas. Ces observations mettent en évidence que les images générées reflètent étroitement les caractéristiques des données

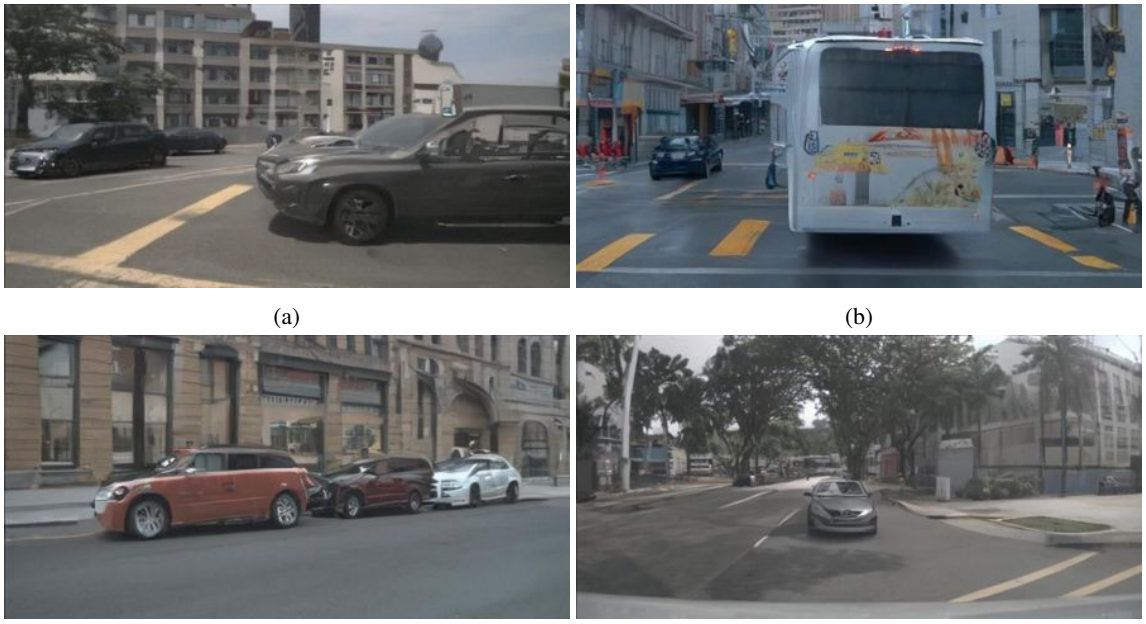
d'entraînement, soulignant ainsi l'importance d'entraîner les modèles de diffusion sur un volume plus important et diversifié de données. En parallèle, il est crucial d'améliorer la cohérence visuelle globale des images générées afin d'en accroître leur réalisme et leur exploitabilité dans des applications pratiques.



(a) (b)

(c) (d)

FIGURE 6.9 – Exemples d'images générées avec le modèle diff_front.



(a) (b)

(c) (d)

FIGURE 6.10 – Exemples d'images générées avec le modèle diff_all.

Une suggestion pour améliorer la qualité et la cohérence sémantiques des images générées serait d'exploiter les annotations disponibles dans les données d'entraînement. Ces annotations pourraient être utilisées en combinaison avec un module CLIP pour conditionner la génération d'images durant l'apprentissage du modèle. En introduisant une correspondance explicite entre les annotations et les caractéristiques visuelles à préserver, le modèle pourrait mieux capturer les éléments essentiels de chaque classe, notamment les minoritaires comme les

cyclistes, tout en garantissant une plus grande précision dans les détails visuels. Après cette phase d'entraînement conditionné, il serait alors possible de générer des images de manière inconditionnelle, tout en bénéficiant de la robustesse et de la cohérence acquises lors de l'apprentissage supervisé.

6.2.4 Conclusion

À ce stade de l'application de la méthodologie d'obtention de garanties de performances, nous disposons des éléments clés constituant le système *Field Monitoring* que nous souhaitons évaluer et garantir. Nous disposons de deux compresseurs entraînés, STF et QRes-VAE, ainsi que de trois détecteurs d'objets, YOLOv7, Faster R-CNN, et RetinaNet. Ces modèles représentent des choix variés en termes d'architectures, de capacités et de compromis entre précision et rapidité, ce qui permet une évaluation complète et diversifiée des performances du système. Enfin, nous avons également intégré des modèles génératifs basés sur la diffusion pour explorer les scénarios critiques et les cas limites susceptibles de révéler les vulnérabilités des modules étudiés. Les performances des compresseurs et des détecteurs ont été établies dans leur configuration actuelle, et notre objectif dans cette phase de l'étude n'est pas de chercher à améliorer ces performances, mais plutôt d'évaluer l'interaction entre la compression et la détection. En d'autres termes, nous allons examiner comment les distorsions induites par la compression affectent les capacités des détecteurs à identifier, localiser et classer correctement les objets présents dans les images. Dans la suite, nous analyserons l'effet de la compression sur les performances des détecteurs d'objets.

6.3 Analyse de l'effet de la compression sur la détection

Avant d'estimer des garanties de performances sur notre application Field Monitoring, nous réalisons une première analyse de l'impact de la compression sur la détection d'objets.

Il est largement établi dans la littérature scientifique que toute distorsion dans une image peut dégrader considérablement les performances des applications basées sur l'apprentissage profond. Qu'il s'agisse de flou, de bruit ou d'autres formes de dégradations, les réseaux neuronaux convolutifs (CNN) subissent une perte systématique de précision. La compression d'image, en introduisant des distorsions, suit cette même tendance. Des études telles que celles de [Feng et al., 2016, Vasiljevic et al., 2017, Poyser et al., 2021, Ehrlich, 2022, Aqqa et al., 2024] ont démontré qu'une compression plus agressive impacte négativement la détection d'objets, réduisant à la fois le nombre d'objets détectés et la confiance dans ces détections. Cette perte de performance est souvent attribuée à une inadéquation entre les données d'entraînement, généralement de haute qualité, et les images compressées plus dégradées. Cependant, ces travaux se concentrent principalement sur des méthodes de compression conventionnelles telles que JPEG, JPEG2000 ou VVC. Ils n'explorent pas l'effet des méthodes de compression par apprentissage profond, qui adaptent dynamiquement la compression à la complexité géométrique et aux informations présentes dans chaque image. Dans notre cas, nous utilisons des modèles de compression basés sur le deep learning, tels que STF et QRes-VAE, qui introduisent des dynamiques de compression et de distorsion fondamentalement différentes.

L'objectif de cette section est d'analyser l'impact des compresseurs basés sur le deep learning, dans leurs configurations agnostique et ROI, sur les performances des modèles de détection d'objets. Pour cela, nous examinons, dans un premier temps, l'évolution de l'Average Precision (AP@.5), une métrique clé présentée dans la Figure 3.35, en fonction du taux de compression (bpp). En suivant le processus illustré dans la Figure 6.11, nous évaluons l'effet de la compression sur chaque classe d'objets (véhicule, piéton, cycliste), permettant ainsi une analyse détaillée de la manière dont les compresseurs STF et QRes-VAE influencent les performances des détecteurs YOLOv7, Faster R-CNN et RetinaNet. Cette étude vise à quantifier précisément les impacts observés, à identifier les classes les plus sensibles à la compression, et à mieux comprendre l'effet de chaque taux de compression sur les résultats.

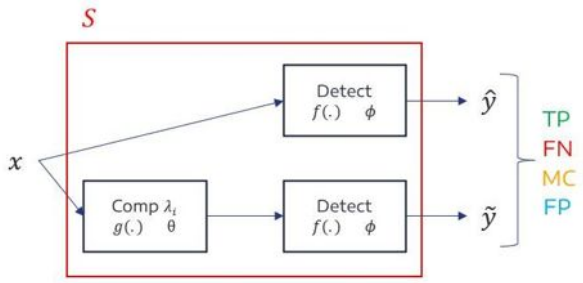
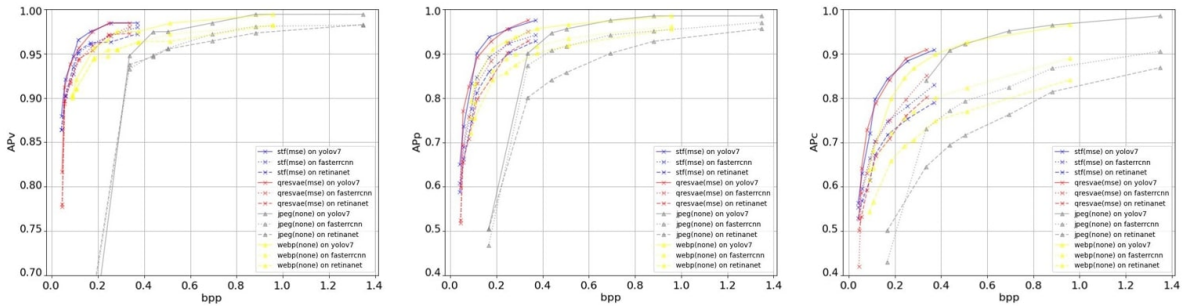


FIGURE 6.11 – Processus d'analyse des erreurs de détection avec et sans la compression.

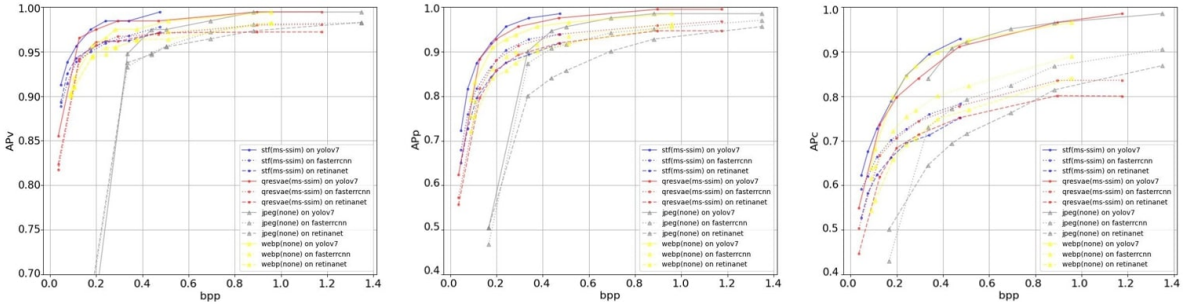
Tout d'abord, comme anticipé, les modèles de compression d'images basés sur l'apprentissage réduisent notablement les performances de tous les détecteurs, mais dans une moindre mesure comparativement aux méthodes classiques comme JPEG et WebP. Dans la Figure 6.12, STF (bleu) et QRes-VAE (rouge) atteignent systématiquement des scores AP@.5 plus élevés que JPEG (gris) et WebP (jaune). Cependant, pour les piétons, illustré dans la Figure 6.12b, WebP rivalise de près avec stf(ms-ssim) et qresvae(ms-ssim). Entre les compresseurs appris, STF (bleu) et QRes-VAE (rouge) montrent des impacts similaires sur tous les détecteurs, avec des fluctuations minimes. Toutefois, stf(ms-ssim) surpasse légèrement qrevae(ms-ssim) sur l'ensemble des classes, indiquant que STF exerce globalement un effet moindre sur les détecteurs.

Ensuite, en comparant les métriques MSE et MS-SSIM entre les Figures 6.12a et 6.12b, on observe que pour YOLOv7, les performances des compresseurs STF et QRes-VAE sont similaires. Cependant, pour la classe des piétons, les compresseurs entraînés avec MSE obtiennent des scores AP@.5 supérieurs pour des débits binaires inférieurs à 0.4 bpp. Cela suggère que l'entraînement avec MSE peut être plus adapté pour réduire l'impact de la compression sur les détecteurs dans certaines classes d'objets.

Enfin, la Figure 6.12 montre qu'à un débit binaire fixé, YOLOv7 (ligne pleine) surpasse constamment Faster R-CNN (ligne pointillée), qui lui-même surpasse RetinaNet (ligne en pointillés). Cela met en évidence une meilleure robustesse de YOLOv7 face aux distorsions induites par la compression. Par ailleurs, une corrélation semble exister entre les performances intrinsèques des détecteurs et leur résistance à la compression. Par exemple, comme les détecteurs excellent dans la détection de véhicules (cf. Table 6.6), ils affichent également une meilleure robustesse dans cette classe. Toutefois, cette corrélation ne s'applique pas systématiquement : YOLOv7 démontre une plus grande robustesse à la compression que Faster R-CNN et RetinaNet, malgré des performances moindres sur certaines classes dans les conditions standards.



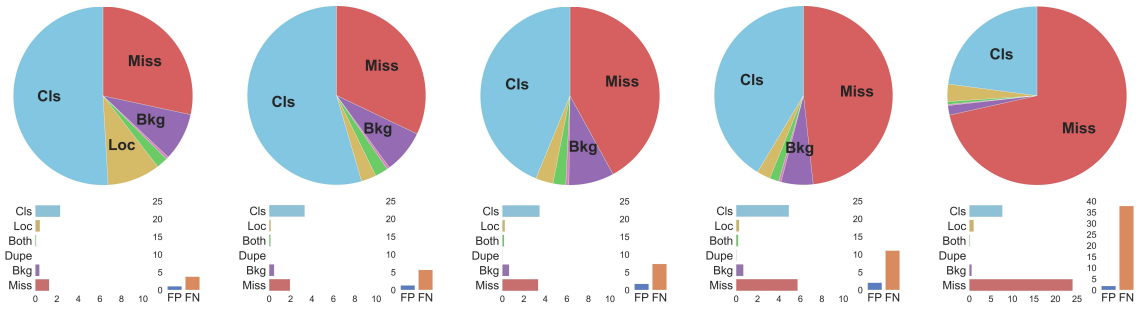
(a) Compresseurs entraînés avec la métrique MSE.



(b) Compresseurs entraînés avec la métrique MS-SSIM.

FIGURE 6.12 – Évolution des scores d'Average Precision pour les véhicules (APv), les piétons (APp), et les cyclistes (APc) en fonction du débit binaire.

En complément du mAP, l'analyse des erreurs à l'aide de la méthode TIDE (cf. section 3.3.2) pour nos trois détecteurs à différents taux de compression avec STF(mse) révèle des tendances intéressantes. Les Figures 6.13, 6.14, et 6.15 montrent que, globalement, la compression accroît significativement la quantité de FN, c'est-à-dire que certains objets détectés dans les images non compressées ne le sont plus après compression. Dans une moindre mesure, la compression augmente également les FP, traduisant la détection d'objets inexistants dans l'image d'origine. Cependant, YOLOv7 se distingue en générant moins de FP comparé à FasterRCNN et RetinaNet. Plus précisément, pour YOLOv7, à des faibles niveaux de compression (bpp élevé, cf. Figures 6.13a et 6.13b), la majorité des erreurs sont dues à des problèmes de classification (Cls). À mesure que le taux de compression augmente, la proportion d'objets non détectés (Miss) devient prédominante, reflétant l'impact croissant de la compression sur les performances de détection.

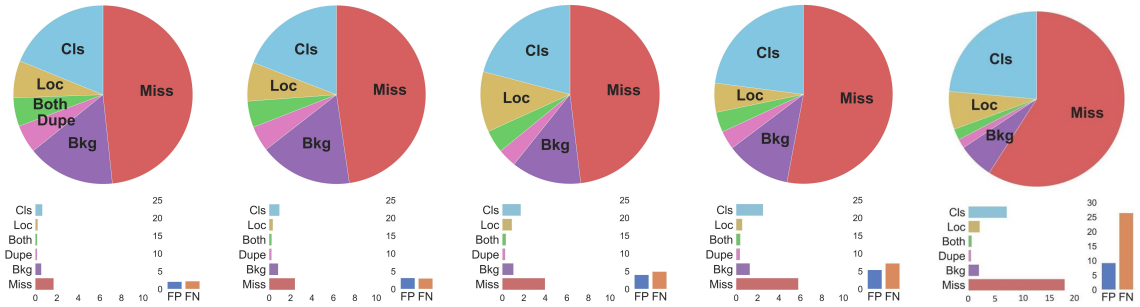


(a) YOLOv7 λ_7 $bpp = 0.348$ (b) YOLOv7 λ_6 $bpp = 0.236$ (c) YOLOv7 λ_5 $bpp = 0.155$ (d) YOLOv7 λ_4 $bpp = 0.104$ (e) YOLOv7 λ_1 $bpp = 0.036$

FIGURE 6.13 – Analyse de l'impact du compresseur stf(mse) sur le détecteur YOLOv7 à l'aide de la méthode TIDE (cf. Figure 8.1 en annexes pour les hyperparamètres λ_2 et λ_3).

Pour FasterRCNN (cf. Figure 6.14), à la différence de YOLOv7, les erreurs de détection manquées (Miss) dominent quel que soit le taux de compression. À mesure que le taux de compression augmente, on observe une augmentation notable des détections manquées (Miss) et des erreurs de classification (Cls). Par ailleurs, les erreurs de localisation (Loc) et celles dues à la confusion avec l'arrière-plan (Bkg) augmentent également, bien

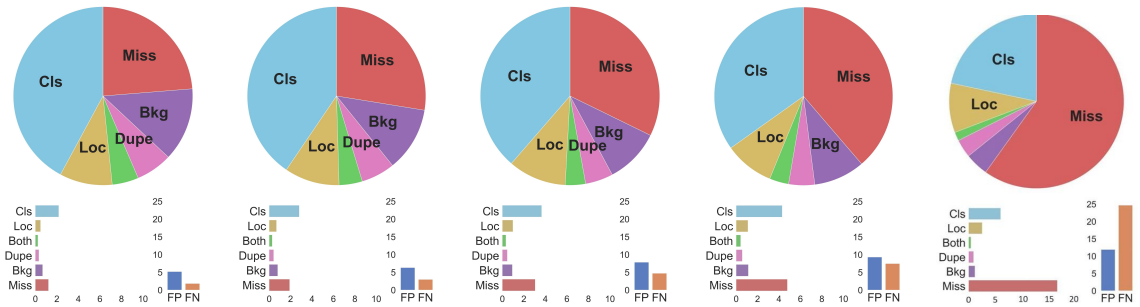
que dans une moindre mesure.



(a) FasterRCNN λ_7 $bpp = 0.348$ (b) FasterRCNN λ_6 $bpp = 0.236$ (c) FasterRCNN λ_5 $bpp = 0.155$ (d) FasterRCNN λ_4 $bpp = 0.104$ (e) FasterRCNN λ_1 $bpp = 0.036$

FIGURE 6.14 – Analyse de l'impact du compresseur stf(mse) sur le détecteur FasterRCNN à l'aide de la méthode TIDE (cf. Figure 8.2 en annexes pour les configurations avec λ_2 et λ_3).

Enfin, pour RetinaNet (cf. Figure 6.15), à faible taux de compression (cf. Figures 6.15a, 6.15b, et 6.15c), les erreurs de fausses détections (FP) constituent la majorité des erreurs. Le comportement global est similaire à celui de YOLOv7, mais RetinaNet présente une proportion légèrement plus élevée d'autres types d'erreurs, notamment les erreurs de localisation (Loc), les erreurs combinées (Both), les doublons (Dupe), et celles dues à la confusion avec l'arrière-plan (Bkg). Ces erreurs augmentent également à mesure que le taux de compression devient plus élevé.



(a) RetinaNet λ_7 $bpp = 0.348$ (b) RetinaNet λ_6 $bpp = 0.236$ (c) RetinaNet λ_5 $bpp = 0.155$ (d) RetinaNet λ_4 $bpp = 0.104$ (e) RetinaNet λ_1 $bpp = 0.036$

FIGURE 6.15 – Analyse de l'effet du compresseur stf(mse) sur le détecteur RetinaNet à l'aide de la méthode TIDE (cf. Figure 8.3 en annexes pour les configurations avec λ_2 et λ_3).

En utilisant de nouveau le processus illustré dans la Figure 6.11 pour évaluer la compression ROI avec les compresseurs stf(mse) et qresvae(mse) appliqués au détecteur YOLOv7, la Figure 6.16 montre une amélioration significative des scores AP@.5 par classe. Cela indique que la compression ROI induit moins d'erreurs de détection comparée à la compression agnostique.

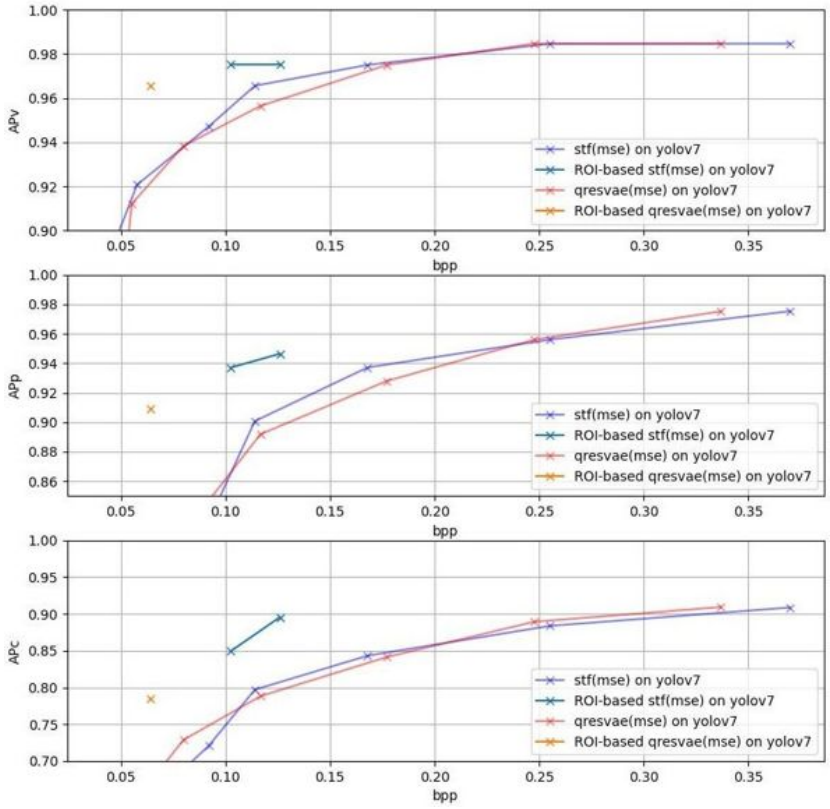
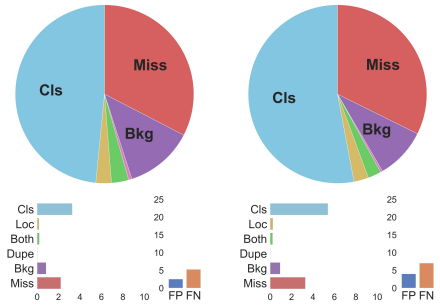


FIGURE 6.16 – Comparison between the impact of our previous STF and QRes-VAE models and the new ROI-based models on YOLOv7 and FasterRCNN over all classes (mAP). The green arrows show the increase of mAP.

Cette observation est corroborée par l'application de la méthode TIDE : pour des taux de compression équivalents, en comparant les Figures 6.13c et 6.13d avec la Figure 6.17, on observe une nette réduction des erreurs, en particulier des FN, bien que les FP augmentent légèrement. Ces résultats démontrent que la compression ROI permet d'atteindre des taux de compression plus élevés tout en maintenant une qualité de détection satisfaisante.



(a) YOLOv7 λ_9
$bpp = 0.125$

(b) YOLOv7 λ_8
$bpp = 0.108$

FIGURE 6.17 – Évaluation avec la méthode TIDE de l'effet du compresseur STF(mse) ROI sur le détecteur YOLOv7.

Des résultats similaires sont observés pour les compresseurs stf(ms-ssim), qresvae(mse) et qresvae(ms-ssim), comme illustré dans les Figures 8.4 à 8.12 en annexes.

En conclusion, bien que la compression par deep learning puisse introduire du flou et des artefacts visibles à des niveaux de compression élevés, elle offre des avantages notables par rapport aux méthodes traditionnelles telles que JPEG et WebP. Ces réseaux de neurones permettent d'atteindre des taux de compression plus élevés tout en maintenant une qualité de reconstruction supérieure. L'impact de cette compression sur les détecteurs

d'objets varie toutefois selon les architectures. YOLOv7 se distingue par une meilleure robustesse, montrant principalement des erreurs sous forme de faux négatifs (FN), même à des taux de compression élevés. En revanche, FasterRCNN et RetinaNet sont davantage affectés, avec un taux élevé de faux positifs (FP) et d'autres erreurs. Ainsi, étant donné sa résilience accrue à la compression et sa capacité à minimiser les erreurs, YOLOv7 sera le détecteur privilégié pour la suite de cette étude.

6.4 Estimation des probabilités d'erreur

Récapitulons les étapes de la méthodologie appliquées jusqu'à présent.

Tout d'abord, nous avons défini le système S dans la section 5.2. Ce système est représenté dans la Figure 6.18 et se compose de deux modules principaux : un module de compression d'image, représenté par la fonction $g(.)$, et un module de détection d'objets, représenté par $f(.)$. L'objectif principal est d'estimer la probabilité que les distorsions causées par la compression modifient les prédictions du détecteur. Plus précisément, nous cherchons à évaluer $P(e(x) \geq \epsilon) = P(e(f(x), f(g(x))) \geq \epsilon)$, où e est une fonction d'erreur et ϵ un seuil de tolérance. Pour ce faire, nous avons défini une fonction d'erreur $e(x) = \#FN + \#FP + \#MC$, qui comptabilise les faux négatifs ($\#FN$), les faux positifs ($\#FP$), et les mauvaises classifications ($\#MC$) observés dans les images compressées par rapport aux images non compressées.

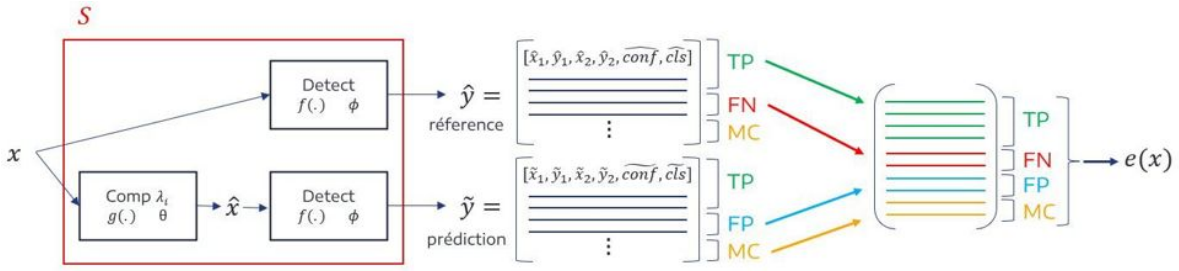


FIGURE 6.18 – Illustration du système S (encadré en rouge) et du processus de calcul de la fonction d'erreur $e(x)$.

Nous avons ensuite établi un ODD couvrant un large éventail de situations pouvant être enregistrées par une caméra embarquée située à l'avant d'un véhicule. Cet ODD inclut des environnements urbains, ruraux et autoroutiers, ainsi que des scénarios variés allant des situations classiques et fréquentes à des cas rares et critiques. En conséquence, nous avons procédé à une acquisition de données combinant des images naturelles issues de bases open source et privées, ainsi que des images synthétiques générées dans un environnement virtuel 3D simulé. Ces données forment la base pour l'entraînement et l'évaluation de notre chaîne compression-détection.

À partir de l'état de l'art, nous avons sélectionné et entraîné différents modèles de compression (STF et QRes-VAE) et de détection (YOLOv7, Faster R-CNN, RetinaNet). Une première analyse a permis de montrer que YOLOv7 est le détecteur le plus robuste face aux distorsions induites par la compression. Par souci de simplicité, nous nous concentrerons exclusivement sur YOLOv7 pour la suite de l'étude. Nos modules de compression ont été entraînés selon des configurations agnostiques et ROI (Region of Interest) avec plusieurs hyperparamètres λ, offrant une large gamme de taux de compression moyens. Ces modèles, associés au détecteur YOLOv7, composent notre chaîne compression-détection, qui est maintenant complètement opérationnelle.

Dans cette section, nous appliquons la dernière étape de la méthodologie pour garantir les performances de cette chaîne. Dans notre cas, la détection porte sur trois classes principales : véhicules, piétons et cyclistes. En appliquant la méthode d'estimation de la probabilité d'erreur $P(e(x) \geq \epsilon)$ dans le cadre avec classes (cf. Section 4.3.2), nous décomposons cette probabilité comme suit :

$$P(e(X) \geq \epsilon) = P(e(X) \geq \epsilon | C \in X) \cdot P(C \in X) \tag{6.6}$$

avec l'évènement $\{C \in X\}$ correspond à l'ensemble des données X dans lesquelles les classes de C interviennent dans les sorties du système S. Cette décomposition permet d'analyser l'effet de la compression de manière globale, mais également pour chaque classe individuellement.

Ensuite, comme décrit dans la section 4.3.2, afin d'estimer ces probabilités, nous procédons en définissant un partitionnement de l'espace ODD (illustré dans la Figure 6.19). Une fois ce partitionnement établi, nous appliquons le théorème de l'espérance totale pour décomposer les calculs et estimer chaque composante.

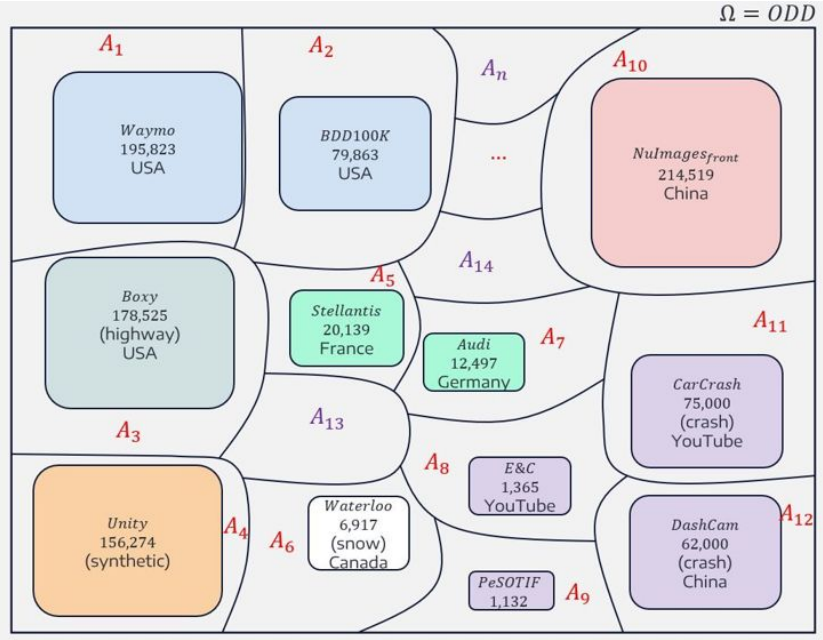


FIGURE 6.19 – Représentation idéalisée et abstraite de nos jeux de données dans l'espace défini par l'ODD et du partitionnage.

Nous rappelons ci-dessous les deux étapes principales d'estimation de la probabilité d'erreur, intra-dataset et inter-dataset, qui seront appliquées dans la suite.

1. **Estimation intra-jeu de données** (*intra-dataset*) par bootstrap (cf. Algorithme 1) avec un intervalle de confiance de niveau $1-\beta$:

$$\mathbb{E}[Z|X \in D_j] \approx \hat{\theta}_j = \frac{1}{|D_j|} \sum_{x \in D_j} \mathbb{1}_{\{e(x) \geq \epsilon\}} \tag{6.7}$$

$$\mathbb{E}[Z|C \in X, X \in D_j] \approx \hat{\theta}_j^{e,C} = \frac{1}{|D_j|} \sum_{x \in D_j} \mathbb{1}_{\{C \in X\} \bigcap \{e(x) \geq \epsilon\}} \tag{6.8}$$

$$\mathbb{E}[V|X \in D_j] \approx \hat{\theta}_j^{C} = \frac{1}{|D_j|} \sum_{x \in D_j} \mathbb{1}_{\{C \in X\}} \tag{6.9}$$

2. **Estimation inter-jeu de données** (*inter-dataset*) par bootstrap épistémique (cf. Algorithme 2) avec un intervalle de confiance de niveau $1-\alpha$:

$$P(e(X) \geq \epsilon) = \mathbb{E}[Z(X)] \approx \hat{\theta} = \sum_{j=1}^{d} \hat{\theta}_j \cdot p_j \tag{6.10}$$

$$P(e(X) \geq \epsilon | C \in X) = \mathbb{E}[Z(X)|C \in X] \approx \hat{\theta}^{e,C} = \sum_{j=1}^{d} \hat{\theta}_j^{e,C} \cdot p_j \tag{6.11}$$

$$P(C \in X) = \mathbb{E}[V(X)] \approx \hat{\theta}^{C} = \sum_{j=1}^{d} \hat{\theta}_j^{C} \cdot p_j \tag{6.12}$$

Nous appliquerons ces deux étapes d'estimation avec un niveau de confiance de $1-\beta = 1-\alpha = 99\%$. Pour garantir une précision optimale, nous utiliserons un échantillonnage bootstrap avec $B = 10,000$ itérations pour l'estimation intra-dataset et $B = 100,000$ itérations pour l'estimation inter-dataset. Ces choix visent à minimiser l'incertitude des estimations tout en assurant leur robustesse statistique.

Dans la suite, nous adopterons un code couleur pour représenter chaque estimation et chaque dataset dans les figures. Les grandeurs $\hat{\theta}_j, \hat{\theta}_j^{e,C}, \hat{\theta}_j^{C}$ et $\hat{\theta}, \hat{\theta}^{e,C}, \hat{\theta}^{C}$ seront distinguées à l'aide de différents types de lignes :

- Ligne pleine : pour représenter $\hat{\theta}_j$ et $\hat{\theta}$.

- Ligne pointillée : pour représenter $\hat{\theta}_j^{e,C}$ et $\hat{\theta}^{e,C}$.
- Ligne en tirets : pour représenter $\hat{\theta}_j^{C}$ et $\hat{\theta}^{C}$.

Ce code visuel permettra une lecture claire des variations de chaque grandeur pour les différents datasets et estimations. Avant d'appliquer ces étapes d'estimation, introduisons quelques notations pour formaliser notre étude. Nous considérons quatre ensembles de classes C définis comme suit :

- $C_0 = \{v\}$,
- $C_1 = \{p\}$,
- $C_2 = \{c\}$,
- $C_3 = \{v, p, c\}$,

avec v, p et c désignent respectivement les classes "véhicule", "piéton" et "cycliste". Un cas particulier d'intérêt est celui où $\epsilon = 1$. En effet, la probabilité $P(e(x) \geq 1)$ représente la probabilité que la compression ait un impact sur la détection, autrement dit, qu'elle entraîne au moins une erreur de détection. Cette probabilité est équivalente à $1 - P(e(x) = 0)$, car dans notre contexte, $e(x) \in \mathbb{N}$. Une telle probabilité est un indicateur direct de la robustesse du système face à la compression, et notre objectif serait idéalement de la minimiser autant que possible. Par ailleurs, étant donné que $e(x) \in \mathbb{N}$, il est possible d'estimer la probabilité que l'erreur soit exactement égale à une valeur donnée ϵ en utilisant la relation suivante :

$$P(e(x) = \epsilon) = P(e(x) \geq \epsilon) - P(e(x) \geq \epsilon + 1) \tag{6.13}$$

Cette décomposition fournit une idée claire de la distribution des erreurs dans les images. En effet, elle permet de comprendre si les erreurs sont concentrées autour de faibles valeurs (par exemple, une seule erreur par image, $e(x) = 1$) ou si elles se répartissent sur un plus large éventail de valeurs, indiquant des cas où plusieurs erreurs peuvent coexister dans une même image. Une telle analyse fine de la distribution des erreurs est cruciale pour évaluer l'impact de la compression et cibler des améliorations potentielles pour le système.

6.4.1 Estimation intra-datasets

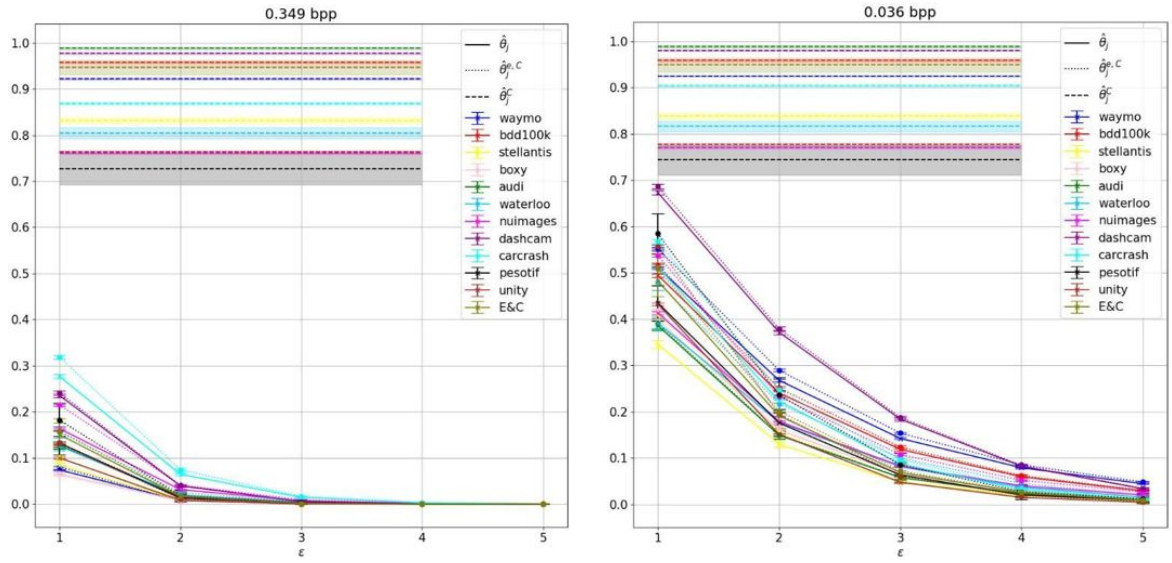
Commençons par l'étape d'estimation intra-datasets afin d'analyser l'évolution de la probabilité d'erreur pour chacun de nos jeux de données isolés.

Prenons comme référence les modèles de compression stf(mse)$_{\lambda_7}$ et stf(mse)$_{\lambda_1}$, correspondant au modèle STF entraîné de manière agnostique avec la métrique de distorsion MSE et les hyperparamètres λ_7 et λ_1. Ces hyperparamètres représentent respectivement les niveaux de compression les plus faibles (0.349 bpp) et les plus élevés (0.036 bpp) disponibles pour ce compresseur. Dans la Figure 6.20, nous présentons les estimations suivantes, conformément aux équations 6.7, 6.8 et 6.9 :

- $\hat{\theta}_j \approx P(e(x) \geq \epsilon | X \in D_j)$, la probabilité d'erreur sur l'ensemble des images d'un dataset.
- $\hat{\theta}_j^{e,C} \approx P(e(x) \geq \epsilon | X \in D_j, C \in X)$, la probabilité d'erreur uniquement sur les images où interviennent les classes de C dans un dataset.
- $\hat{\theta}_j^{C} \approx P(C \in X | X \in D_j)$ la probabilité d'apparition des classes de C dans les images d'un dataset.

Ces probabilités sont reliées par la relation $\hat{\theta}_j \approx \hat{\theta}_j^{e,C} \cdot \hat{\theta}_j^{C}$,illustrant que la probabilité d'erreur globale dépend à la fois des erreurs spécifiques aux classes de C et de leur fréquence d'apparition.

Considérons dans un premier temps l'ensemble des classes $C_3 = \{v, p, c\}$. Dans les Figures 6.20a et 6.20a, on observe que la probabilité d'apparition des classes de C_3 varie entre les datasets. Par exemple, cette probabilité (représentée en pointillés sur la figure) est élevée, entre 0.9 et 1, pour les datasets Waymo, E&C, BDD100K, DashCam et Audi. Cependant, elle est plus faible pour d'autres datasets comme PeSOTIF et NuImages, pouvant descendre jusqu'à 0.7. Cette variabilité impacte directement la relation entre $\hat{\theta}_j$ et $\hat{\theta}_j^{e,C}$, qui deviennent très proches lorsque $\hat{\theta}_j^{C}$ est faible. Pour λ_7 (compression faible), les probabilités d'erreur $\hat{\theta}_j$ restent inférieures à 0.3 (soit 30%) pour tous les datasets et décroissent rapidement avec l'augmentation de ϵ, atteignant quasiment 0 pour $\epsilon = 5$ (cf. Figure 6.20a). Cela indique qu'à faible compression, le nombre d'erreurs par image reste limité, même dans des cas critiques. En revanche, avec λ_1 (compression très forte), les probabilités d'erreur augmentent considérablement pour tous les datasets, atteignant des valeurs beaucoup plus élevées (cf. Figure 6.20b).



(a) YOLOv7, stf(mse) λ_7 (0.349 bpp).

(b) YOLOv7, stf(mse) λ_1 (0.036 bpp).

FIGURE 6.20 – Probabilité intra-dataset $\hat{\theta}_j$, $\hat{\theta}_j^{e,C}$ et $\hat{\theta}_j^{C}$ en fonction de ϵ, pour le système composé de stf(mse) et YOLOv7 pour les niveaux de compression λ_7 (0.349 bpp) et λ_1 (0.036 bpp) dans la configuration $C_3 = \{v, p, c\}$.

En observant le cas particulier où $\epsilon = 1$ (cf. Figure 6.21), on remarque que plus le taux de compression augmente (de λ_7 à λ_1), plus la probabilité d'erreur s'accroît. Cependant, certains datasets se montrent plus sensibles aux taux de compression élevés qu'aux faibles. Par exemple, pour les datasets Stellantis (jaune) et Boxy (rose), à haut débit (λ_7), la probabilité d'erreur de Stellantis est supérieure à celle de Boxy. Mais à mesure que le débit diminue, cette tendance s'inverse : à faible débit (λ_1), Boxy présente une probabilité d'erreur plus élevée que Stellantis.

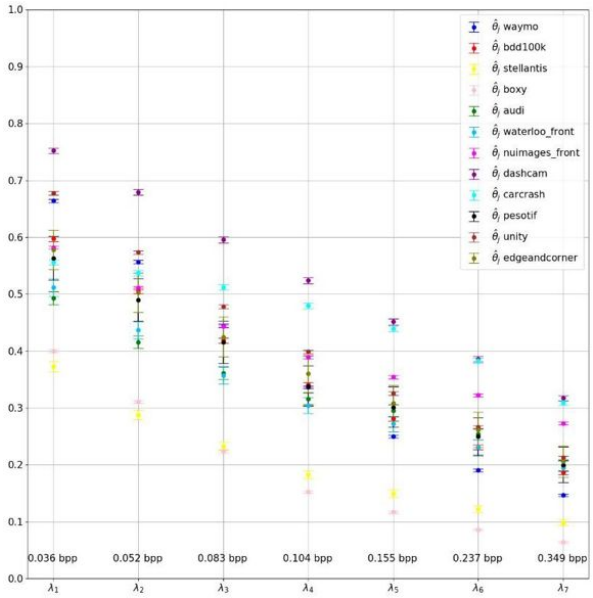
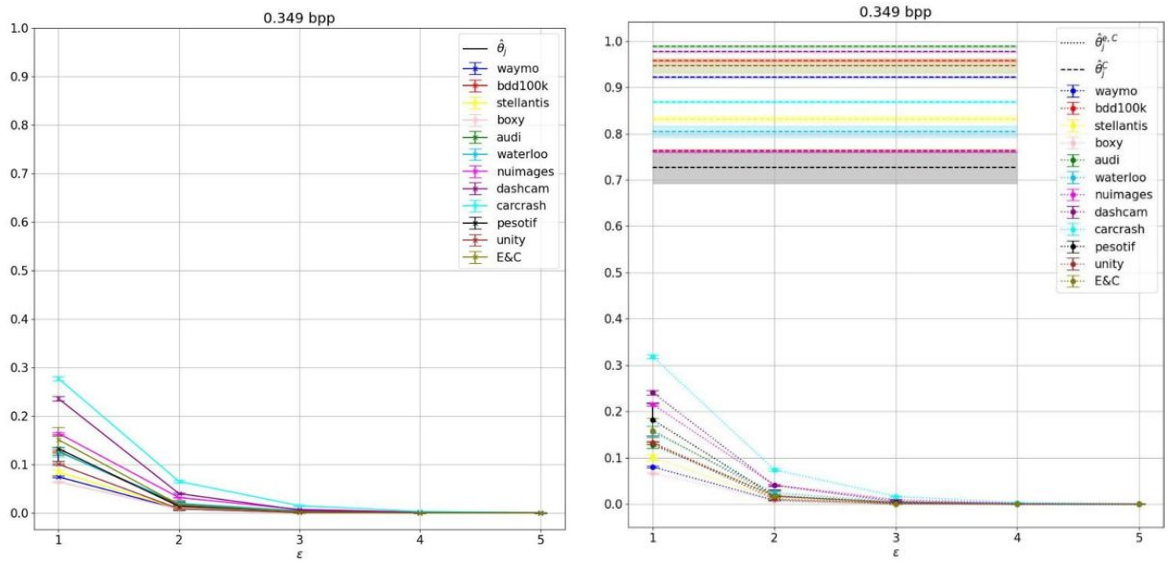


FIGURE 6.21 – Estimations $\hat{\theta}_j$ pour $\epsilon = 1$ pour chaque taux de compression de stf(mse).

Enfin, on observe que pour les datasets contenant le moins d'images ($< 7,000$), tels que Waterloo, E&C et PeSOTIF, les intervalles de confiance pour leurs estimations sont nettement plus larges. Cela est cohérent avec la méthode bootstrap utilisée pour l'estimation. Pour affiner ces estimations, ces datasets nécessiteraient davantage de données, soulignant l'importance d'un échantillonnage suffisant pour garantir des estimations robustes.

Si nous considérons maintenant les ensembles de classes $C_0 = \{v\}$, $C_1 = \{p\}$ et $C_2 = \{c\}$ il devient possible d'évaluer ces probabilités pour chaque classe individuellement. Par exemple, en nous focalisant uniquement sur les véhicules (C_0, cf. Figure 6.22), les résultats obtenus sont très proches de ceux observés avec l'ensemble $C_3 = \{v, p, c\}$. Cette similarité suggère que les véhicules constituent la grande majorité des objets détectés dans les différents datasets et, par conséquent, représentent la majeure partie des erreurs observées. En d'autres termes, la probabilité d'erreur globale est fortement influencée par les erreurs associées aux véhicules, qui dominent les contributions des autres classes. Cela met en évidence l'importance relative de cette classe dans l'évaluation des performances du système.



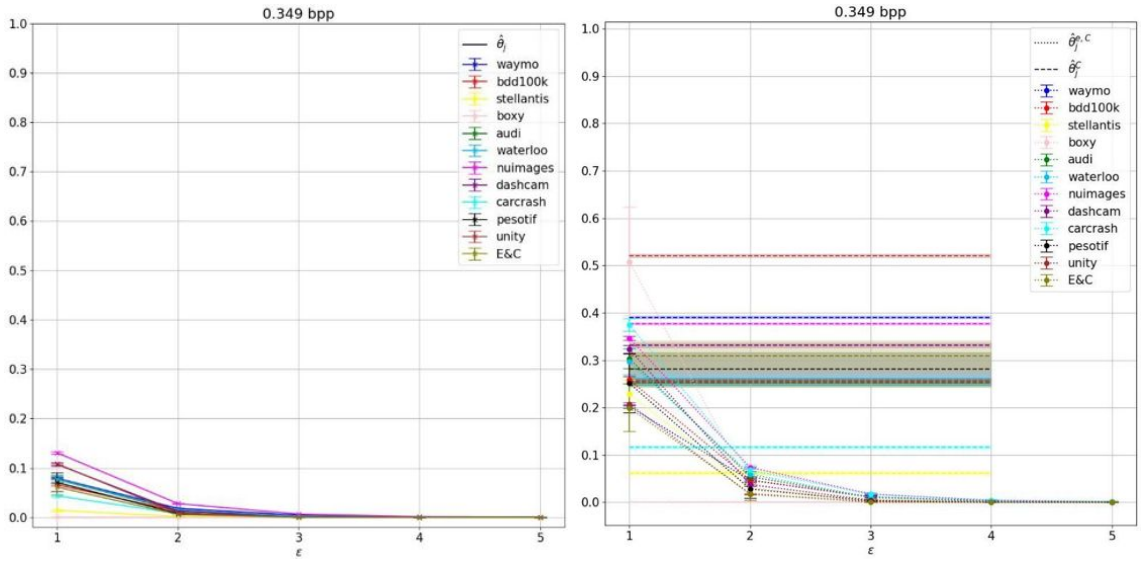
(a) YOLOv7, stf(mse) λ_7 (0.349 bpp).

(b) YOLOv7, stf(mse) λ_7 (0.349 bpp).

FIGURE 6.22 – Probabilité intra-dataset $\hat{\theta}_j$, $\hat{\theta}_j^{e,C}$ et $\hat{\theta}_j^{C}$ en fonction de ϵ, pour le système composé de stf(mse) et YOLOv7 pour le niveau de compression λ_7 (0.349 bpp) dans la configuration $C_0 = \{v\}$.

En analysant spécifiquement les piétons (C_1, cf. Figure 6.23) et les cyclistes (C_2, cf. Figure 6.24), nous constatons que les probabilités d'erreur $\hat{\theta}_j$ pour ces classes sont nettement plus faibles que celles observées pour les véhicules (C_0, cf. Figure 6.22a). Cette différence s'explique en grande partie par la fréquence relative des classes dans les datasets analysés. Les piétons et les cyclistes sont significativement moins fréquents, ce qui limite leur contribution aux probabilités globales d'erreur. En effet, les probabilités d'apparition des piétons dans les datasets restent inférieures à 0.6 (cf. Figure 6.23b), tandis que pour les cyclistes, elles ne dépassent pas 0.4 (cf. Figure 6.24b). Dans certains cas, cette faible fréquence reflète directement la nature des datasets. Par exemple, le dataset Boxy, constitué principalement de scènes autoroutières, montre une probabilité d'apparition presque nulle pour les piétons et les cyclistes. Cela est attendu étant donné le contexte routier où ces classes sont rares. Cependant, lorsqu'un piéton ou un cycliste est détecté dans ce dataset, il est fort probable que l'erreur soit due aux distorsions causées par compression. En revanche, le dataset Unity, issu de simulations, se démarque par les probabilités d'apparition les plus élevées pour ces classes, avec $\hat{\theta}_j^{C} \approx 0.52$ pour les piétons et $\hat{\theta}_j^{C} \approx 0.32$ pour les cyclistes. Cela souligne l'utilité des environnements simulés pour équilibrer la répartition des classes et explorer des scénarios où certaines classes peuvent être sous-représentées dans les données réelles. Les simulations permettent ainsi de combler les lacunes des datasets réels, tout en offrant un contrôle précis sur les conditions expérimentales.

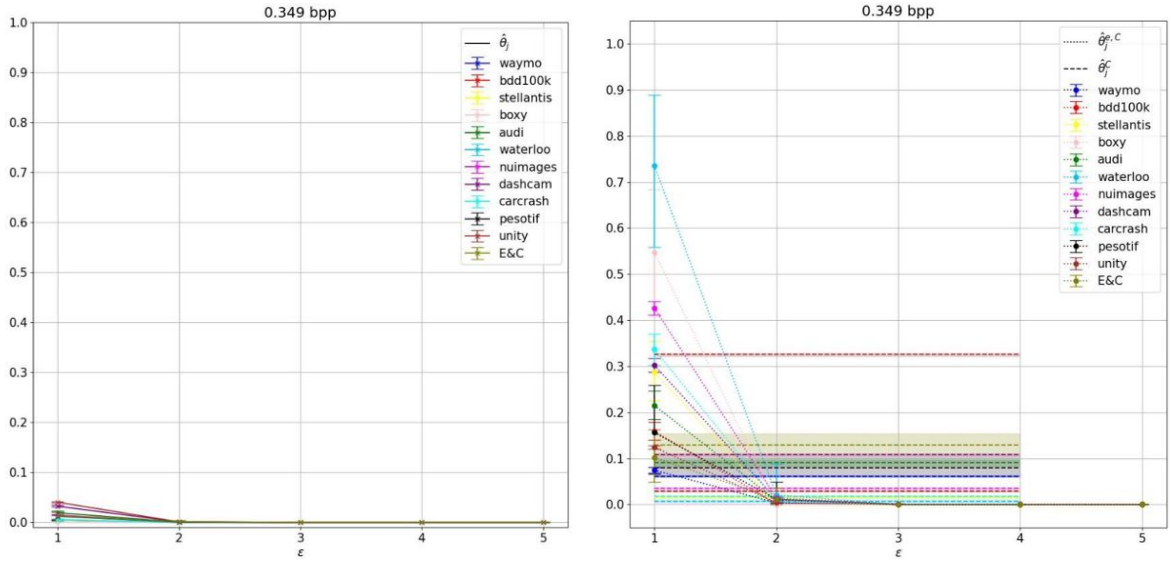
On observe que, en raison de la faible fréquence d'apparition des classes piétons et cyclistes, les intervalles de confiance des estimations $\hat{\theta}_j^{e,C}$ et $\hat{\theta}_j^{C}$ sont sensiblement plus larges que ceux de la classe véhicule. Cette disparité reflète la rareté des échantillons pour ces classes, ce qui engendre une plus grande incertitude dans les estimations probabilistes associées conformément aux principes de la méthode bootstrap.



(a) YOLOv7, stf(mse) λ_7 (0.349 bpp).

(b) YOLOv7, stf(mse) λ_7 (0.349 bpp).

FIGURE 6.23 – Probabilité intra-dataset $\hat{\theta}_j$, $\hat{\theta}_j^{e,C}$ et $\hat{\theta}_j^{C}$ en fonction de ϵ, pour le système composé de stf(mse) et YOLOv7 pour le niveau de compression λ_7 (0.349 bpp) dans la configuration $C_1 = \{p\}$.



(a) YOLOv7, stf(mse) λ_7 (0.349 bpp).

(b) YOLOv7, stf(mse) λ_7 (0.349 bpp).

FIGURE 6.24 – Probabilité intra-dataset $\hat{\theta}_j$, $\hat{\theta}_j^{e,C}$ et $\hat{\theta}_j^{C}$ en fonction de ϵ, pour le système composé de stf(mse) et YOLOv7 pour le niveau de compression λ_7 (0.349 bpp) dans la configurations $C_3 = \{c\}$.

Ces résultats montrent que l'effet de la compression sur la détection d'objets n'est pas négligeable. Pour le niveau de compression le plus faible λ_7, la probabilité d'erreur causée par la compression varie entre 0.05 et 0.3. En d'autres termes, pour un ensemble d'images, on peut s'attendre à ce qu'entre 5% et 30% d'entre elles contiennent au moins une erreur. Cependant, un point rassurant est que ces probabilités diminuent rapidement à mesure que ϵ augmente, ce qui signifie que des erreurs multiples sur une même image deviennent de plus en plus rares. Pour aller plus loin dans l'estimation des probabilités globales, il est nécessaire de passer à la deuxième étape de l'estimation en définissant les hyperparamètres ${p_j}_{j=1}^{d}$. Ces hyperparamètres représentent les probabilités relatives associées à chaque jeu de données, et sont déterminés en fonction d'informations a priori pertinentes pour l'application cible.

6.4.2 Estimation inter-datasets

Pour appliquer la deuxième étape d'estimation, il est essentiel de définir les hyperparamètres $\{p_j\}_{j=1}^{12}$, qui représentent les probabilités associées à chacun des 12 jeux de données. Ces hyperparamètres indiquent la probabilité qu'une donnée aléatoire provienne d'un dataset particulier. Leur définition repose sur des informations a priori, généralement fournies par des experts métiers ayant une connaissance approfondie des scénarios rencontrés par les véhicules et de leur fréquence d'apparition. Dans le secteur automobile, ces experts incluent des ingénieurs en simulation de conduite, des ingénieurs en sécurité active, des spécialistes en modélisation de scénarios ou encore des experts en télémétrie. Leur rôle est crucial pour évaluer la représentativité relative de chaque dataset en fonction des situations réelles ou simulées. En exploitant leurs connaissances, on peut attribuer des poids $\{w_j\}_{j=1}^{12}$ reflétant la probabilité relative de chaque dataset, en tenant compte de leur pertinence pour les cas réels. Ces poids sont ensuite normalisés pour obtenir les hyperparamètres $\{p_j\}_{j=1}^{12}$ conformément à l'équation 4.19, garantissant que $\sum_{j=1}^{12} p_j = 1$. Cependant, dans le cas où de telles informations a priori ne sont pas disponibles, il est toujours possible d'illustrer la méthodologie d'estimation en utilisant des hypothèses simplificatrices.

Distribution uniforme

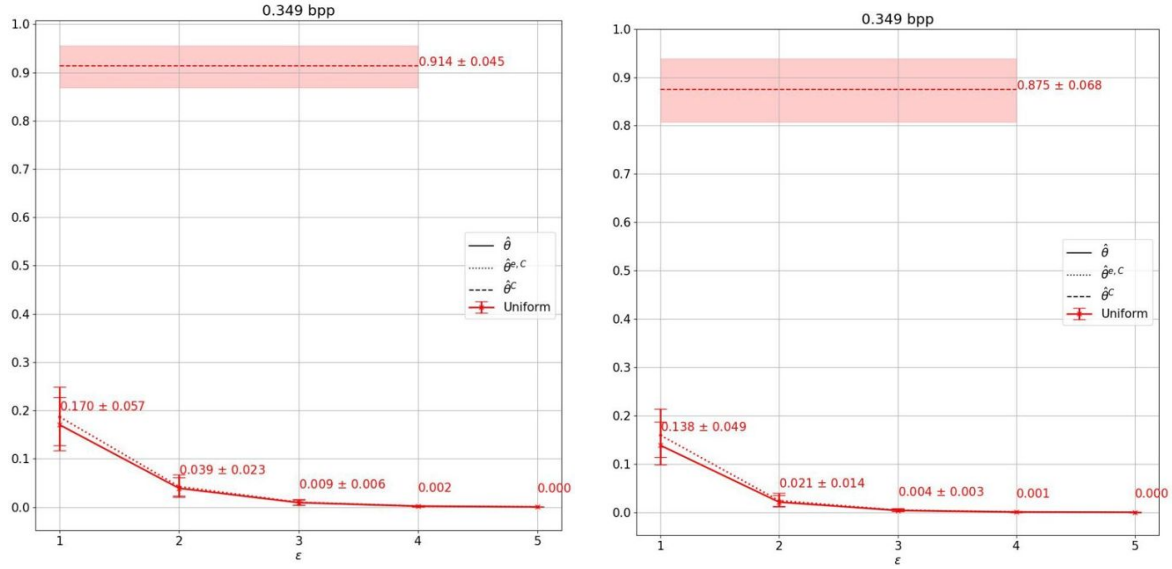
Supposons une distribution uniforme sur l'ensemble des datasets, où chaque jeu de données est considéré comme équiprobable par rapport aux autres. Dans cette hypothèse, les poids $\{w_j\}j = 1^{12}$ sont égaux, indépendamment de leur valeur absolue. Par simplicité, nous attribuons $w_j = 1$ pour $j = 1, \dots, 12$. Les hyperparamètres $\{p_j\}_{j=1:12}$ sont ensuite calculés en normalisant ces poids, aboutissant à une répartition uniforme des probabilités. Cette hypothèse simpliste, bien qu'éloignée de la réalité, fournit une base méthodologique pour les calculs initiaux, comme illustré dans le tableau 6.9. Dans la suite nous verrons une autre distribution plus en adequation avec l'application *Field Monitoring*.

TABLE 6.9 – Tableau des poids w_j et des hyperparamètres p_j selon une distribution uniforme.

Datasets	w_j	p_j
waymo	1	0.083
*bdd*100*k*	1	0.083
stellantis	1	0.083
boxy	1	0.083
audi	1	0.083
waterloo	1	0.083
nuimages	1	0.083
dashcam	1	0.083
carcrash	1	0.083
pesotif	1	0.083
unity	1	0.083
E&C	1	0.083

Concentrons-nous à nouveau sur le modèle de compression stf(mse)$_{\lambda_7}$ (correspondant au plus haut débit, à savoir 0.349 bpp) en combinaison avec YOLOv7. Dans cette analyse, nous appliquons la deuxième étape d'estimation (inter-dataset) en pondérant les estimations intra-dataset obtenues précédemment pour chaque jeu de données à l'aide d'une pondération $\{p_j\}_{j=1:12}$ définie uniformément, conformément au tableau 6.9. Cette hypothèse uniforme attribue une probabilité égale à chaque dataset, reflétant une hypothèse simpliste où tous les datasets ont une importance égale dans l'évaluation globale. Les Figures 6.25 et 6.26 illustrent les estimations globales $\hat{\theta}$, $\hat{\theta}^{e,C}$ et $\hat{\theta}^{C}$ pour les différents ensembles de classes. Pour l'ensembl $C_3 = \{v, p, c\}$ (cf. Figure 6.25a), la probabilité d'erreur globale $\hat{\theta}$ décroît rapidement avec l'augmentation du seuil ϵ, atteignant pratiquement pour $\epsilon \geq 4$. Les erreurs restent majoritairement dominées par la classe des véhicules (v), comme observé dans les estimations intra-dataset, en raison de leur fréquence plus élevée dans les datasets. Pour l'ensemble $C_0 = \{v\}$ (cf. Figure 6.25b), Les résultats pour les véhicules sont similaires à ceux de l'ensemble C_3, confirmant que les véhicules constituent la majorité des occurrences et des erreurs dans les datasets. Cette dominance est évidente

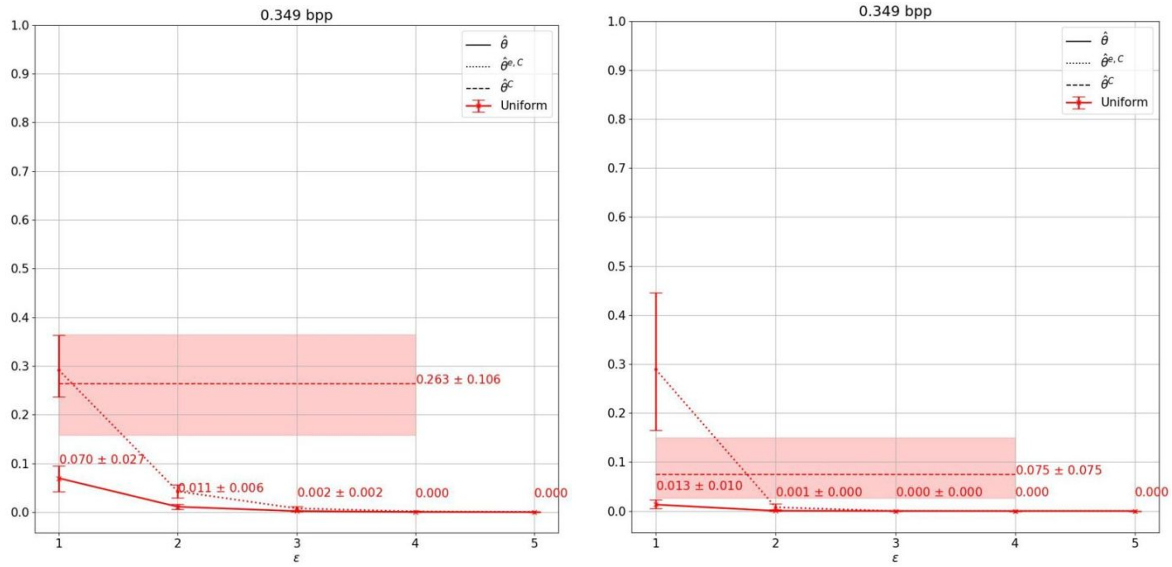
dans les valeurs élevées de $\hat{\theta}^C \approx 0.875$, indiquant que les véhicules apparaissent fréquemment dans presque tous les jeux de données.



(a) $C_3 = \{v, p, c\}$. (b) $C_0 = \{v\}$.

FIGURE 6.25 – Estimation inter-dataset avec une distribution uniforme appliquée au modèle de compression stf(mse)$_{\lambda_7}$ associé à YOLOv7, pour les configurations de classes C_3 et C_0.

Enfin, pour les ensembles C_1 et C_2 (cf. Figure 6.26), représentant respectivement les piétons et les cyclistes, les probabilités d'apparition $\hat{\theta}^C$ sont nettement plus faibles, atteignant 0.263 pour les piétons et 0.075 pour les cyclistes. Ces valeurs reflètent leur rareté relative dans les jeux de données, en comparaison avec les véhicules. L'écart observé entre $\hat{\theta}$ (la probabilité globale d'erreur) et $\hat{\theta}^{e,C}$ (la probabilité d'erreur conditionnelle sur les classes spécifiques) met en évidence que le système est moins performant pour détecter correctement les classes piétons et cyclistes, par rapport aux véhicules. Cependant, leur impact global sur la probabilité d'erreur reste limité, étant donné leur faible fréquence d'apparition dans les jeux de données. De plus, les intervalles de confiance associés à $\hat{\theta}^{e,C}$ sont significativement plus larges pour ces classes, ce qui est cohérent avec la rareté des échantillons disponibles. Cette incertitude accrue dans les estimations souligne la nécessité d'un échantillonnage plus représentatif pour ces classes afin d'améliorer la robustesse statistique des résultats. Ces résultats suggèrent deux axes d'amélioration essentiels. D'une part, il est crucial d'enrichir les jeux de données avec davantage d'images contenant des piétons et des cyclistes, ce qui permettrait de réduire les incertitudes dans les estimations et d'améliorer la représentativité des données pour ces classes. D'autre part, il est impératif d'augmenter la robustesse du détecteur YOLOv7 face aux effets de la compression, en mettant un accent particulier sur les classes de piétons et de cyclistes, qui semblent être plus sensibles aux dégradations liées à la compression. Ces actions combinées permettraient d'améliorer les performances globales du système tout en renforçant sa fiabilité pour ces classes spécifiques.



(a) $C_1 = \{p\}$. (b) $C_2 = \{c\}$.

FIGURE 6.26 – Estimation inter-dataset avec une distribution uniforme appliquée au modèle de compression stf(mse)$_{\lambda_7}$ associé à YOLOv7, pour les configurations de classes C_1 et C_2.

Une autre manière de visualiser ces résultats consiste à appliquer l'équation 6.13 pour déterminer la probabilité d'erreur pour chaque valeur de ϵ. Cela revient à examiner la distribution des erreurs en fonction du seuil ϵ, ce qui offre une perspective plus détaillée sur la performance du système pour différents niveaux de tolérance à l'erreur. La Figure 6.27 illustre l'estimation de la probabilité $P(e(x) = \epsilon)$ (qu'on notera $\hat{P}(e(x) = \epsilon)$)) pour le modèle de compression stf(mse)$_{\lambda_7}$ et YOLOv7 à un débit de 0.349 bpp. On peut observer que la majorité des erreurs se concentrent autour des faibles valeurs de ϵ. En effet, la probabilité que la compression n'engendre aucune sur une image est d'environ 83% avec un intervalle de confiance ± 0.058. Les probabilités décroissent rapidement avec l'augmentation de ϵ, atteignant 13.1% pour $\epsilon = 1$ et seulement 3.0% pour $\epsilon = 2$. Au-delà de $\epsilon = 3$, les probabilités d'erreur deviennent négligeables, ce qui montre que les erreurs graves (avec des écarts significatifs) sont rares, mais pas inexistantes. Ces résultats traduisent une certaine robustesse du système, bien que des erreurs mineures persistent dans une proportion notable des cas. En résumé, cette approche permet d'analyser plus finement la distribution des erreurs en fonction de leur gravité, offrant ainsi une vision complémentaire aux métriques globales comme $\hat{\theta}$.

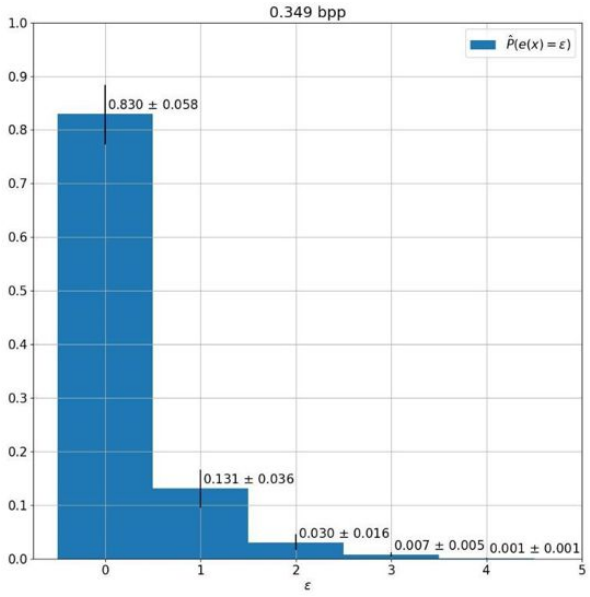


FIGURE 6.27 – Estimation des probabilité

En conclusion, la probabilité d'erreur globale de notre système sur l'ensemble des classes, selon une distribution uniforme des datasets, c'est-à-dire en considérant que chaque ensemble de données a le même poids ou la même probabilité, s'élève à 0.170 ± 0.057. Toutefois, cette estimation uniforme ne reflète pas forcément les spécificités et les priorités opérationnelles liées à une application donnée. Prenons par exemple le cas de l'application *Field Monitoring*.

Distribution pour le Field Monitoring

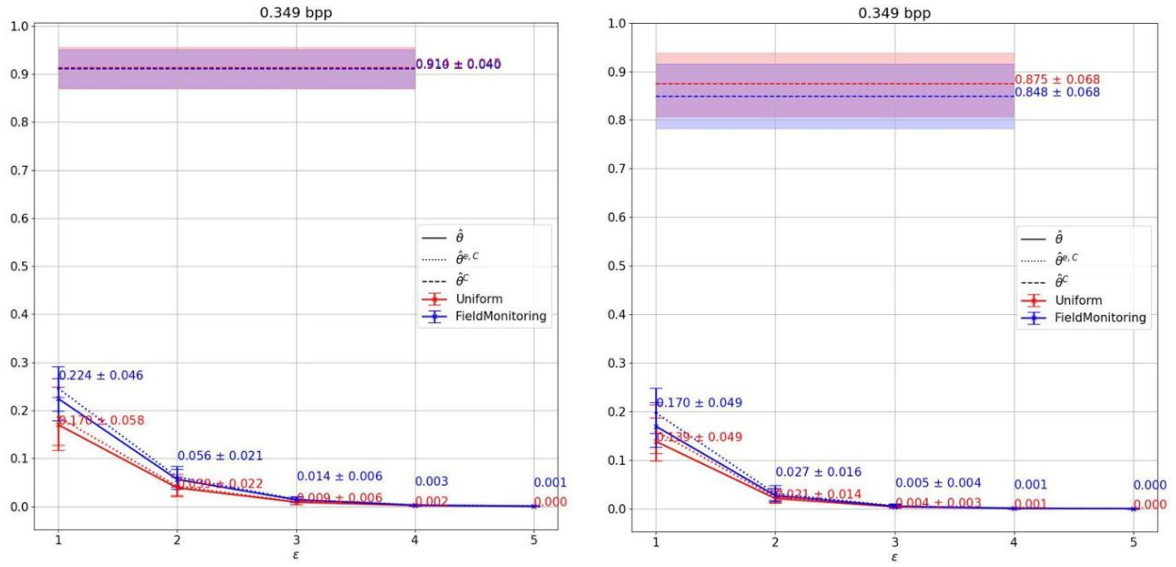
Dans le contexte de l'application *Field Monitoring*, l'enregistrement et la compression des images sont déclenchés par l'émission et la réception d'un *Trigger*. Ce *Trigger* se produit typiquement aux limites de capacité fonctionnelle du système de conduite autonome. En outre, l'objectif principal du *Field Monitoring* est d'acquérir des données dans des scénarios rares et critiques, notamment ceux qui pourraient révéler des failles potentielles ou des cas limites dans le système. En d'autres termes, il est plus probable que les *Triggers* correspondent à des scènes et des scénarios de type *Edge et Corner*. Ces types de scénarios incluent, par exemple, des accidents, des piétons en situation dangereuse (chute ou traversée inattendue), des véhicules en feu, des animaux ou obstacles sur la route, ainsi que des conditions météorologiques sévères comme de fortes pluies, la neige ou brouillard. Ces situations correspondent typiquement aux caractéristiques des jeux de données tels que Waterloo, Dashcam, Carcrash, PeSOTIF, Unity et E&C. Ces datasets capturent des scénarios critiques ou rares qui sont directement liés aux objectifs de l'application *Field Monitoring*. En tenant compte de cette spécificité, il est judicieux de pondérer davantage ces datasets dans notre estimation globale. Ainsi, nous posons $w_j = 100$ pour les datasets Waterloo, Dashcam, Carcrash, PeSOTIF, Unity et E&C, et $w_j = 1$ pour les autres datasets. Cela suppose que ces datasets critiques sont 100 fois plus probables que les autres (cf. Tableau 6.10). Une telle pondération reflète mieux la réalité opérationnelle du *Field Monitoring*, où les données issues de scénarios critiques sont beaucoup plus susceptibles d'être collectées et analysées. Cette adaptation des pondérations permet d'obtenir une estimation des performances du système plus représentative des conditions spécifiques à cette application.

TABLE 6.10 – Tableau des poids w_j et des hyperparamètres p_j selon une distribution pour le *Field Monitoring*.

Datasets	w_j	p_j
waymo	1	0.0017
bdd100k	1	0.0017
stellantis	1	0.0017
boxy	1	0.0017
audi	1	0.0017
waterloo	100	0.165
nuimages	1	0.0017
dashcam	100	0.165
carcrash	100	0.165
pesotif	100	0.165
unity	100	0.165
E&C	100	0.165

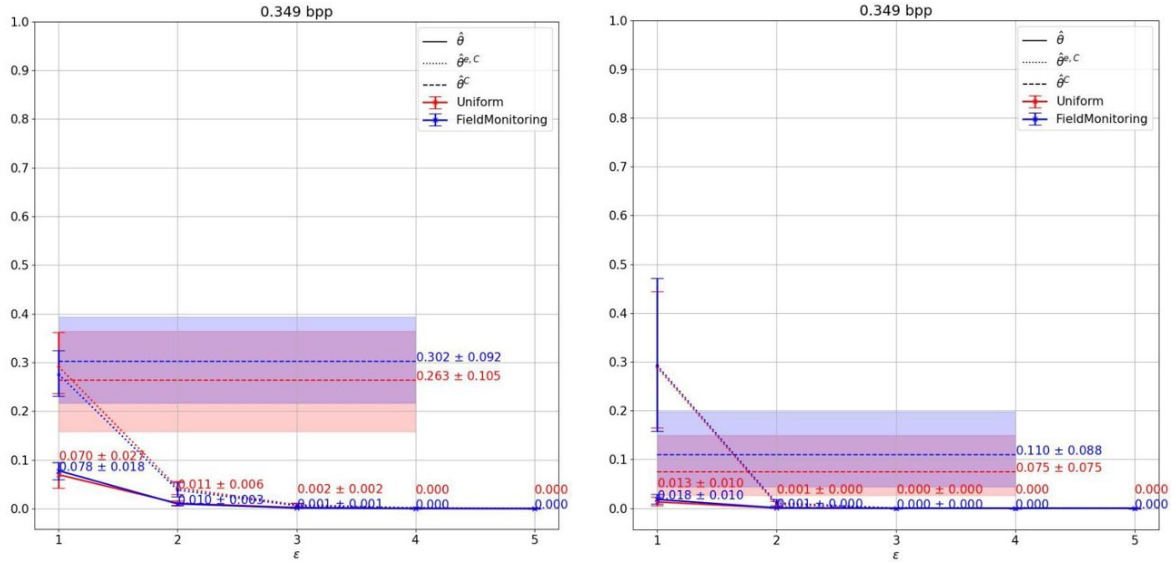
Nous appliquons à nouveau la deuxième étape d'estimation des grandeurs $\hat{\theta}$, $\hat{\theta}^{e,C}$ et $\hat{\theta}^{C}$, cette fois en utilisant les pondérations définies dans le tableau 6.10, qui reflètent la priorisation des datasets critiques dans le contexte de l'application *Field Monitoring*.

Pour l'ensemble $C_3 = \{v, p, c\}$ (cf. Figure 6.28a), on observe que, comparé à la distribution uniforme, le risque d'erreur causé par la compression augmente sensiblement. En effet, la probabilité d'erreur globale $\hat{\theta}$ passe de 0.170 ± 0.057 à 0.224 ± 0.046. Cette augmentation indique que les jeux de données de type *Edge and Corner*, tels que Dashcam, Waterloo, Unity, ou E&C, sont plus sensibles à la compression, ce qui est cohérent avec leur contenu représentant des scènes rares, complexes, ou critiques. Ces données, par nature, mettent davantage en difficulté les détecteurs d'objets après compression. De même, pour les classes véhicules (C_0), piétons (C_1) et cyclistes (C_3), on constate également une augmentation notable des estimations $\hat{\theta}$ et de $\hat{\theta}^{e,C}$ (cf. Figures 6.28b, 6.29a et 6.29b). Contrairement aux véhicules pour lesquels la probabilité d'apparition a diminuée (0.848 par rapport à 0.875), pour les piétons (C_1) et les cyclistes (C_2), leur probabilité d'apparition $\hat{\theta}^{C}$ est respectivement de 0.302 et 0.110, ce qui est légèrement supérieur dans cette configuration par rapport à la distribution uniforme. En résumé, cette estimation pondérée met en évidence que, dans un contexte applicatif spécifique comme le *Field Monitoring*, où les scénarios critiques sont prédominants, la probabilité d'erreur causée par la compression est amplifiée, en particulier pour les piétons et les cyclistes. Ces résultats soulignent l'importance d'adapter les pondérations des datasets en fonction de leur représentativité réelle pour obtenir des estimations fiables et pertinentes. Cela renforce également la nécessité d'améliorer la robustesse des algorithmes de compression et de détection pour ces types de scénarios, notamment en intégrant des solutions mieux adaptées aux défis posés par les données rares et critiques.



(a) $C_3 = \{v, p, c\}$. (b) $C_0 = \{v\}$.

FIGURE 6.28 – Estimation inter-dataset avec une distribution pour le Field Monitoring appliquée au modèle de compression stf(mse)$_{\lambda_7}$ associé à YOLOv7, pour les configurations de classes C_3 et C_0.



(a) $C_1 = \{p\}$. (b) $C_2 = \{c\}$.

FIGURE 6.29 – Estimation inter-dataset avec une distribution pour le Field Monitoring appliquée au modèle de compression stf(mse)$_{\lambda_7}$ associé à YOLOv7, pour les configurations de classes C_1 et C_2.

De manière similaire à l'analyse effectuée avec la distribution uniforme, nous pouvons visualiser les résultats obtenus en appliquant l'équation 6.13 pour déterminer la probabilité d'erreur en fonction de différentes valeurs de ϵ. Les probabilités estimées, présentées dans la Figure 6.30, montrent qu'en moyenne, 77% des images d'un ensemble de données ne contiendront aucune erreur, tandis que 95% des images contiendront au plus une seule erreur.

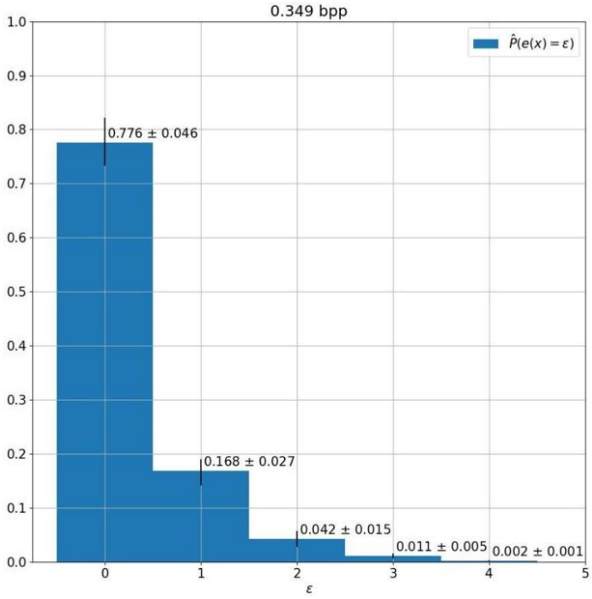


FIGURE 6.30 – Estimation des probabilité pour le *Field Monitoring* avec YOLOv7 et stf(mse)$_{\lambda_7}$.

Pour justifier la méthodologie proposée dans la section 4.3.2, rappelons qu'une approche simplifiée consisterait à estimer directement l'espérance $\mathbb{E}[Z(X)]$ via une moyenne empirique sur l'ensemble des datasets, comme vu dans la section 4.2.2. Cette méthode peut s'exprimer par :

$$\mathbb{E}[Z(X)] \approx \frac{1}{\sum_{i=1}^{d} |D_i|} \sum_{i=1}^{d} \sum_{x \in D_i} \mathbb{1}_{\{e(x) \geq \epsilon\}} \tag{6.14}$$

où $|D_i|$ représente le cardinal de D_i, c'est-à-dire le nombre d'échantillons dans le dataset D_i. Cependant, cette estimation suppose implicitement que chaque donnée, quel que soit le dataset auquel elle appartient, contribue de manière équitable à l'espérance globale. Cela revient à attribuer des poids proportionnels à la taille des datasets, sans tenir compte des disparités dans les conditions de collecte, la représentativité des scénarios ou la fréquence relative des événements dans le monde réel. Une telle approche peut donc être fortement biaisée si ces disparités ne sont pas prises en compte. Pour vérifier cet effet, nous avons comparé les estimations obtenues en utilisant une moyenne empirique non pondérée (méthode simplifiée) avec celles obtenues via la méthodologie proposée, qui intègre des pondérations adaptées aux caractéristiques et à la représentativité des datasets (cf. Figure 6.31). Les résultats mettent en évidence que la méthode simplifiée sous-estime significativement les risques pour les scénarios critiques lorsqu'elle ne tient pas compte des pondérations adaptées. En effet, les probabilités d'erreur causées par la compression sont estimées à seulement 0.027, ce qui constitue une sous-évaluation majeure. Ces résultats soulignent l'importance cruciale d'intégrer les disparités entre les datasets dans le processus d'estimation pour garantir des évaluations fiables et représentatives de la réalité opérationnelle du système étudié.

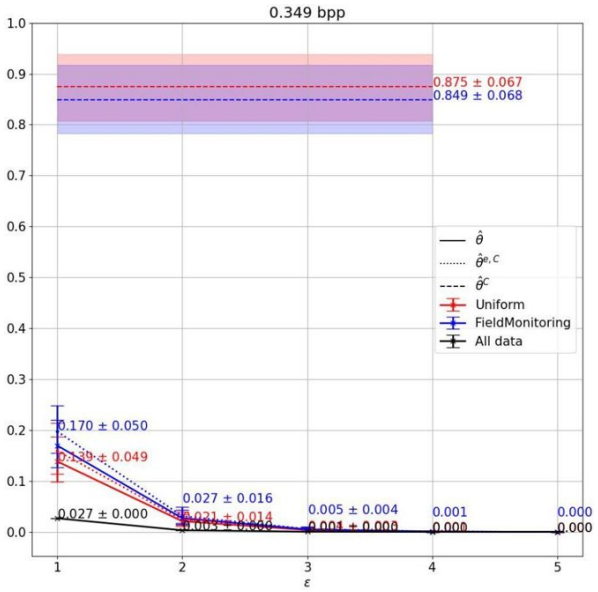


FIGURE 6.31 – Comparaison des estimations entre la méthodologie proposée (Uniform et Field Monitoring) et une estimation simplifiée basée sur l'ensemble des données (All Data).

Comparons les garanties obtenues pour chaque compresseur pour $\epsilon = 1$ et la configuration $C_3 = \{v, p, c\}$, en utilisant la distribution spécifique au *Field Monitoring* (cf. Tableau 6.10).

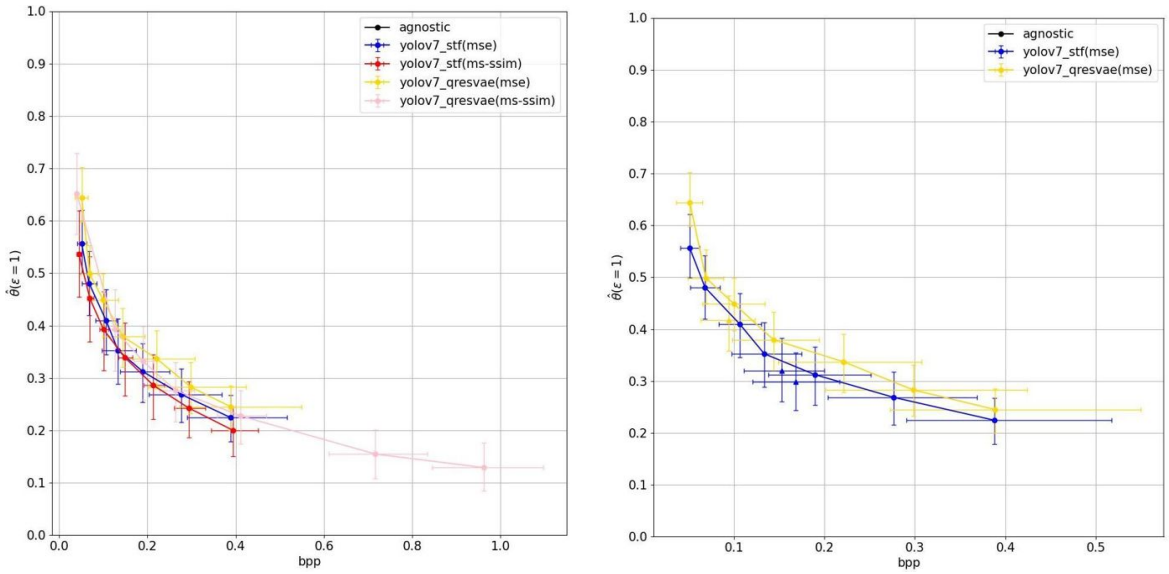
La Figure 6.32 présente les estimations de $\hat{\theta}$ pour chacun des compresseurs entraînés (cf. Tables 6.3 et 6.4), en fonction des différents hyperparamètres λ. Ces estimations permettent de mettre en évidence l'impact de la compression sur les probabilités d'erreur, tout en comparant les performances relatives des différents modèles. La sélection du modèle de compression le plus adapté repose sur un compromis subtil entre la probabilité d'erreur $\hat{\theta}$ causée par la compression et le taux de compression (bpp) requis par l'application visée. Par exemple, un modèle avec un faible taux de compression (bpp élevé) réduit considérablement les erreurs de détection, mais peut être inadapté dans des contextes où l'efficacité de stockage ou de transmission est cruciale. À l'inverse, un modèle offrant une compression plus agressive (bpp faible) permet un gain important en espace mémoire ou en bande passante, mais cela au prix d'une augmentation significative des erreurs.

D'après la Figure 6.32a, le modèle de compression le plus prometteur pour notre application semble être le modèle stf(ms-ssim) (indiqué en rouge). Ce dernier minimise efficacement la probabilité d'erreur $\hat{\theta}$ tout en maintenant un bpp compétitif. Si l'on devait sélectionner une configuration optimale, on privilégierait le modèle stf(ms-ssim) entraîné avec l'hyperparamètre λ_7 (représenté par le point rouge le plus à droite dans la Figure 6.32a). En effet, pour un taux de compression de $bpp \approx 0.39$, la probabilité d'erreur est estimée à $\hat{\theta} \approx 0.2$. Il est également à noter qu'au-delà de $bpp \approx 0.4$, les avantages des modèles de compression basés sur le deep learning par rapport aux méthodes standards, telles que JPEG ou WebP, deviennent marginaux.

Cette dernière analyse entre en contradiction avec les conclusions établies dans la section 6.3. À partir de la Figure 6.12, nous avions conclu que l'entraînement des compresseurs, notamment STF, avec la métrique MSE réduisait l'impact de la compression sur la détection. Cependant, cette conclusion était basée uniquement sur le dataset D_{test}, composé des jeux de données Waymo et BDD100K, qui ne reflètent qu'une partie des scénarios possibles. Désormais, grâce à l'estimation des probabilités d'erreur $\hat{\theta}$ sur l'ensemble de nos jeux de données, en prenant en compte la pondération spécifique au *Field Monitoring*, nous arrivons à une conclusion différente. Cette nouvelle analyse révèle que la métrique MS-SSIM semble mieux adaptée pour préserver la robustesse du système dans des contextes critiques. En effet, en intégrant des scénarios variés et des poids plus représentatifs des cas réels, les performances des modèles entraînés avec MS-SSIM surpassent celles obtenues avec MSE, particulièrement pour les situations de type *Edge and Corner*. Cette divergence souligne l'importance de considérer une méthodologie globale et pondérée pour évaluer les performances, car une analyse restreinte à un seul dataset peut mener à des conclusions partielles, voire biaisées, qui ne reflètent pas la réalité opérationnelle du système étudié.

En examinant les compresseurs agnostiques et ROI dans la Figure 6.32b, il est clair que les modèles ROI présentent des probabilités d'erreur légèrement inférieures à celles des modèles agnostiques. Cette observation suggère que les modèles ROI préservent mieux les détails critiques, notamment pour les objets. Une optimisation

supplémentaire, par réentraînement spécifique des modèles ROI, pourrait réduire encore davantage l'impact de la compression sur les performances de détection.



(a) Compression agnostique.

(b) Comparaison entre la compression agnostique et ROI.

FIGURE 6.32 – Estimation de la probabilité d'erreur $\hat{\theta}$ pour chaque compresseur et pour chaque niveau de compression.

En conclusion, le choix du compresseur le plus adapté dépendra du contexte applicatif et des exigences spécifiques, notamment la tolérance à l'erreur acceptable (ϵ), le taux de compression souhaité, et la nécessité de préserver des détails critiques dans certaines zones de l'image. Ces éléments devront être soigneusement pris en compte pour garantir un équilibre optimal entre efficacité et robustesse dans le cadre du système étudié. Cependant, les niveaux d'erreur estimés restent actuellement trop élevés pour répondre aux exigences de l'application *Field Monitoring*. Cette application, dédiée à la sécurité des véhicules autonomes, requiert des probabilités d'erreur significativement plus faibles, idéalement de l'ordre de 1% voire 1 ‰. De tels seuils sont cruciaux pour garantir que les images compressées conservent une qualité suffisante pour une analyse précise des scénarios critiques. Dans les sections suivantes, nous explorerons des stratégies visant à réduire davantage les probabilités d'erreur estimées. Ces stratégies incluront notamment le réentraînement des modèles de compression avec des objectifs spécifiques, une optimisation des détecteurs pour mieux gérer les distorsions induites par la compression, ainsi que des ajustements des hyperparamètres pour adapter les modèles à des exigences plus strictes.

6.4.3 Conclusion

Dans cette section, nous avons appliqué la méthodologie d'estimation des garanties de performances à notre système S, composé d'un module de compression d'image et d'un module de détection d'objets. Tout d'abord, l'estimation intra-dataset a révélé une variabilité importante de la probabilité d'erreur d'un dataset à l'autre. Certains jeux de données, notamment ceux contenant des scènes critiques ou des conditions particulières, comme les datasets de type *Edge and Corner*, se sont avérés beaucoup plus sensibles à la compression. Ces résultats soulignent l'importance de la spécificité des données dans l'évaluation des performances du système. Nous avons ensuite analysé l'impact de la distribution ou pondération p_j définie sur les datasets dans le cadre de l'estimation inter-dataset. Il est apparu que ces pondérations influencent fortement les probabilités d'erreur globales estimées. Nous avons illustré cela à l'aide de deux exemples : une distribution uniforme, où tous les datasets sont considérés équiprobables, et une distribution spécifique à l'application *Field Monitoring*, où les datasets *Edge and Corner* (scènes rares et critiques) sont considérés comme beaucoup plus probables. Cette dernière configuration, qui reflète mieux les conditions réelles du système, a mis en évidence des risques accrus dans les scénarios critiques. En séparant les analyses des probabilités par classes d'objets, nous avons constaté que, par rapport à la classe véhicule, le système était moins performant sur les classes piétons et cyclistes. Cependant, ces classes, étant moins fréquentes dans l'ensemble des datasets, ont un impact relativement limité sur la probabilité

globale d'erreur. De plus, à travers les différents compresseurs testés, nous avons estimé qu'avec des taux de compression inférieurs à 0.4 bpp, le taux d'erreur global atteint un minimum d'environ 20%. Cela signifie que, sur un nouveau jeu de données, on peut s'attendre à ce qu'environ 20% des images contiennent au moins une erreur dans les prédictions du système S. Cependant, il a également été observé que, parmi ces 20% d'images, la grande majorité ne contient qu'une seule erreur.

Il est également important de noter une divergence significative entre les probabilités d'erreur estimées dans cette section et les analyses de l'effet de la compression réalisées dans la section 6.3, où des valeurs de mAP atteignant 95 à 99% avaient été observées (cf. Figure 6.12). Cette différence provient des divergences méthodologiques entre les deux approches. En effet, les évaluations conventionnelles des détecteurs d'objets utilisent généralement des métriques comme le mAP. Le mAP compare une liste de boîtes prédites par le modèle à une liste de boîtes de référence et mesure la proportion de boîtes correctement prédites par rapport au nombre total de boîtes. Cependant, le mAP ne tient pas compte de la répartition des erreurs parmi les images. En d'autres termes, cette métrique informe sur la proportion d'erreurs globales, mais ne permet pas d'évaluer leur impact à l'échelle des images. En revanche, la méthodologie d'estimation des garanties de performances utilisée ici se concentre sur la probabilité d'erreur par image. Cette approche est particulièrement pertinente pour des applications critiques, où la distribution des erreurs sur les images est aussi importante que leur fréquence globale.

Ainsi, cette méthodologie offre une évaluation complémentaire, mais plus stricte et opérationnelle, des performances du système, en prenant en compte non seulement la qualité globale des prédictions, mais aussi leur impact potentiel dans un contexte applicatif donné. Elle met en lumière les limites des analyses basées uniquement sur des métriques comme le mAP et souligne la nécessité d'une évaluation plus fine des erreurs dans des environnements critiques.

6.5 Optimisation des performances

Dans la section 6.4.2, nous avons mis en évidence les effets négatifs de la compression sur la détection et l'estimation des garanties de performance du système S. Cependant, les niveaux d'erreur actuellement estimés restent trop élevés pour répondre aux exigences strictes de l'application *Field Monitoring*. Cette application, dédiée à la sécurité des véhicules autonomes, impose des probabilités d'erreur nettement plus faibles, idéalement de l'ordre de 1% voire 1 ‰. Pour atteindre cet objectif, nous proposons d'explorer plusieurs pistes visant à réduire ces probabilités d'erreur tout en maintenant l'efficacité globale du système.

L'une des premières approches consiste à réentraîner le détecteur avec davantage de données naturelles, synthétiques, ou en encore des images présentant des distorsions similaires à celles introduites par la compression. Cependant, cette stratégie n'est pas sans risques : il est crucial de veiller à ce que l'amélioration de la robustesse vis-à-vis de la compression ne dégrade pas les performances intrinsèques du détecteur dans des conditions normales. Un équilibre entre robustesse et précision doit être maintenu pour garantir que les performances globales du système ne soient pas compromises.

En ce qui concerne les compresseurs entraînés de manière agnostique (cf. Figure 6.4), il a été observé qu'ils engendrent généralement des distorsions visuelles globales, souvent sous la forme de flous perceptibles. Ces modèles, lorsqu'ils sont optimisés avec la métrique MSE, allouent davantage de bits aux régions contenant des contours prononcés ou des transitions marquées, ce qui peut favoriser une certaine structure géométrique mais au détriment d'autres parties de l'image. Les modèles optimisés avec la métrique MS-SSIM, en revanche, adoptent une répartition des bits plus homogène à travers l'image, ce qui favorise la préservation de la structure globale et des détails visuels. Ces observations montrent que la compression agnostique se base principalement sur une analyse géométrique pour décider des zones à privilégier lors de la compression. Cela étant, réentraîner ces compresseurs agnostiques avec davantage de données ne serait probablement pas très utile, car leur objectif n'est pas directement aligné avec la détection d'objets. Une autre possibilité pourrait être de réentraîner ces compresseurs avec des hyperparamètres λ plus élevés pour réduire les distorsions dans les images. Toutefois, cela augmenterait significativement le débit (bpp), ce qui va à l'encontre des contraintes de compression strictes imposées par l'application *Field Monitoring*.

En revanche, les compresseurs basés sur les régions d'intérêt (ROI) offrent une opportunité intéressante. Contrairement aux compresseurs agnostiques, ces modèles apprennent à détecter des objets spécifiques et à allouer davantage de ressources de compression aux zones d'intérêt, comme les objets critiques dans une scène. Réentraîner ces compresseurs ROI avec davantage de données pertinentes pourrait améliorer leur capacité à détecter et à préserver les objets importants dans les images compressées. En conséquence, les objets resteraient plus nets après la compression, ce qui réduirait les probabilités d'erreur associées à leur détection par le système.

Ainsi, dans la suite de cette section, nous explorerons ces différentes stratégies, en mettant un accent particulier sur le réentraînement des détecteurs et l'optimisation des compresseurs ROI pour mieux répondre aux exigences de l'application *Field Monitoring*. Nous analyserons également l'impact de ces approches sur la probabilité d'erreur globale et le compromis entre compression et précision.

6.5.1 Réentrainer la détection

L'objectif principal du réentraînement des modèles de détection est d'améliorer leurs performances tout en augmentant leur robustesse vis-à-vis des variations de données et des distorsions causées par la compression. Pour ce faire, nous proposons de réentraîner YOLOv7 selon trois approches distinctes, en diversifiant et en enrichissant les données utilisées lors de la phase d'apprentissage.

Dans un premier temps, nous intégrerons dans l'ensemble des données d'apprentissage des données naturelles supplémentaires provenant de la base NuImagesLabels. Ces données incluent non seulement des images issues des caméras frontales, mais également des caméras latérales avant, latérales arrière et arrière. Cette diversification permettra de couvrir un éventail plus large de perspectives, enrichissant ainsi le détecteur en termes de variations d'angles de vue, de configurations de scène et de conditions environnementales. Ensuite, dans un deuxième temps, nous ajouterons des données synthétiques générées à l'aide de notre simulateur décrit dans la section 5.3.2. Ces données simulées permettront d'introduire des scénarios critiques ou rares, tels que des conditions météorologiques extrêmes (brouillard, pluie intense, neige) ou des événements inhabituels (obstacles inattendus, accidents, animaux sur la route). Ces situations, bien que rares, sont essentielles pour tester la capacité du modèle à gérer des scénarios hors distribution. Enfin, pour évaluer et améliorer la robustesse du détecteur face à la compression, nous intégrerons dans l'apprentissage des données ayant été compressées par notre modèle de compression stf(mse), entraîné avec deux configurations d'hyperparamètres : λ_7 (faible compression, haute qualité) et λ_3 (forte compression, faible qualité). L'objectif de cette étape est de familiariser le détecteur avec les artefacts induits par la compression, afin qu'il puisse mieux s'adapter aux distorsions visuelles dans des contextes réels, tels que ceux rencontrés dans l'application *Field Monitoring*.

Pour chaque approche, nous analyserons non seulement l'effet de l'augmentation des données sur les performances globales du détecteur (mAP), mais également son impact sur la robustesse vis-à-vis des scénarios spécifiques et des distorsions liées à la compression. Ces analyses permettront de déterminer dans quelle mesure ces stratégies de réentraînement contribuent à réduire les probabilités d'erreur estimées et à répondre aux exigences de l'application en termes de fiabilité et de performance.

Ajout d'images naturelles

Dans un premier temps, nous réentraînons YOLOv7 en enrichissant le dataset d'entraînement D_{train} avec des données supplémentaires issues de la base NuImagesLabels, afin d'augmenter la diversité des scénarios et des perspectives dans les données d'apprentissage. Pour ce faire, nous construisons deux nouvelles partitions du dataset D. La première, nommée $D_{\text{train_aug_front}}$, est constituée en augmentant la partition d'entraînement originale D_{train} avec des images issues des caméras frontales de la base NuImagesLabels. La seconde partition, nommée $D_{\text{train_aug}}$, est construite en enrichissant D_{train} avec des images issues de toutes les caméras de la base NuImagesLabels, y compris les caméras latérales avant, latérales arrière, et arrière (cf. Table 6.11).

TABLE 6.11 – Partitions augmentées du dataset D.

Nom	Partition	Bases de données	Volumétrie (#images)	
			Par base	Total
D	$train_aug_front$	Waymo	124,668	198,333
		BDD100K	55,890	
		NuImagesLabels (Front)	17,774	
	$train_aug$	Waymo	124,668	271,797
		BDD100K	55,890	
		NuImagesLabels (Toutes)	91,238	

Ensuite, nous entraînons YOLOv7 sur les deux partitions augmentées, $D_{train_aug_front}$ et D_{train_aug}, en utilisant D_{valid} comme ensemble de validation. Comme précédemment, nous testons ces modèles sur D_{test}, et leurs performances sont répertoriées dans le Tableau 6.12. Les résultats montrent que l'ajout de données a significativement amélioré les performances de détection pour chacune des classes : véhicule, piéton et cycliste.En

particulier, l'ajout d'images provenant de toutes les caméras dans D_{train_aug} s'est révélé être la configuration la plus performante. Cela s'explique par la diversité accrue des perspectives et des situations, ce qui enrichit l'apprentissage et améliore la généralisation du modèle. À l'inverse, bien que l'ajout des seules images frontales ($D_{train_aug_front}$) ait également conduit à une amélioration, les gains sont moins marqués, reflétant une couverture plus limitée des cas de figure.

TABLE 6.12 – Performances du detecteurs YOLOv7 entrainé en augmentant la base d'apprentissage avec les données NuImagesLabels.

Datasets d'entrainement	Nom	Évaluation sur D_{test}			
		APv@.5	APp@.5	APc@.5	mAP@.5
D_{train} D_{valid}	yolov7	0.9298 ± 0.0001	0.834 ± 0.007	0.60 ± 0.01	0.788 ± 0.004
$D_{train_aug_front}$ D_{valid}	yolo_aug_front	0.9302 ± 0.0001	0.846 ± 0.001	0.62 ± 0.01	0.7997 ± 0.0035
D_{train_aug} D_{valid}	yolo_aug	0.936 ± 0.006	0.855 ± 0.008	0.63 ± 0.01	0.806 ± 0.005

Enfin, nous appliquons les deux étapes d'estimation, intra-dataset et inter-dataset, aux nouveaux détecteurs *yolo_aug_front* et *yolo_aug*, combinés au modèle de compression stf(mse) agnostique. Comparé aux probabilités obtenues avec le modèle de détection YOLOv7 de référence, les résultats, présentés dans la Figure 6.33, montrent que les probabilités d'erreur n'ont significativement diminué que pour le modèle *yolo_aug* (bleu clair). Cette réduction reflète l'impact positif de l'ajout d'images issues de multiples caméras, offrant une variété accrue de points de vue et de scénarios. À l'inverse, le modèle *yolo_aug_front*, bien qu'il bénéficie de l'ajout d'images frontales, ne présente qu'une amélioration marginale, soulignant les limites d'une diversité réduite des données. Ces résultats indiquent que l'intégration de données provenant de perspectives variées joue un rôle crucial dans l'amélioration des performances globales du système.

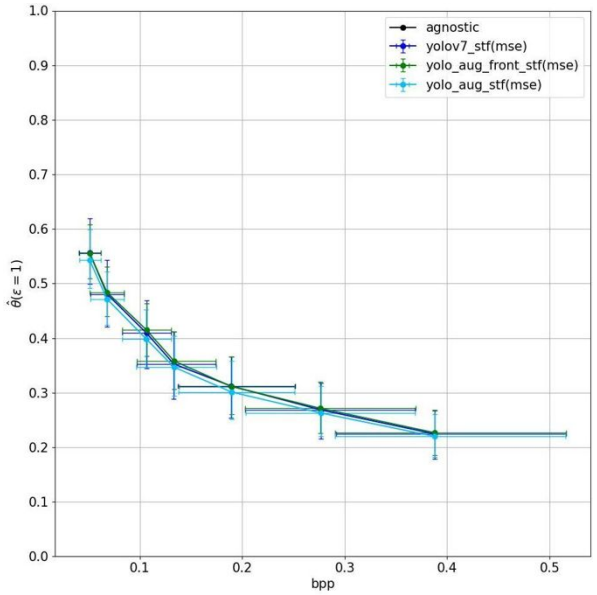


FIGURE 6.33 – Estimation de la probabilité d'erreur $\hat{\theta}$ pour le modèle de compression stf(mse) agnostique avec les modèles de détection YOLOv7, *yolo_aug_front* et *yolo_aug*.

En conclusion, l'acquisition de données issues des autres caméras pourrait constituer une solution efficace pour enrichir les jeux de données avec davantage de situations de type *Edge and Corner*. En effet, ces images offrent des perspectives complémentaires qui peuvent être interprétées comme des extensions des situations critiques rencontrées par la caméra frontale. Cela permet de mieux capturer des scénarios complexes ou rares, tout en augmentant la diversité des données. Par exemple, les images issues de la caméra arrière peuvent être assimilées à des données provenant de la caméra frontale, mais dans des contextes particuliers, tels que des situations où le véhicule autonome croise d'autres véhicules roulant à contre-sens. De même, les caméras latérales peuvent refléter des cas où des obstacles surgissent sur les côtés, tels qu'un piéton débouchant brusquement entre deux véhicules stationnés ou un cycliste effectuant un dépassement inattendu. Cette approche permet non seulement

d'accroître la couverture des scénarios critiques dans les datasets, mais également d'augmenter la robustesse du système en l'exposant à une variété plus large de points de vue et de conditions environnementales. En intégrant ces données dans le cadre d'un entraînement optimisé, le système peut mieux appréhender les variations spatiales et comportementales, renforçant ainsi sa capacité à réagir correctement dans des situations inattendues.

Ajout d'images synthétiques

Dans la section 5.3.3, nous avions déjà entraîné des modèles YOLOv7 sur des jeux de données augmentés avec des données synthétiques issues de notre simulateur. Pour ce faire, nous avions construit trois nouvelles partitions du dataset D à partir de la partition D_{train}, chacune enrichie avec respectivement 25%, 33%, et 50% de données synthétiques (cf. Table 5.7). Ces données supplémentaires provenaient de scénarios simulés conçus pour refléter des conditions variées et complexes, incluant des situations critiques. Après entraînement, nous avions observé que les trois nouveaux modèles de détection, $yolo_025unity$, $yolo_033unity$, et $yolo_055unity$, surpassaient le modèle de référence YOLOv7 sur les trois classes principales : véhicules, piétons et cyclistes (cf. Table 5.8). Les augmentations de données synthétiques semblaient donc contribuer efficacement à améliorer les performances globales du détecteur, notamment sur les classes moins fréquentes dans les données réelles. En appliquant les deux étapes d'estimation, intra-dataset et inter-dataset, à ces nouveaux modèles de détection augmentés, nous constatons une réduction significative des probabilités d'erreur estimées $\hat{\theta}$ par rapport au modèle de référence YOLOv7. Ces résultats confirment que l'ajout de données synthétiques renforce non seulement les performances du détecteur sur les métriques standards comme le mAP, mais réduit également la probabilité d'erreur par image, un critère essentiel dans des applications critiques comme le *Field Monitoring*. Parmi les trois modèles, les détecteurs $yolo_025unity$ et $yolo_033unity$ affichent des résultats particulièrement prometteurs, avec des courbes de probabilités d'erreur $\hat{\theta}$ quasiment superposées, suggérant une efficacité de l'augmentation de données à ce niveau.

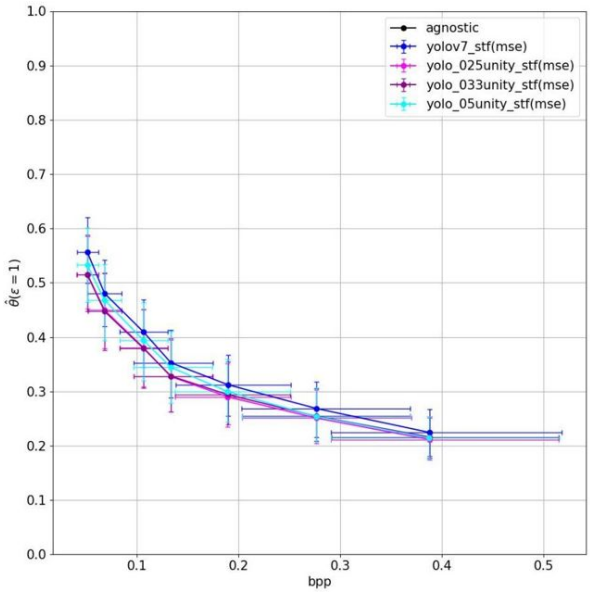


FIGURE 6.34 – Estimation de la probabilité d'erreur $\hat{\theta}$ pour le modèle de compression stf(mse) agnostique avec les modèles de détection YOLOv7, $yolo_025unity$, $yolo_033unity$ et $yolo_055unity$.

En conclusion, à l'instar de l'ajout de données naturelles, l'intégration de données synthétiques dans le processus d'entraînement contribue également à une réduction significative de la probabilité d'erreur $\hat{\theta}$. Cette approche s'avère particulièrement efficace pour renforcer les performances globales du système, en améliorant sa robustesse face aux scénarios critiques et en enrichissant la diversité des situations rencontrées pendant l'entraînement.

Ajout d'images comprimées

Pour améliorer la robustesse du modèle de détection face à la compression, une dernière option consiste à entraîner de nouveaux détecteurs sur un mélange de données non compressées et compressées. Cette approche vise

à rendre le modèle plus résilient aux distorsions causées par la compression, en l'exposant ce distorsions pendant l'entraînement. Pour vérifier cette hypothèse, nous avons entraîné deux modèles YOLOv7. Le premier modèle a été entrainé sur la partition D_{train} augmentée avec les mêmes images compressées par le modèle stf(mse)$_{\lambda_7}$, que nous notons $STF^{\lambda_7}_{\text{mse}}(D_{train})$. Le second modèle a été formé sur D_{train} augmenté avec les mêmes images compressées par le modèle stf(mse)$_{\lambda_3}$, noté $STF^{\lambda_3}_{\text{mse}}(D_{train})$. De manière similaire, les partitions D_{valid} ont été enrichies avec les données compressées correspondantes pour valider les modèles. Nous nommons ces deux nouveaux détecteurs yolostfmseq7 et yolostfmseq3. Après entraînement, nous avons évalué ces modèles sur la partition D_{test}. Les résultats, présentés dans le Tableau 6.13, montrent que les performances des nouveaux modèles sont presque identiques à celles du modèle de référence YOLOv7. Cette similitude suggère que l'exposition à des données compressées n'a pas entraîné de dégradation notable des performances de détection.

TABLE 6.13 – Performances du detecteurs YOLOv7 entrainé en augmentant la base d'apprentissage avec des données comprimées.

Datasets d'entrainement	Nom	Évaluation sur D_{test}			
		APv@.5	APp@.5	APc@.5	mAP@.5
D_{train} D_{valid}	yolov7	0.9298 ± 0.0001	0.834 ± 0.007	0.60 ± 0.01	0.788 ± 0.004
$D_{train} + \text{STF}^{\lambda_7}_{mse}(D_{train})$ $D_{valid} + \text{STF}^{\lambda_7}_{mse}(D_{valid})$	yolostfmseq7	0.9299 ± 0.0001	0.830 ± 0.006	0.61 ± 0.01	0.792 ± 0.004
$D_{train} + \text{STF}^{\lambda_3}_{mse}(D_{train})$ $D_{valid} + \text{STF}^{\lambda_3}_{mse}(D_{valid})$	yolostfmseq3	0.9299 ± 0.0001	0.830 ± 0.006	0.61 ± 0.01	0.792 ± 0.004

Ensuite, comme précédemment, nous avons appliqué les étapes d'estimation intra-dataset et inter-dataset aux nouveaux détecteurs yolostfmseq7 et yolostfmseq3, combinés au modèle de compression stf(mse) agnostique. Les résultats, présentés dans la Figure 6.35, montrent une diminution significative des probabilités d'erreur par rapport au modèle de détection YOLOv7 de référence. En particulier, les probabilités d'erreur $\hat{\theta}$ sont nettement plus faibles pour le modèle yolostfmseq3 que pour le modèle yolostfmseq7, indiquant que l'entraînement sur des images fortement compressées permet de renforcer considérablement la robustesse du détecteur à tous les niveaux de compression. Cette amélioration est particulièrement marquée pour des taux de compression élevés, où les modèles précédents montraient une sensibilité accrue aux distorsions. Cette réduction des probabilités d'erreur reflète l'impact positif de l'ajout d'images compressées pendant l'entraînement, en exposant le modèle à des scénarios variés de dégradations visuelles. Ce processus semble jouer un rôle crucial pour augmenter la résilience du détecteur. Notamment, la diminution des probabilités d'erreur pour le modèle yolostfmseq3 dépasse même celle obtenue avec les modèles augmentés avec des données naturelles ($yolo_aug$) ou synthétiques ($yolo_025unity$ et $yolo_033unity$). Cela démontre que l'utilisation de données directement affectées par les distorsions caractéristiques de la compression est une approche particulièrement efficace pour optimiser la robustesse du détecteur.

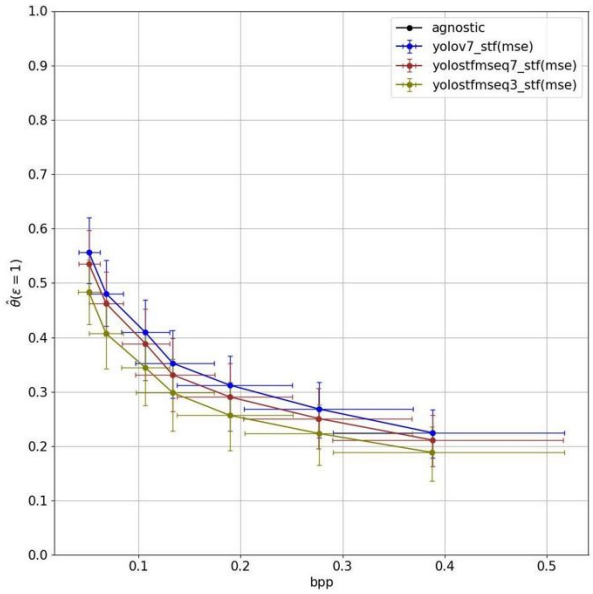


FIGURE 6.35 – Estimation de la probabilité d'erreur $\hat{\theta}$ pour le modèle de compression stf(mse) agnostique avec les modèles de détection YOLOv7, yolostfmseq7 et yolostfmseq3.

En conclusion, l'entraînement de YOLOv7 sur des images compressées, en particulier avec des taux de compression élevés, constitue une solution évidente et performante pour renforcer la robustesse à la compression. Parmi les différentes méthodes explorées pour optimiser le détecteur, cette stratégie s'est révélée être la plus prometteuse et efficace, offrant une amélioration notable des garanties de performance du système.

6.5.2 Ré-entrainer la compression

Nous avons observé que les modèles de compression appris de manière agnostique allouent un budget de bits limité à certaines zones de l'image, en privilégiant principalement les contours et les transitions marquées. Cette approche repose exclusivement sur des critères géométriques et ne tient aucunement compte du contenu sémantique de l'image. Par conséquent, ces modèles ne font aucune distinction basée sur la signification ou la pertinence des éléments présents dans l'image. De plus, lorsqu'ils sont testés sur des images dites "hors domaine", c'est-à-dire des images totalement différentes de celles utilisées lors de leur entraînement, ces modèles de compression montrent une efficacité remarquable. Malgré l'absence de connaissance préalable du domaine, la compression s'effectue de manière fluide et conserve des performances globales satisfaisantes, comme le montre la Figure 6.36. Ce résultat illustre la robustesse de ces compresseurs dans des contextes variés, bien que leur fonctionnement reste aveugle aux informations contextuelles ou sémantiques des images qu'ils compressent.



FIGURE 6.36 – Compression et allocation de bits pour une image "hors domaine" avec le compresseur stf(mse)$_{\lambda_7}$.

Il est donc peu probable que réentraîner les modèles de compression agnostique sur un volume accru d'images soit réellement bénéfique pour notre système de compression-détection. En effet, leur approche basée uniquement

sur des caractéristiques géométriques (telles que les contours) n'exploite pas directement les informations sémantiques ou les structures contextuelles des objets présents dans l'image. En revanche, pour les modèles de compression ROI, la situation est différente. Ces derniers, en apprenant indirectement à détecter les objets dans les images, peuvent potentiellement mieux préserver les informations critiques pour la détection. Plus ces modèles sont performants dans leur capacité de détection, moins la compression risque d'affecter négativement les performances des modules de détection en aval. Ainsi, réentraîner ces modèles sur un volume accru d'images annotées semble pouvoir réduire les probabilités d'erreur et améliorer la robustesse globale du système. C'est pourquoi nous avons réentraîné notre modèle STF en mode ROI avec les hyperparamètres $\{\lambda^- = 1.25 \cdot 10^{-5}, \lambda^+ = 0.08\}$, en utilisant les datasets augmentés $D_{train_aug_front}$ et D_{train_aug}, partitions utilisées précédemment augmentées avec les données de NuImagesLabels (cf. Table 6.14). Cette stratégie vise à tirer parti des informations supplémentaires fournies par les annotations des nouvelles images, renforçant ainsi la capacité du modèle à préserver les zones liées aux objets lors de la compression.

TABLE 6.14 – Modèles de compression STF entrainé avec la metrique MSE appris ROI augmentés.

Architecture	Datasets d'entrainement	Nom	Hyperparamètres λ
STF	$D_{train_aug_front}$ D_{valid}	stf(mse)	$\lambda_{10} = \{\lambda^- = 1.25 \cdot 10^{-5}, \lambda^+ = 0.08\}$
	D_{train_aug} D_{valid}		$\lambda_{11} = \{\lambda^- = 1.25 \cdot 10^{-5}, \lambda^+ = 0.08\}$

En appliquant les étapes d'estimation intra-dataset et inter-dataset pour calculer les probabilités d'erreur $\hat{\theta}$, nous constatons, comme le montre la Figure 6.37, que cette optimisation n'a pas produit les résultats escomptés. Dans la Figure 6.37, il est évident qu'il n'y a pratiquement aucune différence notable entre les performances des compresseurs ROI associés aux hyperparamètres λ_8 et λ_9 (représentés par des triangles) et celles des nouveaux compresseurs entraînés avec λ_{10} et λ_{11} (représentés par des croix). Les probabilités d'erreur restent quasiment superposées, indiquant que l'ajout de données annotées n'a pas significativement amélioré la robustesse de YOLOv7 à la compression des modèles ROI.

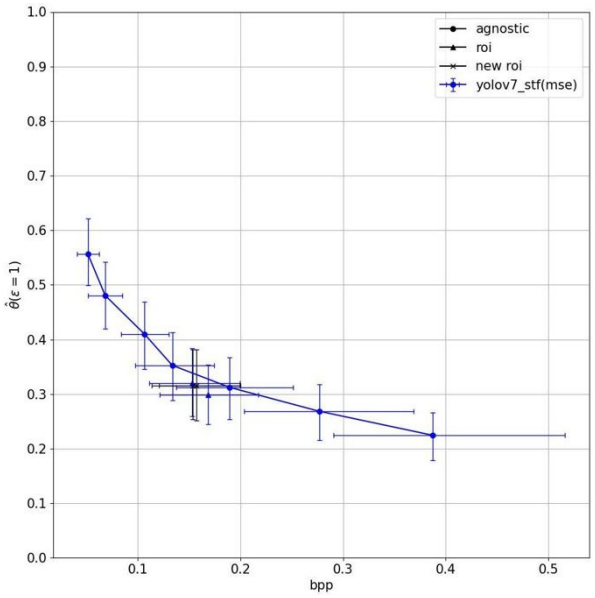


FIGURE 6.37 – Estimation de la probabilité d'erreur $\hat{\theta}$ pour le modèle de compression stf(mse) agnostique et ROI augmenté avec les modèles de détection yolov7.

En conclusion, cette tentative de réentraînement des modèles ROI avec un volume accru de données annotées n'a pas conduit à une réduction significative des probabilités d'erreur estimées. Ces résultats indiquent que les compresseurs ROI n'ont pas améliorer de manière notable leur capacité à détecter efficacement les objets d'intérêt.

6.6 Génération d'images par diffusion

Pour compléter la stratégie d'obtention de garanties de performance, nous avions proposé, dans la section 4.3.3, de générer des données, et en particulier des images, à partir de modèles génératifs orientés via un gradient. L'objectif principal de cette approche par diffusion est de produire des images qui sont particulièrement difficiles à traiter ou susceptibles d'induire des erreurs dans le système étudié. Ces images, souvent appelées *Natural Adversarial Examples*, permettent d'explorer les limites opérationnelles du système, de révéler des scénarios critiques et d'identifier des vulnérabilités potentielles dans les modules d'IA. En dirigeant le processus de génération vers des configurations qui maximisent les erreurs, cette méthode offre un outil puissant pour tester rigoureusement les performances des modules, tels que la compression ou la détection, dans des conditions extrêmes. En outre, les données générées peuvent être exploitées pour réentraîner les modèles, renforçant ainsi leur robustesse face à des cas complexes et atypiques. Cette approche constitue un levier essentiel pour améliorer la fiabilité et la performance des systèmes intégrant des modules d'intelligence artificielle.

Les modèles de diffusion, que nous avons précédemment entraînés, génèrent des images à partir d'un bruit gaussien initial, un processus itératif permettant de transformer un échantillon aléatoire en une image réaliste. L'idée clé réside dans la manipulation ciblée de ce bruit ou de son équivalent dans l'espace latent, afin d'orienter la génération vers des images maximisant les erreurs du système. Ce mécanisme repose sur l'utilisation du gradient d'une fonction d'erreur l pour modifier de manière itérative l'entrée latente z_T, comme illustré dans la Figure 6.38. Ce procédé s'inspire des approches décrites dans la section 3.4.6, mais l'adapte au cadre de la génération d'images via modèles de diffusion. Une contrainte majeure du processus de diffusion repose sur le caractère probabiliste des modèles tels que le DDPM (Denoising Diffusion Probabilistic Models). En effet, ce caractère stochastique peut entraîner une instabilité dans la génération, limitant la capacité à produire des scénarios spécifiques ou des exemples adversariaux reproductibles. Pour contourner cette limitation, nous adoptons l'approche DDIM (Denoising Diffusion Implicit Models), qui offre une meilleure contrôle sur la trajectoire de génération tout en réduisant la variabilité des résultats. Le DDIM, grâce à sa formulation déterministe, permet ainsi de générer des images plus cohérentes et alignées avec les objectifs d'exploration des erreurs du système.

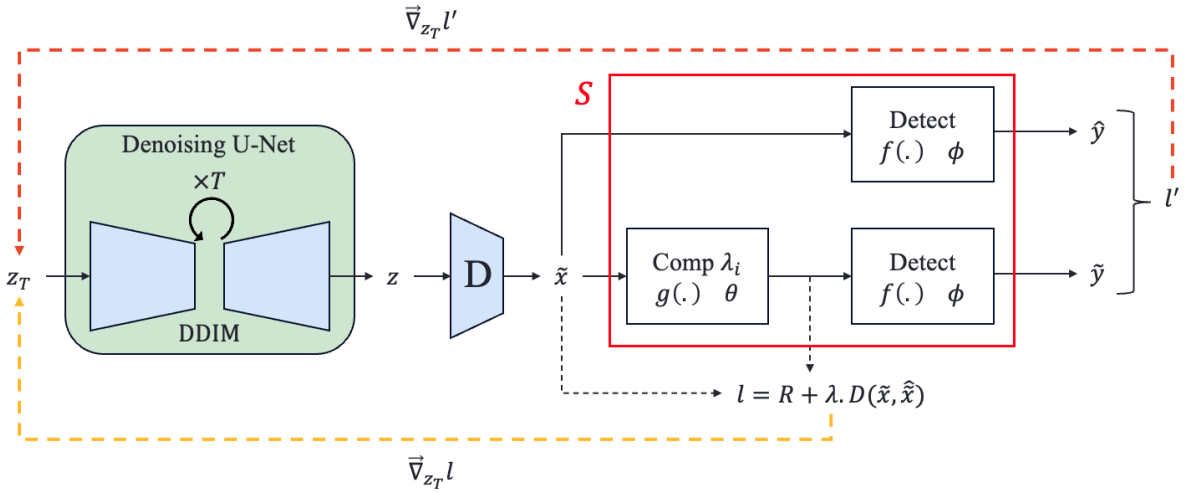


FIGURE 6.38 – Processus de génération d'image orienté pour augmenter une fonction d'erreur.

Dans ce contexte, le bruit initial ou latent, noté z_T, est ajusté pour influencer la génération d'images de manière à maximiser les erreurs dans le système étudié. Ce processus d'ajustement est similaire à celui utilisé pour modifier les poids d'un réseau de neurones durant l'entraînement. À chaque itération, une fonction de perte l est calculée, suivie du calcul de son gradient par rapport au latent d'origine z_T, soit $\vec{\nabla}_{z_T} l$. Le latent z_T est ensuite mis à jour par rétropropagation à l'aide de l'optimisateur Adam avec un pas lr. Dans le cadre de notre application et dans le contexte du *Field Monitoring*, plusieurs options sont envisageables pour définir la fonction de perte l. Cette fonction peut être adaptée en fonction des besoins spécifiques, notamment pour tester les limites du système ou simuler des scénarios critiques. Voici les principales approches possibles pour définir l :

1. Maximiser l'erreur liée à la compression : Dans cette approche, la fonction de perte l est conçue pour amplifier les erreurs induites par la compression. Deux objectifs principaux peuvent être poursuivis :
   - Maximisation du taux de compression R (mesuré en bits par pixel, bpp).

- Minimisation de la qualité de l'image reconstruite $D(\tilde{x}, \hat{\tilde{x}})$, généralement évaluée par des métriques telles que le MSE ou le MS-SSIM.

Ces définitions permettent de générer des images qui, lorsqu'elles sont compressées, mettent en évidence les faiblesses des modules de compression et de reconstruction.

2. Maximiser l'erreur de détection induite par la compression : Cette approche se concentre sur l'impact de la compression sur les performances des systèmes de détection. La fonction l est définie comme l'erreur entre les détections effectuées sur une image compressée et celles effectuées sur une image non compressée. Étant donné que la métrique d'erreur $e(x)$ définie précédemment n'est pas différentiable, nous utilisons à la place la fonction de perte de YOLOv7 employée durant l'entraînement :

$$l = 0.05 \cdot l_{box} + 0.7 \cdot l_{obj} + 0.3 \cdot l_{cls} \tag{6.15}$$

Cette fonction se décompose en trois termes principaux, chacun correspondant à un aspect critique de la détection d'objets :

- Perte de localisation (l_{box}) : La perte de localisation mesure la précision des boîtes englobantes prédites par rapport aux boîtes de références. Elle utilise la CIoU (*Complete Intersection over Union*), une variante avancée de l'IoU, pour prendre en compte à la fois le recouvrement des boîtes, leur distance relative, et leur rapport d'aspect :

$$l_{box} = \frac{1}{N} \sum_{(i,j) \in \text{correspondances}} \left(1 - \boldsymbol{CIoU}\left(\tilde{x}_i^{box}, \hat{\tilde{x}}_j^{box}\right)\right) \tag{6.16}$$

où $\tilde{x}_i^{box}$ et $\hat{\tilde{x}}_j^{box}$ représentent respectivement les coordonnées des boîtes englobantes dans l'image non compressée et l'image compressée, et N est le nombre de boîtes totales.

- Perte d'*objectness* (l_{obj}) : La perte d'objectivité (*objectness*) quantifie la confiance du modèle concernant la présence d'un objet dans une cellule donnée. Elle est calculée à l'aide de l'entropie croisée binaire (BCE) entre les niveaux de confiance prédits et ceux de références :

$$l_{obj} = \frac{1}{N} \sum_{(i,j) \in \text{correspondances}} \boldsymbol{BCE}\left(\tilde{x}_i^{conf}, \hat{\tilde{x}}_j^{conf}\right) \tag{6.17}$$

où $\tilde{x}_i^{conf}$ et $\hat{\tilde{x}}_j^{conf}$ représentent respectivement les niveaux de confiance des boîtes englobantes dans l'image non compressée et l'image compressée, et N est le nombre de boîtes totales.

- Perte de classification (l_{cls}) : Cette perte mesure l'erreur dans la classification des objets détectés. Elle compare les distributions de probabilités des classes prédites avec celles de références, également à l'aide de l'entropie croisée binaire :

$$l_{cls} = \frac{1}{N} \sum_{(i,j) \in \text{correspondances}} \boldsymbol{BCE}\left(\tilde{x}_i^{cls}, \hat{\tilde{x}}_j^{cls}\right) \tag{6.18}$$

où $\tilde{x}_i^{cls}$ et $\hat{\tilde{x}}_j^{cls}$ représentent respectivement les probabilités prédites pour chaque classe dans l'image non compressée et l'image compressée, et N est le nombre de boîtes totales.

Si l'on souhaite explorer davantage l'espace latent tout en contrôlant l'amplitude des déplacements dans celui-ci, il peut être intéressant d'ajouter à chacune des fonctions de perte ci-dessus un terme de régularisation de la forme :

$$l = l + \eta \cdot \parallel z_{origine} - z \parallel_2 \tag{6.19}$$

où η est un hyperparamètre qui contrôle le poids de la régularisation, $z_{origine}$ représente le bruit initial ou le point de départ dans l'espace latent, et z correspond au point courant après une ou plusieurs étapes d'optimisation. Ce terme permet de limiter les déplacements excessifs dans l'espace latent, garantissant que les images générées restent proches des distributions apprises par le modèle de diffusion, tout en maximisant les erreurs ciblées. Cette approche équilibre ainsi l'exploration des cas et la cohérence des données générées avec les structures visuelles attendues. Dans la suite, nous explorerons plusieurs applications illustrant l'intérêt de ces méthodes dans des contextes variés.

Pour vérifier le bon fonctionnement du procédé, nous avons conçu une série d'expériences rigoureuses visant à évaluer la capacité de notre système à générer des images orientées qui maximisent la fonction de perte cible.

Pour la génération, nous avons choisi d'utiliser exclusivement le modèle de diffusion diff_front. Ce choix repose sur une observation clé : les images issues de ce modèle présentent une fréquence plus élevée d'objets d'intérêt, comme des véhicules, des piétons et des cyclistes, par rapport au modèle diff_all. Cette différence s'explique par la nature des données utilisées pour l'entraînement. En effet, dans les ensembles de données disponibles, la caméra avant capture bien plus fréquemment des scènes contenant des objets pertinents pour notre étude, contrairement aux autres caméras embarquées. Pour nos expériences, nous utiliserons le modèle de compression $\text{stf(mse)}_{\lambda_7}$ et le modèle de détection YOLOv7. Par rapport à la méthode classique DDPM, DDIM permet de produire des images de haute qualité tout en réduisant significativement le nombre d'itérations nécessaires. Pour nos expériences, nous avons fixé $T = 100$, ce qui représente une réduction notable des étapes de débruitage sans compromettre la qualité visuelle des images générées. En termes de performance, chaque cycle complet de génération d'images, incluant le calcul de l'erreur, l'évaluation du gradient, et la rétropropagation, requiert environ 25 secondes par image. Ce temps comprend à la fois les calculs liés au modèle de diffusion et les étapes d'évaluation nécessaires pour orienter la génération des données vers des cas spécifiques maximisant la fonction de perte. Ce compromis en termes de temps de calcul est acceptable dans notre contexte, car il permet d'obtenir des résultats assez rapidement.

6.6.1 Maximiser l'erreur liée à la compression

Dans un premier temps, l'objectif est de générer des images qui maximisent l'erreur liée à la compression. Pour cela, nous définissons une fonction de perte $l = \text{MSE}(\tilde{x}, \hat{\tilde{x}})$, où $\tilde{x}$ est l'image générée et $\hat{\tilde{x}}$ est son approximation après compression et décompression. Les hyperparamètres utilisés pour orienter la génération incluent un pas fixé à $lr = 0.0005$ et un paramètre de régularisation $\eta = 0.001$, qui limite la magnitude des modifications apportées au latent initial. D'après la Figure 6.39, on observe que, à chaque étape de génération, la perte MSE augmente progressivement, indiquant une dégradation accrue de la qualité de l'image reconstruite après compression. En parallèle, la distance entre le latent d'origine z_{origine} et le latent courant z s'accroît continuellement, ce qui reflète les ajustements successifs du latent pour maximiser l'erreur cible. Ces résultats confirment que l'orientation du processus de diffusion permet de générer des images présentant une complexité croissante pour le module de compression, rendant plus difficile une reconstruction fidèle des données originales.

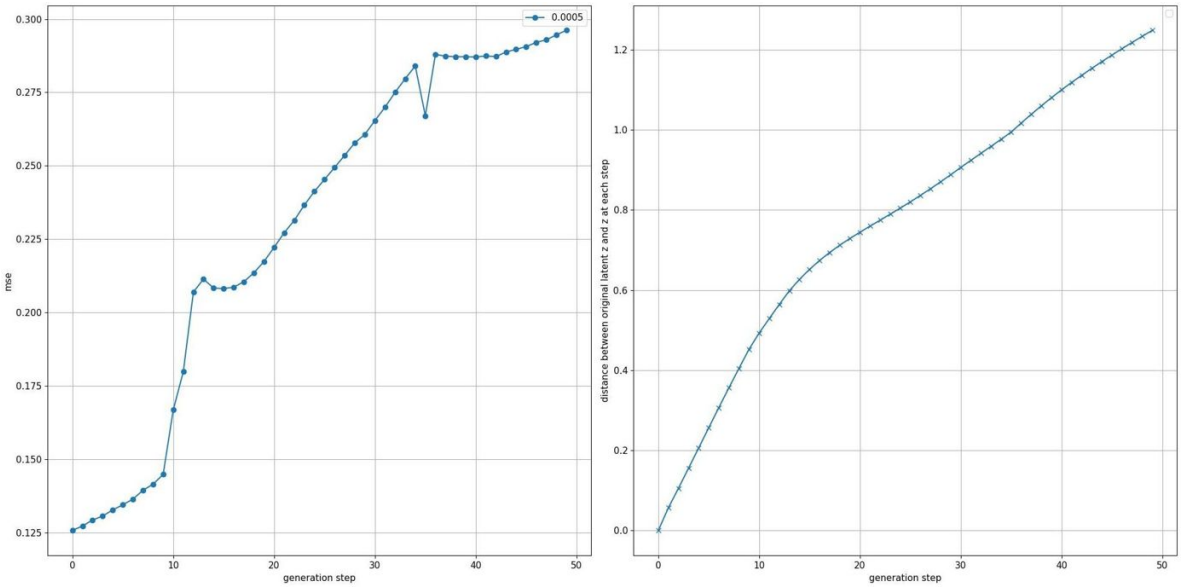


FIGURE 6.39 – Évolution de la fonction de perte l (gauche) et de la distance L_2 entre le latent d'origine $z_{origine}$ et le latent courant z (droite) à chaque étape de génération.

On observe que le processus de génération offre un certain contrôle sur le contenu sémantique des images. En effet, dans les images de la Figure 6.44, la première image présente ce qui semble être un camion au premier plan. Progressivement, au fil des étapes de génération, le camion disparaît pour laisser place à une route vide bordée par un ciel bleu et davantage de végétation. Ce phénomène illustre la manière dont le modèle peut être orienté pour modifier des éléments spécifiques dans l'image. Notamment, la présence accrue de végétation, qui

est connue pour introduire des textures complexes et irrégulières, est un facteur bien identifié pour augmenter le MSE dans les images compressées. Cela met en évidence l'utilité de la génération orientée : en adaptant le contenu des images, il devient possible de cibler des caractéristiques spécifiques du système étudié, comme les zones où la compression est susceptible de causer des distorsions plus importantes.



(a) 1er image générée : mse=0.1257.



(b) 9e image générée : mse=0.1394.



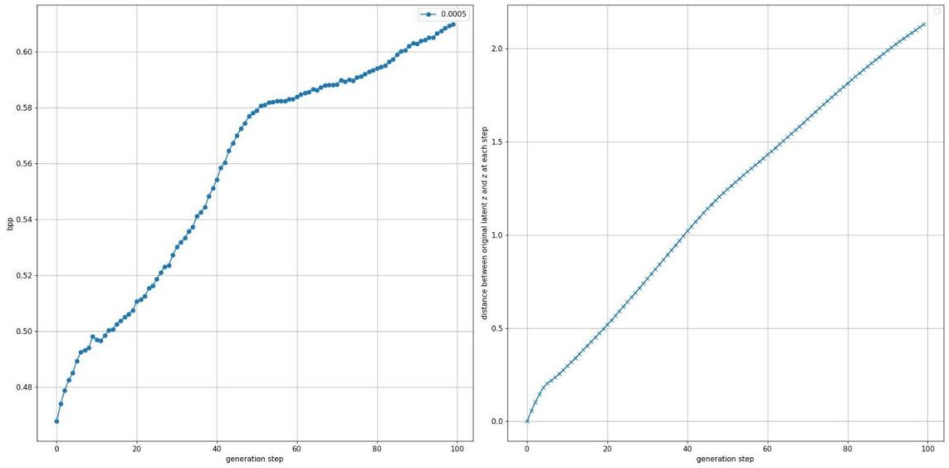
(c) 11e image générée : mse=0.1643.



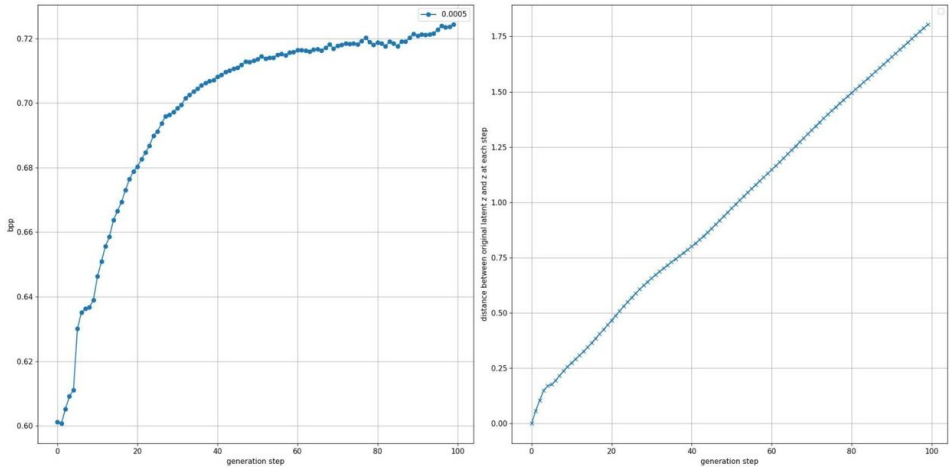
(d) 36e image générée : mse=0.2621.

FIGURE 6.40 – Images générées par diffusion avec orientation pour maximiser le MSE.

De manière similaire, nous appliquons le processus de génération à la fonction de perte $l = bpp$ (bits par pixel), en conservant les mêmes hyperparamètres ($lr = 0.0005$ et $\eta = 0.001$). Deux exemples d'images générées sont présentés dans les Figures 6.41 et 6.42. Dans les deux cas, la distance entre le latent d'origine $z_{origine}$ et le latent courant z augmente progressivement de manière presque linéaire. Cependant, l'évolution du bpp suit une tendance différente. Contrairement à une augmentation linéaire, le bpp croît de manière non linéaire, avec des variations plus significatives à certains points. Cela pourrait indiquer que le modèle de diffusion modifie certaines zones spécifiques de l'image pour allouer davantage de bits, par exemple en augmentant les détails visuels ou les textures dans des régions clés. Ces résultats mettent en évidence que, bien que le processus latent soit contrôlé de manière uniforme, la réponse en termes de bpp peut être influencée par des interactions complexes entre les régions visuelles et les priorités d'allocation des bits du compresseur.



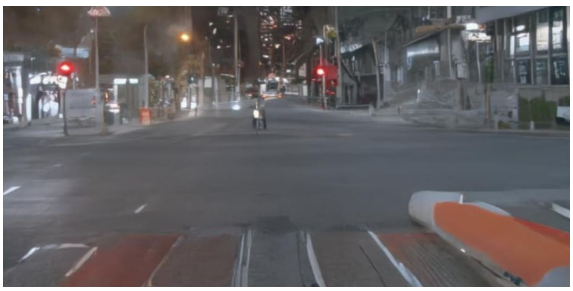
(a) Exemple 1.



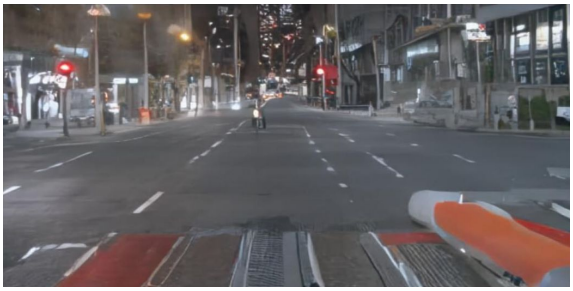
(b) Exemple 2.

FIGURE 6.41 – Évolution de la fonction de perte l (gauche) et de la distance L_2 entre le latent d'origine $z_{origine}$ et le latent courant z (droite) à chaque étape de génération.

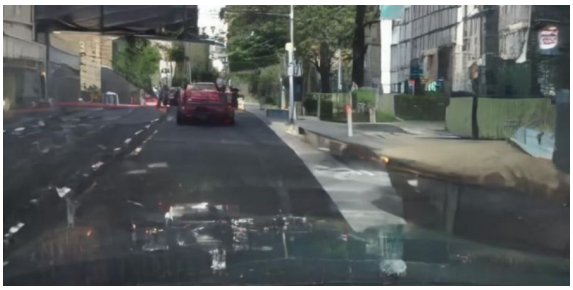
Similaire à l'exemple précédent avec le MSE, on peut observer que les modifications apportées par le processus de diffusion orienté influencent à la fois les détails visuels et, dans certains cas, le contenu sémantique des images générées. Dans le premier exemple (cf. Figures 6.42a et 6.42b), on constate que, tout au long du processus de génération, l'image gagne progressivement en détails, notamment les lignes de circulation au sol qui deviennent plus nettes et mieux définies. Toutefois, aucun changement significatif dans le contenu sémantique de l'image n'est visible, ce qui suggère que l'augmentation du bpp résulte principalement d'une amélioration des textures et des éléments visuels présents dans la scène. En revanche, dans le second exemple (cf. Figures 6.42c et 6.42d), un changement sémantique majeur est observé. On assiste à l'apparition de multiples véhicules sur la voie, modifiant ainsi de manière significative le contexte de l'image. Ces modifications sémantiques, associées à une augmentation des détails visuels, contribuent fortement à une élévation du bpp, car les véhicules ajoutés introduisent des zones complexes qui nécessitent une allocation accrue de bits pour préserver leur structure lors de la compression.



(a) 1er image générée : bpp=0.4678.



(b) 100e image générée : bpp=0.6098.



(c) 1er image générée : bpp=0.6012.



(d) 100e image générée : bpp=0.7243.

FIGURE 6.42 – Deux exemples d'images générées par diffusion avec orientation pour maximiser le bpp.

En conclusion, les résultats mettent en évidence des comportements distincts dans les processus de génération orientés par MSE et par bpp, soulignant leurs dynamiques spécifiques et leurs impacts sur le contenu visuel et sémantique des images générées. La génération orientée par MSE tend à produire des images avec des couleurs vives, une augmentation notable des détails visuels, et des modifications fréquentes du contenu sémantique. L'apparition de végétation, comme des buissons, arbres ou pelouses, est l'une des modifications les plus courantes. Ce comportement peut être expliqué par le fait que les textures complexes et les variations chromatiques introduites par ces éléments augmentent considérablement le MSE lorsqu'elles sont compressées. Ces changements sémantiques marqués indiquent que le processus oriente les images vers des scénarios visuellement plus complexes et riches, susceptibles de mettre en difficulté les algorithmes de compression. En revanche, la génération orientée par bpp est plus subtile et présente une instabilité notable. Contrairement à la génération par MSE, elle ne se traduit pas nécessairement par des changements significatifs du contenu sémantique. Dans de nombreux cas, l'augmentation du bpp résulte principalement de l'ajout de détails visuels, tels que des lignes de circulation ou des textures améliorées, sans altérer fondamentalement la composition de la scène. Cependant, dans certains exemples, des changements sémantiques majeurs peuvent apparaître, comme l'introduction de nouveaux objets dans l'image. Cette instabilité souligne la difficulté de contrôler de manière cohérente le contenu généré en orientant uniquement la génération par bpp. Ces observations suggèrent que, bien que les deux méthodes permettent de manipuler le processus de génération pour influencer des propriétés spécifiques des images, leur efficacité dépend des objectifs visés. La génération par MSE est particulièrement adaptée pour explorer des scénarios visuellement complexes et diversifiés, tandis que la génération par bpp pourrait être exploitée pour tester des subtilités liées à la gestion des ressources de compression sans altérer drastiquement le contenu sémantique de l'image.

6.6.2 Maximiser l'erreur de détection induite par la compression

Ici, nous cherchons à générer des images qui maximisent l'erreur de détection induite par la compression, en ciblant spécifiquement les faiblesses du système de détection après passage par le module de compression.

Similairement aux expériences menées dans les sections précédentes, nous appliquerons des critères stricts pour l'évaluation des détections. Plus précisément, seules les boîtes prédites avec un niveau de confiance conf $\geq$ 0.5 seront prises en compte. De plus, pour associer les boîtes prédites aux boîtes de référence, nous utiliserons une condition d'IoU avec un seuil fixé à IoU ≥ 0.5. Dans les précédentes expériences, réalisées sur des images de taille 640×1280, nous avions défini des seuils de taille minimale pour les boîtes en fonction des classes d'objets : 1200 pixels pour les véhicules, et 500 pixels pour les piétons et cyclistes. Ces seuils ont été établis

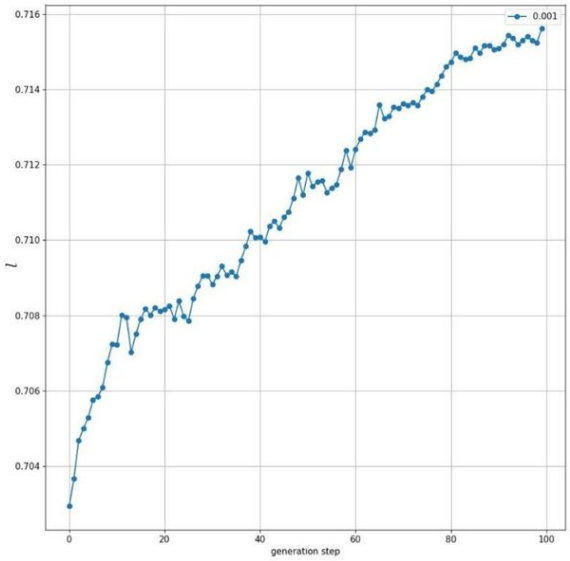
afin de filtrer les détections non pertinentes, notamment celles correspondant à des objets trop petits ou trop loin pour être significatifs. Cependant, dans le cadre des présentes expériences, les modèles de diffusion utilisés génèrent des images avec des dimensions réduites de 320×600, soit la moitié des dimensions précédentes. En conséquence, les seuils de taille minimale des boîtes doivent être ajustés proportionnellement. Ainsi, nous considérons des tailles minimales de 600 pixels pour les véhicules et 250 pixels pour les piétons et cyclistes. Ces ajustements garantissent que les critères de détection restent cohérents avec la résolution des images générées, tout en maintenant une évaluation robuste et équitable des performances.

Pour orienter la génération des images, nous utilisons la fonction de perte du modèle YOLOv7, définie comme suit :

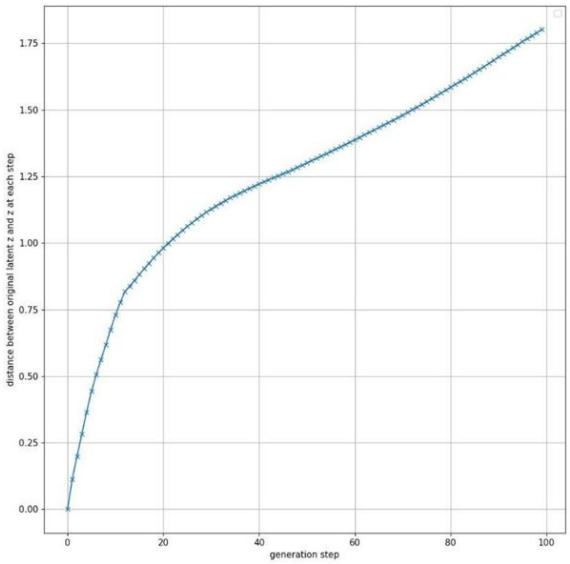
$$l = 0.05 \cdot l_{box} + 0.7 \cdot l_{obj} + 0.3 \cdot l_{cls} \tag{6.20}$$

où l_{box} représente la perte liée aux coordonnées des boîtes prédites, l_{obj} la perte associée à la détection de la présence d'objets, et l_{cls} la perte relative à la classification des objets détectés. Pour l'optimisation, nous fixons les hyperparamètres suivants : un taux d'apprentissage $lr = 0.001$ et un paramètre de régulation $\eta = 0.001$.

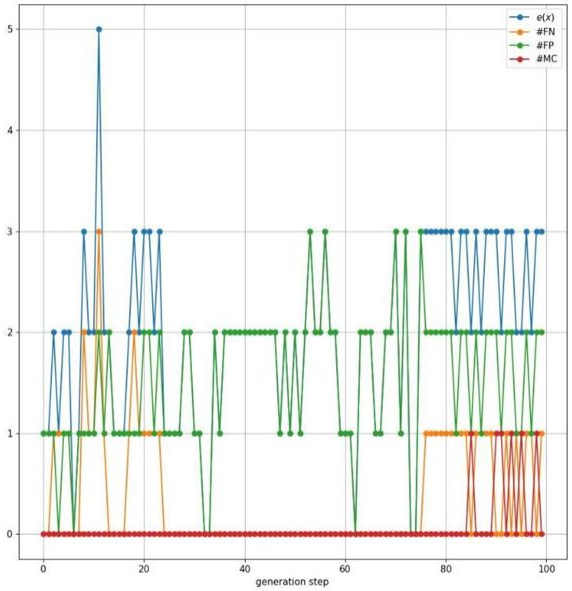
Comme illustré dans les Figures 6.43a et 6.43b, la fonction de perte l augmente de manière progressive à chaque étape de génération, ce qui indique que le processus oriente efficacement les images vers des cas plus difficiles pour le modèle de détection. Cette augmentation contrôlée de la perte reflète une optimisation cohérente pour générer des scénarios qui exacerbent les faiblesses du système. Parallèlement, la distance entre le latent initial z_{origine} et le latent courant z croît de façon continue, traduisant l'évolution graduelle des caractéristiques visuelles et sémantiques des images générées. Cette dynamique démontre que chaque étape de génération contribue à orienter les données vers des configurations critiques pour le système. Cependant, comme le montre la Figure 6.43, on n'observe pas de croissance notable de l'erreur de détection. Bien que les images générées soient modifiées pour tenter d'augmenter les erreurs, le système de détection semble capable de s'adapter à ces changements, limitant ainsi l'impact sur la quantité totale d'erreurs. Cette stabilité pourrait être liée à la robustesse inhérente du modèle de détection, qui parvient à maintenir ses performances malgré les variations visuelles induites par le processus de génération.



(a) Évolution de la fonction de perte l à chaque étape de génération.



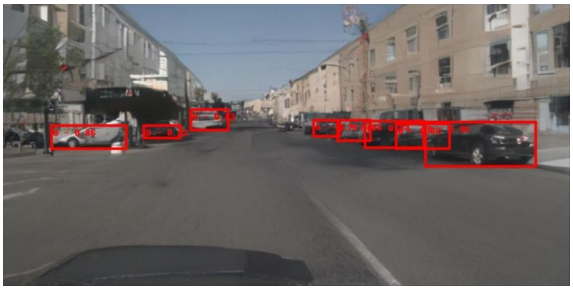
(b) Évolution de la distance L_2 entre le latent d'origine $z_{origine}$ et le latent courant z à chaque étape de génération.



(c) Évolution de l'erreur $e(x)$, ainsi que des valeurs de $\#FN$ (orange), $\#FP$ (vert) et $\#MC$ (rouge) à chaque étape de génération.

FIGURE 6.43 – Évolution de la fonction de perte l (haut gauche), de la distance L_2 entre le latent d'origine z_{origine} et le latent courant z (haut droite), ainsi que de la quantité d'erreur (bas), à chaque étape de génération.

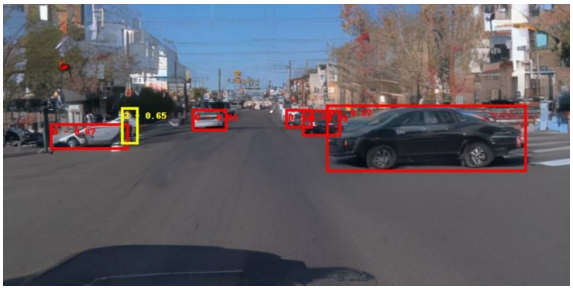
Dans les images générées (cf. Figure 6.44), on observe des changements significatifs de la sémantique au fil des étapes de génération. Par exemple, on constate une augmentation progressive de la végétation, avec l'apparition d'arbres et de buissons de part et d'autre de l'image. Parallèlement, le ciel devient plus clair et uniformément bleu, traduisant une simplification visuelle de l'arrière-plan. Les lignes de marquage au sol, initialement présentes et bien définies, tendent à disparaître progressivement, contribuant à une modification notable de la structure de la scène. Enfin, un véhicule au premier plan fait son apparition, ajoutant un nouvel élément sémantique à l'image. Ces transformations illustrent la capacité du modèle de diffusion à générer des scénarios visuellement distincts et à orienter le contenu généré vers des caractéristiques susceptibles de poser des défis au système étudié.



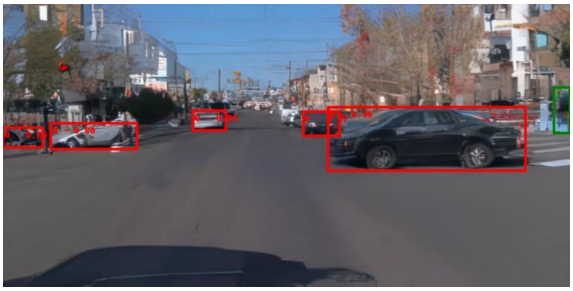
(a) 1er image générée.



(b) 1er image générée et compressée : e(x)=1, #FN=0, #FP=1, #MC=0.



(c) 99e image générée.

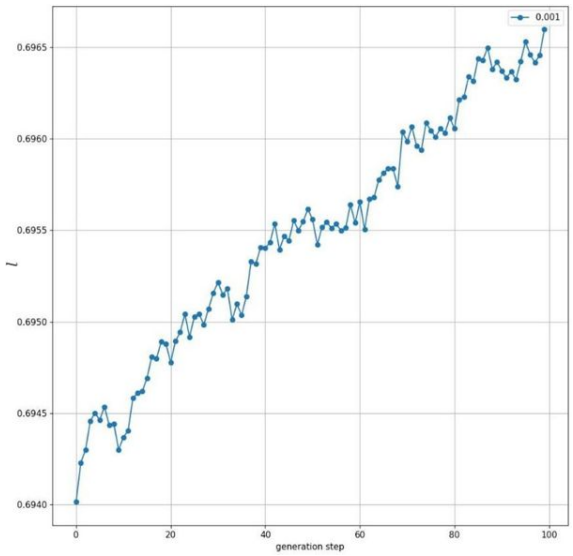


(d) 99e image générée et compressée : e(x)=3, #FN=1, #FP=2, #MC=0.

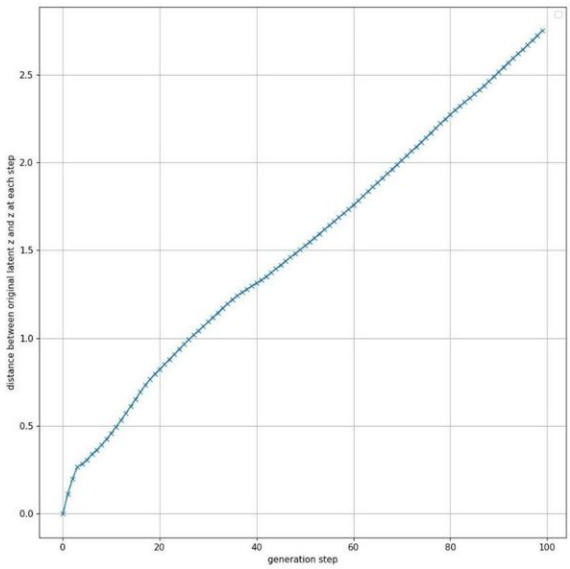
FIGURE 6.44 – Détections sur images générées et compressées, orientées pour maximiser la fonction de perte de YOLOv7. Code couleur : véhicule (rouge), piéton (vert), cycliste (jaune).

Il est complexe de comprendre comment les trois composantes de la fonction de coût (l_{obj}, l_{cls} et l_{box}) interagissent individuellement sur le processus de génération d'images et leur impact sur l'erreur $e(x)$. Pour clarifier ces interactions, nous avons entrepris des expériences où chacune des composantes est isolée et utilisée indépendamment comme fonction de perte pour orienter la génération d'images. Ces expériences visent à mieux comprendre les types d'erreurs induits par chaque composante.

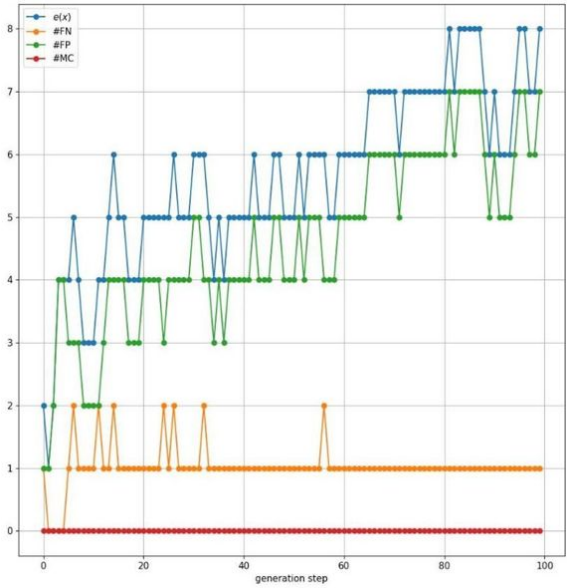
Lorsque nous utilisons $l = l_{obj}$ comme fonction de perte, l'objectif est d'amplifier l'erreur liée au niveau de confiance des prédictions. Dans ce cas, on s'attend à une augmentation des erreurs de type faux négatifs (FN) et faux positifs (FP). En pratique, cela signifie que certaines boîtes de détection initialement présentes avant compression pourraient disparaître après compression en raison d'une diminution excessive de leur niveau de confiance (FN), tandis que d'autres boîtes pourraient apparaître de manière erronée en raison d'une surévaluation du niveau de confiance (FP). C'est précisément ce qui est observé dans la Figure 6.45. À chaque étape de génération, la quantité d'erreurs augmente de manière significative, avec une prédominance des faux positifs. Cela indique que le processus de génération oriente efficacement les images vers des scénarios où la compression perturbe les prédictions en introduisant de nouvelles boîtes incorrectes.



(a) Évolution de la fonction de perte l à chaque étape de génération.



(b) Évolution de la distance L_2 entre le latent d'origine $z_{origine}$ et le latent courant z à chaque étape de génération.



(c) Évolution de l'erreur $e(x)$, ainsi que des valeurs de $\#FN$ (orange), $\#FP$ (vert) et $\#MC$ (rouge) à chaque étape de génération.

FIGURE 6.45 – Évolution de la fonction de perte l (haut gauche), de la distance L_2 entre le latent d'origine z_{origine} et le latent courant z (haut droite), ainsi que de la quantité d'erreur (bas), à chaque étape de génération.

Dans les images générées (cf. Figure 6.46), on observe que le contenu sémantique global reste relativement stable. Les éléments principaux de la scène, tels que les véhicules, la route ou les panneaux, ne subissent pas de modifications significatives au niveau de leur disposition ou de leur type. Pourtant, malgré cette stabilité sémantique apparente, les erreurs de détection augmentent de manière significative. Cela suggère que l'augmentation des erreurs est davantage liée à des perturbations subtiles dans les caractéristiques visuelles ou structurelles des images plutôt qu'à une altération du contenu sémantique lui-même. Par exemple, des variations dans l'éclairage, des changements mineurs dans les contours des objets, ou des détails ajoutés dans l'arrière-plan pourraient affecter la manière dont le modèle de détection interprète les niveaux de confiance. Ces changements visuels, bien que presque imperceptibles à l'œil humain, semblent suffire pour perturber le détecteur et engendrer une augmentation marquée des FP et FN.



(a) 1er image générée.



(b) 1er image générée et compressée : e(x)=2, #FN=1, #FP=1, #MC=0.



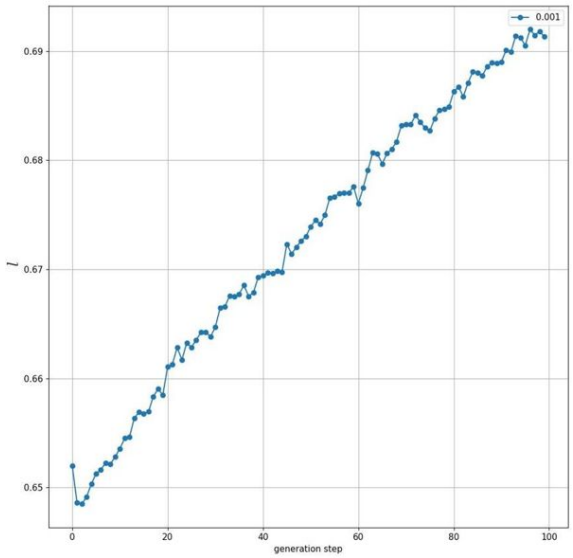
(c) 99e image générée.



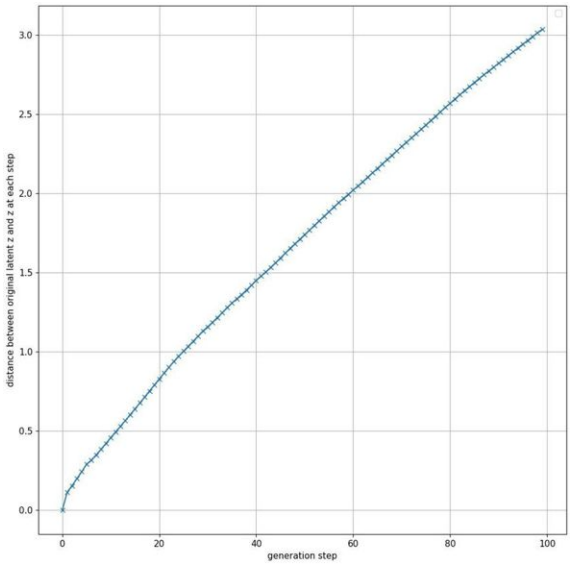
(d) 99e image générée et compressée : e(x)=8, #FN=1, #FP=7, #MC=0.

FIGURE 6.46 – Détections sur images générées et compressées, orientées pour maximiser la fonction de perte l_{obj}. Code couleur : véhicule (rouge), piéton (vert), cycliste (jaune).

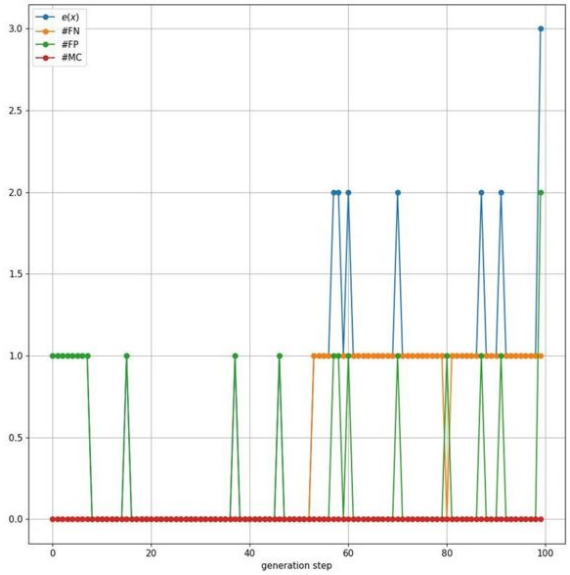
Avec $l = l_{cls}$, la perte cible concerne la classification des objets détectés. Théoriquement, cela devrait induire une augmentation des erreurs de type MC (mauvaise classification), où un objet est détecté mais mal classé (par exemple, un cycliste identifié comme un piéton). Cependant, les résultats observés ne correspondent pas toujours à cette hypothèse. Cela suggère que l'impact de l_{cls} sur la génération d'images est moins direct ou moins dominant que prévu. Dans la Figure 6.47, on observe une augmentation significative des erreurs de type FN (faux négatifs) et FP (faux positifs), mais pas d'augmentation notable des erreurs de type MC. Cela indique que, bien que l_{cls} cible théoriquement la classification, son effet sur la génération d'images semble davantage lié à des perturbations globales affectant la présence ou l'absence d'objets détectés. Cette observation suggère que la contribution de l_{cls} à la génération d'images critiques n'est pas isolée, mais pourrait interagir de manière complexe avec d'autres composantes du système, telles que les niveaux de confiance (l_{obj}) ou les ajustements des boîtes englobantes (l_{box}). Une autre explication plausible est que les effets de l_{cls} ne sont pas immédiatement perceptibles. En d'autres termes, avant qu'un objet ne soit mal classé (erreur de type MC), son niveau de confiance peut diminuer au point où il n'est plus détécté du tout, conduisant plutôt à une erreur de type FN. Cette dynamique expliquerait pourquoi les erreurs de type MC restent rares, tandis que les FN et FP augmentent, même lorsque la génération est orientée par l_{cls}.



(a) Évolution de la fonction de perte l à chaque étape de génération.



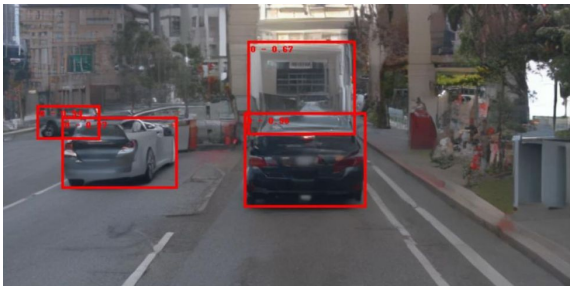
(b) Évolution de la distance L_2 entre le latent d'origine $z_{origine}$ et le latent courant z à chaque étape de génération.



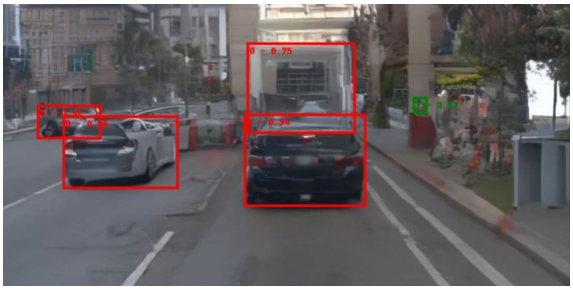
(c) Évolution de l'erreur $e(x)$, ainsi que des valeurs de $\#FN$ (orange), $\#FP$ (vert) et $\#MC$ (rouge) à chaque étape de génération.

FIGURE 6.47 – Évolution de la fonction de perte l (haut gauche), de la distance L_2 entre le latent d'origine z_{origine} et le latent courant z (haut droite), ainsi que de la quantité d'erreur (bas), à chaque étape de génération.

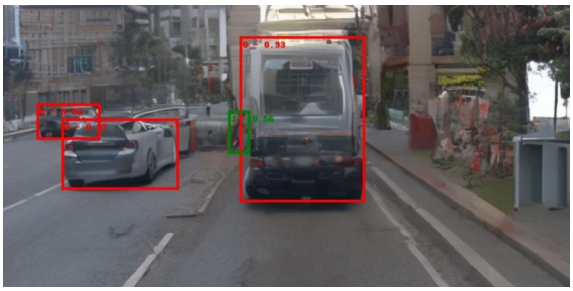
Dans les images générées (cf. Figure 6.48), les modifications visuelles des scènes restent globalement subtiles. On peut toutefois observer un changement notable : une voiture initialement présente dans l'image semble se transformer progressivement en camion au fil des étapes de génération. Ce type de modification illustre que la fonction de perte l_{cls} peut effectivement influer sur le contenu sémantique en ciblant des erreurs de classification. Cependant, ces transformations restent rares et relativement limitées par rapport aux autres fonctions de perte testées, telles que l_{obj}.



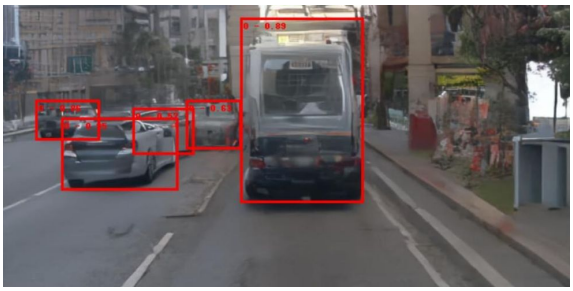
(a) 1er image générée.



(b) 1er image générée et compressée : e(x)=1, #FN=0, #FP=1, #MC=0.



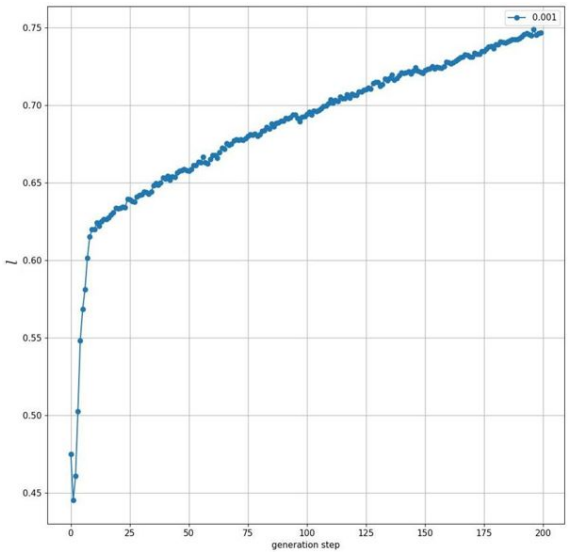
(c) 99e image générée.



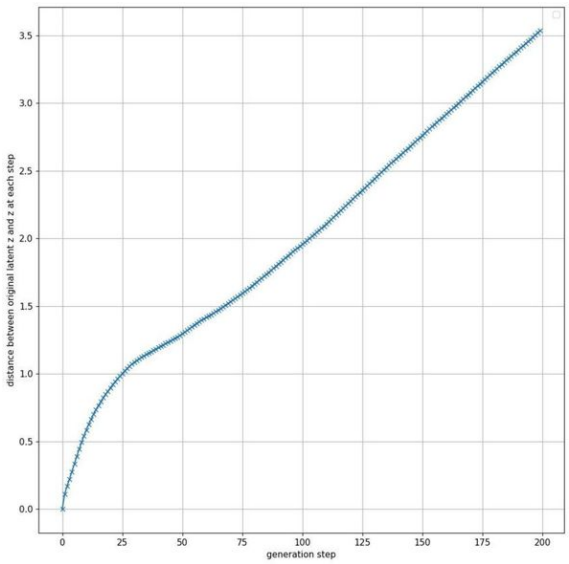
(d) 99e image générée et compressée : e(x)=3, #FN=1, #FP=2, #MC=0.

FIGURE 6.48 – Détections sur images générées et compressées, orientées pour maximiser la fonction de perte l_{cls}. Code couleur : véhicule (rouge), piéton (vert), cycliste (jaune).

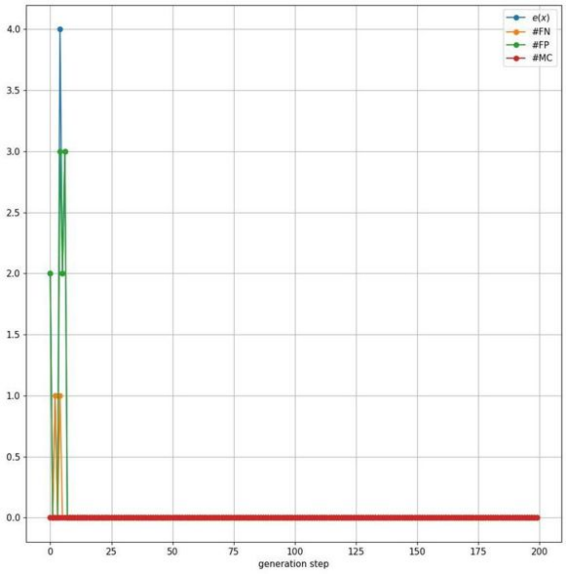
Enfin, lorsque $l = l_{box}$ est utilisé, la fonction de perte cible les paramètres géométriques des boîtes de détection, notamment leur position et leurs dimensions. Théoriquement, on pourrait s'attendre à ce qu'une augmentation de l_{box} entraîne une augmentation des erreurs de type FN et FP. Par exemple, si une boîte associée dans l'image non compressée et sa correspondance dans l'image compressée deviennent désalignées en raison de modifications géométriques, cela pourrait provoquer un désappairage, générant ainsi un FN pour l'objet initial et un FP pour l'objet nouvellement interprété. Cependant, les résultats observés ne corroborent pas cette hypothèse. Dans la Figure 6.49, on remarque que maximiser l_{box} ne conduit pas à une augmentation des erreurs de détection, mais semble plutôt avoir l'effet inverse : une réduction globale du contenu sémantique de l'image, ce qui fait diminuer la quantité d'erreurs. Ce phénomène peut s'expliquer par le fait que l'optimisation de l_{box} conduit à une simplification ou une suppression des objets dans la scène, réduisant ainsi le nombre de boîtes générées, et par conséquent, les FN et FP associés.



(a) Évolution de la fonction de perte l à chaque étape de génération.



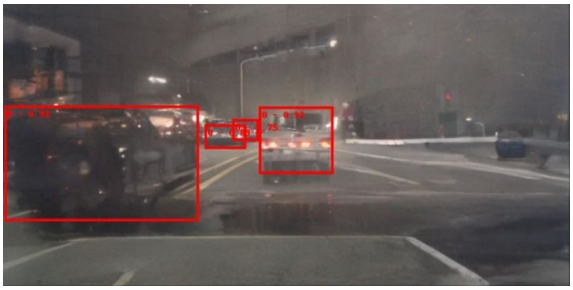
(b) Évolution de la distance L_2 entre le latent d'origine $z_{origine}$ et le latent courant z à chaque étape de génération.



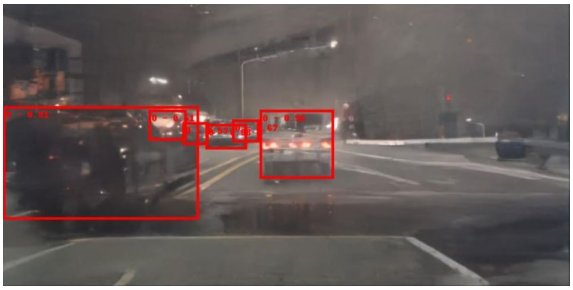
(c) Évolution de l'erreur $e(x)$, ainsi que des valeurs de $\#FN$ (orange), $\#FP$ (vert) et $\#MC$ (rouge) à chaque étape de génération.

FIGURE 6.49 – Évolution de la fonction de perte l (haut gauche), de la distance L_2 entre le latent d'origine z_{origine} et le latent courant z (haut droite), ainsi que de la quantité d'erreur (bas), à chaque étape de génération.

Dans les images générées (cf. Figure 6.50), on observe effectivement que le contenu sémantique, tel que les véhicules ou autres objets détectables, tend à disparaître progressivement. Par exemple, des véhicules présents dans les étapes initiales de la génération deviennent de moins en moins discernables, et certaines régions de l'image se transforment en éléments uniformes, comme des surfaces de route ou de ciel. Ces observations suggèrent que, contrairement aux attentes initiales, maximiser l_{box} agit davantage comme un mécanisme de simplification sémantique que comme perturbation géométrique des boîtes.



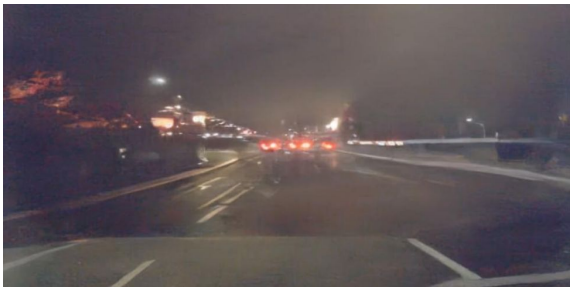
(a) 1er image générée.



(b) 1er image générée et compressée : e(x)=2, #FN=0, #FP=2, #MC=0.



(c) 99e image générée.



(d) 99e image générée et compressée : e(x)=0, #FN=0, #FP=0, #MC=0.

FIGURE 6.50 – Détections sur images générées et compressées, orientées pour maximiser la fonction de perte l_{box}. Code couleur : véhicule (rouge), piéton (vert), cycliste (jaune).

Ces expériences mettent en lumière l'interdépendance complexe des composantes de la fonction de coût dans le processus de génération. Bien que chaque composante cible un aspect spécifique de la détection, leurs effets sur l'erreur $e(x)$ ne sont pas toujours isolés et peuvent s'entremêler. Pour mieux comprendre ces interactions, des expériences complémentaires pourraient consister à moduler les pondérations de ces composantes ou à analyser leurs effets combinés dans des configurations contrôlées. En outre, l'étude de la relation entre les caractéristiques visuelles des images générées et les erreurs de détection pourrait fournir des indices supplémentaires pour optimiser la génération orientée.

En conclusion, ce type de génération orientée montre un potentiel prometteur pour identifier des scénarios critiques et explorer les faiblesses de notre système *Field Monitoring*. En particulier, la génération orientée par le MSE et l_{obj} s'est révélée efficace pour produire des images qui augmentent significativement les erreurs respectivement liées à la compression et au niveau de confiance des prédictions. Cependant, certaines limitations subsistent, notamment pour les fonctions de perte l_{box} et surtout l_{cls}. Leur impact sur la génération est encore mal maîtrisé, et des études plus approfondies seront nécessaires pour exploiter pleinement leur potentiel. Cela inclut la nécessité de mieux comprendre leurs interactions avec les autres composantes de la fonction de perte et leur influence sur les caractéristiques des images générées. Ces efforts permettront de compléter l'analyse des lacunes du système *Field Monitoring* et d'améliorer la pertinence des images critiques produites. Il convient de noter que toutes les générations d'images ne sont pas fructueuses. Certaines tentatives, non présentées ici, aboutissent à des résultats inutilisables en raison d'images trop éloignées de la réalité ou présentant des artefacts visuels. Cette limitation est en grande partie due à la qualité insuffisante des modèles de diffusion utilisés. Ces modèles, bien qu'efficaces, génèrent parfois des anomalies ou des incohérences visuelles, notamment lorsque le processus de génération est fortement poussé pour maximiser les erreurs. Dans ces cas, l'objectif d'augmenter l'erreur peut prendre le pas sur la nécessité de conserver un réalisme visuel, ce qui limite l'exploitabilité des images. En perspective, l'amélioration de la qualité des modèles de diffusion, combinée à des stratégies d'entraînement adaptées, devrait permettre de produire des images générées à la fois plus réalistes et plus critiques. Ces avancées contribueront à renforcer la méthodologie d'analyse des performances et à affiner les garanties apportées du système *Field Monitoring*. Plus performants seront les modèles de génération, plus riche sera la diversité et la pertinence des données critiques générées, offrant ainsi un outil encore plus puissant pour l'évaluation et l'optimisation des systèmes intégrant des modules d'IA.

6.7 Conclusion et perspectives

Dans cette section, nous avons appliqué la méthodologie décrite au chapitre 4 à l'application *Field Monitoring*, un système composé de deux modules d'IA : un compresseur d'image et un détecteur d'objets.

Nous avons tout d'abord défini le système dont nous souhaitions garantir les performances. Pour cela, nous avons réalisé une acquisition de données en respectant l'ODD, combinant des images naturelles issues d'acquisitions par caméras et des données synthétiques simulées. Cette acquisition a suivi un ODD spécifique, intégrant des données open source et des données privées, afin de couvrir autant que possible les scénarios pertinents. Bien que ces données soient suffisantes pour une première illustration de la méthodologie, elles restent limitées, et des acquisitions supplémentaires seront nécessaires pour obtenir des estimations plus complètes et robustes. Ensuite, nous avons entraîné les différents modules d'IA composant le système, à savoir les modèles de compression STF et QRes-VAE, ainsi que les modèles de détection d'objets YOLOv7, Faster-RCNN et Retinanet. En complément, des modèles de diffusion, basés sur la méthodologie Stable Diffusion, ont été entraînés à partir de divers ensembles de données pour explorer les capacités de génération d'images. Nous avons ensuite analysé l'effet des différents modèles de compression sur les performances des modèles de détection. Cette analyse a montré que, selon le taux de compression, les distorsions induites par la compression peuvent fortement impacter les performances de détection. Parmi les modèles évalués, YOLOv7 s'est révélé être le plus robuste à ces distorsions. Nous avons appliqué la méthodologie d'estimation des probabilités d'erreur décrite à la section 6.4, en utilisant les deux étapes d'estimation intra-dataset et inter-dataset. Les résultats ont mis en évidence des variations importantes des probabilités d'erreur selon les datasets et les pondérations définies par l'application. La pondération $\{p_j\}$ permet une estimation spécifique des probabilités d'erreur, reflétant les exigences propres au *Field Monitoring*. Le système le plus performant, c'est-à-dire celui ayant la probabilité d'erreur globale la plus faible, est le modèle composé du compresseur STF (entraîné avec la métrique MS-SSIM) et YOLOv7. Avec des taux de compression inférieurs à 0.4 bpp, la probabilité d'erreur globale atteint un minimum d'environ 20%. Cependant, parmi ces 20%, la majorité des images contiennent une seule erreur, ce qui atténue l'impact perçu de cette probabilité élevée. En séparant les classes, nous avons constaté que les probabilités globales pour les classes piétons et cyclistes étaient particulièrement faibles, en raison de leur rareté dans les datasets. Toutefois, cette rareté masque une performance moins robuste du système sur ces classes, accompagnée d'une incertitude importante dans les estimations. Cela met en évidence la nécessité d'acquérir davantage de données pertinentes pour améliorer les performances et réduire les incertitudes d'estimation. Pour réduire les probabilités d'erreur estimées, nous avons exploré plusieurs stratégies, notamment le réentraînement du détecteur d'objets sur un plus grand nombre d'images, en particulier des images compressées. Ce réentraînement a montré des améliorations significatives dans la performance du système. Enfin, dans une étape complémentaire à la méthodologie, nous avons proposé d'utiliser des modèles génératifs, notamment des modèles de diffusion, pour explorer les cas critiques susceptibles de provoquer des erreurs importantes dans le système. L'approche repose sur une génération d'images guidée par le gradient, visant à produire des scénarios problématiques tout en restant réalistes. Cependant, cette étape est encore à un stade préliminaire et nécessite des recherches supplémentaires pour optimiser les hyperparamètres du processus de génération, tels que le nombre d'étapes de débruitage T, le pas d'apprentissage lr, ou encore le paramètre de régulation η. De plus, les modèles de diffusion utilisés actuellement ne produisent pas encore des images de qualité suffisante pour des analyses rigoureuses. Il est important de noter que les données synthétiques issues de simulateurs et celles générées par des modèles génératifs répondent à des problématiques différentes. Les données simulées permettent de couvrir des scénarios rares ou impossibles à obtenir via une acquisition classique (par exemple, des accidents ou des conditions météorologiques extrêmes). Elles comblent les lacunes de l'ODD et renforcent la robustesse du système sur des situations critiques. À l'inverse, les données générées par des modèles génératifs, produites selon la distribution des données d'apprentissage, ont pour objectif d'identifier et d'explorer des cas critiques spécifiques au système étudié. En ciblant les scénarios susceptibles de provoquer des erreurs ou de mettre en difficulté les modules d'IA, ces données synthétiques offrent une opportunité précieuse d'optimiser le système en renforçant sa robustesse et en améliorant ses performances face à des situations critiques.

Dans la suite, nous verrons comment généraliser cette méthodologie à d'autres systèmes complexes composés de modules d'IA, en adaptant les étapes d'acquisition de données, d'entraînement des modèles et d'estimation des garanties pour répondre aux exigences de différentes applications.

Chapitre 7

Vers une generalisation de la methodologie

Contents

7.1 Flexibilité et limites de la méthodologie 239
7.2 Recommandations pour l'adaptation 240
7.3 Perspectives 242
7.4 Conclusion 243

L'évaluation des performances des systèmes intégrant des modules d'intelligence artificielle (IA) constitue une étape cruciale pour leur déploiement en contexte opérationnel. Dans ce cadre, nous avons développé une méthodologie d'obtention de garanties statistiques de performance, illustrée par son application au système *Field Monitoring*, combinant un compresseur d'image et un détecteur d'objets. Cette méthodologie repose sur des étapes structurées et adaptables, visant à fournir des garanties probabilistes sur les erreurs potentielles et à identifier les vulnérabilités du système. La méthodologie s'articule autour de plusieurs étapes essentielles :

1. Elle débute par une définition rigoureuse du système étudié, incluant la formulation d'une métrique d'évaluation claire et interprétable, ainsi que la définition précise de son ODD.
2. Elle se poursuit par l'acquisition de données pertinentes pour recouvrir autant que possible cet ODD. Une attention particulière est accordée aux cas critiques, dits *Edge and Corner*, souvent négligés mais susceptibles d'exposer les limites opérationnelles du système.
3. Ces données sont ensuite utilisées pour estimer les probabilités d'erreur via des étapes d'estimations intra-dataset et inter-dataset, permettant une quantification des risques.
4. Enfin, des modèles génératifs orientés sont employés pour explorer des scénarios critiques, augmentant ainsi la capacité du système à identifier et à gérer des cas limites

Les résultats obtenus pour le système *Field Monitoring* démontrent l'efficacité de cette approche. La méthodologie a permis d'identifier les points faibles du système, de quantifier les probabilités d'erreur dans des conditions variées, et de proposer des stratégies d'amélioration ciblées. Par exemple, l'estimation des erreurs a montré que, pour des taux de compression inférieurs à 0,4 bpp, le taux global d'erreur atteint environ 20%, mais que la plupart des images contenant des erreurs présentaient des cas isolés. De plus, les analyses ont révélé que les classes minoritaires, comme les piétons et les cyclistes, souffrent d'une plus grande incertitude, nécessitant des données additionnelles et un renforcement ciblé du système.

L'objectif de ce chapitre est de montrer comment la méthodologie développée peut être généralisée à d'autres systèmes intégrant des modules d'IA dans des contextes variés. Cette généralisation repose sur l'adaptabilité de la méthodologie, qui peut être modulée pour répondre aux exigences spécifiques et aux contraintes opérationnelles propres à chaque système. En effet, bien que les principes fondamentaux – définition du système, acquisition de données, estimation probabiliste des erreurs, et exploration des scénarios critiques – soient universels, leur mise en œuvre doit être ajustée en fonction des caractéristiques du domaine d'application. Nous définissons les garanties de performance comme des estimations probabilistes des erreurs et des vulnérabilités potentielles des systèmes étudiés. Ces garanties s'appuient sur une analyse statistique rigoureuse et des hypothèses explicitement définies. Cependant, elles restent conditionnées par la qualité et la représentativité des datasets, ainsi que par la

validité des modèles statistiques employés. Les marges d'incertitude, reflétées par les intervalles de confiance associés aux estimations, soulignent la nature probabiliste des résultats. Dans ce contexte, l'objectif est d'identifier les vulnérabilités potentielles, de quantifier les risques d'erreur, et de proposer des stratégies d'amélioration basées sur des analyses probabilistes robustes. Cette méthodologie permet non seulement de garantir les performances d'un système dans son ODD, mais aussi de poser les bases pour des adaptations futures en cas de modification des exigences ou des conditions opérationnelles.

7.1 Flexibilité et limites de la méthodologie

La méthodologie proposée se distingue par sa modularité, offrant une grande flexibilité dans son application à divers systèmes et domaines. Cette modularité se manifeste par la facilité d'intégration de nouveaux types de données, de métriques ou de modèles, ce qui en fait un outil adaptable à des besoins variés. Par exemple, elle peut être appliquée aussi bien à des systèmes embarqués, comme les véhicules autonomes, qu'à des dispositifs médicaux ou des infrastructures industrielles. Les étapes clés de la méthodologie, définition du système et de son ODD, acquisition de données, évaluation statistique des erreurs, et exploration des cas critiques, peuvent être ajustées pour s'adapter aux spécificités des systèmes étudiés. Cette flexibilité permet d'appliquer la méthodologie à différents domaines où des garanties de performances sont critiques. La possibilité d'intégrer des simulateurs ou des modèles génératifs rend cette méthodologie particulièrement efficace pour combler les lacunes dans les ensembles de données, notamment dans des scénarios rares ou critiques.

Cependant, malgré sa flexibilité, la méthodologie repose fortement sur la qualité et la représentativité des ensembles de données utilisés. Si les données ne couvrent pas de manière adéquate l'ODD ou sont biaisées, les estimations probabilistes obtenues risquent de ne pas refléter les performances réelles du système. Cette dépendance met en lumière la nécessité d'une acquisition de données rigoureuse, incluant des scénarios *Edge and Corner*, afin d'assurer une évaluation fiable et exhaustive. Les résultats de la méthodologie sont basés sur des hypothèses statistiques explicites, telles que la représentativité des données ou l'indépendance des échantillons. Si ces hypothèses ne sont pas vérifiées dans un contexte donné, les garanties obtenues pourraient être invalides ou peu pertinentes. Par conséquent, il est essentiel de valider les hypothèses utilisées à chaque étape de la méthodologie et de prendre en compte les marges d'incertitude associées aux résultats. Certaines étapes de la méthodologie peuvent être très coûteuses en termes de calcul et de ressources. Par exemple, les techniques de bootstrap pour l'évaluation statistique et l'exploration des cas critiques à l'aide de modèles génératifs nécessitent des infrastructures matérielles performantes et des temps de calcul importants. Les modèles de diffusion, bien qu'efficaces pour générer des données critiques, illustrent cette contrainte avec leurs processus de génération itératifs qui peuvent être particulièrement longs. Cette complexité opérationnelle peut limiter l'applicabilité de la méthodologie dans des environnements où les ressources informatiques sont restreintes. De plus, les modèles génératifs offrent des outils puissants pour explorer les scénarios critiques, mais leurs résultats peuvent manquer d'interprétabilité. Par exemple, relier une modification dans l'espace latent à une vulnérabilité spécifique du système étudié reste difficile. Cette limite complique parfois l'identification précise des causes des erreurs ou des scénarios critiques générés. Une amélioration de l'interprétabilité pourrait renforcer l'utilité pratique de ces modèles dans l'évaluation des systèmes complexes. Enfin, bien que la méthodologie soit flexible, son application peut nécessiter des ajustements significatifs dans certains contextes spécifiques ou pour des systèmes ayant des exigences très différentes. Cela pourrait limiter la généralisation directe des résultats obtenus. Par exemple, pour des systèmes complexes à forte intégration modulaire, l'interaction entre les modules pourrait rendre difficile l'identification précise des vulnérabilités. Une approche plus personnalisée serait alors nécessaire pour adapter la méthodologie aux spécificités du système.

La méthodologie proposée constitue un cadre robuste et adaptable pour l'obtention de garanties de performances dans des systèmes intégrant des modules d'IA. Sa flexibilité lui permet de répondre à des exigences variées, et son approche modulaire facilite son extension à de nouveaux contextes. Toutefois, pour garantir la pertinence des résultats, il est indispensable de tenir compte des limites identifiées, telles que la dépendance à la qualité des données, les hypothèses statistiques, et les contraintes opérationnelles. Une compréhension approfondie du système étudié, une validation rigoureuse des hypothèses, et une planification adéquate des ressources sont essentielles pour maximiser l'efficacité de la méthodologie et répondre aux défis spécifiques de chaque application.

7.2 Recommandations pour l'adaptation

L'application de la méthodologie d'obtention de garanties de performances à différents systèmes nécessite des ajustements pour répondre aux spécificités de chaque contexte. Ces ajustements concernent principalement la définition du système étudié, l'acquisition des données, l'évaluation des performances, et l'utilisation de modèles génératifs. L'objectif est de conserver la rigueur et la validité des garanties obtenues tout en tenant compte des exigences et contraintes propres à chaque domaine.

Un premier aspect fondamental est de comprendre en détail le système cible et son domaine d'application. Chaque système intégrant des modules d'intelligence artificielle (IA) a des objectifs opérationnels spécifiques et évolue dans un environnement unique. Il est crucial de définir clairement ces éléments, notamment les objectifs de performance, les contraintes d'utilisation, et les scénarios critiques qui composent son domaine d'opération (ODD). Par exemple, dans un système médical, l'ODD incluera des paramètres tels que les caractéristiques démographiques des patients (âge, sexe, antécédents médicaux), les pathologies cibles (par exemple, cancers spécifiques, maladies cardiaques ou neurologiques), et les modalités d'imagerie utilisées (radiographies, IRM, scanners). En revanche, dans un système de surveillance industrielle, l'ODD concernera les types d'équipements surveillés (par exemple, turbines, pompes ou circuits électroniques), les anomalies détectables (usure, surchauffe, vibrations excessives), ainsi que les conditions environnementales dans lesquelles ces équipements fonctionnent, comme la température, l'humidité ou les niveaux de bruit ambiant. Chaque ODD définit ainsi les limites opérationnelles dans lesquelles le système est censé fonctionner de manière fiable et efficace, tout en guidant les étapes d'acquisition et de validation des données. L'acquisition de données pertinentes constitue une étape clé. Les données utilisées doivent couvrir de manière exhaustive les cas d'utilisation définis dans l'ODD. Cela inclut des données représentatives des conditions normales d'utilisation, mais aussi des scénarios critiques ou "*Edge and Corner*", souvent rares ou difficiles à collecter. Par exemple, dans le domaine médical, des cas impliquant des pathologies rares, des images médicales de faible qualité (par exemple, en raison d'artefacts ou de bruit), ou des variations anatomiques atypiques peuvent être essentiels. Lorsque des données réelles ne sont pas disponibles ou sont insuffisantes, des données synthétiques générées par des simulateurs ou des modèles génératifs peuvent être utilisées pour identifier et combler les lacunes. Contrairement au cadre applicatif Field Monitoring, dans lequel l'estimation des garanties n'a pas nécessité d'annotations, mais uniquement des images, il sera dans de nombreux cas indispensable de fournir des annotations exhaustives et pertinentes pour chaque donnée collectée, adaptées aux objectifs spécifiques du système étudié. Ces annotations doivent non seulement refléter les caractéristiques clés de chaque donnée (par exemple, les classes d'objets pour un détecteur ou les contours précis pour une segmentation), mais également permettre une interprétation claire et cohérente des performances du système dans son contexte opérationnel. Ces annotations doivent non seulement refléter les caractéristiques clés de chaque donnée (par exemple, les classes d'objets pour un détecteur, ou les contours précis pour une segmentation), mais aussi permettre une interprétation claire des performances du système dans son contexte opérationnel. Ces annotations sont cruciales à plusieurs égards. D'une part, elles permettent de constituer des ensembles de données d'entraînement de haute qualité, garantissant ainsi que les modèles d'intelligence artificielle apprennent de manière efficace et pertinente. D'autre part, elles sont indispensables pour évaluer les performances du système avec précision, en servant de référence pour mesurer les erreurs. Dans notre méthodologie, l'importance des annotations varie selon les étapes. Par exemple, dans l'analyse des erreurs causées par la compression, nous avons pu comparer directement les prédictions du système sur des images compressées et non compressées, sans dépendre fortement d'annotations externes. Cependant, pour d'autres applications ou contextes, comme la segmentation sémantique ou la classification médicale, des annotations détaillées deviennent incontournables. Ces annotations doivent être cohérentes, précises, et si possible validées par des experts du domaine pour garantir leur fiabilité. Enfin, il convient de noter que la collecte d'annotations peut être une tâche complexe et coûteuse, en particulier dans des contextes où l'expertise humaine est requise (par exemple, pour l'analyse d'images médicales ou l'identification de défauts industriels rares). Il sera donc essentiel d'optimiser cette étape, notamment en explorant des approches semi-supervisées ou basées sur l'apprentissage actif, où les annotations se concentrent sur les données les plus informatives pour le système. À cet égard, les données simulées offrent un avantage considérable : elles permettent de générer des annotations parfaites et gratuites, éliminant ainsi les coûts et les erreurs associées à l'annotation manuelle. En outre, ces données peuvent être créées pour couvrir des scénarios rares ou critiques qui sont difficilement observables dans des données réelles, renforçant ainsi la couverture de l'ODD et la robustesse des estimations de performance du système.

Après l'acquisition, il est essentiel d'organiser ces données en différents ensembles de données (datasets) regroupant des caractéristiques similaires. Cette organisation facilite leur exploitation ultérieure et permet une évaluation plus précise des performances du système dans des contextes variés. Les critères de regroupement

peuvent inclure des aspects tels que les types de scènes (urbaines, rurales, autoroutes), les environnements (intérieurs, extérieurs, sous-marins, aériens), les conditions météorologiques (ensoleillé, pluvieux, neigeux), ou encore des spécificités géographiques liées aux lieux d'acquisition (par exemple, villes, pays, ou continents). De plus, il peut être pertinent de segmenter les données en fonction des paramètres techniques de l'acquisition, comme les types de capteurs utilisés (caméras RGB, capteurs infrarouges, LIDAR), les résolutions des images, ou encore les configurations temporelles (données collectées de jour ou de nuit). Cette structuration méthodique permet non seulement de couvrir de manière ciblée l'ODD du système étudié, mais aussi de mieux cibler les estimations de garanties pour l'application visée. En effet, elle facilite l'étape d'estimation inter-dataset en permettant de pondérer précisément chaque dataset selon son importance relative dans le contexte opérationnel, via les coefficients $\{p_j\}$. Ces pondérations permettent d'adapter les probabilités d'erreur globales à l'application spécifique du système, garantissant ainsi une évaluation plus représentative et alignée avec les priorités du domaine étudié. Cela simplifie également l'identification des scénarios critiques ou des lacunes potentielles dans les données, tout en rendant les analyses comparatives plus cohérentes et interprétables.

Ensuite, l'évaluation des performances du système nécessite également des ajustements adaptés au domaine d'application, notamment dans le choix des métriques. Ces métriques doivent non seulement être adaptées aux objectifs spécifiques du système, mais également être interprétables pour faciliter l'analyse et les recommandations. De plus, elles doivent permettre une estimation rigoureuse de la probabilité que l'erreur dépasse un certain seuil critique, notée $P(\text{erreur}(x) \geq \epsilon)$, afin de fournir des garanties quantitatives sur les performances. Dans l'application *Field Monitoring*, nous avions défini une métrique d'erreur $e(x)$ adaptée au contexte de détection d'objets sous compression, en comptabilisant les erreurs comme suit : $e(x) = \#\text{FP} + \#\text{FN} + \#\text{MC}$. Cette métrique reflète directement les types d'erreurs les plus critiques dans ce cadre applicatif. Pour un système de classification, les métriques adaptées incluront la précision (*accuracy*), qui mesure la proportion d'exemples correctement classés, ainsi que des métriques plus détaillées comme la précision (*precision*), le rappel (*recall*) ou le F1-score. Pour un système de segmentation sémantique, des métriques comme l'IoU (*Intersection over Union*) ou la mIoU (*mean IoU*) sont couramment utilisées pour évaluer la qualité des segmentations. Ces métriques comparent les zones segmentées par le modèle à celles de référence pour mesurer le recouvrement Ou plus simplement, on peut juste considérer la quantité d'erreur, c'est-à-dire le nombre de pixels mal classifiés. Une autre métrique pertinente est le score de Dice, qui est particulièrement adapté pour des applications médicales où les frontières des régions segmentées sont cruciales, par exemple pour la détection de tumeurs ou de lésions. Pour un système de reconnaissance vocale, des métriques comme le WER (*Word Error Rate*) sont essentielles pour quantifier les erreurs de transcription. Le WER prend en compte les substitutions, les omissions et les insertions de mots dans la transcription, offrant ainsi une évaluation complète de la qualité de la sortie du modèle. L'objectif, dans chaque cas, est de sélectionner une métrique qui non seulement évalue précisément les performances du système, mais qui est également en lien direct avec les exigences opérationnelles et les risques spécifiques de l'application. Cette interprétabilité facilite l'identification des vulnérabilités et oriente efficacement les stratégies d'amélioration.

La méthodologie statistique employée doit être soigneusement adaptée aux caractéristiques spécifiques des données et du système étudié, afin de garantir la pertinence et la fiabilité des estimations obtenues. Cela implique de valider plusieurs hypothèses fondamentales sur lesquelles reposent les analyses. Par exemple, la représentativité des datasets est une hypothèse clé : les données utilisées pour l'évaluation doivent être suffisamment variées et couvrir de manière exhaustive l'espace des scénarios possibles, tel que défini par l'ODD. Si les données ne reflètent pas fidèlement les conditions réelles d'utilisation du système, les garanties obtenues pourraient ne pas être applicables dans des contextes opérationnels.Les marges d'incertitude associées aux estimations jouent également un rôle central. Elles sont généralement exprimées sous forme d'intervalles de confiance, qui fournissent une mesure quantitative de la précision et de la fiabilité des résultats. Ces intervalles permettent de contextualiser les probabilités d'erreur obtenues en indiquant une fourchette dans laquelle la véritable performance du système est susceptible de se situer avec une certaine probabilité. Mentionner explicitement ces marges est essentiel pour informer sur la nature probabiliste des garanties obtenues, car elles soulignent que les résultats ne sont pas des certitudes absolues, mais des estimations basées sur les données disponibles et les hypothèses faites. Enfin, l'intégration explicite de ces marges d'incertitude dans l'interprétation des résultats favorise une prise de décision plus éclairée. Par exemple, dans un contexte critique comme le médical ou l'automobile, une marge d'incertitude importante pourrait indiquer la nécessité de collecter davantage de données ou de revoir les hypothèses pour améliorer la robustesse des garanties. En résumé, valider les hypothèses statistiques, garantir la représentativité des données et interpréter les résultats en tenant compte des incertitudes sont des éléments indispensables pour

assurer la pertinence et la fiabilité des garanties de performance obtenues.

Dans le cadre des modèles génératifs, il est essentiel de sélectionner un modèle qui soit non seulement adapté au type de données à produire, mais également aux objectifs spécifiques du système étudié. Par exemple, pour générer des images, les modèles de diffusion ou les GANs sont fréquemment utilisés en raison de leur capacité à produire des données visuellement réalistes. En revanche, pour des données textuelles, des modèles autoregressifs comme GPT sont mieux adaptés, car ils excellent dans la génération séquentielle et contextuelle. Dans tous les cas, les données générées doivent être réalistes, c'est-à-dire indiscernables de données réelles pour le système ou les utilisateurs finaux, et représentatives des scénarios critiques identifiés, notamment les cas rares ou extrêmes qui pourraient révéler des vulnérabilités du système. Pour garantir la qualité des données produites, il est crucial de configurer correctement les hyperparamètres des modèles génératifs. Cela inclut le nombre d'étapes de génération, les paramètres de régularisation, ou encore les critères de convergence. Par exemple, dans les modèles de diffusion, un nombre excessif d'étapes de débruitage pourrait entraîner des dégradations qualitatives des données, tandis qu'un nombre insuffisant pourrait limiter leur réalisme. De même, dans les GANs, un équilibre subtil entre le générateur et le discriminateur est nécessaire pour éviter des phénomènes tels que le mode collapse (génération de données peu diversifiées). Dans le contexte de la génération orientée par le gradient, des considérations supplémentaires entrent en jeu. Ce type de génération vise à produire des données spécifiques qui maximisent une fonction de perte définie en fonction des objectifs du système. Pour que cette approche soit efficace, la fonction de perte utilisée pour orienter la génération doit être différentiable par rapport aux données d'entrée, afin de permettre le calcul du gradient. Par exemple, pour un détecteur d'objets, la fonction de perte pourrait inclure des termes liés à la classification, à la localisation des objets ou au niveau de confiance des prédictions. Si la fonction de perte n'est pas différentiable, il serait nécessaire d'utiliser des approximations ou des techniques alternatives, ce qui pourrait diminuer la précision de l'orientation. Il s'agira ensuite de sélectionner avec soin les hyperparamètres du processus de génération, car ils influencent directement la qualité et la pertinence des données produites. Parmi ces hyperparamètres, le pas de génération lr est crucial. Un pas trop grand risque d'entraîner des modifications excessives dans l'espace latent, générant des données irréalistes ou incohérentes, tandis qu'un pas trop petit pourrait ralentir considérablement le processus sans garantir une convergence vers des données pertinentes. Un autre paramètre essentiel est le poids η, qui pondère la régularisation sur la distance à laquelle on s'éloigne de la donnée d'origine. Cette régularisation est souvent définie à l'aide d'une norme, telle que la norme L_1, L_2, ou L_∞.

Enfin, l'intégration de cette méthodologie dans le cycle de vie opérationnel du système nécessite une planification rigoureuse. Il est important de s'assurer que les garanties obtenues sont régulièrement réévaluées à chaque mise à jour ou modification du système. Cela inclut également une documentation claire des hypothèses et des limites des analyses réalisées, afin que les garanties soient interprétées de manière adéquate par les parties prenantes. La méthodologie doit être intégrée dans un processus itératif permettant d'améliorer en continu les performances et la robustesse du système. Ces recommandations offrent un cadre structuré pour adapter la méthodologie à divers contextes tout en maintenant la rigueur et la pertinence des garanties de performances obtenues. En ajustant les étapes de la méthodologie en fonction des spécificités de chaque système, il devient possible d'évaluer de manière fiable les performances et de renforcer la confiance dans les modules d'IA utilisés dans des applications critiques.

7.3 Perspectives

La méthodologie proposée pour l'obtention de garanties de performances peut être enrichie par de nombreuses améliorations et extensions, qui visent à la rendre plus robuste, efficace et généralisable. Ces perspectives d'amélioration s'organisent autour de plusieurs axes clés.

L'un des premiers défis est de développer des méthodes systématiques pour évaluer si l'ODD est correctement couvert par les données disponibles. À l'heure actuelle, cette évaluation repose souvent sur des analyses qualitatives ou sur des hypothèses implicites. Des approches quantitatives, basées sur des métriques formelles ou des outils de clustering, pourraient être introduites pour déterminer objectivement le degré de couverture de l'ODD. Ces méthodes permettraient également d'identifier les lacunes ou les biais dans les ensembles de données, en mettant en lumière des scénarios sous-représentés ou manquants.

En ce qui concerne l'acquisition de nouvelles données, il est essentiel de mieux comprendre comment les intégrer efficacement dans les ensembles existants. Une question cruciale est de savoir si ces données doivent être

ajoutées à un dataset existant ou bien si elles nécessitent la création d'un nouveau dataset. Cette décision pourrait être guidée par des techniques comme le clustering ou l'analyse de distribution des données, permettant de préserver l'homogénéité des ensembles tout en améliorant leur diversité. De plus, des approches semi-supervisées ou adaptatives pourraient être explorées pour optimiser l'ajout des données.

La détermination des hyperparamètres p_j, représentant les pondérations des datasets dans l'estimation inter-datasets, demeure un défi majeur. Dans certains cas, comme le nôtre, il est difficile d'utiliser la taille des datasets comme un proxy fiable pour définir ces pondérations. En raison des divergences dans les méthodes d'acquisition, les datasets présentent souvent des disparités importantes, ce qui peut biaiser l'estimation de l'erreur. Actuellement, l'approche la plus courante repose sur l'utilisation d'informations expertes pour déterminer ces pondérations, en s'appuyant sur une analyse qualitative des données et des scénarios représentés dans chaque dataset. Toutefois, cette méthode, bien qu'efficace dans de nombreux cas, présente des limites, notamment en termes de subjectivité et de dépendance aux connaissances humaines. Identifier des alternatives permettrait de rendre la méthodologie nettement plus objective et conduirait à des estimations plus robustes.

Concernant la génération d'images critiques, plusieurs axes d'amélioration peuvent être explorés pour rendre ce processus plus performant. D'abord, il serait nécessaire de rendre les images générées plus réalistes, ce qui pourrait être réalisé en entraînant les modèles génératifs sur des ensembles d'images plus diversifiés et de meilleure qualité. L'intégration d'annotations ou de conditions dans l'apprentissage des modèles de diffusion permettrait également de générer des images de manière conditionnelle, augmentant ainsi leur pertinence pour des cas spécifiques. Enfin, le processus de génération lui-même pourrait être automatisé. Actuellement, il n'existe pas de convergence garantie lors de la génération, et les itérations peuvent produire des résultats sous-optimaux ou manquer des cas critiques. En introduisant un paramètre de régulation η, il est possible de limiter les déplacements dans l'espace latent, mais cela peut entraîner une perte d'images pertinentes. Une solution idéale serait de définir un critère de convergence basé sur l'atteinte d'un maximum local ou global de la fonction de perte, plutôt qu'une simple contrainte sur les déplacements.

Enfin, une perspective particulièrement ambitieuse consiste à étendre la méthodologie à des systèmes encore plus complexes. Par exemple, des systèmes multi-agents ou des systèmes intégrant plusieurs types d'IA collaboratives présentent des défis uniques en termes de garanties de performances. La méthodologie pourrait être adaptée pour prendre en compte les interactions entre les différents agents ou modules d'IA, ainsi que les dépendances complexes qui en résultent. Cela nécessiterait de développer des outils pour modéliser ces interactions et pour évaluer leur impact sur les performances globales du système.

Ces améliorations et extensions permettraient non seulement de renforcer la pertinence et l'efficacité de la méthodologie proposée, mais aussi de l'appliquer à des domaines et des systèmes toujours plus variés et exigeants. Elles constituent des étapes essentielles pour transformer cette méthodologie en un cadre universel pour l'évaluation et la garantie des performances des systèmes intégrant des modules d'IA.

7.4 Conclusion

La généralisation de la méthodologie d'obtention de garanties de performances est essentielle pour garantir la robustesse et la fiabilité des systèmes intégrant des modules d'intelligence artificielle (IA) dans des contextes variés. L'évaluation de tels systèmes, souvent déployés dans des environnements critiques, nécessite des approches capables de s'adapter à des domaines et des exigences opérationnelles multiples, tout en maintenant une rigueur scientifique dans l'analyse des performances. Cette généralisation est également clé pour démontrer que la méthodologie peut offrir une valeur ajoutée dans des secteurs industriels diversifiés. Pour favoriser l'adoption de cette méthodologie à l'échelle industrielle, il devient impératif de développer des standards ou des guides sectoriels. Ces outils fourniraient des cadres pratiques et normalisés permettant d'intégrer la méthodologie dans les cycles de développement et d'évaluation des systèmes, tout en répondant aux besoins spécifiques de chaque industrie. Ces standards pourraient inclure des recommandations sur les types de données à acquérir, les métriques adaptées à chaque domaine, et les protocoles d'estimation probabiliste des performances. Dans le cadre de notre application spécifique au système *Field Monitoring*, nous avons pu bénéficier d'un avantage notable : l'absence d'un besoin critique d'annotations pour l'évaluation des performances. Étant donné que l'approche repose sur une comparaison entre les détections effectuées sur des images non compressées et celles obtenues sur des images compressées, il n'est pas nécessaire de connaître la vérité terrain pour procéder à l'évaluation des erreurs. Les annotations sont uniquement requises lors des phases d'entraînement des modèles, ce qui simplifie considérablement le processus dans notre cas. Cependant, cette situation ne s'applique pas à toutes les applications. Dans de nombreux domaines, des annotations précises et exhaustives seront indispensables, notamment

lorsqu'il s'agit de systèmes opérant dans des environnements complexes ou avec des données fortement hétérogènes. Par exemple, dans les applications médicales, les annotations doivent être effectuées par des experts qualifiés, ce qui les rend coûteuses et chronophages. Dans d'autres contextes, comme la surveillance industrielle ou les systèmes multi-agents, les annotations doivent souvent inclure des informations sur les relations ou interactions entre les entités présentes dans les données. La quantité et la qualité des annotations nécessaires pourraient ainsi représenter un défi conséquent, soulignant la nécessité de développer des outils automatisés ou semi-automatisés pour faciliter cette tâche. En conclusion, bien que la méthodologie d'obtention de garanties de performances ait démontré sa pertinence dans le cadre de notre application, sa généralisation requiert des adaptations spécifiques pour répondre aux contraintes de domaines variés. La mise en place de standards sectoriels, ainsi que des solutions pour gérer efficacement les besoins en annotations, seront des éléments déterminants pour garantir une adoption réussie et une applicabilité étendue de cette méthodologie.

Chapitre 8

Conclusion et perspectives

Au cours de cette thèse, nous avons abordé l'un des défis majeurs de l'intégration des systèmes d'intelligence artificielle (IA) dans des applications critiques : l'obtention de garanties de performances. Les systèmes intégrant des modules d'IA, bien qu'efficaces dans de nombreuses tâches, sont souvent difficiles à évaluer et à valider de manière rigoureuse, en raison de leur complexité, de leur sensibilité aux variations des données et de leur comportement parfois imprévisible. Ces limitations constituent un frein à leur adoption dans des domaines exigeant des niveaux élevés de fiabilité, comme l'automobile, la santé ou la finance. Pour répondre à ce défi, nous avons proposé une méthodologie complète et modulable d'obtention de garanties de performances statistiques pour les systèmes d'IA. Développée et testée dans le cadre du système *Field Monitoring* appliqué au domaine des véhicules autonomes, cette méthodologie repose sur plusieurs étapes clés : la définition et l'exploration de l'ODD (*Operational Design Domain*), l'acquisition de données pertinentes, l'estimation probabiliste des erreurs, l'analyse des scénarios critiques, et enfin, la proposition de stratégies d'amélioration des performances. Ces étapes permettent une évaluation exhaustive et une compréhension approfondie des limites et des vulnérabilités des systèmes étudiés.

Étant donné les limitations actuelles de l'état de l'art dans le domaine des garanties pour les systèmes d'IA, cette thèse propose une avancée significative en formalisant une méthodologie novatrice. La première contribution majeure réside dans le développement d'une approche méthodologique qui intègre harmonieusement des outils statistiques rigoureux, des techniques avancées d'apprentissage profond, et des modèles génératifs. Cette méthodologie se distingue par sa flexibilité, permettant son adaptation à des systèmes variés intégrant des modules d'IA, et par sa modularité, qui facilite l'incorporation de nouvelles métriques, types de données ou modèles selon les besoins spécifiques. Elle constitue ainsi une réponse pertinente aux défis de validation et d'évaluation des systèmes d'IA dans des contextes applicatifs diversifiés.

L'application concrète de cette méthodologie au système *Field Monitoring* a permis d'illustrer sa pertinence et son efficacité. En combinant un compresseur d'images et un détecteur d'objets, nous avons démontré que la méthodologie permet non seulement d'identifier les points faibles du système, mais aussi de proposer des améliorations ciblées. Par exemple, nos résultats montrent que, malgré de bonnes performances globales, certaines classes d'objets (comme les cyclistes et les piétons) sont particulièrement sensibles à la compression, ce qui a conduit à des recommandations pour améliorer la robustesse des modules concernés. Si la fréquence d'échantillonnage des vidéos dans le cadre de l'application *Field Monitoring* le permet, il serait pertinent, à l'avenir, d'intégrer la dimension temporelle en travaillant sur des vidéos plutôt que sur des images individuelles. Cette transition offrirait de nouvelles opportunités en matière de compression : les vidéos permettent de mieux exploiter les redondances spatiales et temporelles, ce qui pourrait conduire à des compromis plus optimaux entre taux de compression et distorsion par rapport aux compresseurs dédiés uniquement aux images. Cependant, cette approche pose également des défis supplémentaires. Les vidéos sont des données plus complexes à acquérir, et l'espace des vidéos, combinant plusieurs dimensions temporelles et spatiales, est considérablement plus vaste que celui des images fixes. En conséquence, la définition et la couverture de l'ODD deviennent encore plus exigeantes, car il faut inclure des scénarios et des dynamiques temporelles variées pour refléter fidèlement les situations critiques.

Une autre contribution importante réside dans l'utilisation de modèles génératifs, comme les modèles de diffusion, pour explorer les scénarios critiques et générer des données synthétiques pertinentes. Ces modèles permettent de simuler des cas rares ou extrêmes qui ne sont pas forcément représentés dans les ensembles de données réels, renforçant ainsi la couverture de l'ODD et l'efficacité des tests. Nous avons également enrichi l'analyse en proposant une estimation probabiliste des erreurs basée sur des approches intra-dataset et inter-

dataset. Cette double approche permet une évaluation à la fois locale (au sein de chaque dataset) et globale (à l'échelle du système), tout en tenant compte des spécificités des scénarios d'application.

Malgré les avancées réalisées, cette thèse a également mis en évidence plusieurs défis qui méritent d'être explorés dans des travaux futurs. L'un des principaux défis est la dépendance de la méthodologie à la qualité et à la représentativité des données. Une couverture insuffisante de l'ODD ou des données biaisées peut limiter la pertinence des garanties obtenues. Par ailleurs, certaines étapes, comme l'utilisation de modèles génératifs, restent coûteuses en termes de calculs et nécessitent des infrastructures matérielles adaptées. Un autre défi est lié à l'interprétabilité des modèles génératifs et à leur capacité à produire des données synthétiques suffisamment réalistes pour être pertinentes. Par exemple, nos expériences ont montré que les modèles de diffusion, bien qu'efficaces, génèrent parfois des artefacts ou des anomalies visuelles. Il sera crucial de poursuivre les efforts pour améliorer la qualité des données générées et leur exploitabilité. Ensuite, des travaux supplémentaires pourraient porter sur l'automatisation de certaines étapes de la méthodologie, comme la génération de données synthétiques ou l'ajustement des hyperparamètres p_j.

En conclusion, cette thèse propose une contribution significative à la validation et à l'évaluation des systèmes intégrant des modules d'IA. En combinant des approches statistiques rigoureuses, des outils génératifs avancés et une analyse ciblée des scénarios critiques, la méthodologie développée offre une solution robuste et adaptable pour garantir les performances des systèmes d'IA dans des contextes variés. Elle constitue une avancée importante vers une intégration plus sûre et plus fiable de ces systèmes dans des applications industrielles et sociétales. Cependant, le chemin vers une adoption généralisée de cette méthodologie reste encore long. Les travaux futurs devront relever les défis identifiés, affiner les outils développés, et adapter la méthodologie à des cas d'usage encore plus diversifiés. Ce faisant, ils contribueront à faire de l'intelligence artificielle un outil de confiance, capable de répondre aux exigences des environnements critiques où la robustesse et la fiabilité sont des impératifs absolus.

Bibliographie

[Kod,]

[CLI,] Challenge on learned image compression (clic). https://www.compression.cc/.

[Fli,] Flickr2k dataset. https://github.com/LimBee/NTIRE2017?tab=readme-ov-file.

[eta,] GitHub - eth-sri/eran : ETH Robustness Analyzer for Deep Neural Networks. https://github.com/eth-sri/eran/?tab=readme-ov-file.

[yol,] super-gradients/YOLONAS.md at master · Deci-AI/super-gradients.

[VP8, 2008] (2008). Vp8.

[web, 2010] (2010). Webp : Un format d'image pour le Web.

[bpg, 2014] (2014). BPG Image format.

[aut, 2020] (2020). An Analysis of ISO 26262 : Machine Learning and Safety in Automotive Software. In *Safety of the Intended Functionality*, pages 13–25. SAE. Conference Name : Safety of the Intended Functionality.

[sae, 2021] (2021). J3016_202104 : Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International. https://www.sae.org/standards/content/j3016_202104/.

[ODD, 2021] (2021). ODD : Définition & Canevas de description. https://www.irt-systemx.fr/projets/sam/.

[iso, 2022] (2022). ISO 21448 : Road vehicles - safety of the intended functionality. https://www.iso.org/fr/standard/77490.html.

[iso, 2023] (2023). ISO 34503 : Road vehicles - test scenarios for automated driving systems - specification for operational design domain. https://www.iso.org/fr/standard/78952.html.

[Adedjouma et al., 2024] Adedjouma, M., Botella, B., Ibanez-Guzman, J., Mantissa, K., Proum, C.-M., and Smaoui, A. (2024). Defining operational design domain for autonomous systems : A domain-agnostic and risk-based approach. In *SOSE 2024 - 19th Annual System of Systems Engineering Conference*, Tacoma, United States.

[Agustsson and Timofte, 2017] Agustsson, E. and Timofte, R. (2017). Ntire 2017 challenge on single image super-resolution : Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135.

[Akutsu and Naruko, 2020] Akutsu, H. and Naruko, T. (2020). End-to-End Deep ROI Image Compression. *IEICE Transactions on Information and Systems*, E103.D(5) :1031–1038.

[Alfarra et al., 2022a] Alfarra, M., Bibi, A., Khan, N., Torr, P. H. S., and Ghanem, B. (2022a). DeformRS : Certifying Input Deformations with Randomized Smoothing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(6) :6001–6009. Number : 6.

[Alfarra et al., 2022b] Alfarra, M., Bibi, A., Torr, P. H. S., and Ghanem, B. (2022b). Data dependent randomized smoothing. In Cussens, J. and Zhang, K., editors, *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*, volume 180 of *Proceedings of Machine Learning Research*, pages 64–74. PMLR.

[Alquier, 2024] Alquier, P. (2024). User-friendly Introduction to PAC-Bayes Bounds. *MAL*, 17(2) :174–303. Publisher : Now Publishers, Inc.

[Alvey et al., 2021] Alvey, B., Anderson, D. T., Buck, A., Deardorff, M., Scott, G., and Keller, J. M. (2021). Simulated Photorealistic Deep Learning Framework and Workflows to Accelerate Computer Vision and Unmanned Aerial Vehicle Research. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3882–3891, Montreal, BC, Canada. IEEE.

[Andeol et al., 2023] Andeol, L., Fel, T., Grancey, F. d., and Mossina, L. (2023). Confident Object Detection via Conformal Prediction and Conformal Risk Control : an Application to Railway Signaling. In *Proceedings of the Twelfth Symposium on Conformal and Probabilistic Prediction with Applications*, pages 36–55. PMLR. ISSN : 2640-3498.

[Anderson et al., 2022] Anderson, B. G., Pfrommer, S., and Sojoudi, S. (2022). Towards Optimal Randomized Smoothing : A Semi-Infinite Linear Programming Approach. In *The Thirty-Ninth International Conference on Machine Learning (ICML 2022)*.

[Anderson and Sojoudi, 2022] Anderson, B. G. and Sojoudi, S. (2022). Certified Robustness via Locally Biased Randomized Smoothing. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, pages 207–220. PMLR. ISSN : 2640-3498.

[Anderson et al., 2020] Anderson, R., Huchette, J., Ma, W., Tjandraatmadja, C., and Vielma, J. P. (2020). Strong mixed-integer programming formulations for trained neural networks. *Math. Program.*, 183(1) :3–39.

[Angelopoulos and Bates, 2023] Angelopoulos, A. N. and Bates, S. (2023). Conformal prediction : A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4) :494–591.

[Angelopoulos et al., 2021] Angelopoulos, A. N., Bates, S., Candès, E. J., Jordan, M. I., and Lei, L. (2021). Learn then test : Calibrating predictive algorithms to achieve risk control. *arXiv preprint arXiv :2110.01052*.

[Angelopoulos et al., 2024] Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., and Schuster, T. (2024). Conformal risk control. In *The Twelfth International Conference on Learning Representations*.

[Aqqa et al., 2024] Aqqa, M., Mantini, P., and Shah, S. K. (2024). Understanding How Video Quality Affects Object Detection Algorithms. pages 96–104.

[Asuni and Giachetti, 2014] Asuni, N. and Giachetti, A. (2014). Testimages : a large-scale archive for testing visual devices and basic image processing algorithms. In *Smart Tools and Applications in Graphics*.

[Avrahami et al., 2022] Avrahami, O., Lischinski, D., and Fried, O. (2022). Blended Diffusion for Text-driven Editing of Natural Images. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18187–18197. ISSN : 2575-7075.

[Balestriero et al., 2021] Balestriero, R., Pesenti, J., and LeCun, Y. (2021). Learning in High Dimension Always Amounts to Extrapolation. Number : arXiv :2110.09485 arXiv :2110.09485 [cs].

[Ballé et al., 2016] Ballé, J., Laparra, V., and Simoncelli, E. P. (2016). End-to-end optimization of nonlinear transform codes for perceptual quality. In *2016 Picture Coding Symposium (PCS)*, pages 1–5. IEEE.

[Ballé et al., 2017] Ballé, J., Laparra, V., and Simoncelli, E. P. (2017). End-to-end optimized image compression. In *International Conference on Learning Representations*.

[Ballé et al., 2018] Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. (2018). Variational image compression with a scale hyperprior. In *International Conference on Learning Representations*.

[Balunovic and Vechev, 2020] Balunovic, M. and Vechev, M. (2020). Adversarial training and provable defenses : Bridging the gap. In *International Conference on Learning Representations*.

[Bansal et al., 2024] Bansal, A., Borgnia, E., Chu, H.-M., Li, J., Kazemi, H., Huang, F., Goldblum, M., Geiping, J., and Goldstein, T. (2024). Cold diffusion : Inverting arbitrary image transforms without noise. *Advances in Neural Information Processing Systems*, 36.

[Bao et al., 2021] Bao, F., Li, C., Zhu, J., and Zhang, B. (2021). Analytic-DPM : an Analytic Estimate of the Optimal Reverse Variance in Diffusion Probabilistic Models.

[Bao et al., 2023] Bao, F., Nie, S., Xue, K., Cao, Y., Li, C., Su, H., and Zhu, J. (2023). All are Worth Words : A ViT Backbone for Diffusion Models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22669–22679. ISSN : 2575-7075.

[Bao et al., 2024] Bao, Q., Wei, K., Xu, J., and Jiang, W. (2024). Application of Deep Learning in Financial Credit Card Fraud Detection. *Journal of Economic Theory and Business Management*, 1(2) :51–57.

[Bao et al., 2020] Bao, W., Yu, Q., and Kong, Y. (2020). Uncertainty-based traffic accident anticipation with spatio-temporal relational learning.

[Bartlett and Mendelson, 2003] Bartlett, P. L. and Mendelson, S. (2003). Rademacher and gaussian complexities : Risk bounds and structural results. *J. Mach. Learn. Res.*, 3 :463–482.

[Bégaint et al., 2020] Bégaint, J., Racapé, F., Feltman, S., and Pushparaja, A. (2020). Compressai : a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv :2011.03029*.

[Behrendt, 2019] Behrendt, K. (2019). Boxy vehicle detection in large images. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, volume 4, pages 840–846. IEEE.

[Bińkowski et al., 2018] Bińkowski, M., Sutherland, D. J., Arbel, M., and Gretton, A. (2018). Demystifying MMD GANs. In *International Conference on Learning Representations*.

[Blattmann et al., 2023] Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S. W., Fidler, S., and Kreis, K. (2023). Align Your Latents : High-Resolution Video Synthesis with Latent Diffusion Models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22563–22575. ISSN : 2575-7075.

[Bochkovskiy et al., 2020] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4 : Optimal speed and accuracy of object detection. *arXiv preprint arXiv :2004.10934*.

[Bolya et al., 2020] Bolya, D., Foley, S., Hays, J., and Hoffman, J. (2020). TIDE : A General Toolbox for Identifying Object Detection Errors. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pages 558–573, Cham. Springer International Publishing.

[Boniol et al., 2020] Boniol, F., Chan-Hon-Tong, A., Eudes, A., Herbin, S., Le Besnerais, G., Pagetti, C., and Sanfourche, M. (2020). Challenges in the Certification of Computer Vision-Based Systems. *Aerospace Lab*. Publisher : Alain Appriou.

[Boopathy et al., 2021] Boopathy, A., Weng, L., Liu, S., Chen, P.-Y., Zhang, G., and Daniel, L. (2021). Fast Training of Provably Robust Neural Networks by SingleProp. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6803–6811. ISSN : 2374-3468, 2159-5399 Issue : 8 Journal Abbreviation : AAAI.

[Boopathy et al., 2019] Boopathy, A., Weng, T.-W., Chen, P.-Y., Liu, S., and Daniel, L. (2019). CNN-Cert : An Efficient Framework for Certifying Robustness of Convolutional Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01) :3240–3247. Number : 01.

[Borkman et al., 2021] Borkman, S., Crespi, A., Dhakad, S., Ganguly, S., Hogins, J., Jhang, Y.-C., Kamalzadeh, M., Li, B., Leal, S., Parisi, P., Romero, C., Smith, W., Thaman, A., Warren, S., and Yadav, N. (2021). Unity Perception : Generate Synthetic Data for Computer Vision. arXiv :2107.04259 [cs].

[Botoeva et al., 2020] Botoeva, E., Kouvaros, P., Kronqvist, J., Lomuscio, A., and Misener, R. (2020). Efficient Verification of ReLU-Based Neural Networks via Dependency Analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04) :3291–3299. Number : 04.

[Brooks et al., 2023] Brooks, T., Holynski, A., and Efros, A. A. (2023). InstructPix2Pix : Learning to Follow Image Editing Instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18392–18402. ISSN : 2575-7075.

[Brown et al., 2020] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

[Brummer and De Vleeschouwer, 2021] Brummer, B. and De Vleeschouwer, C. (2021). End-to-end optimized image compression with competition of prior distributions. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1890–1894. ISSN : 2160-7516.

[Bunel et al., 2020a] Bunel, R., Palma, A. D., Desmaison, A., Dvijotham, K., Kohli, P., Torr, P., and Kumar, M. P. (2020a). Lagrangian Decomposition for Neural Network Verification. In *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 370–379. PMLR. ISSN : 2640-3498.

[Bunel et al., 2020b] Bunel, R., Turkaslan, I., Torr, P. H. S., Kumar, M. P., Lu, J., and Kohli, P. (2020b). Branch and bound for piecewise linear neural network verification. *J. Mach. Learn. Res.*, 21(1) :42 :1574–42 :1612.

[Bunel et al., 2018] Bunel, R. R., Turkaslan, I., Torr, P., Kohli, P., and Mudigonda, P. K. (2018). A Unified View of Piecewise Linear Neural Network Verification. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[Caesar et al., 2020] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. (2020). nuScenes : A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[Cai et al., 2019] Cai, C., Chen, L., Zhang, X., and Gao, Z. (2019). Efficient variable rate image compression with multi-scale decomposition network. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(12) :3687–3700.

[Cai et al., 2022a] Cai, S., Zhang, Z., Chen, L., Yan, L., Zhong, S., and Zou, X. (2022a). High-Fidelity Variable-Rate Image Compression via Invertible Activation Transformation. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, pages 2021–2031, New York, NY, USA. Association for Computing Machinery.

[Cai et al., 2022b] Cai, Z., Kwon, G., Ravichandran, A., Bas, E., Tu, Z., Bhotika, R., and Soatto, S. (2022b). X-detr : A versatile architecture for instance-wise vision-language tasks. In *European Conference on Computer Vision*, pages 290–308. Springer.

[Cai and Vasconcelos, 2018] Cai, Z. and Vasconcelos, N. (2018). Cascade R-CNN : Delving Into High Quality Object Detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6154–6162. ISSN : 2575-7075.

[Cao et al., 2023] Cao, J., Yao, X., Zhang, H., Jin, J., Zhang, Y., and Ling, B. W.-K. (2023). Slimmable Multi-Task Image Compression for Human and Machine Vision. *IEEE Access*, 11 :29946–29958. Conference Name : IEEE Access.

[Careil et al., 2024] Careil, M., Muckley, M. J., Verbeek, J., and Lathuilière, S. (2024). Towards image compression with perfect realism at ultra-low bitrates. In *The Twelfth International Conference on Learning Representations*.

[Carion et al., 2020] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer.

[Carlini et al., 2023] Carlini, N., Tramer, F., Dvijotham, K. D., Rice, L., Sun, M., and Kolter, J. Z. (2023). (certified ! !) adversarial robustness for free ! In *The Eleventh International Conference on Learning Representations*.

[Catoni, 2003] Catoni, O. (2003). A pac-bayesian approach to adaptive classification. *preprint*, 840(2) :6.

[Che et al., 2019] Che, Z., Li, G., Li, T., Jiang, B., Shi, X., Zhang, X., Lu, Y., Wu, G., Liu, Y., and Ye, J. (2019). D^2-city : A large-scale dashcam video dataset of diverse traffic scenarios.

[Chen et al., 2023a] Chen, Q., Chen, X., Wang, J., Zhang, S., Yao, K., Feng, H., Han, J., Ding, E., Zeng, G., and Wang, J. (2023a). Group detr : Fast detr training with group-wise one-to-many assignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6633–6642.

[Chen et al., 2021] Chen, T., Liu, H., Ma, Z., Shen, Q., Cao, X., and Wang, Y. (2021). End-to-End Learnt Image Compression via Non-Local Attention Optimization and Improved Context Modeling. *IEEE Transactions on Image Processing*, 30 :3179–3191. Conference Name : IEEE Transactions on Image Processing.

[Chen and Kloul, 2018] Chen, W. and Kloul, L. (2018). An Ontology-based Approach to Generate the Advanced Driver Assistance Use Cases of Highway Traffic. In *10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management.*, Seville, Spain.

[Chen et al., 2016] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. (2016). InfoGAN : Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

[Chen et al., 2023b] Chen, X., Gao, X., Zhao, J., Ye, K., and Xu, C.-Z. (2023b). Advdiffuser : Natural adversarial example synthesis with diffusion models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4539–4549.

[Chen et al., 2023c] Chen, Y.-H., Weng, Y.-C., Kao, C.-H., Chien, C., Chiu, W.-C., and Peng, W.-H. (2023c). TransTIC : Transferring Transformer-based Image Compression from Human Perception to Machine Perception. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23240–23250. ISSN : 2380-7504.

[Cheng et al., 2018] Cheng, Z., Sun, H., Takeuchi, M., and Katto, J. (2018). Deep convolutional autoencoder-based lossy image compression. In *2018 Picture Coding Symposium (PCS)*, pages 253–257.

[Cheng et al., 2020a] Cheng, Z., Sun, H., Takeuchi, M., and Katto, J. (2020a). Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7939–7948.

[Cheng et al., 2020b] Cheng, Z., Sun, H., Takeuchi, M., and Katto, J. (2020b). Learned Image Compression With Discretized Gaussian Mixture Likelihoods and Attention Modules. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7936–7945. ISSN : 2575-7075.

[Chiang et al., 2020] Chiang, P.-y., Curry, M., Abdelkader, A., Kumar, A., Dickerson, J., and Goldstein, T. (2020). Detection as Regression : Certified Object Detection with Median Smoothing. In *Advances in Neural Information Processing Systems*, volume 33, pages 1275–1286. Curran Associates, Inc.

[Choi et al., 2021] Choi, J., Kim, S., Jeong, Y., Gwon, Y., and Yoon, S. (2021). ILVR : Conditioning Method for Denoising Diffusion Probabilistic Models. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14347–14356. ISSN : 2380-7504.

[Choi et al., 2019] Choi, Y., El-Khamy, M., and Lee, J. (2019). Variable Rate Deep Image Compression With a Conditional Autoencoder. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3146–3154. ISSN : 2380-7504.

[Chugg et al., 2023] Chugg, B., Wang, H., and Ramdas, A. (2023). A Unified Recipe for Deriving (Time-Uniform) PAC-Bayes Bounds. *Journal of Machine Learning Research*, 24(372) :1–61.

[Chérief-Abdellatif et al., 2022] Chérief-Abdellatif, B.-E., Shi, Y., Doucet, A., and Guedj, B. (2022). On PAC-Bayesian reconstruction guarantees for VAEs. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, pages 3066–3079. PMLR. ISSN : 2640-3498.

[Cohen et al., 2019] Cohen, J., Rosenfeld, E., and Kolter, Z. (2019). Certified Adversarial Robustness via Randomized Smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1310–1320. PMLR. ISSN : 2640-3498.

[Cohen et al., 2024] Cohen, K. M., Park, S., Simeone, O., and Shitz, S. S. (2024). Cross-validation conformal risk control. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 250–255.

[Cui et al., 2021] Cui, Z., Wang, J., Gao, S., Guo, T., Feng, Y., and Bai, B. (2021). Asymmetric Gained Deep Image Compression With Continuous Rate Adaptation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10527–10536. ISSN : 2575-7075.

[Dai et al., 2024a] Dai, C., Song, T., Jin, Y., Ren, Y., Yang, B., and Song, G. (2024a). Masked Feature Compression for Object Detection. *Mathematics*, 12(12) :1848. Number : 12 Publisher : Multidisciplinary Digital Publishing Institute.

[Dai et al., 2016] Dai, J., Li, Y., He, K., and Sun, J. (2016). R-FCN : Object Detection via Region-based Fully Convolutional Networks. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

[Dai et al., 2022] Dai, L., Liu, H., Tang, H., Wu, Z., and Song, P. (2022). Ao2-detr : Arbitrary-oriented object detection transformer. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(5) :2342–2356.

[Dai et al., 2024b] Dai, X., Liang, K., and Xiao, B. (2024b). AdvDiff : Generating Unrestricted Adversarial Examples Using Diffusion Models. In Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., and Varol, G., editors, *Computer Vision – ECCV 2024*, pages 93–109, Cham. Springer Nature Switzerland.

[Dai et al., 2021] Dai, Z., Cai, B., Lin, Y., and Chen, J. (2021). Up-detr : Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1601–1610.

[Daniel and Tamar, 2021] Daniel, T. and Tamar, A. (2021). Soft-IntroVAE : Analyzing and Improving the Introspective Variational Autoencoder. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4389–4398. ISSN : 2575-7075.

[Dathathri et al., 2020] Dathathri, S., Dvijotham, K., Kurakin, A., Raghunathan, A., Uesato, J., Bunel, R. R., Shankar, S., Steinhardt, J., Goodfellow, I., Liang, P. S., and Kohli, P. (2020). Enabling certification of verification-agnostic networks via memory-efficient semidefinite programming. In *Advances in Neural Information Processing Systems*, volume 33, pages 5318–5331. Curran Associates, Inc.

[De Palma et al., 2022] De Palma, A., Bunel, R., Dvijotham, K., Kumar, M. P., and Stanforth, R. (2022). IBP regularization for verified adversarial robustness via branch-and-bound. abs/2206.14772.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet : A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[Denton et al., 2015] Denton, E. L., Chintala, S., szlam, a., and Fergus, R. (2015). Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

[Ding et al., 2023] Ding, T., Angelopoulos, A., Bates, S., Jordan, M., and Tibshirani, R. J. (2023). Class-conditional conformal prediction with many classes. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 64555–64576. Curran Associates, Inc.

[Donahue et al., 2018] Donahue, C., McAuley, J., and Puckette, M. (2018). Adversarial audio synthesis. In *International Conference on Learning Representations*.

[Du et al., 2024] Du, R., Chang, D., Hospedales, T., Song, Y.-Z., and Ma, Z. (2024). DemoFusion : Democratising High-Resolution Image Generation With No $$$. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6159–6168. ISSN : 2575-7075.

[Duan et al., 2023] Duan, Z., Lu, M., Ma, Z., and Zhu, F. (2023). Lossy image compression with quantized hierarchical VAEs. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 198–207. ISSN : 2642-9381.

[Dutta et al., 2018] Dutta, S., Jha, S., Sankaranarayanan, S., and Tiwari, A. (2018). Output Range Analysis for Deep Feedforward Neural Networks. In Dutle, A., Muñoz, C., and Narkawicz, A., editors, *NASA Formal Methods*, pages 121–138, Cham. Springer International Publishing.

[Dvijotham et al., 2018] Dvijotham, K., Stanforth, R., Gowal, S., Mann, T. A., and Kohli, P. (2018). A dual approach to scalable verification of deep networks. In *Conference on Uncertainty in Artificial Intelligence*.

[Dvijotham et al., 2020] Dvijotham, K. D., Hayes, J., Balle, B., Kolter, Z., Qin, C., Gyorgy, A., Xiao, K., Gowal, S., and Kohli, P. (2020). A framework for robustness certification of smoothed classifiers using f-divergences. In *International Conference on Learning Representations*.

[Ebadi et al., 2022] Ebadi, S. E., Dhakad, S., Vishwakarma, S., Wang, C., Jhang, Y.-C., Chociej, M., Crespi, A., Thaman, A., and Ganguly, S. (2022). PSP-HDRI$+$: A synthetic dataset generator for pre-training of human-centric computer vision models. In *First Workshop on Pre-training : Perspectives, Pitfalls, and Paths Forward at ICML 2022*.

[Efron, 1979] Efron, B. (1979). Bootstrap methods : Another look at the jackknife. 7(1) :1–26. Publisher : Institute of Mathematical Statistics.

[Ehlers, 2017] Ehlers, R. (2017). Formal Verification of Piece-Wise Linear Feed-Forward Neural Networks. In D'Souza, D. and Narayan Kumar, K., editors, *Automated Technology for Verification and Analysis*, pages 269–286, Cham. Springer International Publishing.

[Ehrlich, 2022] Ehrlich, M. (2022). *The First Principles of Deep Learning and Compression*. PhD thesis. pp. 93-99.

[Eiras et al., 2021] Eiras, F., Alfarra, M., Kumar, M. P., Torr, P. H. S., Dokania, P., Ghanem, B., and Bibi, A. (2021). ANCER : Anisotropic Certification via Sample-wise Volume Maximization. *Trans. Mach. Learn. Res.*

[Elboher et al., 2020] Elboher, Y. Y., Gottschlich, J., and Katz, G. (2020). An Abstraction-Based Framework for Neural Network Verification. In Lahiri, S. K. and Wang, C., editors, *Computer Aided Verification*, Lecture Notes in Computer Science, pages 43–65, Cham. Springer International Publishing.

[Erfanian Ebadi et al., 2022] Erfanian Ebadi, S., Jhang, Y.-C., Zook, A., Dhakad, S., Crespi, A., Parisi, P., Borkman, S., Hogins, J., and Ganguly, S. (2022). PeopleSansPeople : A Synthetic Data Generator for Human-Centric Computer Vision. arXiv :2112.09290 [cs].

[Esser et al., 2021] Esser, P., Rombach, R., and Ommer, B. (2021). Taming Transformers for High-Resolution Image Synthesis. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12868–12878. ISSN : 2575-7075.

[Fabbri et al., 2021] Fabbri, M., Braso, G., Maugeri, G., Cetintas, O., Gasparini, R., Osep, A., Calderara, S., Leal-Taixe, L., and Cucchiara, R. (2021). MOTSynth : How can synthetic data help pedestrian detection and tracking ? In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10829–10839. IEEE.

[Fang et al., 2021] Fang, Y., Liao, B., Wang, X., Fang, J., Qi, J., Wu, R., Niu, J., and Liu, W. (2021). You Only Look at One Sequence : Rethinking Transformer in Vision through Object Detection. In *Advances in Neural Information Processing Systems*, volume 34, pages 26183–26197. Curran Associates, Inc.

[Farhadi and Redmon, 2018] Farhadi, A. and Redmon, J. (2018). Yolov3 : An incremental improvement. In *Computer vision and pattern recognition*, volume 1804, pages 1–6. Springer Berlin/Heidelberg, Germany.

[Fazlyab et al., 2019] Fazlyab, M., Robey, A., Hassani, H., Morari, M., and Pappas, G. (2019). Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[Feng et al., 2020] Feng, H., Wu, C., Chen, G., Zhang, W., and Ning, Y. (2020). Regularized Training and Tight Certification for Randomized Smoothed Classifier with Provable Robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3858–3865. ISSN : 2374-3468, 2159-5399 Issue : 04 Journal Abbreviation : AAAI.

[Feng et al., 2016] Feng, W.-c., Feng, R., Wyatt, P., and Liu, F. (2016). Understanding the Impact of Compression on Feature Detection and Matching in Computer Vision. In *2016 IEEE International Symposium on Multimedia (ISM)*, pages 457–462.

[Ferrari et al., 2021] Ferrari, C., Mueller, M. N., Jovanović, N., and Vechev, M. (2021). Complete Verification via Multi-Neuron Relaxation Guided Branch-and-Bound.

[Fischer et al., 2020] Fischer, M., Baader, M., and Vechev, M. (2020). Certified Defense to Image Transformations via Randomized Smoothing. In *Advances in Neural Information Processing Systems*, volume 33, pages 8404–8417. Curran Associates, Inc.

[Fontana et al., 2023] Fontana, M., Zeni, G., and Vantini, S. (2023). Conformal prediction : A unified review of theory and new challenges. *Bernoulli*, 29(1) :1–23. Publisher : Bernoulli Society for Mathematical Statistics and Probability.

[Fu et al., 2024] Fu, H., Liang, F., Liang, J., Fang, Z., Zhang, G., and Han, J. (2024). Learned image compression with dual-branch encoder and conditional information coding. In *2024 Data Compression Conference (DCC)*, pages 173–182.

[Fu et al., 2023] Fu, H., Liang, F., Lin, J., Li, B., Akbari, M., Liang, J., Zhang, G., Liu, D., Tu, C., and Han, J. (2023). Learned Image Compression With Gaussian-Laplacian-Logistic Mixture Model and Concatenated Residual Modules. *IEEE Transactions on Image Processing*, 32 :2063–2076. Conference Name : IEEE Transactions on Image Processing.

[Gal et al., 2023] Gal, R., Alaluf, Y., Atzmon, Y., Patashnik, O., Bermano, A. H., Chechik, G., and Cohen-or, D. (2023). An image is worth one word : Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*.

[Gao et al., 2023] Gao, C., Liu, D., Li, L., and Wu, F. (2023). Towards task-generic image compression : A study of semantics-oriented metrics. *IEEE Transactions on Multimedia*, 25 :721–735.

[Gao et al., 2022a] Gao, C., Xu, T., He, D., Wang, Y., and Qin, H. (2022a). Flexible neural image compression via code editing. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 12184–12196. Curran Associates, Inc.

[Gao et al., 2021] Gao, G., You, P., Pan, R., Han, S., Zhang, Y., Dai, Y., and Lee, H. (2021). Neural image compression via attentional multi-scale back projection and frequency decomposition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14657–14666.

[Gao et al., 2024] Gao, R., Chen, K., Xie, E., HONG, L., Li, Z., Yeung, D.-Y., and Xu, Q. (2024). Magicdrive : Street view generation with diverse 3d geometry control. In *The Twelfth International Conference on Learning Representations*.

[Gao et al., 2022b] Gao, Z., Wang, L., Han, B., and Guo, S. (2022b). Adamixer : A fast-converging query-based object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5364–5373.

[Garza and Mergenthaler-Canseco, 2023] Garza, A. and Mergenthaler-Canseco, M. (2023). Timegpt-1.

[Ge et al., 2021] Ge, Z., Liu, S., Wang, F., Li, Z., and Sun, J. (2021). Yolox : Exceeding yolo series in 2021. *ArXiv*, abs/2107.08430.

[Gehr et al., 2018] Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., and Vechev, M. (2018). AI2 : Safety and Robustness Certification of Neural Networks with Abstract Interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, San Francisco, CA. IEEE.

[Geiger et al., 2012] Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving ? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Geyer et al., 2020] Geyer, J., Kassahun, Y., Mahmudi, M., Ricou, X., Durgesh, R., Chung, A. S., Hauswald, L., Pham, V. H., Mühlegg, M., Dorn, S., Fernandez, T., Jänicke, M., Mirashi, S., Savani, C., Sturm, M., Vorobiov, O., Oelker, M., Garreis, S., and Schuberth, P. (2020). A2d2 : Audi autonomous driving dataset.

[Gibbs et al., 2024] Gibbs, I., Cherian, J. J., and Candès, E. J. (2024). Conformal Prediction With Conditional Guarantees. arXiv :2305.12616 [stat].

[Girshick, 2015] Girshick, R. (2015). Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448. ISSN : 2380-7504.

[Girshick et al., 2014] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587. ISSN : 1063-6919.

[Gluch and Urbanke, 2020] Gluch, G. and Urbanke, R. (2020). Constructing a provably adversarially-robust classifier from a high accuracy one. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, pages 3674–3684. PMLR. ISSN : 2640-3498.

[Goodfellow et al., 2014] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

[Gowal et al., 2019] Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T. A., and Kohli, P. (2019). Scalable verified training for provably robust image classification. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4841–4850.

[Gregor et al., 2015] Gregor, K., Danihelka, I., Graves, A., Rezende, D., and Wierstra, D. (2015). DRAW : A Recurrent Neural Network For Image Generation. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1462–1471. PMLR. ISSN : 1938-7228.

[Grzegorzewski and Romaniuk, 2022] Grzegorzewski, P. and Romaniuk, M. (2022). Bootstrap methods for epistemic fuzzy data. 32(2) :285–297.

[Guo et al., 2023] Guo, J., Xu, D., and Lu, G. (2023). CBANet : Toward Complexity and Bitrate Adaptive Deep Image Compression Using a Single Network. *IEEE Transactions on Image Processing*, 32 :2049–2062. Conference Name : IEEE Transactions on Image Processing.

[Guo et al., 2024] Guo, X., Huang, Q., Wang, Y., and Sun, H. (2024). End-to-End Variable-Rate Image Compression with Bi-Resolution Spatial-Channel Context Aggregation. In *Proceedings of the 5th ACM International Conference on Multimedia in Asia*, MMAsia '23, pages 1–7, New York, NY, USA. Association for Computing Machinery.

[Guo et al., 2022] Guo, Z., Zhang, Z., Feng, R., and Chen, Z. (2022). Causal Contextual Prediction for Learned Image Compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(4) :2329–2341. Conference Name : IEEE Transactions on Circuits and Systems for Video Technology.

[Gupta et al., 2022] Gupta, A., Narayan, S., Joseph, K., Khan, S., Khan, F. S., and Shah, M. (2022). Ow-detr : Open-world detection transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9235–9244.

[Haddouche and Guedj, 2023] Haddouche, M. and Guedj, B. (2023). PAC-Bayes Generalisation Bounds for Heavy-Tailed Losses through Supermartingales. *Transactions on Machine Learning Research*.

[Hannuksela et al., 2015] Hannuksela, M. M., Lainema, J., and Malamal Vadakital, V. K. (2015). The High Efficiency Image File Format Standard [Standards in a Nutshell]. *IEEE Signal Processing Magazine*, 32(4) :150–156. Conference Name : IEEE Signal Processing Magazine.

[Haque et al., 2024] Haque, N., Li, X., Chen, Z., Wu, Y., Yu, L., Iyengar, A., and Wei, W. (2024). Boosting imperceptibility of stable diffusion-based adversarial examples generation with momentum. *arXiv preprint arXiv :2410.13122*.

[Hasanzadeh et al., 2019] Hasanzadeh, A., Hajiramezanali, E., Narayanan, K., Duffield, N., Zhou, M., and Qian, X. (2019). Semi-implicit graph variational auto-encoders. *Advances in neural information processing systems*, 32.

[He et al., 2021] He, D., Zheng, Y., Sun, B., Wang, Y., and Qin, H. (2021). Checkerboard Context Model for Efficient Learned Image Compression. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14766–14775. ISSN : 2575-7075.

[He et al., 2017] He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

[He et al., 2014] He, K., Zhang, X., Ren, S., and Sun, J. (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., editors, *Computer Vision – ECCV 2014*, pages 346–361, Cham. Springer International Publishing.

[Henriksen and Lomuscio, 2020] Henriksen, P. and Lomuscio, A. (2020). Efficient Neural Network Verification via Adaptive Refinement and Adversarial Search. In *ECAI 2020*, pages 2513–2520. IOS Press.

[Henriksen and Lomuscio, 2021] Henriksen, P. and Lomuscio, A. (2021). DEEPSPLIT : An Efficient Splitting Method for Neural Network Verification via Indirect Effect Analysis. volume 3, pages 2549–2555. ISSN : 1045-0823.

[Heusel et al., 2017] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

[Heyn et al., 2022] Heyn, H.-M., Subbiah, P., Linder, J., Knauss, E., and Eriksson, O. (2022). Setting ai in context : A case study on defining the context and operational design domain for automated driving. In Gervasi, V. and Vogelsang, A., editors, *Requirements Engineering : Foundation for Software Quality*, pages 199–215. Springer International Publishing.

[Ho et al., 2020] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc.

[Ho et al., 2021a] Ho, Y.-H., Chan, C.-C., Peng, W.-H., and Hang, H.-M. (2021a). End-to-end learned image compression with augmented normalizing flows. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1931–1935.

[Ho et al., 2021b] Ho, Y.-H., Chan, C.-C., Peng, W.-H., Hang, H.-M., and Domański, M. (2021b). ANFIC : Image Compression Using Augmented Normalizing Flows. *IEEE Open Journal of Circuits and Systems*, 2 :613–626. Conference Name : IEEE Open Journal of Circuits and Systems.

[Horváth et al., 2022a] Horváth, M. Z., Mueller, M. N., Fischer, M., and Vechev, M. (2022a). Boosting randomized smoothing with variance reduced classifiers. In *International Conference on Learning Representations*.

[Horváth et al., 2022b] Horváth, M. Z., Müller, M. N., Fischer, M., and Vechev, M. (2022b). Robust and accurate - compositional architectures for randomized smoothing. In *ICLR 2022 Workshop on Socially Responsible Machine Learning*.

[Hu et al., 2023] Hu, A., Russell, L., Yeo, H., Murez, Z., Fedoseev, G., Kendall, A., Shotton, J., and Corrado, G. (2023). Gaia-1 : A generative world model for autonomous driving. *arXiv preprint arXiv :2309.17080*.

[Hu et al., 2024] Hu, Q., Yuille, A., and Zhou, Z. (2024). Synthetic Data as Validation. Submitted for publication.

[Hu et al., 2020] Hu, Y., Yang, W., and Liu, J. (2020). Coarse-to-Fine Hyper-Prior Modeling for Learned Image Compression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(07) :11013–11020. Number : 07.

[Huang et al., 2018] Huang, H., li, z., He, R., Sun, Z., and Tan, T. (2018). IntroVAE : Introspective Variational Autoencoders for Photographic Image Synthesis. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[Huang et al., 2020] Huang, X., Khetan, A., Cvitkovic, M., and Karnin, Z. (2020). Tabtransformer : Tabular data modeling using contextual embeddings. *arXiv preprint arXiv :2012.06678*.

[Huang et al., 2017] Huang, X., Kwiatkowska, M., Wang, S., and Wu, M. (2017). Safety Verification of Deep Neural Networks. In Majumdar, R. and Kunčak, V., editors, *Computer Aided Verification*, pages 3–29, Cham. Springer International Publishing.

[Inazu and Kimata, 2024] Inazu, Y. and Kimata, H. (2024). Study on Learned RGBA Image Compression Using Loss Function Based on Alpha Channel. In *Proceedings of the 2024 6th International Conference on Image, Video and Signal Processing*, IVSP '24, pages 104–110, New York, NY, USA. Association for Computing Machinery.

[Isola et al., 2017] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976. ISSN : 1063-6919.

[Jamil et al., 2023] Jamil, S., Piran, M. J., Rahman, M., and Kwon, O.-J. (2023). Learning-driven lossy image compression : A comprehensive survey. *Engineering Applications of Artificial Intelligence*, 123 :106361.

[Jang et al., 2023] Jang, K., Jun, K.-S., Kuzborskij, I., and Orabona, F. (2023). Tighter PAC-Bayes Bounds Through Coin-Betting. In *Proceedings of Thirty Sixth Conference on Learning Theory*, pages 2240–2264. PMLR. ISSN : 2640-3498.

[Jaspar et al., 2024] Jaspar, J., Viennet, E., Gualandris, D., Sauvaget, J.-L., and Fogelman-Soulié, F. (2024). Using synthetic images to improve and test object detection in the context of the autonomous vehicle. In *2024 12th European Workshop on Visual Information Processing (EUVIP)*, pages 1–6.

[Jatzkowski et al., 2021] Jatzkowski, I., Menzel, T., Bock, A., and Maurer, M. (2021). A knowledge-based approach for the automatic construction of skill graphs for online monitoring. In *2021 IEEE Intelligent Vehicles Symposium (IV)*, pages 142–149. IEEE.

[Jeon et al., 2023] Jeon, S., Choi, K. P., Park, Y., and Kim, C.-S. (2023). Context-Based Trit-Plane Coding for Progressive Image Compression. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14348–14357. ISSN : 2575-7075.

[Jeong et al., 2021] Jeong, J., Park, S., Kim, M., Lee, H.-C., Kim, D.-G., and Shin, J. (2021). SmoothMix : Training Confidence-calibrated Smoothed Classifiers for Certified Robustness. In *Advances in Neural Information Processing Systems*, volume 34, pages 30153–30168. Curran Associates, Inc.

[Jia et al., 2019] Jia, C., Liu, Z., Wang, Y., Ma, S., and Gao, W. (2019). Layered Image Compression Using Scalable Auto-Encoder. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 431–436.

[Jiang et al., 2024a] Jiang, J., Hong, G., Zhou, L., Ma, E., Hu, H., xia zhou, Xiang, J., Liu, F., Yu, K., Sun, H., Zhan, K., Jia, P., and Zhang, M. (2024a). DiVE : Dit-based video generation with enhanced control. In *ECCV 2024 Workshop on Multimodal Perception and Comprehension of Corner Cases in Autonomous Driving*.

[Jiang et al., 2024b] Jiang, W., Ning, P., Yang, J., Zhai, Y., Gao, F., and Wang, R. (2024b). LLIC : Large Receptive Field Transform Coding with Adaptive Weights for Learned Image Compression. *IEEE Transactions on Multimedia*, pages 1–15. Conference Name : IEEE Transactions on Multimedia.

[Jiang et al., 2023] Jiang, X., Tan, W., Tan, T., Yan, B., and Shen, L. (2023). Multi-Modality Deep Network for Extreme Learned Image Compression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(1) :1033–1041. Number : 1.

[Jiang et al., 2021] Jiang, Y., Chang, S., and Wang, Z. (2021). TransGAN : Two Pure Transformers Can Make One Strong GAN, and That Can Scale Up. In *Advances in Neural Information Processing Systems*, volume 34, pages 14745–14758. Curran Associates, Inc.

[Jocher et al., 2023] Jocher, G., Qiu, J., and Chaurasia, A. (2023). Ultralytics YOLO. original-date : 2022-09-11T16 :39 :45Z.

[Jovanovi'c et al., 2021] Jovanovi'c, N., Balunovic, M., Baader, M., and Vechev, M. T. (2021). On the Paradox of Certified Training. *Trans. Mach. Learn. Res.*

[Kao et al., 2023] Kao, C.-H., Chen, Y.-H., Chien, C., Chiu, W.-C., and Peng, W.-H. (2023). Transformer-based Image Compression with Variable Image Quality Objectives. In *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1718–1725. ISSN : 2640-0103.

[Karras et al., 2018] Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2018). Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*.

[Karras et al., 2021] Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., and Aila, T. (2021). Alias-Free Generative Adversarial Networks. In *Advances in Neural Information Processing Systems*, volume 34, pages 852–863. Curran Associates, Inc.

[Karras et al., 2019] Karras, T., Laine, S., and Aila, T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4396–4405. ISSN : 2575-7075.

[Karras et al., 2020] Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., and Aila, T. (2020). Analyzing and Improving the Image Quality of StyleGAN. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8107–8116. ISSN : 2575-7075.

[Kateb et al., 2021] Kateb, F. A., Monowar, M. M., Hamid, M. A., Ohi, A. Q., and Mridha, M. F. (2021). Fruitdet : Attentive feature aggregation for real-time fruit detection in orchards. *Agronomy*, 11(12) :2440.

[Katz et al., 2017] Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J. (2017). Reluplex : An Efficient SMT Solver for Verifying Deep Neural Networks. In Majumdar, R. and Kunčak, V., editors, *Computer Aided Verification*, pages 97–117. Springer International Publishing.

[Katz et al., 2019] Katz, G., Huang, D. A., Ibeling, D., Julian, K., Lazarus, C., Lim, R., Shah, P., Thakoor, S., Wu, H., Zeljić, A., Dill, D. L., Kochenderfer, M. J., and Barrett, C. (2019). The Marabou Framework for Verification and Analysis of Deep Neural Networks. In Dillig, I. and Tasiran, S., editors, *Computer Aided Verification*, pages 443–452. Springer International Publishing.

[Khanam and Hussain, 2024] Khanam, R. and Hussain, M. (2024). Yolov11 : An overview of the key architectural enhancements. *arXiv preprint arXiv :2410.17725*.

[Kim et al., 2022] Kim, G., Kwon, T., and Ye, J. C. (2022). DiffusionCLIP : Text-Guided Diffusion Models for Robust Image Manipulation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2416–2425. ISSN : 2575-7075.

[Kim et al., 2021] Kim, S. W., Philion, J., Torralba, A., and Fidler, S. (2021). DriveGAN : Towards a Controllable High-Quality Neural Simulation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5816–5825, Nashville, TN, USA. IEEE.

[Kingma et al., 2021] Kingma, D., Salimans, T., Poole, B., and Ho, J. (2021). Variational Diffusion Models. In *Advances in Neural Information Processing Systems*, volume 34, pages 21696–21707. Curran Associates, Inc.

[Kong et al., 2020] Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. (2020). DiffWave : A Versatile Diffusion Model for Audio Synthesis.

[Kotyan et al., 2025] Kotyan, S., Mao, P., and Vargas, D. V. (2025). Breaking free : Hacking diffusion models for generating adversarial examples and bypassing safety guardrails. In *Submitted to The Thirteenth International Conference on Learning Representations*. under review.

[Kou et al., 2022] Kou, Y., Zheng, Q., and Wang, Y. (2022). Certified Adversarial Robustness Under the Bounded Support Set. In *Proceedings of the 39th International Conference on Machine Learning*, pages 11559–11597. PMLR. ISSN : 2640-3498.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

[Kumar and Goldstein, 2021] Kumar, A. and Goldstein, T. (2021). Center Smoothing : Certified Robustness for Networks with Structured Outputs. In *Advances in Neural Information Processing Systems*, volume 34, pages 5560–5575. Curran Associates, Inc.

[Kumar et al., 2020] Kumar, A., Levine, A., Feizi, S., and Goldstein, T. (2020). Certifying Confidence via Randomized Smoothing. In *Advances in Neural Information Processing Systems*, volume 33, pages 5165–5177. Curran Associates, Inc.

[Kumar et al., 2019] Kumar, K., Kumar, R., De Boissiere, T., Gestin, L., Teoh, W. Z., Sotelo, J., De Brebisson, A., Bengio, Y., and Courville, A. C. (2019). Melgan : Generative adversarial networks for conditional waveform synthesis. *Advances in neural information processing systems*, 32.

[Kuznetsova et al., 2020] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T., and Ferrari, V. (2020). The open images dataset v4 : Unified image classification, object detection, and visual relationship detection at scale. *IJCV*.

[Kynkäänniemi et al., 2019] Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., and Aila, T. (2019). Improved precision and recall metric for assessing generative models. *Advances in neural information processing systems*, 32.

[Labarbarie et al., 2022] Labarbarie, P., Hajri, H., and Arnaudon, M. (2022). Riemannian data-dependent randomized smoothing for neural networks certification. *International Conference on Machine Learning Workshop on New Frontiers in Adversarial Machine Learning*.

[Law and Deng, 2020] Law, H. and Deng, J. (2020). CornerNet : Detecting Objects as Paired Keypoints. *Int J Comput Vis*, 128(3) :642–656.

[Leclerc et al., 2022] Leclerc, G., Salman, H., Ilyas, A., Vemprala, S., Engstrom, L., Vineet, V., Xiao, K., Zhang, P., Santurkar, S., Yang, G., Kapoor, A., and Madry, A. (2022). 3DB : A Framework for Debugging Computer Vision Models. pages 8498–8511. NeurIPS.

[Lee et al.,] Lee, H., Kim, Y., and Lee, S.-G. Multi-scale contrastive learning for complex scene generation. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 764–774. IEEE.

[Lee et al., 2018] Lee, J., Cho, S., and Beack, S. (2018). Context-adaptive entropy model for end-to-end optimized image compression. In *International Conference on Learning Representations*.

[Lee et al., 2022a] Lee, J., Jeon, S., Choi, K. P., Park, Y., and Kim, C.-S. (2022a). DPICT : Deep Progressive Image Compression Using Trit-Planes. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16092–16101. ISSN : 2575-7075.

[Lee et al., 2022b] Lee, J., Jeong, S., and Kim, M. (2022b). Selective compression learning of latent representations for variable-rate image compression. *Advances in Neural Information Processing Systems*, 35 :13146–13157.

[Lee et al., 2022c] Lee, K., Chang, H., Jiang, L., Zhang, H., Tu, Z., and Liu, C. (2022c). ViTGAN : Training GANs with vision transformers. In *International Conference on Learning Representations*.

[Lee et al., 2021] Lee, S., Lee, W., Park, J., and Lee, J. (2021). Towards Better Understanding of Training Certifiably Robust Models against Adversarial Examples. In *Advances in Neural Information Processing Systems*, volume 34, pages 953–964. Curran Associates, Inc.

[Lei et al., 2023] Lei, Z., Duan, P., Hong, X., Mota, J. F. C., Shi, J., and Wang, C.-X. (2023). Progressive Deep Image Compression for Hybrid Contexts of Image Classification and Reconstruction. *IEEE Journal on Selected Areas in Communications*, 41(1) :72–89. Conference Name : IEEE Journal on Selected Areas in Communications.

[Levine and Feizi, 2021] Levine, A. J. and Feizi, S. (2021). Improved, Deterministic Smoothing for L_1 Certified Robustness. In *Proceedings of the 38th International Conference on Machine Learning*, pages 6254–6264. PMLR. ISSN : 2640-3498.

[Li et al., 2022] Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W., et al. (2022). Yolov6 : A single-stage object detection framework for industrial applications. *arXiv preprint arXiv :2209.02976*.

[Li et al., 2024a] Li, H., Li, S., Ding, S., Dai, W., Cao, M., Li, C., Zou, J., and Xiong, H. (2024a). Image compression for machine and human vision with spatial-frequency adaptation. *ECCV. Springer*, 2.

[Li et al., 2021] Li, L., Weber, M., Xu, X., Rimanic, L., Kailkhura, B., Xie, T., Zhang, C., and Li, B. (2021). TSS : Transformation-Specific Smoothing for Robustness Certification. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, pages 535–557, New York, NY, USA. Association for Computing Machinery.

[Li et al., 2018] Li, M., Zuo, W., Gu, S., Zhao, D., and Zhang, D. (2018). Learning Convolutional Networks for Content-Weighted Image Compression. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3214–3223. ISSN : 2575-7075.

[Li et al., 2024b] Li, X., Zhang, Y., and Ye, X. (2024b). Drivingdiffusion : Layout-guided multi-view driving scenarios video generation with latent diffusion model. In *European Conference on Computer Vision*, pages 469–485. Springer.

[Liang et al., 2023] Liang, J., Liu, M., Yao, C., Lin, C., and Zhao, Y. (2023). SIGVIC : Spatial Importance Guided Variable-Rate Image Compression. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. ISSN : 2379-190X.

[Liebender et al., 2024] Liebender, C., Bezerra, R., Ohno, K., and Tadokoro, S. (2024). Region of interest loss for anonymizing learned image compression. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pages 3569–3576.

[Lin et al., 2017a] Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017a). Feature Pyramid Networks for Object Detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944. ISSN : 1063-6919.

[Lin et al., 2017b] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017b). Focal Loss for Dense Object Detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007. ISSN : 2380-7504.

[Lin et al., 2014] Lin, T.-Y., Maire, M., Belongie, S. J., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco : Common objects in context. In *European Conference on Computer Vision*.

[Lin et al., 2024] Lin, Y., Zhang, J., Chen, Y., and Li, H. (2024). SD-NAE : Generating natural adversarial examples with stable diffusion. In *The Second Tiny Papers Track at ICLR 2024*.

[Liu et al., 2024] Liu, J., Feng, R., Qi, Y., Chen, Q., Chen, Z., Zeng, W., and Jin, X. (2024). Rate-distortion-cognition controllable versatile neural image compression. In *European Conference on Computer Vision*, pages 329–348. Springer.

[Liu et al., 2022a] Liu, J., Sun, H., and Katto, J. (2022a). Improving multiple machine vision tasks in the compressed domain. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 331–337.

[Liu et al., 2023a] Liu, J., Sun, H., and Katto, J. (2023a). Learned Image Compression with Mixed Transformer-CNN Architectures. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14388–14397. ISSN : 2575-7075.

[Liu et al., 2016] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). SSD : Single Shot MultiBox Detector. In Leibe, B., Matas, J., Sebe, N., and Welling, M., editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham. Springer International Publishing.

[Liu et al., 2023b] Liu, Y., Schiele, B., Vedaldi, A., and Rupprecht, C. (2023b). Continual detection transformer for incremental object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23799–23808.

[Liu et al., 2021] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021). Swin Transformer : Hierarchical Vision Transformer using Shifted Windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002. ISSN : 2380-7504.

[Liu et al., 2022b] Liu, Z., Wang, H., and Su, T. (2022b). Learned image compression with multi-scale spatial and contextual information fusion. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 706–710.

[Lu et al., 2022] Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., and Zhu, J. (2022). DPM-Solver : A Fast ODE Solver for Diffusion Probabilistic Model Sampling in Around 10 Steps. *Advances in Neural Information Processing Systems*, 35 :5775–5787.

[Lu and Kumar, 2019] Lu, J. and Kumar, M. P. (2019). Neural Network Branching for Neural Network Verification. *ArXiv*.

[Lu et al., 2021] Lu, Y., Zhu, Y., Yang, Y., Said, A., and Cohen, T. S. (2021). Progressive Neural Image Compression With Nested Quantization And Latent Ordering. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 539–543. ISSN : 2381-8549.

[Luo et al., 2023] Luo, A., Sun, H., Liu, J., Lin, F., and Katto, J. (2023). PTS-LIC : Pruning Threshold Searching for Lightweight Learned Image Compression. In *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5. ISSN : 2642-9357.

[Lyu et al., 2020] Lyu, Z., Ko, C.-Y., Kong, Z., Wong, N., Lin, D., and Daniel, L. (2020). Fastened CROWN : Tightened Neural Network Robustness Certificates. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04) :5037–5044. Number : 04.

[Lyu et al., 2022] Lyu, Z., Xu, X., Yang, C., Lin, D., and Dai, B. (2022). Accelerating diffusion models via early stop of the diffusion process. *arXiv preprint arXiv :2205.12524*.

[Ma et al., 2021] Ma, C., Wang, Z., Liao, R., and Ye, Y. (2021). A cross channel context model for latents in deep image compression. *arXiv preprint arXiv :2103.02884*.

[Ma et al., 2022a] Ma, C., Wang, Z., Liao, R.-L., and Ye, Y. (2022a). Improved deep image compression with joint optimization of cross channel context model and generalized loop filter. In *2022 Data Compression Conference (DCC)*, pages 23–32.

[Ma et al., 2020] Ma, H., Liu, D., Xiong, R., and Wu, F. (2020). iWave : CNN-Based Wavelet-Like Transform for Image Compression. *IEEE Transactions on Multimedia*, 22(7) :1667–1679. Conference Name : IEEE Transactions on Multimedia.

[Ma et al., 2022b] Ma, H., Liu, D., Yan, N., Li, H., and Wu, F. (2022b). End-to-end optimized versatile image compression with wavelet-like transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3) :1247–1263.

[Ma et al., 2023] Ma, S., Wang, Y., Wei, Y., Fan, J., Li, T. H., Liu, H., and Lv, F. (2023). Cat : Localization and identification cascade detection transformer for open-world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19681–19690.

[Ma and Sojoudi, 2021] Ma, Z. and Sojoudi, S. (2021). A Sequential Framework Towards an Exact SDP Verification of Neural Networks. In *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–8.

[Madry et al., 2018] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*.

[Mansimov et al., 2016] Mansimov, E., Parisotto, E., Ba, J., and Salakhutdinov, R. (2016). Generating images from captions with attention. In *ICLR*.

[McAllester, 1999] McAllester, D. A. (1999). Some PAC-Bayesian Theorems. *Machine Learning*, 37(3) :355–363.

[Mehlhorn et al., 2023] Mehlhorn, M. A., Richter, A., and Shardt, Y. A. (2023). Ruling the operational boundaries : A survey on operational design domains of autonomous driving systems. *IFAC-PapersOnLine*, 56(2) :2202–2213. 22nd IFAC World Congress.

[Mei et al., 2024] Mei, J., Ma, Y., Yang, X., Wen, L., Wei, T., Dou, M., Shi, B., and Liu, Y. (2024). Dreamforge : Motion-aware autoregressive video generation for multi-view driving scenes. In *ECCV 2024 Workshop on Multimodal Perception and Comprehension of Corner Cases in Autonomous Driving*.

[Meng et al., 2021a] Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S. (2021a). SDEdit : Guided Image Synthesis and Editing with Stochastic Differential Equations.

[Meng et al., 2021b] Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., and Wang, J. (2021b). Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3651–3660.

[Mentzer et al., 2018] Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., and Gool, L. V. (2018). Conditional Probability Models for Deep Image Compression. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4394–4402. ISSN : 2575-7075.

[Mentzer et al., 2020] Mentzer, F., Toderici, G. D., Tschannen, M., and Agustsson, E. (2020). High-Fidelity Generative Image Compression. In *Advances in Neural Information Processing Systems*, volume 33, pages 11913–11924. Curran Associates, Inc.

[Minnen et al., 2018a] Minnen, D., Ballé, J., and Toderici, G. D. (2018a). Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31.

[Minnen et al., 2018b] Minnen, D., Ballé, J., and Toderici, G. D. (2018b). Joint Autoregressive and Hierarchical Priors for Learned Image Compression. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[Minnen and Singh, 2020] Minnen, D. and Singh, S. (2020). Channel-Wise Autoregressive Entropy Models for Learned Image Compression. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 3339–3343. ISSN : 2381-8549.

[Mirman et al., 2018] Mirman, M., Gehr, T., and Vechev, M. (2018). Differentiable abstract interpretation for provably robust neural networks. In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3578–3586. PMLR.

[Mirza and Osindero, 2014] Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *CoRR*, abs/1411.1784.

[Mou et al., 2024] Mou, C., Wang, X., Song, J., Shan, Y., and Zhang, J. (2024). Dragondiffusion : Enabling drag-style manipulation on diffusion models. In *The Twelfth International Conference on Learning Representations*.

[Mueller et al., 2021] Mueller, M. N., Balunovic, M., and Vechev, M. (2021). Certify or predict : Boosting certified robustness with compositional architectures. In *International Conference on Learning Representations*.

[Mueller et al., 2023] Mueller, M. N., Eckert, F., Fischer, M., and Vechev, M. (2023). Certified training : Small boxes are all you need. In *The Eleventh International Conference on Learning Representations*.

[Müller et al., 2021] Müller, C., Serre, F., Singh, G., Püschel, M., and Vechev, M. (2021). Scaling Polyhedral Neural Network Verification on GPUs. *Proceedings of Machine Learning and Systems*, 3 :733–746.

[Nash et al., 2021] Nash, C., Menick, J., Dieleman, S., and Battaglia, P. (2021). Generating images with sparse representations. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7958–7968. PMLR.

[Natarajan and Madden, 2023] Natarajan, S. A. and Madden, M. G. (2023). Hybrid synthetic data generation pipeline that outperforms real data. *JEI*, 32(2) :023011. Publisher : SPIE.

[Nichol and Dhariwal, 2021] Nichol, A. Q. and Dhariwal, P. (2021). Improved Denoising Diffusion Probabilistic Models. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8162–8171. PMLR. ISSN : 2640-3498.

[Nichol et al., 2022] Nichol, A. Q., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., Mcgrew, B., Sutskever, I., and Chen, M. (2022). GLIDE : Towards photorealistic image generation and editing with text-guided diffusion models. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S., editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16784–16804. PMLR.

[Ouyang, 2022] Ouyang, H. (2022). Deyo : Detr with yolo for step-by-step object detection. *arXiv preprint arXiv :2211.06588*.

[Ouyang, 2023a] Ouyang, H. (2023a). Deyov2 : Rank feature with greedy matching for end-to-end object detection. *arXiv preprint arXiv :2306.09165*.

[Ouyang, 2023b] Ouyang, H. (2023b). Deyov3 : Detr with yolo for real-time object detection. *arXiv preprint arXiv :2309.11851*.

[Pakdaman et al., 2024] Pakdaman, F., Nami, S., and Gabbouj, M. (2024). Perceptual learned image compression via end-to-end jnd-based optimization. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 1146–1151.

[Palma et al., 2021] Palma, A. D., Behl, H., Bunel, R. R., Torr, P., and Kumar, M. P. (2021). Scaling the convex barrier with active sets. In *International Conference on Learning Representations*.

[Park et al., 2023] Park, J., Lee, J., and Kim, M. (2023). COMPASS : High-Efficiency Deep Image Compression with Arbitrary-scale Spatial Scalability. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12780–12789. ISSN : 2380-7504.

[Park et al., 2020] Park, K., Lee, H., Yang, H., and Oh, S.-Y. (2020). Improving Instance Segmentation using Synthetic Data with Artificial Distractors. *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, pages 22–26. Conference Name : 2020 20th International Conference on Control, Automation and Systems (ICCAS) ISBN : 9788993215205 Place : Busan, Korea (South) Publisher : IEEE.

[Paul et al., 2021] Paul, S., Riffat, M., Yasir, A., Mahim, M. N., Sharnali, B. Y., Naheen, I. T., Rahman, A., and Kulkarni, A. (2021). Industry 4.0 Applications for Medical/Healthcare Services. *Journal of Sensor and Actuator Networks*, 10(3) :43. Number : 3 Publisher : Multidisciplinary Digital Publishing Institute.

[Peng et al., 2023] Peng, L., Li, J., Shao, W., and Wang, H. (2023). PeSOTIF : a challenging visual dataset for perception SOTIF problems in long-tail traffic scenarios. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pages 1–8. ISSN : 2642-7214.

[Pfrommer et al., 2023a] Pfrommer, S., Anderson, B., Piet, J., and Sojoudi, S. (2023a). Asymmetric Certified Robustness via Feature-Convex Neural Networks. *Advances in Neural Information Processing Systems*, 36 :52365–52400.

[Pfrommer et al., 2023b] Pfrommer, S., Anderson, B. G., and Sojoudi, S. (2023b). Projected Randomized Smoothing for Certified Adversarial Robustness. *Transactions on Machine Learning Research*.

[Pitropov et al., 2021] Pitropov, M., Garcia, D., Rebello, J., Smart, M., Wang, C., Czarnecki, K., and Waslander, S. (2021). Canadian adverse driving conditions dataset. 40(4) :681–690.

[Podell et al., 2024] Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. (2024). SDXL : Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*.

[Poyser et al., 2021] Poyser, M., Atapour-Abarghouei, A., and Breckon, T. P. (2021). On the Impact of Lossy Image and Video Compression on the Performance of Deep Convolutional Neural Network Architectures. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2830–2837. ISSN : 1051-4651.

[Prakash et al., 2019] Prakash, A., Boochoon, S., Brophy, M., Acuna, D., Cameracci, E., State, G., Shapira, O., and Birchfield, S. (2019). Structured domain randomization : Bridging the reality gap by context-aware synthetic data. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7249–7255. ISSN : 2577-087X.

[Prativadibhayankaram et al., 2023] Prativadibhayankaram, S., Richter, T., Sparenberg, H., and Foessel, S. (2023). Color learning for image compression. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 2330–2334.

[Pu et al., 2024] Pu, Y., Liang, W., Hao, Y., Yuan, Y., Yang, Y., Zhang, C., Hu, H., and Huang, G. (2024). Rank-detr for high quality object detection. *Advances in Neural Information Processing Systems*, 36.

[Qian et al., 2021] Qian, Y., Tan, Z., Sun, X., Lin, M., Li, D., Sun, Z., Li, H., and Jin, R. (2021). Learning accurate entropy model with global reference for image compression. In *International Conference on Learning Representations*.

[Qiao et al., 2021] Qiao, S., Chen, L.-C., and Yuille, A. (2021). DetectoRS : Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10208–10219. ISSN : 2575-7075.

[Qin et al., 2024] Qin, S., Zhou, Y.-M., Wang, J.-P., Chen, B., An, B.-Y., Dai, T., and Xia, S.-T. (2024). Progressive learning with visual prompt tuning for variable-rate image compression. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 1767–1773. IEEE.

[Qin et al., 2019] Qin, Z., Li, Z., Zhang, Z., Bao, Y., Yu, G., Peng, Y., and Sun, J. (2019). ThunderNet : Towards Real-Time Generic Object Detection on Mobile Devices. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6717–6726. ISSN : 2380-7504.

[Rabbani and Joshi, 2002] Rabbani, M. and Joshi, R. (2002). An overview of the JPEG 2000 still image compression standard. *Signal Processing : Image Communication*, 17(1) :3–48.

[Radford et al., 2016] Radford, A., Metz, L., and Chintala, S. (2016). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv. arXiv :1511.06434.

[Raghunathan et al., 2018] Raghunathan, A., Steinhardt, J., and Liang, P. S. (2018). Semidefinite relaxations for certifying robustness to adversarial examples. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[Ramesh et al., 2022] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv :2204.06125*.

[Ramesh et al., 2021] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-Shot Text-to-Image Generation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8821–8831. PMLR. ISSN : 2640-3498.

[Razavi et al., 2019] Razavi, A., Van den Oord, A., and Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32.

[Redmon et al., 2016] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You Only Look Once : Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788. ISSN : 1063-6919.

[Redmon and Farhadi, 2017] Redmon, J. and Farhadi, A. (2017). YOLO9000 : Better, Faster, Stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525. ISSN : 1063-6919.

[Ren et al., 2016] Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6) :1137–1149. Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence.

[Richter et al., 2023] Richter, S. R., Alhaija, H. A., and Koltun, V. (2023). Enhancing Photorealism Enhancement. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(2) :1700–1715.

[Rippel and Bourdev, 2017] Rippel, O. and Bourdev, L. (2017). Real-Time Adaptive Image Compression. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2922–2930. PMLR. ISSN : 2640-3498.

[Rivasplata et al., 2020] Rivasplata, O., Kuzborskij, I., Szepesvari, C., and Shawe-Taylor, J. (2020). PAC-Bayes Analysis Beyond the Usual Bounds. In *Advances in Neural Information Processing Systems*, volume 33, pages 16833–16845. Curran Associates, Inc.

[Rombach et al., 2022] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685. IEEE.

[Ruan et al., 2023a] Ruan, H., Wang, F., and Wang, Y. (2023a). An efficient convnet for learned image compression with transformer-style architecture. In *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5.

[Ruan et al., 2023b] Ruan, H., Wang, F., Xu, T., Tan, Z., and Wang, Y. (2023b). MIXLIC : Mixing Global and Local Context Model for learned Image Compression. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 684–689. ISSN : 1945-788X.

[Ruiz et al., 2023] Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., and Aberman, K. (2023). DreamBooth : Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22500–22510. ISSN : 2575-7075.

[Rumezhak et al., 2023] Rumezhak, T., Eiras, F. G., S. Torr, P. H., and Bibi, A. (2023). RANCER : Non-Axis Aligned Anisotropic Certification with Randomized Smoothing. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4661–4669, Waikoloa, HI, USA. IEEE.

[Saharia et al., 2022] Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. (2022). Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding. *Advances in Neural Information Processing Systems*, 35 :36479–36494.

[Saito et al., 2020] Saito, M., Saito, S., Koyama, M., and Kobayashi, S. (2020). Train sparsely, generate densely : Memory-efficient unsupervised training of high-resolution temporal gan.

[Salimans et al., 2016] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. *Advances in neural information processing systems*, 29.

[Santurkar et al., 2018] Santurkar, S., Budden, D., and Shavit, N. (2018). Generative Compression. In *2018 Picture Coding Symposium (PCS)*, pages 258–262. ISSN : 2472-7822.

[Sauer et al., 2022] Sauer, A., Schwarz, K., and Geiger, A. (2022). StyleGAN-XL : Scaling StyleGAN to Large Diverse Datasets. In *ACM SIGGRAPH 2022 Conference Proceedings*, SIGGRAPH '22, pages 1–10, New York, NY, USA. Association for Computing Machinery.

[Serradilla et al., 2022] Serradilla, O., Zugasti, E., Rodriguez, J., and Zurutuza, U. (2022). Deep learning models for predictive maintenance : a survey, comparison, challenges and prospects. *Appl Intell*, 52(10) :10934–10964.

[Shen and Liu, 2023] Shen, T. and Liu, Y. (2023). Learned image compression with transformers. In *Big Data V : Learning, Analytics, and Applications*, volume 12522, pages 10–20. SPIE.

[Shi et al., 2023] Shi, Y., Zhang, K., Wang, J., Ling, N., and Yin, B. (2023). Variable-rate image compression based on side information compensation and r-λ model rate control. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(7) :3488–3501.

[Shi et al., 2024] Shi, Z., Jin, Q., Kolter, Z., Jana, S., Hsieh, C.-J., and Zhang, H. (2024). Neural Network Verification with Branch-and-Bound for General Nonlinearities.

[Shi et al., 2021] Shi, Z., Wang, Y., Zhang, H., Yi, J., and Hsieh, C.-J. (2021). Fast certified robust training with short warmup. In *ICML 2021 Workshop on Adversarial Machine Learning*.

[Shindo et al., 2024a] Shindo, T., Watanabe, T., Yamada, K., and Watanabe, H. (2024a). Image coding for machines with object region learning. In *2024 IEEE 21st Consumer Communications and Networking Conference (CCNC)*, pages 1040–1041.

[Shindo et al., 2024b] Shindo, T., Yamada, K., Watanabe, T., and Watanabe, H. (2024b). Image coding for machines with edge information learning using segment anything. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 3702–3708.

[Shizhan, 2023] Shizhan, L. (2023). MTE : Learned Image Compression with a Merge-Then-Estimate Entropy Model. In *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5. ISSN : 2642-9357.

[Simonyan and Zisserman, 2015] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. Number arXiv :1409.1556. arXiv.

[Singh et al., 2019a] Singh, G., Ganvir, R., Püschel, M., and Vechev, M. (2019a). Beyond the Single Neuron Convex Barrier for Neural Network Certification. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[Singh et al., 2018] Singh, G., Gehr, T., Mirman, M., Püschel, M., and Vechev, M. T. (2018). Fast and effective robustness certification. In *NeurIPS*, pages 10825–10836.

[Singh et al., 2019b] Singh, G., Gehr, T., Püschel, M., and Vechev, M. (2019b). Boosting robustness certification of neural networks. In *International Conference on Learning Representations*.

[Singh et al., 2019c] Singh, G., Gehr, T., Püschel, M., and Vechev, M. T. (2019c). An abstract domain for certifying neural networks. *Proc. ACM Program. Lang.*, 3(POPL) :41 :1–41 :30.

[Sinha et al., 2021] Sinha, A., Song, J., Meng, C., and Ermon, S. (2021). D2C : Diffusion-Decoding Models for Few-Shot Conditional Generation. In *Advances in Neural Information Processing Systems*, volume 34, pages 12533–12548. Curran Associates, Inc.

[Sohn et al., 2015] Sohn, K., Lee, H., and Yan, X. (2015). Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

[Song et al., 2021a] Song, J., Meng, C., and Ermon, S. (2021a). Denoising diffusion implicit models. In *International Conference on Learning Representations*.

[Song et al., 2021b] Song, M., Choi, J., and Han, B. (2021b). Variable-rate deep image compression through spatially-adaptive feature transform. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2360–2369.

[Song et al., 2021c] Song, M., Choi, J., and Han, B. (2021c). Variable-Rate Deep Image Compression through Spatially-Adaptive Feature Transform. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2360–2369. ISSN : 2380-7504.

[Strohmayer and Kampel, 2021] Strohmayer, J. and Kampel, M. (2021). Data synthesis for large-scale supermarket product recognition. pages 64–66. Verlag der TU Graz. Accepted : 2022-08-04T16 :03 :38Z.

[Su et al., 2024] Su, J., Gu, S., Duan, Y., Chen, X., and Luo, J. (2024). Text2street : Controllable text-to-image generation for street views. *arXiv preprint arXiv :2402.04504*.

[Sui et al., 2024] Sui, Y., Ding, D., Pan, X., Xu, X., Liu, S., Yuan, B., and Chen, Z. (2024). Corner-to-Center long-range context model for efficient learned image compression. *Journal of Visual Communication and Image Representation*, 98 :103990.

[Sun et al., 2020] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., and Anguelov, D. (2020). Scalability in Perception for Autonomous Driving : Waymo Open Dataset. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2443–2451. ISSN : 2575-7075.

[Sun et al., 2021] Sun, Z., Tan, Z., Sun, X., Zhang, F., Qian, Y., Li, D., and Li, H. (2021). Interpolation Variable Rate Image Compression. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, pages 5574–5582, New York, NY, USA. Association for Computing Machinery.

[Tan et al., 2020] Tan, M., Pang, R., and Le, Q. V. (2020). EfficientDet : Scalable and Efficient Object Detection. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10778–10787. ISSN : 2575-7075.

[Tang et al., 2023] Tang, Z., Wang, H., Yi, X., Zhang, Y., Kwong, S., and Kuo, C.-C. J. (2023). Joint graph attention and asymmetric convolutional neural network for deep image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(1) :421–433.

[Tao et al., 2023] Tao, L., Gao, W., Li, G., and Zhang, C. (2023). AdaNIC : Towards Practical Neural Image Compression via Dynamic Transform Routing. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16833–16842. ISSN : 2380-7504.

[Teneggi et al., 2023] Teneggi, J., Tivnan, M., Stayman, W., and Sulam, J. (2023). How to trust your diffusion model : A convex optimization approach to conformal risk control. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J., editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 33940–33960. PMLR.

[Tian et al., 2019] Tian, Z., Shen, C., Chen, H., and He, T. (2019). FCOS : Fully Convolutional One-Stage Object Detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9626–9635. ISSN : 2380-7504.

[Timans et al., 2024] Timans, A., Straehle, C.-N., Sakmann, K., and Nalisnick, E. (2024). Adaptive bounding box uncertainties via two-step conformal prediction. In *Proceedings of the European Conference on Computer Vision*.

[Tjeng et al., 2019] Tjeng, V., Xiao, K., and Tedrake, R. (2019). Evaluating Robustness of Neural Networks with Mixed Integer Programming. In *International Conference on Learning Representations (ICLR)*.

[Tong et al., 2023] Tong, K., Wu, Y., Li, Y., Zhang, K., Zhang, L., and Jin, X. (2023). QVRF : A Quantization-Error-Aware Variable Rate Framework for Learned Image Compression. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 1310–1314.

[Tremblay et al., 2018] Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., and Birchfield, S. (2018). Training Deep Networks with Synthetic Data : Bridging the Reality Gap by Domain Randomization. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1082–10828, Salt Lake City, UT. IEEE.

[Tschannen et al., 2018] Tschannen, M., Agustsson, E., and Lucic, M. (2018). Deep Generative Models for Distribution-Preserving Lossy Compression. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[Tulyakov et al., 2018] Tulyakov, S., Liu, M.-Y., Yang, X., and Kautz, J. (2018). MoCoGAN : Decomposing motion and content for video generation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1526–1535.

[van den Oord et al., 2016] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. (2016). Wavenet : A generative model for raw audio. In *Speech Synthesis Workshop*.

[van den Oord et al., 2017] van den Oord, A., Vinyals, O., and kavukcuoglu, k. (2017). Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Vapnik and Chervonenkis, 2015] Vapnik, V. N. and Chervonenkis, A. Y. (2015). On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. In Vovk, V., Papadopoulos, H., and Gammerman, A., editors, *Measures of Complexity : Festschrift for Alexey Chervonenkis*, pages 11–30. Springer International Publishing, Cham.

[Vasiljevic et al., 2017] Vasiljevic, I., Chakrabarti, A., and Shakhnarovich, G. (2017). Examining the Impact of Blur on Recognition by Convolutional Networks. Unpublished.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

[Vazquez and Facelli, 2022] Vazquez, J. and Facelli, J. C. (2022). Conformal Prediction in Clinical Medical Sciences. *J Healthc Inform Res*, 6(3) :241–252.

[Viallard et al., 2021] Viallard, P., VIDOT, E. G., Habrard, A., and Morvant, E. (2021). A PAC-Bayes Analysis of Adversarial Robustness. In *Advances in Neural Information Processing Systems*, volume 34, pages 14421–14433. Curran Associates, Inc.

[Wallace, 1992] Wallace, G. (1992). The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1) :xviii–xxxiv. Conference Name : IEEE Transactions on Consumer Electronics.

[Wang et al., 2024a] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., and Ding, G. (2024a). Yolov10 : Real-time end-to-end object detection. *arXiv preprint arXiv :2405.14458*.

[Wang et al., 2021a] Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2021a). Scaled-YOLOv4 : Scaling Cross Stage Partial Network. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13024–13033. ISSN : 2575-7075.

[Wang et al., 2023a] Wang, C.-Y., Bochkovskiy, A., and Liao, H.-Y. M. (2023a). YOLOv7 : Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7464–7475. ISSN : 2575-7075.

[Wang et al., 2021b] Wang, C.-Y., Yeh, I.-H., and Liao, H. (2021b). You only learn one representation : Unified network for multiple tasks. *J. Inf. Sci. Eng.*, 39 :691–709.

[Wang et al., 2024b] Wang, C.-Y., Yeh, I.-H., and Liao, H.-Y. M. (2024b). Yolov9 : Learning what you want to learn using programmable gradient information. *arXiv preprint arXiv :2402.13616*.

[Wang et al., 2022a] Wang, D., Yang, W., Hu, Y., and Liu, J. (2022a). Neural data-dependent transform for learned image compression. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 17379–17388.

[Wang et al., 2022b] Wang, F., Chen, J., and Wang, R. (2022b). Entropy-Reduced Attention for Image Compression. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 2401–2405. ISSN : 2381-8549.

[Wang et al., 2023b] Wang, H., Wang, H., Xue, J., Lu, E., Wang, H., Wu, Z., and Song, Y. (2023b). Improving Image Compression Performance by Spatial-Channel Context Adaptive Model. In *2023 4th International Conference on Electronic Communication and Artificial Intelligence (ICECAI)*, pages 124–130.

[Wang et al., 2018a] Wang, S., Pei, K., Whitehouse, J., Yang, J., and Jana, S. (2018a). Efficient Formal Safety Analysis of Neural Networks. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[Wang et al., 2018b] Wang, S., Pei, K., Whitehouse, J., Yang, J., and Jana, S. (2018b). Formal Security Analysis of Neural Networks using Symbolic Intervals. pages 1599–1614.

[Wang and Wang, 2022] Wang, S. and Wang, H. (2022). Discretized gaussian mixture hyperprior for learned image compression with mask module. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

[Wang et al., 2021c] Wang, S., Zhang, H., Xu, K., Lin, X., Jana, S., Hsieh, C.-J., and Kolter, J. Z. (2021c). Beta-CROWN : Efficient Bound Propagation with Per-neuron Split Constraints for Neural Network Robustness Verification. In *Advances in Neural Information Processing Systems*, volume 34, pages 29909–29921. Curran Associates, Inc.

[Wang et al., 2018c] Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., and Catanzaro, B. (2018c). Video-to-video synthesis. In *Neural Information Processing Systems*.

[Wang et al., 2021d] Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. (2021d). Pyramid Vision Transformer : A Versatile Backbone for Dense Prediction without Convolutions. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 548–558. ISSN : 2380-7504.

[Wang et al., 2024c] Wang, X., Zhu, Z., Huang, G., Chen, X., Zhu, J., and Lu, J. (2024c). Drivedreamer : Towards real-world-driven world models for autonomous driving. *The 18th European Conference on Computer Vision ECCV 2024*.

[Wang et al., 2024d] Wang, Y., Liang, F., Liang, J., and Fu, H. (2024d). S2lic : Learned image compression with the swinv2 block, adaptive channel-wise and global-inter attention context. *arXiv preprint arXiv :2403.14471*.

[Wang et al., 2022c] Wang, Y., Zhang, X., Yang, T., and Sun, J. (2022c). Anchor detr : Query design for transformer-based detector. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 2567–2575.

[Weng et al., 2018a] Weng, L., Zhang, H., Chen, H., Song, Z., Hsieh, C.-J., Daniel, L., Boning, D., and Dhillon, I. (2018a). Towards Fast Computation of Certified Robustness for ReLU Networks. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5276–5285. PMLR. ISSN : 2640-3498.

[Weng et al., 2018b] Weng, T.-W., Zhang, H., Chen, P.-Y., Lozano, A., Hsieh, C.-J., and Daniel, L. (2018b). ON EXTENSIONS OF CLEVER : A NEURAL NETWORK ROBUSTNESS EVALUATION ALGORITHM. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1159–1163.

[Weng et al., 2018c] Weng, T.-W., Zhang, H., Chen, P.-Y., Yi, J., Su, D., Gao, Y., Hsieh, C.-J., and Daniel, L. (2018c). Evaluating the Robustness of Neural Networks : An Extreme Value Theory Approach. *ArXiv*.

[Westerski and Teck, 2023] Westerski, A. and Teck, F. W. (2023). Synthetic Data for Object Detection with Neural Networks : State of the Art Survey of Domain Randomisation Techniques. *ACM Trans. Multimedia Comput. Commun. Appl.*, page 3637064.

[Wikipedia, 2024] Wikipedia (2024). Vapnik–chervonenkis dimension. https://en.wikipedia.org/wiki/Vapnik%E2%80%93Chervonenkis_dimension.

[Wong and Kolter, 2018] Wong, E. and Kolter, Z. (2018). Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5286–5295. PMLR. ISSN : 2640-3498.

[Wu et al., 2024a] Wu, C., Wu, Q., Wei, H., Chen, S., Wang, L., Ngan, K. N., Meng, F., and Li, H. (2024a). On the adversarial robustness of learning-based image compression against rate-distortion attacks. *arXiv preprint arXiv :2405.07717*.

[Wu et al., 2024b] Wu, H., Isac, O., Zeljić, A., Tagomori, T., Daggitt, M., Kokke, W., Refaeli, I., Amir, G., Julian, K., Bassan, S., Huang, P., Lahav, O., Wu, M., Zhang, M., Komendantskaya, E., Katz, G., and Barrett, C. (2024b). Marabou 2.0 : A Versatile Formal Analyzer of Neural Networks. In Gurfinkel, A. and Ganesh, V., editors, *Computer Aided Verification*, pages 249–264, Cham. Springer Nature Switzerland.

[Xiao et al., 2019] Xiao, K. Y., Tjeng, V., Shafiullah, N. M. M., and Madry, A. (2019). Training for faster adversarial robustness verification via inducing reLU stability. In *International Conference on Learning Representations*.

[Xie et al., 2023] Xie, E., Yao, L., Shi, H., Liu, Z., Zhou, D., Liu, Z., Li, J., and Li, Z. (2023). DiffFit : Unlocking Transferability of Large Diffusion Models via Simple Parameter-Efficient Fine-Tuning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4207–4216. ISSN : 2380-7504.

[Xie et al., 2021] Xie, Y., Cheng, K. L., and Chen, Q. (2021). Enhanced Invertible Encoding for Learned Image Compression. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, pages 162–170, New York, NY, USA. Association for Computing Machinery.

[Xu et al., 2020] Xu, K., Zhang, H., Wang, S., Wang, Y., Jana, S., Lin, X., and Hsieh, C.-J. (2020). Fast and Complete : Enabling Complete Neural Network Verification with Rapid and Massively Parallel Incomplete Verifiers.

[Xu et al., 2023a] Xu, S., Li, Y., Lin, M., Gao, P., Guo, G., Lü, J., and Zhang, B. (2023a). Q-detr : An efficient low-bit quantized detection transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3842–3851.

[Xu et al., 2022] Xu, X., Jiang, Y., Chen, W., Huang, Y., Zhang, Y., and Sun, X. (2022). Damo-yolo : A report on real-time object detection design. *arXiv preprint arXiv :2211.15444*.

[Xu et al., 2023b] Xu, Y., Guo, W., and Wei, Z. (2023b). Conformal risk control for ordinal classification. In Evans, R. J. and Shpitser, I., editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 2346–2355. PMLR.

[Xu et al., 2021] Xu, Y., Zhang, Q., Zhang, J., and Tao, D. (2021). Vitae : Vision transformer advanced by exploring intrinsic inductive bias. *Advances in neural information processing systems*, 34 :28522–28535.

[Xue et al., 2023a] Xue, D., Dong, C., Ye, F., Chen, H., Kang, B., Li, L., and Liu, D. (2023a). iWavePro : An improved framework for iWave++. In *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. ISSN : 2158-1525.

[Xue et al., 2023b] Xue, H., Araujo, A., Hu, B., and Chen, Y. (2023b). Diffusion-based adversarial sample generation for improved stealthiness and controllability. *Advances in Neural Information Processing Systems*, 36 :2894–2921.

[Xue et al., 2019] Xue, T., Chen, B., Wu, J., Wei, D., and Freeman, W. T. (2019). Video Enhancement with Task-Oriented Flow. *Int J Comput Vis*, 127(8) :1106–1125.

[Yang et al., 2021] Yang, F., Herranz, L., Cheng, Y., and Mozerov, M. G. (2021). Slimmable Compressive Autoencoders for Practical Neural Image Compression. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4996–5005. ISSN : 2575-7075.

[Yang and Wang, 2023] Yang, M. and Wang, Z. (2023). Image synthesis under limited data : A survey and taxonomy.

[Yang et al., 2024] Yang, M., Yang, F., Murn, L., Blanch, M. G., Sock, J., Wan, S., Yang, F., and Herranz, L. (2024). Task-Switchable Pre-Processor for Image Compression for Multiple Machine Vision Tasks. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(7) :6416–6429. Conference Name : IEEE Transactions on Circuits and Systems for Video Technology.

[Yang and Mandt, 2023] Yang, R. and Mandt, S. (2023). Lossy image compression with conditional diffusion models. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems*, volume 36, pages 64971–64995. Curran Associates, Inc.

[Ye et al., 2023] Ye, M., Ke, L., Li, S., Tai, Y.-W., Tang, C.-K., Danelljan, M., and Yu, F. (2023). Cascade-detr : delving into high-quality universal object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6704–6714.

[Yilmaz et al., 2023] Yilmaz, C., MARAŞ, B., Yilmaz, G., Ceylan, G., HAMAMCIOĞLU, Ö., Arica, N., and ERTÜZÜN, A. B. (2023). Enhancing object detection algorithms by synthetic aerial images. In *2023 31st Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE.

[Yin et al., 2022] Yin, S., Li, C., Bao, Y., Liang, Y., Meng, F., and Liu, W. (2022). Universal Efficient Variable-Rate Neural Image Compression. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2025–2029. ISSN : 2379-190X.

[Yu et al., 2020] Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., and Darrell, T. (2020). BDD100k : A diverse driving dataset for heterogeneous multitask learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[Yu et al., 2022] Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., Hutchinson, B., Han, W., Parekh, Z., Li, X., Zhang, H., Baldridge, J., and Wu, Y. (2022). Scaling Autoregressive Models for Content-Rich Text-to-Image Generation. *Trans. Mach. Learn. Res.* Publisher : arXiv Version Number : 1.

[Yu et al., 2023] Yu, Y., Wang, Y., Yang, W., Lu, S., Tan, Y.-P., and Kot, A. C. (2023). Backdoor Attacks Against Deep Image Compression via Adaptive Frequency Trigger. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12250–12259. ISSN : 2575-7075.

[Zang et al., 2022] Zang, Y., Li, W., Zhou, K., Huang, C., and Loy, C. C. (2022). Open-vocabulary detr with conditional matching. In *European Conference on Computer Vision*, pages 106–122. Springer.

[Zeng et al., 2024] Zeng, Y., Chen, Y., Yang, X., Li, Q., and Yan, J. (2024). Ars-detr : Aspect ratio-sensitive detection transformer for aerial oriented object detection. *IEEE Transactions on Geoscience and Remote Sensing*, 62 :1–15.

[Zhai et al., 2019] Zhai, R., Dan, C., He, D., Zhang, H., Gong, B., Ravikumar, P., Hsieh, C.-J., and Wang, L. (2019). MACER : Attack-free and Scalable Robust Training via Maximizing Certified Radius.

[Zhang et al., 2021a] Zhang, B., Cai, T., Lu, Z., He, D., and Wang, L. (2021a). Towards certifying l-infinity robustness using neural networks with l-inf-dist neurons. In Meila, M. and Zhang, T., editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12368–12379. PMLR.

[Zhang et al., 2022a] Zhang, G., Luo, Z., Cui, K., Lu, S., and Xing, E. P. (2022a). Meta-detr : Image-level few-shot detection with inter-class correlation exploitation. *IEEE transactions on pattern analysis and machine intelligence*, 45(11) :12832–12843.

[Zhang et al., 2020a] Zhang, H., Chen, H., Xiao, C., Gowal, S., Stanforth, R., Li, B., Boning, D., and Hsieh, C.-J. (2020a). Towards stable and efficient training of verifiably robust neural networks. In *International Conference on Learning Representations*.

[Zhang et al., 2019] Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A. (2019). Self-Attention Generative Adversarial Networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 7354–7363. PMLR. ISSN : 2640-3498.

[Zhang et al., 2023a] Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L., and Shum, H.-Y. (2023a). DINO : DETR with improved denoising anchor boxes for end-to-end object detection. In *The Eleventh International Conference on Learning Representations*.

[Zhang et al., 2022b] Zhang, H., Wang, S., Xu, K., Li, L., Li, B., Jana, S., Hsieh, C.-J., and Kolter, J. Z. (2022b). General cutting planes for bound-propagation-based neural network verification. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K., editors, *Advances in Neural Information Processing Systems*.

[Zhang et al., 2018a] Zhang, H., Weng, T.-W., Chen, P.-Y., Hsieh, C.-J., and Daniel, L. (2018a). Efficient Neural Network Robustness Certification with General Activation Functions. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[Zhang et al., 2023b] Zhang, J., Lin, X., Zhang, W., Wang, K., Tan, X., Han, J., Ding, E., Wang, J., and Li, G. (2023b). Semi-detr : Semi-supervised object detection with detection transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 23809–23818.

[Zhang et al., 2023c] Zhang, L., Rao, A., and Agrawala, M. (2023c). Adding Conditional Control to Text-to-Image Diffusion Models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3813–3824. ISSN : 2380-7504.

[Zhang et al., 2023d] Zhang, Q., Tao, M., and Chen, Y. (2023d). gDDIM : Generalized denoising diffusion implicit models. In *The Eleventh International Conference on Learning Representations*.

[Zhang et al., 2023e] Zhang, Q., Xu, Y., Zhang, J., and Tao, D. (2023e). Vitaev2 : Vision transformer advanced by exploring inductive bias for image recognition and beyond. *International Journal of Computer Vision*, 131(5) :1141–1162.

[Zhang et al., 2018b] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018b). The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595. IEEE.

[Zhang et al., 2020b] Zhang, R., Li, W., Wang, P., Guan, C., Fang, J., Song, Y., Yu, J., Chen, B., Xu, W., and Yang, R. (2020b). AutoRemover : Automatic Object Removal for Autonomous Driving Videos. *AAAI*, 34(07) :12853–12861.

[Zhang and Wu, 2023] Zhang, X. and Wu, X. (2023). LVQAC : Lattice Vector Quantization Coupled with Spatially Adaptive Companding for Efficient Learned Image Compression. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10239–10248. ISSN : 2575-7075.

[Zhang and Lin, 2024] Zhang, Y. and Lin, K. (2024). End-to-end optimized image compression with the frequency-oriented transform. *Machine Vision and Applications*, 35(2) :27.

[Zhang et al., 2023f] Zhang, Z., Chen, B., Lin, H., Lin, J., Wang, X., and Zhao, T. (2023f). ELFIC : A Learning-based Flexible Image Codec with Rate-Distortion-Complexity Optimization. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pages 9252–9261, New York, NY, USA. Association for Computing Machinery.

[Zhang et al., 2021b] Zhang, Z., Lu, X., Cao, G., Yang, Y., Jiao, L., and Liu, F. (2021b). ViT-YOLO :Transformer-Based YOLO for Object Detection. In *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2799–2808. ISSN : 2473-9944.

[Zhao et al., 2024a] Zhao, G., Wang, X., Zhu, Z., Chen, X., Huang, G., Bao, X., and Wang, X. (2024a). Drivedreamer-2 : Llm-enhanced world models for diverse driving video generation. *arXiv preprint arXiv :2403.06845*.

[Zhao et al., 2024b] Zhao, Y., Lv, W., Xu, S., Wei, J., Wang, G., Dang, Q., Liu, Y., and Chen, J. (2024b). Detrs beat yolos on real-time object detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16965–16974.

[Zhong et al., 2021] Zhong, Y., Ta, Q.-T., Luo, T., Zhang, F., and Khoo, S.-C. (2021). Scalable and Modular Robustness Analysis of Deep Neural Networks. In Oh, H., editor, *Programming Languages and Systems*, pages 3–22, Cham. Springer International Publishing.

[Zhou et al., 2023] Zhou, H., Xiong, F., and Chen, H. (2023). A Comprehensive Survey of Recommender Systems Based on Deep Learning. *Applied Sciences*, 13(20) :11378. Number : 20 Publisher : Multidisciplinary Digital Publishing Institute.

[Zhou et al., 2018] Zhou, L., Cai, C., Gao, Y., Su, S., and Wu, J. (2018). Variational autoencoder for low bit-rate image compression. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2617–2620.

[Zhou et al., 2019] Zhou, X., Wang, D., and Krähenbühl, P. (2019). Objects as points. In *arXiv preprint arXiv :1904.07850*.

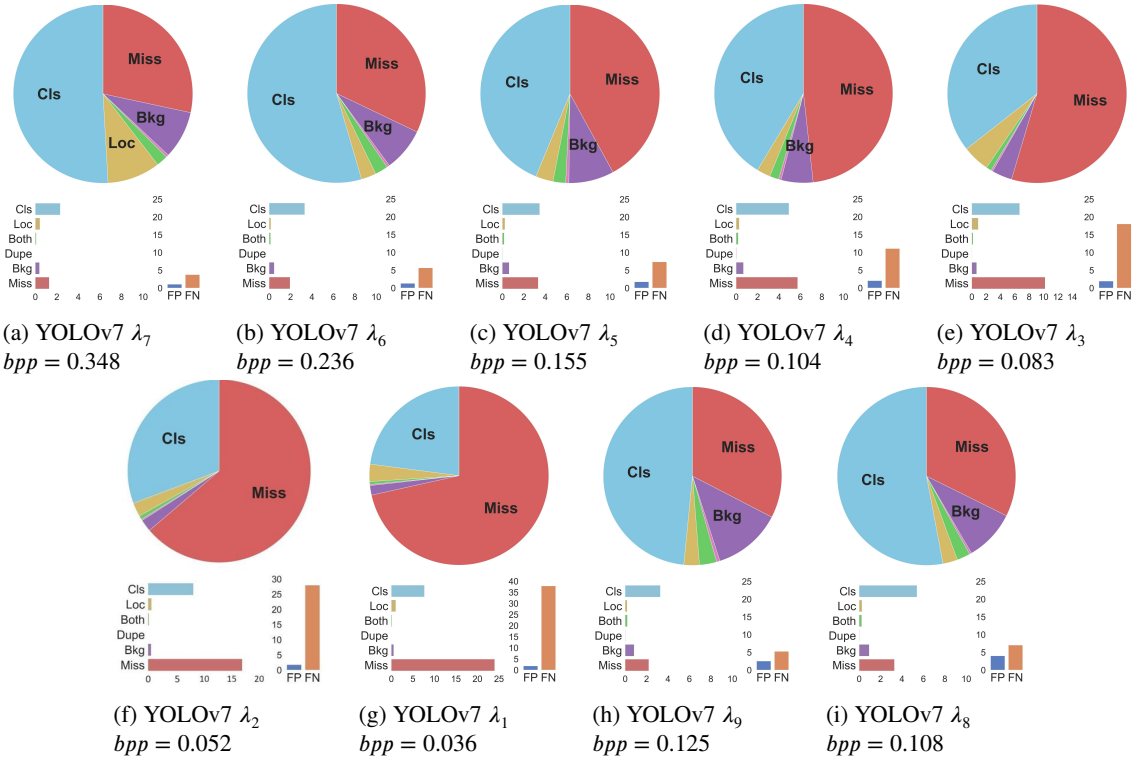
[Zhu et al., 2017] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251. ISSN : 2380-7504.

[Zhu et al., 2021] Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. (2021). Deformable {detr} : Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*.

[Zhu et al., 2022] Zhu, Y., Yang, Y., and Cohen, T. (2022). Transformer-based transform coding. In *International Conference on Learning Representations*.

[Zou et al., 2022] Zou, R., Song, C., and Zhang, Z. (2022). The devil is in the details : Window-based attention for image compression. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17471–17480. ISSN : 2575-7075.

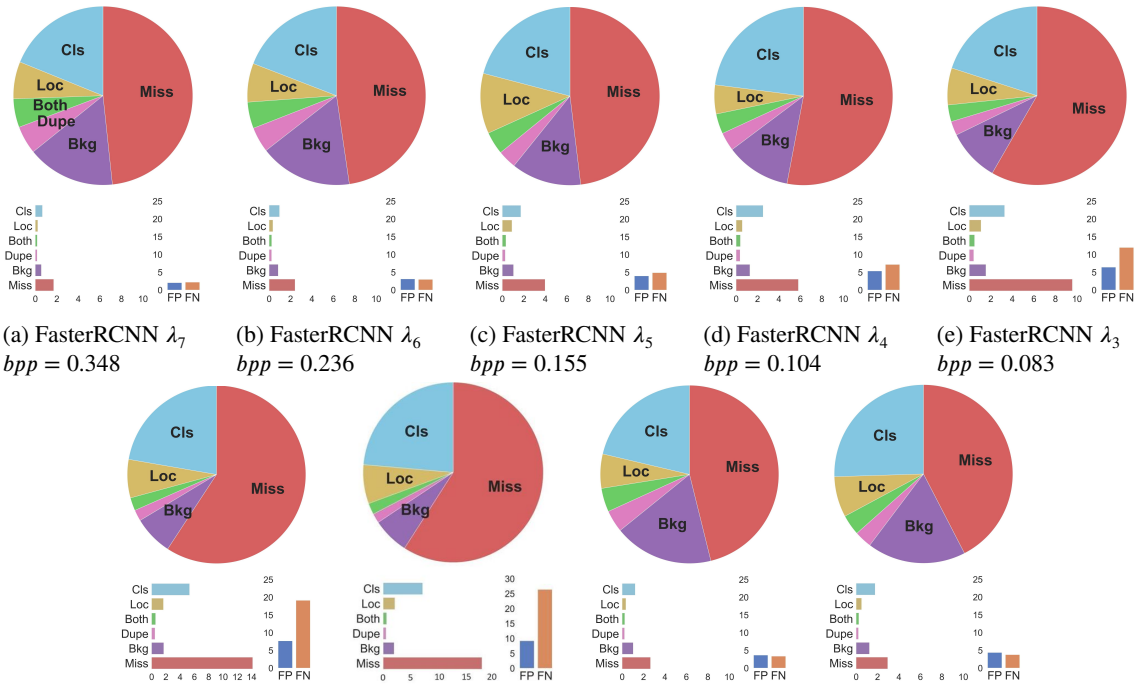
Analyse de l'impact de la compression sur la détection à l'aide de la méthode TIDE



(a) YOLOv7 λ_7 $bpp = 0.348$ (b) YOLOv7 λ_6 $bpp = 0.236$ (c) YOLOv7 λ_5 $bpp = 0.155$ (d) YOLOv7 λ_4 $bpp = 0.104$ (e) YOLOv7 λ_3 $bpp = 0.083$

(f) YOLOv7 λ_2 $bpp = 0.052$ (g) YOLOv7 λ_1 $bpp = 0.036$ (h) YOLOv7 λ_9 $bpp = 0.125$ (i) YOLOv7 λ_8 $bpp = 0.108$

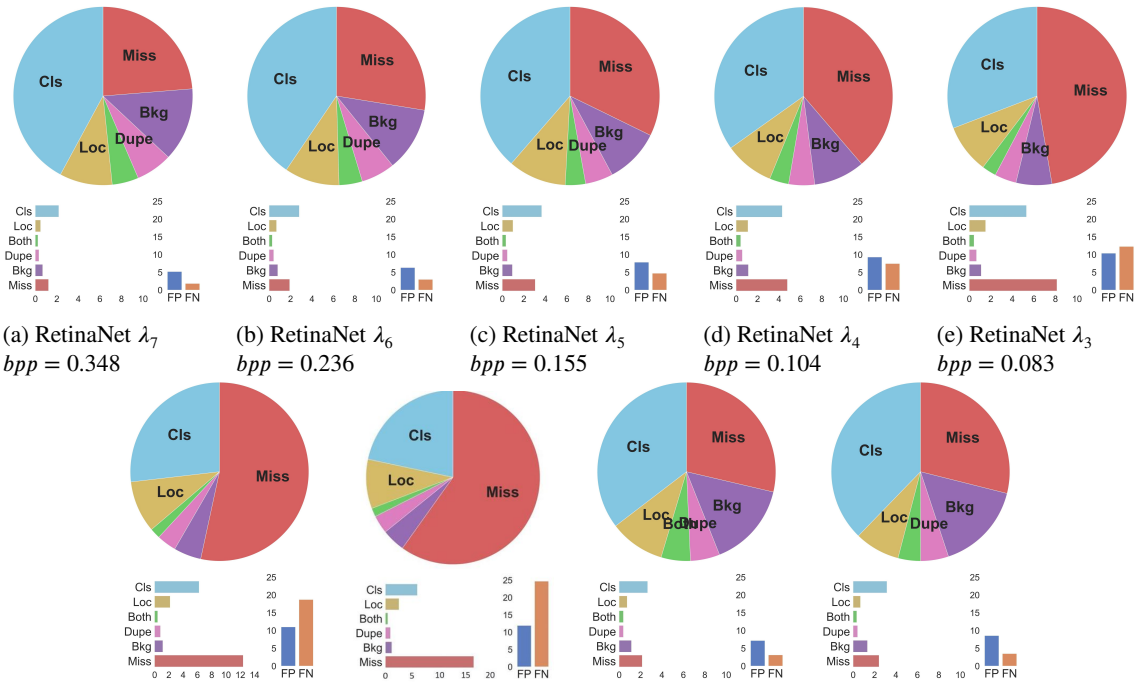
FIGURE 8.1 – Évaluation TIDE sur D_{test} avec STF(mse) et YOLOv7.



(a) FasterRCNN λ_7 $bpp = 0.348$ (b) FasterRCNN λ_6 $bpp = 0.236$ (c) FasterRCNN λ_5 $bpp = 0.155$ (d) FasterRCNN λ_4 $bpp = 0.104$ (e) FasterRCNN λ_3 $bpp = 0.083$

(f) FasterRCNN λ_2 $bpp = 0.052$ (g) FasterRCNN λ_1 $bpp = 0.036$ (h) FasterRCNN λ_9 $bpp = 0.125$ (i) FasterRCNN λ_8 $bpp = 0.108$

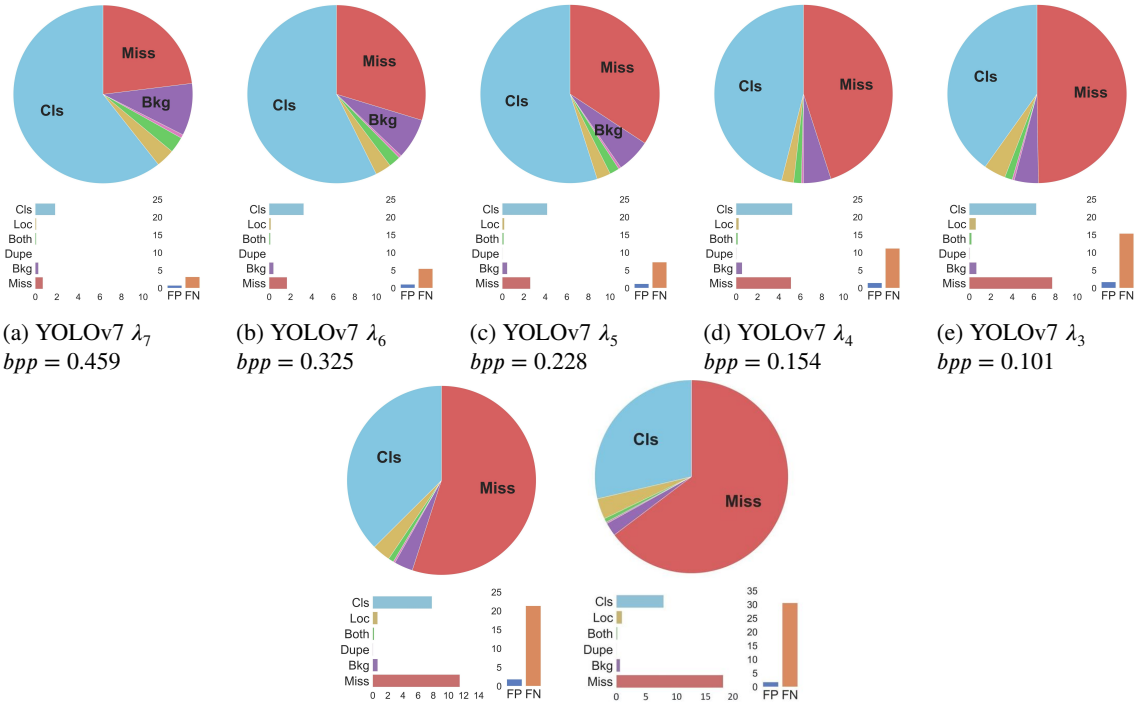
FIGURE 8.2 – Évaluation TIDE sur D_{test} avec STF(mse) et FasterRCNN.



(a) RetinaNet λ_7 $bpp = 0.348$ (b) RetinaNet λ_6 $bpp = 0.236$ (c) RetinaNet λ_5 $bpp = 0.155$ (d) RetinaNet λ_4 $bpp = 0.104$ (e) RetinaNet λ_3 $bpp = 0.083$

(f) RetinaNet λ_2 $bpp = 0.052$ (g) RetinaNet λ_1 $bpp = 0.036$ (h) RetinaNet λ_9 $bpp = 0.125$ (i) RetinaNet λ_8 $bpp = 0.108$

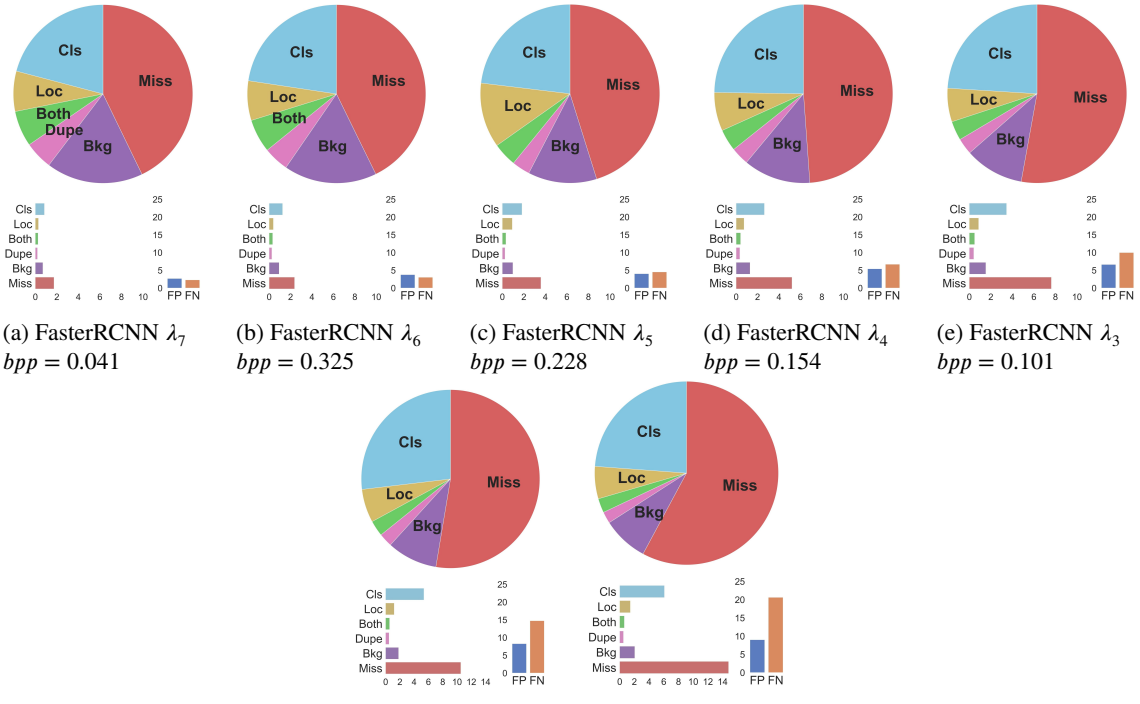
FIGURE 8.3 – Évaluation TIDE sur D_{test} avec STF(mse) et RetinaNet.



(a) YOLOv7 λ_7 $bpp = 0.459$ (b) YOLOv7 λ_6 $bpp = 0.325$ (c) YOLOv7 λ_5 $bpp = 0.228$ (d) YOLOv7 λ_4 $bpp = 0.154$ (e) YOLOv7 λ_3 $bpp = 0.101$

(f) YOLOv7 λ_2 $bpp = 0.067$ (g) YOLOv7 λ_1 $bpp = 0.041$

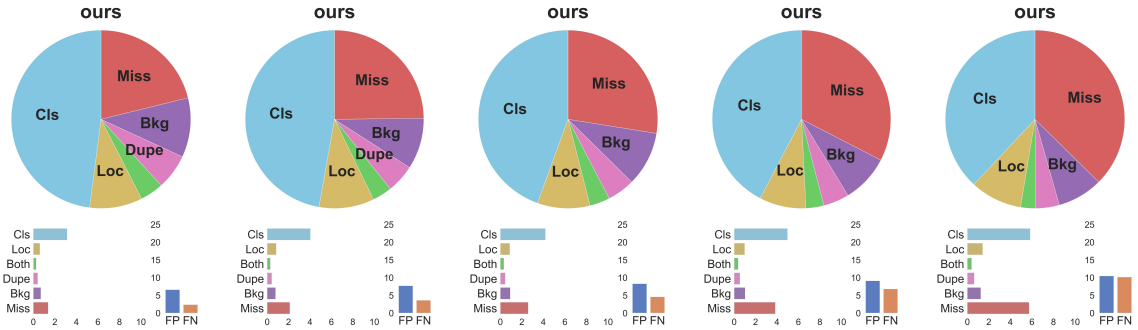
FIGURE 8.4 – Évaluation TIDE sur D_{test} avec STF(ms-ssim) et YOLOv7.



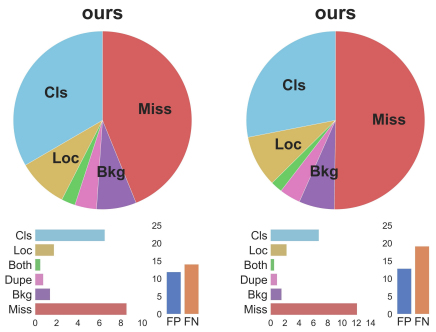
(a) FasterRCNN λ_7 $bpp = 0.041$ (b) FasterRCNN λ_6 $bpp = 0.325$ (c) FasterRCNN λ_5 $bpp = 0.228$ (d) FasterRCNN λ_4 $bpp = 0.154$ (e) FasterRCNN λ_3 $bpp = 0.101$

(f) FasterRCNN λ_2 $bpp = 0.067$ (g) FasterRCNN λ_1 $bpp = 0.041$

FIGURE 8.5 – Évaluation TIDE sur D_{test} avec STF(ms-ssim) et FasterRCNN.

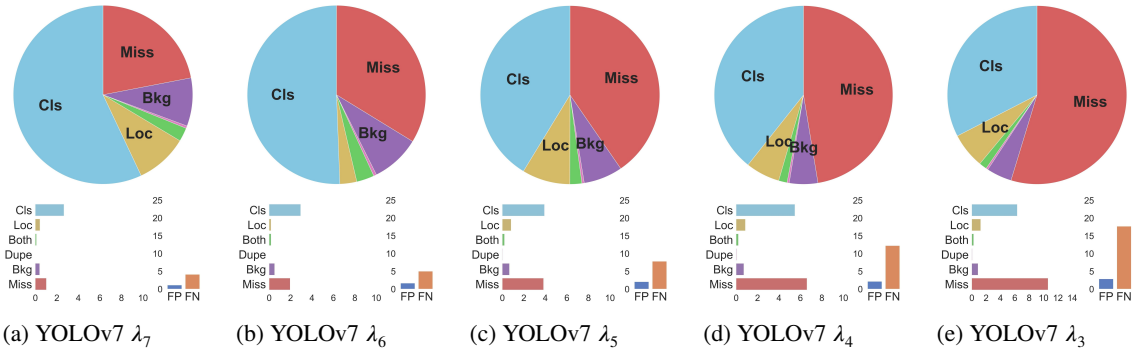


(a) RetinaNet λ_7 $bpp = 0.459$ (b) RetinaNet λ_6 $bpp = 0.325$ (c) RetinaNet λ_5 $bpp = 0.228$ (d) RetinaNet λ_4 $bpp = 0.154$ (e) RetinaNet λ_3 $bpp = 0.101$

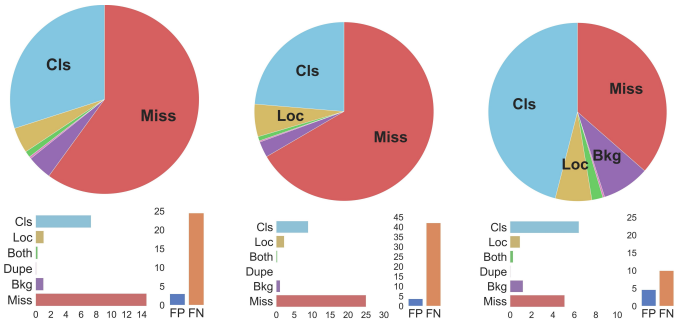


(f) RetinaNet λ_2 $bpp = 0.067$ (g) RetinaNet λ_1 $bpp = 0.041$

FIGURE 8.6 – Évaluation TIDE sur D_{test} avec STF(ms-ssim) et RetinaNet.

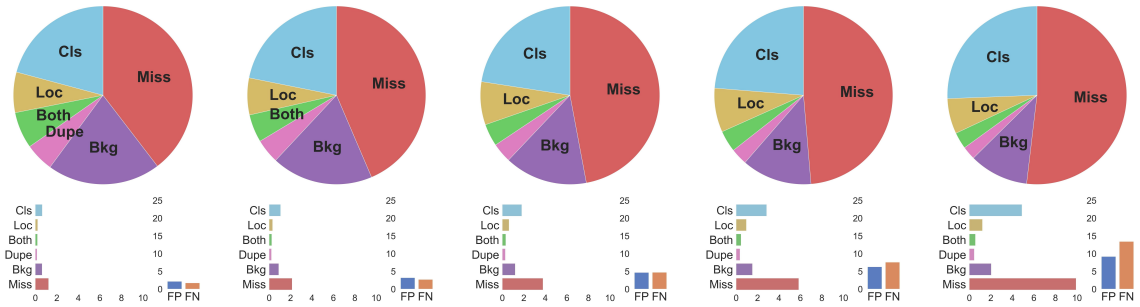


(a) YOLOv7 λ_7 $bpp = 0.331$ (b) YOLOv7 λ_6 $bpp = 0.241$ (c) YOLOv7 λ_5 $bpp = 0.169$ (d) YOLOv7 λ_4 $bpp = 0.108$ (e) YOLOv7 λ_3 $bpp = 0.073$

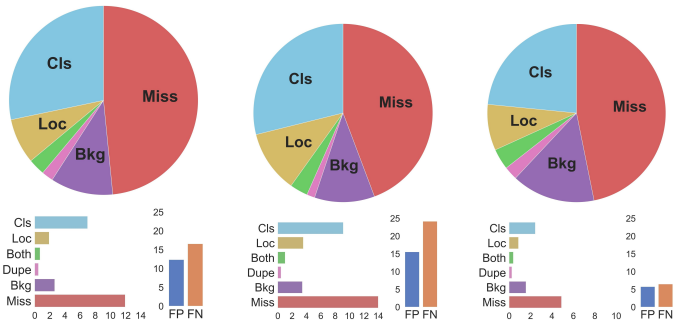


(f) YOLOv7 λ_2 $bpp = 0.051$ (g) YOLOv7 λ_1 $bpp = 0.036$ (h) YOLOv7 λ_8 $bpp = 0.067$

FIGURE 8.7 – Évaluation TIDE sur D_{test} avec QRes-VAE(mse) et YOLOv7.

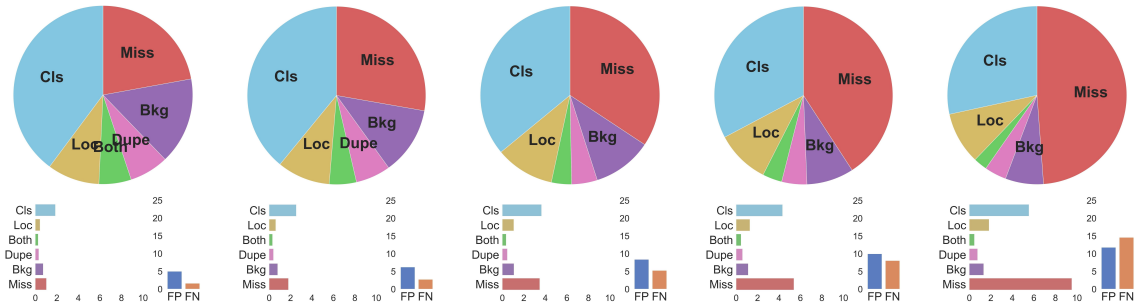


(a) FasterRCNN λ_7 $bpp = 0.331$ (b) FasterRCNN λ_6 $bpp = 0.241$ (c) FasterRCNN λ_5 $bpp = 0.169$ (d) FasterRCNN λ_4 $bpp = 0.108$ (e) FasterRCNN λ_3 $bpp = 0.073$

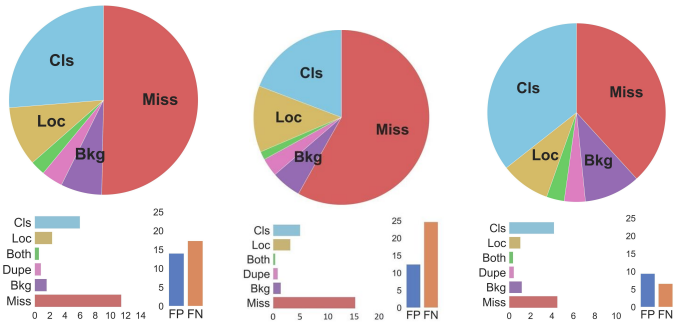


(f) FasterRCNN λ_2 $bpp = 0.051$ (g) FasterRCNN λ_1 $bpp = 0.036$ (h) FasterRCNN λ_8 $bpp = 0.067$

FIGURE 8.8 – Évaluation TIDE sur D_{test} avec QRes-VAE(mse) et FasterRCNN.

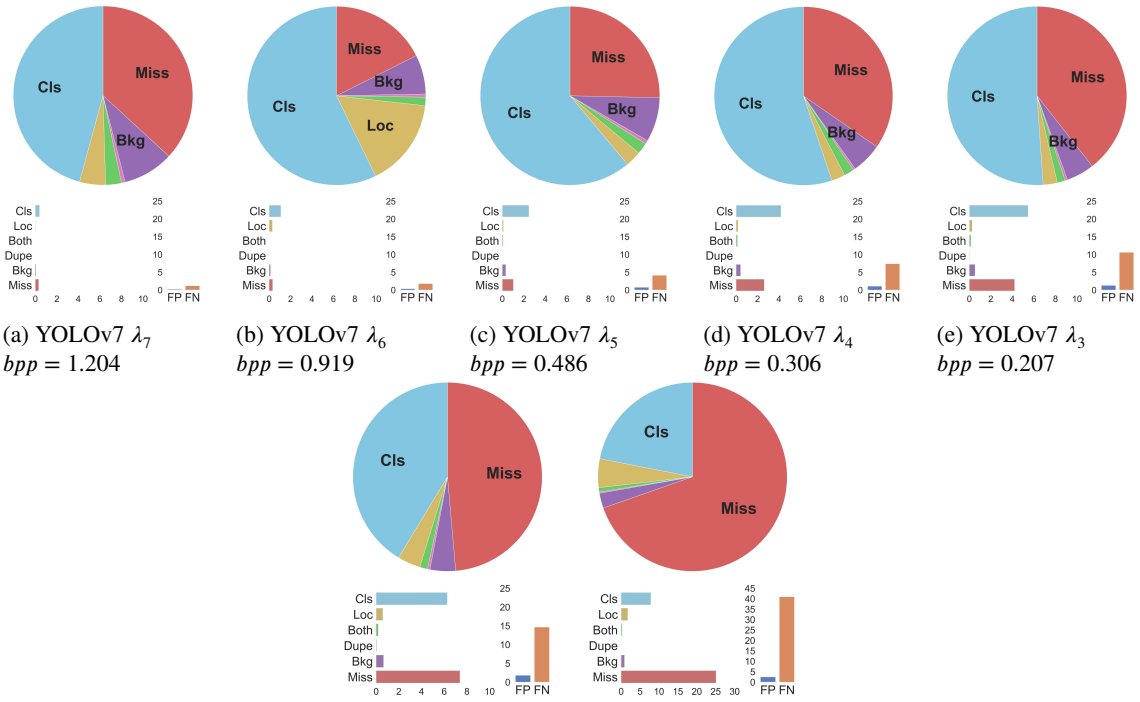


(a) RetinaNet λ_7 $bpp = 0.331$ (b) RetinaNet λ_6 $bpp = 0.241$ (c) RetinaNet λ_5 $bpp = 0.169$ (d) RetinaNet λ_4 $bpp = 0.108$ (e) RetinaNet λ_3 $bpp = 0.073$



(f) RetinaNet λ_2 $bpp = 0.051$ (g) RetinaNet λ_1 $bpp = 0.036$ (h) RetinaNet λ_8 $bpp = 0.067$

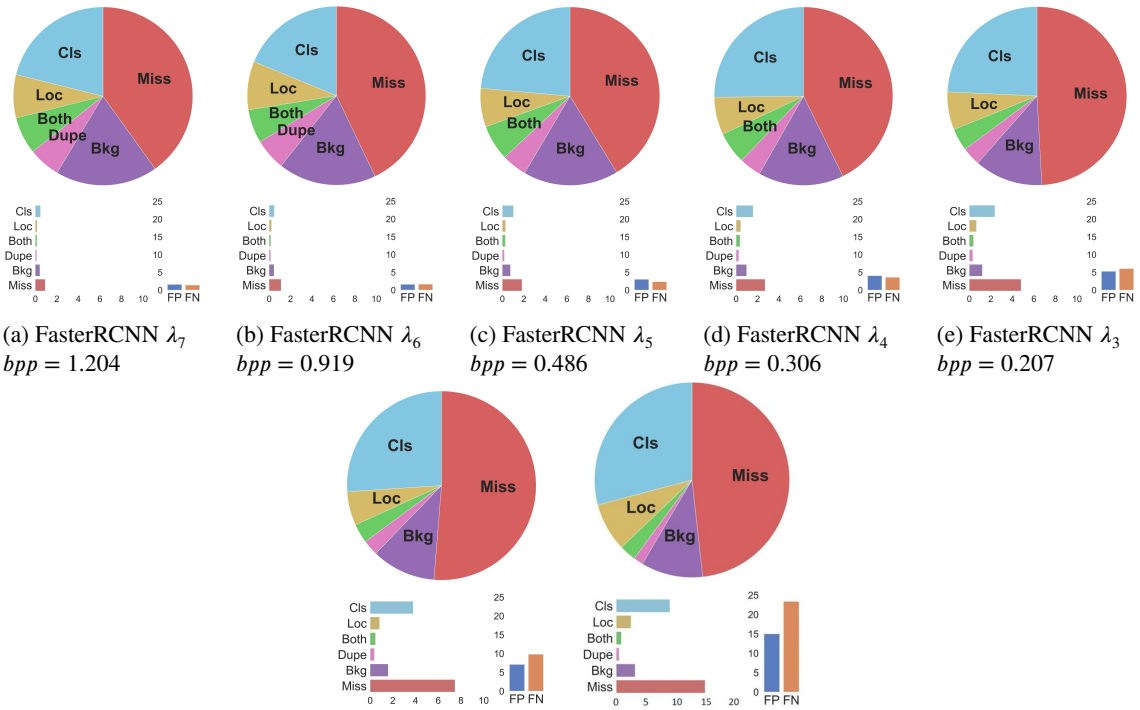
FIGURE 8.9 – Évaluation TIDE sur D_{test} avec QRes-VAE(mse) et RetinaNet.



(a) YOLOv7 λ_7 $bpp = 1.204$ (b) YOLOv7 λ_6 $bpp = 0.919$ (c) YOLOv7 λ_5 $bpp = 0.486$ (d) YOLOv7 λ_4 $bpp = 0.306$ (e) YOLOv7 λ_3 $bpp = 0.207$

(f) YOLOv7 λ_2 $bpp = 0.132$ (g) YOLOv7 λ_1 $bpp = 0.035$

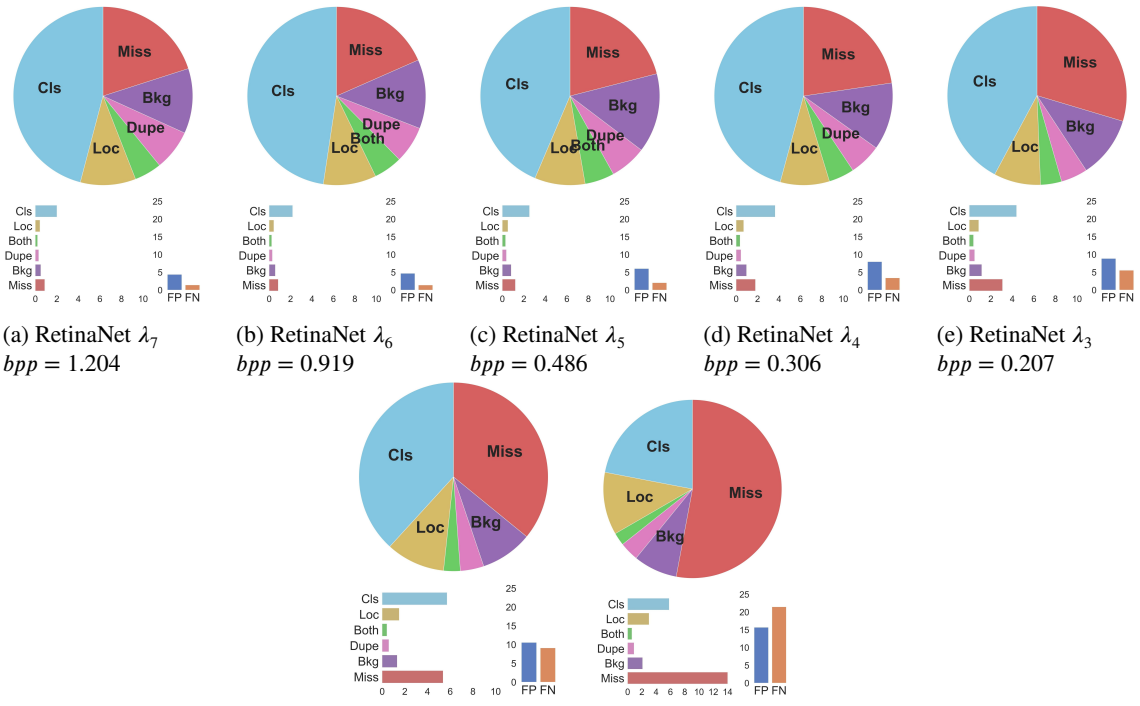
FIGURE 8.10 – Évaluation TIDE sur D_{test} avec QRes-VAE(ms-ssim) et YOLOv7.



(a) FasterRCNN λ_7 $bpp = 1.204$ (b) FasterRCNN λ_6 $bpp = 0.919$ (c) FasterRCNN λ_5 $bpp = 0.486$ (d) FasterRCNN λ_4 $bpp = 0.306$ (e) FasterRCNN λ_3 $bpp = 0.207$

(f) FasterRCNN λ_2 $bpp = 0.132$ (g) FasterRCNN λ_1 $bpp = 0.035$

FIGURE 8.11 – Évaluation TIDE sur D_{test} avec QRes-VAE(ms-ssim) et FasterRCNN.



(a) RetinaNet λ_7 $bpp = 1.204$ (b) RetinaNet λ_6 $bpp = 0.919$ (c) RetinaNet λ_5 $bpp = 0.486$ (d) RetinaNet λ_4 $bpp = 0.306$ (e) RetinaNet λ_3 $bpp = 0.207$

(f) RetinaNet λ_2 $bpp = 0.132$ (g) RetinaNet λ_1 $bpp = 0.035$

FIGURE 8.12 – Évaluation TIDE sur D_{test} avec QRes-VAE(ms-ssim) et RetinaNet.

Récapitulatif des modèles entraînés

TABLE 8.1 – Récapitulatif des modèles de Compression.

Architecture	Datasets d'entrainement	Nom	Hyperparamètres λ	
			Agnostique	ROI
STF	D_{train} D_{valid}	stf(mse)	$\lambda_1 = 0.0009$, $\lambda_2 = 0.0018$, $\lambda_3 = 0.0035$, $\lambda_4 = 0.0067$, $\lambda_5 = 0.013$, $\lambda_6 = 0.025$, $\lambda_7 = 0.0483$	$\lambda_8 = \{\lambda^- = 1.25 \cdot 10^{-5}, \lambda^+ = 0.08\}$, $\lambda_9 = \{\lambda^- = 10^{-3}, \lambda^+ = 0.08\}$
	$D_{train_aug_front}$ D_{valid}		n/a	$\lambda_{10} = \{\lambda^- = 1.25 \cdot 10^{-5}, \lambda^+ = 0.08\}$
	D_{train_aug} D_{valid}		n/a	$\lambda_{11} = \{\lambda^- = 1.25 \cdot 10^{-5}, \lambda^+ = 0.08\}$
	D_{train} D_{valid}	stf(ms-ssim)	$\lambda_1 = 1.2$, $\lambda_2 = 2.4$, $\lambda_3 = 4.58$, $\lambda_4 = 8.73$, $\lambda_5 = 16.64$, $\lambda_6 = 31.73$, $\lambda_7 = 60.5$	n/a
QRes-VAE	D_{train} D_{valid}	qresvae(mse)	$\lambda_1 = 4$, $\lambda_2 = 8$, $\lambda_3 = 16$, $\lambda_4 = 32$, $\lambda_5 = 64$, $\lambda_6 = 128$, $\lambda_7 = 256$	$\lambda_8 = \{\lambda^- = 2, \lambda^+ = 128\}$
		qresvae(ms-ssim)	$\lambda_1 = 0.1$, $\lambda_2 = 1$, $\lambda_3 = 2$, $\lambda_4 = 4$, $\lambda_5 = 8$, $\lambda_6 = 32$, $\lambda_7 = 64$	n/a

TABLE 8.2 – Récapitulatif des modèles de détection et de leurs performances sur D_{test} : APv@.5, APp@.5 et APc@.5 correspondant respectivement aux AP@.5 pour les classes véhicules, piétons et cyclistes.

Architecture	Datasets d'entrainement	Nom	Évaluation sur D_{test}			
			APv@.5	APp@.5	APc@.5	mAP@.5
FasterRCNN	D_{train} D_{valid}	fasterrcnn	0.9434 ± 0.0003	0.847 ± 0.004	0.61 ± 0.01	0.802 ± 0.004
RetinaNet		retinanet	0.936 ± 0.004	0.878 ± 0.001	0.65 ± 0.013	0.821 ± 0.0047
YOLOv7		yolov7	0.9298 ± 0.0001	0.834 ± 0.007	0.60 ± 0.01	0.788 ± 0.004
		$\{\text{yolov7}_i\}_{i=1:6}$	n/a	n/a	n/a	n/a
	$D_{train_aug_front}$ D_{valid}	yolo_aug_front	0.9302 ± 0.0001	0.846 ± 0.001	0.62 ± 0.01	0.7997 ± 0.0035
	D_{train_aug} D_{valid}	yolo_aug	0.936 ± 0.006	0.855 ± 0.008	0.63 ± 0.01	0.806 ± 0.005
	$DU25_{train}$ D_{valid}	yolo_025unity	0.9302 ± 0.0001	0.8462 ± 0.0009	0.62 ± 0.01	0.797 ± 0.003
	$DU33_{train}$ D_{valid}	yolo_033unity	0.9297 ± 0.0001	0.845 ± 0.007	0.6 ± 0.01	0.792 ± 0.004
	$DU50_{train}$ D_{valid}	yolo_05unity	0.930 ± 0.008	0.849 ± 0.005	0.61 ± 0.01	0.797 ± 0.004
	U_{train} U_{valid}	yolo_1unity	0.7172 ± 0.0006	0.565 ± 0.003	0.458 ± 0.02	0.580 ± 0.008
	$D_{train} + \text{STF}^{\lambda_7}_{mse}(D_{train})$ $D_{valid} + \text{STF}^{\lambda_7}_{mse}(D_{valid})$	yolostfmseq7	0.9299 ± 0.0001	0.830 ± 0.006	0.61 ± 0.01	0.792 ± 0.004
	$D_{train} + \text{STF}^{\lambda_3}_{mse}(D_{train})$ $D_{valid} + \text{STF}^{\lambda_3}_{mse}(D_{valid})$	yolostfmseq3	0.9299 ± 0.0001	0.830 ± 0.006	0.61 ± 0.01	0.792 ± 0.004

TABLE 8.3 – Récapitulatif des modèles de diffusion utilisés.

Architecture	Datasets d'entrainement	Nom
Stable Diffusion	$Diffusion_Front_{train}$ $Diffusion_Front_{valid}$	diff_front
	$Diffusion_All_{train}$ $Diffusion_All_{valid}$	diff_all

Table des figures

1.1 Le véhicule autonome (illustration générée par IA). . . . 15
1.2 Objectif du Field Monitoring (Source ISO 21448 [iso, 2022]). . . . 16
1.3 Fonctionnement complet du *Field Monitoring*. L'application étudiée dans cette thèse concerne les zones encadrées en rouge. . . . 17

2.1 Illustration de la vérification formelle d'un réseau de neurones avec X et Y représentant les plages d'entrées, et F la plage de sortie correspondante. . . . 22
2.2 Effet des fonctions non-linéaires comme ReLU sur la sortie des réseaux de neurones. . . . 23
2.3 Illustration des méthodes de vérification formelles. . . . 23
2.4 Exemples montrant (a) une extension d'intervalle où l'intervalle de sortie est très large, car elle ignore l'interdépendance des variables d'entrée, (b) l'utilisation de l'analyse d'intervalle symbolique pour suivre certaines des dépendances (source [Wang et al., 2018b]). . . . 25
2.5 Processus du Branch and Bound. . . . 27
2.6 Relaxations adaptative proposées par CROWN [Zhang et al., 2018a]. . . . 29
2.7 Illustration de la vérification par Interprétation Abstraite (source [eta,]). . . . 30
2.8 Exemple d'approximation zonotopique pour la fonction ReLU paramétrées par la pente λ, avec l_x et u_x les bornes du zonotope d'entrée $Z(c_x, G_x)$ (source [Singh et al., 2018]). . . . 31
2.9 Interpretation de la constante de Lipschitz (source [Weng et al., 2018c]). . . . 32
2.10 Évaluation du classifieur lissé pour une entrée x. À gauche : les régions de décision du classifieur de base f sont représentées par différentes couleurs. Les lignes pointillées indiquent les niveaux de la distribution $\mathcal{N}(x, \sigma^2 I)$. À droite : la distribution $f(\mathcal{N}(x, \sigma^2 I))$, $\underline{p_A}$ est une borne inférieure de la probabilité de la classe principale, et p_B est une borne supérieure de la probabilité de chaque autre classe. Ici, $g(x)$ est "bleu" (source [Cohen et al., 2019]). . . . 35
2.11 Illustration du paysage de prédiction des modèles de base et du modèle d'ensemble, où les couleurs représentent les classes (source [Horváth et al., 2022a]). . . . 35
2.12 Détermination de la VC-dim pour un classifieur linéaire dans un espace bidimensionnel, VC-dim= 3 (source [Wikipedia, 2024]). . . . 37
2.13 Trois exemples de plus en plus difficiles de la classe écureuil fauve et les ensembles de prédiction (c'est-à-dire, $C(X_{test})$) générés par la prédiction conforme (source [Angelopoulos and Bates, 2023]). 40
2.14 Processus pour créer l'ensemble $C(X_{test})$ (source [Angelopoulos and Bates, 2023]). . . . 40
2.15 Schéma de l'approche proposée par [Timans et al., 2024]. . . . 41

3.1 Schéma de la compression et décompression JPEG (source [Jamil et al., 2023]). . . . 48
3.2 Pour calculer une distance d_0 entre deux zones x et x_0, étant donné un réseau de neurones F (souvent VGG [Simonyan and Zisserman, 2015] ou AlexNet [Krizhevsky et al., 2012]), nous commençons par calculer les embeddings profonds, normaliser les activations dans la dimension des canaux, mettre à l'échelle chaque canal par le vecteur w, et prendre la distance l_2. Nous faisons ensuite une moyenne à travers la dimension spatiale et à travers toutes les couches (source [Zhang et al., 2018b]). . . . 51
3.3 Exemple de courbes taux-distorsion. Les méthodes incluent QRes-VAE [Duan et al., 2023], Wang 2022 Data-Depend.+ [Wang et al., 2022a], Xie 2021 Invertible Enc. [Xie et al., 2021], Qian 2021 Global Ref. [Qian et al., 2021], Minnen 2020 Channel-wise [Minnen and Singh, 2020], VVC intro (VTM 12.1), Cheng 2020 LIC [Cheng et al., 2020a], Minnen 2018 Joint AR & H [Minnen et al., 2018a], Minnen 2018 M & S Hyper [Ballé et al., 2018] et Balle 2017 Factorized [Ballé et al., 2016] (source [Duan et al., 2023]). . . . 52

3.4 Modèle entropique Factorisé. Q représente la quantification, et AE et AD représentent respectivement l'encodeur arithmétique et le décodeur arithmétique. 53
3.5 Modèle entropique Hyperprior. Q représente la quantification, et AE et AD représentent respectivement l'encodeur arithmétique et le décodeur arithmétique. 54
3.6 Modèle entropique Contextuel. Q représente la quantification, et AE et AD représentent respectivement l'encodeur arithmétique et le décodeur arithmétique. 55
3.7 Modèle entropique Hybride. Q représente la quantification, et AE et AD représentent respectivement l'encodeur arithmétique et le décodeur arithmétique (source [Minnen et al., 2018b]). . . 55
3.8 Architecture du réseau du modèle Hyperprior. Comme décrit dans la Section 3.2.3, le côté gauche montre une architecture d'auto-encodeur d'image, et le côté droit correspond à l'auto-encodeur implémentant l'hyperprior. Q représente la quantification, et AE et AD représentent respectivement l'encodeur arithmétique et le décodeur arithmétique (source [Ballé et al., 2018]). 57
3.9 Schéma de compression AE avec concurrence des distributions a priori. Les indices i désignent les indices des tables CDF qui minimisent le nombre de bits pour chaque dimension spatiale latente. L'entrainement se fait selon Loss = Distortion+λbitCost (source [Brummer and De Vleeschouwer, 2021]). 57
3.10 Convolutions masquées modélisant le contexte causal spatial. Les blocs rouges représentent les éléments à encoder/décoder. Les latents aux emplacements jaunes et bleus sont actuellement visibles (ils sont tous visibles pendant l'encodage, et ceux qui ont été décodés sont visibles pendant le décodage). Une modélisation de contexte peut être effectuée en utilisant une convolution masquée centrée sur l'emplacement rouge et convoluant uniquement avec les latents jaunes. (a)(b) Instances 3×3 et 5×5 du modèle de contexte en série largement utilisé dans la littérature. Ce modèle limite l'efficacité en termes de calcul. (c)(d) Modèle de contexte en damier proposé avec une taille de noyau de 3×3 et 5×5. Après le décodage de tous les éléments d'ancrage (bleus et jaunes), le calcul du contexte pour tous les non-ancrages peut être exécuté en parallèle (source [He et al., 2021]). 58
3.11 Pipeline de compression d'image appris proposé par [Sui et al., 2024]. 59
3.12 Architecture proposée par [Hu et al., 2020]. La distribution de probabilité de la couche la plus interne de l'hyperprior est approximée par une distribution gaussienne à moyenne nulle, où les valeurs d'échelle σ sont des paramètres entraînables indépendants par canal et partagés spatialement. 60
3.13 Vue d'ensemble de QRes-VAE. En (a), le cadre général des ResNet VAEs. Les blocs latents se comportent différemment pour l'entraînement, la compression et la décompression, illustrés en (b), (c) et (d) respectivement (source [Duan et al., 2023]). 61
3.14 Illustration de l'architecture CNN pour la compression d'image pondérée par le contenu proposé par [Li et al., 2018]. 62
3.15 Vue d'ensemble du modèle proposé par [Tang et al., 2023]. 62
3.16 (a) L'architecture de STF. La couche de fusion de patchs et la couche de division de patchs sont composées de couches linéaires et de couches de normalisation (LN). (b) W-MSA est le *window-based multi-head self-attention*, et SW-MSA est le *shifted window-based self-attention*, respectivement (source [Zou et al., 2022]). 63
3.17 Architecture proposée par [Kao et al., 2023]. G_{enc} et G_{dec} génèrent des tokens de prompt comme entrée supplémentaire afin de s'adapter à des objectifs de qualité d'image variables. 64
3.18 Architecture de HiFiC (source [Mentzer et al., 2020]) 65
3.19 Architecture proposée par [Jiang et al., 2023]. AE et AD représentent l'encodage et le décodage arithmétique, et Q correspond à la quantification. ITA est introduit pour fusionner les caractéristiques d'image et de texte basées sur le mécanisme d'attention, et IRC est conçu pour utiliser de manière adaptative les caractéristiques textuelles en complément sémantique de l'image. . . . 66
3.20 Vue d'ensemble de PerCo (source [Careil et al., 2024]). 66
3.21 Vue d'ensemble du réseau de codage inversible proposé par [Xie et al., 2021]. 67
3.22 Architecture du réseau du modèle d'apprentissage des couleurs proposé par [Prativadibhayankaram et al., 2023]. La partie supérieure représente la branche de luminance et la partie inférieure la branche de chrominance. 68
3.23 Le cadre de la compression d'image basée sur le SAE proposé par [Jia et al., 2019]. 69
3.24 Approche proposée par [Yang et al., 2021]. 70
3.25 Architecture proposée par [Song et al., 2021b]. 71
3.26 Vue d'ensemble de SigVIC (source [Liang et al., 2023]). 71

3.27 Architecture de AG-VAE. L'adaptation du taux est réalisée en insérant une unité de gain après l'encodeur et une unité de gain inverse avant le décodeur. Le taux de bits peut être ajusté de manière continue en modifiant l'indice du vecteur de gain s et le coefficient d'interpolation l. Le modèle d'entropie gaussien asymétrique estime avec précision l'entropie de la représentation latente après gain et quantification (source [Cui et al., 2021]). 72
3.28 Architecture proposée par [Qin et al., 2024]. CAM est le module d'attention causale, FEM est le modèle d'entropie factorisé. 73
3.29 Architecture proposée par [Ballé et al., 2017]. 74
3.30 Approche proposée par [Liu et al., 2022a]. 75
3.31 Architecture proposée par [Cao et al., 2023]. "Q" représente la quantification. "AE" et "AD" désignent respectivement l'encodeur et le décodeur arithmétique. 75
3.32 Vue d'ensemble du cadre Adapt-ICMH proposé. Plusieurs adaptateurs de modulation spatial-fréquence (SFMA) sont insérés dans l'encodeur g_a et le décodeur g_s du codec de base pour l'adaptation des caractéristiques. Lors de l'adaptation à la tâche de vision par ordinateur, le codec de base est gelé et seuls ces adaptateurs sont entraînables (source [Li et al., 2024a]). . . . 76
3.33 Illustration de la détection d'objets. 78
3.34 Composition des détecteurs d'objets (source [Kateb et al., 2021]). 78
3.35 Processus de calcul du mAP. 80
3.36 *Intersection over Union* (IoU) est defini comme le rapport entre l'intersection et l'union des deux boîtes. Si ce dernier est supérieur à un certain seuil (en général 0.5) on considère que les deux boites encadrent le même objet. 80
3.37 Exemple de courbe Precision-Rappel (Precision-Recall). 81
3.38 Les couleurs des boîtes sont définies comme suit : ■ = détection de faux positif ; ■ = vérité terrain ; ■ = détection de vrai positif. L'IoU avec la vérité terrain pour chaque type d'erreur est indiqué par une surbrillance orange et montré dans la rangée inférieur (dans notre cas $t_b = 0,1$ et $t_f = 0,5$) 82
3.39 Illustration de R-CNN (source [Girshick et al., 2014]). 82
3.40 Illustration de SPP. Regroupement des caractéristiques à partir de fenêtres arbitraires sur les cartes de caractéristiques. Les cartes de caractéristiques sont calculées à partir de l'image entière. Le regroupement est effectué dans les fenêtres candidates (source [He et al., 2014]). 83
3.41 Illustration de Fast R-CNN. Une image d'entrée et plusieurs régions d'intérêt (RoIs) sont introduites dans un réseau entièrement convolutionnel. Chaque RoI est mise en commun dans une carte de caractéristiques de taille fixe, puis transformée en un vecteur de caractéristiques par des couches entièrement connectées (FCs). Le réseau produit deux vecteurs de sortie par RoI : des probabilités Softmax et des décalages de régression de boîte englobante pour chaque classe (source [Girshick, 2015]). 84
3.42 Illustration de Faster R-CNN (source [Ren et al., 2016]). 84
3.43 Architecture générale de R-FCN. Un Réseau de Proposition de Régions (RPN) propose des RoIs candidates, qui sont ensuite appliquées sur les cartes de scores. Toutes les couches à poids ajustables sont convolutionnelles et sont calculées sur l'image entière ; le coût de calcul par RoI est négligeable (source [Dai et al., 2016]). 85
3.44 Feature Pyramid Network (source [Lin et al., 2017a]). 85
3.45 Architecture de Mask R-CNN (source [He et al., 2017]). 86
3.46 L'architecture générale de ThunderNet. ThunderNet utilise une résolution d'entrée de 320×320 pixels. Le backbone SNet est basée sur ShuffleNetV2 et spécialement conçu pour la détection d'objets. Dans la partie détection, le RPN est compressé, et le sous-réseau R-CNN utilise une couche fc de dimension 1024 pour une meilleure efficacité. Le module d'amélioration du contexte exploite les informations sémantiques et contextuelles à partir de multiples échelles. Le module d'attention spatiale introduit les informations du RPN pour affiner la distribution des caractéristiques (source [Qin et al., 2019]). 86
3.47 Illustration du modèle YOLOv1 (source [Redmon et al., 2016]). 87
3.48 Illustration de la différence entre la tête YOLOv3 et la tête découplée proposée par YOLOX (source [Ge et al., 2021]). 88
3.49 Architecture de YOLOv6 (source [Li et al., 2022]). 89
3.50 Comparaison entre les modèles de détection SSD et YOLO (source [Liu et al., 2016]). 90

3.51 Vue d'ensemble de CornerNet. Le réseau backbone est suivi de deux modules de prédiction, l'un pour les coins supérieurs gauches et l'autre pour les coins inférieurs droits. En utilisant les prédictions des deux modules, nous localisons et regroupons les coins (source [Law and Deng, 2020]). 91
3.52 Architecture de EfficientDet (source [Tan et al., 2020]). 91
3.53 Vue d'ensemble de DETR. Ce dernier utilise un réseau CNN classique en backbone pour apprendre une représentation 2D d'une image d'entrée. Le modèle aplatit cette représentation et lui ajoute un encodage positionnel avant de la passer dans un encodeur Transformer. Un décodeur Transformer prend ensuite en entrée un petit nombre fixe d'embeddings positionnels appris, appelés requêtes d'objet, et se concentre également sur la sortie de l'encodeur. Chaque embedding de sortie du décodeur est transmis à un réseau de neurones à propagation directe (FFN) partagé, qui prédit soit une détection (classe et boîte englobante), soit une classe « aucun objet » (source [Carion et al., 2020]). 92
3.54 Architecture générale du *Pyramid Vision Transformer* (PVT). Le modèle complet est divisé en quatre étapes, chacune composée d'une couche d'incorporation de patchs et d'un encodeur Transformer de L_i couches. Suivant une structure pyramidale, la résolution de sortie des quatre étapes diminue progressivement, passant d'une haute résolution (stride de 4) à une basse résolution (stride de 32) (source [Wang et al., 2021d]). 93
3.55 Définition de la précision et du rappel pour les distributions : (b) La précision est la probabilité qu'une image aléatoire de P_g se trouve dans le support de P_r, (c) Le rappel est la probabilité qu'une image aléatoire de P_r se trouve dans le support de P_g (source [Kynkäänniemi et al., 2019]). 97
3.56 Fonctionnement du VAE (source [Yang and Wang, 2023]). 97
3.57 À gauche : Une figure décrivant le VQ-VAE. À droite : Visualisation de l'espace d'incorporation. La sortie de l'encodeur z est mappée au point le plus proche e_2. Le gradient $\nabla_z L$ (en rouge) poussera l'encodeur à modifier sa sortie, ce qui pourrait altérer la configuration lors du passage suivant (source [van den Oord et al., 2017]). 98
3.58 Schéma de l'entraînement du GAN. Les flèches en pointillé représentent la mise à jour des poids via le gradient de l'erreur (source `https://blogs.mathworks.com/deep-learning/2021/12/02/synthetic-image-generation-using-gans/`). 99
3.59 Procédure d'échantillonnage du modèle LAPGAN. Le modèle commence avec un échantillon de bruit z_3 (coté droit) et utilisons un modèle génératif G_3 pour générer $\tilde{I}_3$. Celui-ci est suréchantillonné (flèche verte) puis utilisé comme variable de conditionnement l_2 (flèche orange) pour le modèle génératif au niveau suivant, G_2. En combinaison avec un autre échantillon de bruit z_2, G_2 génère une image de différence $\tilde{h}_2$, qui est ajoutée à l_2 pour créer $\tilde{I}_2$. Ce processus se répète sur deux niveaux supplémentaires pour produire un échantillon final à pleine résolution I_0 (source [Denton et al., 2015]). 99
3.60 (a) CycleGAN contient deux fonctions de mapping, $G : X \rightarrow Y$ et $F : Y \rightarrow X$, ainsi que des discriminateurs adversaires associés D_Y et D_X. D_Y incite G à traduire X en des sorties indiscernables du domaine Y, et inversement pour D_X et F. Pour régulariser davantage les mappings, le modèle introduit deux pertes de cohérence cyclique. (b) perte de cohérence cyclique directe : $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$, et (c) perte de cohérence cyclique inverse : $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$ (source [Zhu et al., 2017]). 100
3.61 Générateur dans StyleGAN. Alors qu'un générateur traditionnel transmet le code latent uniquement à travers la couche d'entrée, StyleGAN mappe d'abord l'entrée vers un espace latent intermédiaire W, qui contrôle ensuite le générateur via une normalisation adaptative par instance (AdaIN) à chaque couche de convolution. Du bruit gaussien est ajouté après chaque convolution, avant d'appliquer la non-linéarité (source [Karras et al., 2019]). 101
3.62 Vue d'ensemble du ViTGAN (source [Lee et al., 2022c]). 101
3.63 Processus probabiliste de diffusion par débruitage (source [Ho et al., 2020]). 102
3.64 Structure du LDM (source [Rombach et al., 2022]). 103
3.65 Vue d'ensemble du modèle autoregressif séquence-à-séquence Parti (à gauche) pour la génération d'images à partir de texte, avec ViT-VQGAN utilisé comme tokenizer d'image (à droite) (source [Yu et al., 2022]). 104
3.66 Fonctionnement du modèle DriveGAN et résultats (source [Kim et al., 2021]). 105

3.67 Vue d'ensemble de DriveDreamer. Le cadre commence avec une image de référence I_0 et des informations structurelles de la route (c'est-à-dire la carte HD H_0 et la boîte 3D B_0). DriveDreamer utilise ActionFormer pour prédire les futures caractéristiques structurelles de la route, qui servent de conditions fournies à Auto-DM, permettant de générer des vidéos de conduite futures $\hat{I}_i^{N-1}$ pour $i = 0$. De plus, des invites textuelles permettent d'ajuster dynamiquement le style des scénarios. Le modèle intègre les actions de conduite passées et des caractéristiques multi-échelles provenant d'Auto-DM pour générer des actions de conduite futures plausibles $\hat{A}_i^{N+M}$ pour $i \geq N$ (source [Wang et al., 2024c]). 106
3.68 Architecture de GAIA-1 (source [Hu et al., 2023]). 107
3.69 Vue d'ensemble de MagicDrive (source [Gao et al., 2024]). 107
3.70 Vue d'ensemble de Text2Street (source [Su et al., 2024]). 108
3.71 Architecture de DrivingDiffusion (source [Li et al., 2024b]). 108
3.72 Architecture de DreamForge (source [Mei et al., 2024]). 109
3.73 Vue d'ensemble de DiVE (source [Jiang et al., 2024a]). 109
3.74 Exemples adversariaux naturels. Dans chaque paire, l'image d'initialisation à gauche est correctement classifiée, tandis que celle de droite (adversariale) est mal classifiée avec la prédiction incorrecte indiquée en rouge (source [Lin et al., 2024]). 110
3.75 Le cadre Diff-PGD : p est le prompt, q représente la diffusion directe, R_ϕ correspond au débruitage inverse, et x_0 est l'échantillon débruité (source [Xue et al., 2023b]). 111
3.76 Les deux techniques de guidage dans AdvDiff pour générer des exemples adversariaux non restreints. Pendant le processus de génération inverse, le guidage adversarial (*adversarial guidance*) est ajouté à l'étape x_t, ce qui injecte l'objectif adversarial y_a dans le processus de diffusion. Le guidage de l'échantillonnage du bruit (*noise sampling guidance*) modifie le bruit initial en augmentant la probabilité conditionnelle de y_a (source [Dai et al., 2024b]). 111
3.77 Guidé par le gradient de la perte rétropropagé depuis le classificateur, SD-NAE génère des NAEs en optimisant uniquement l'embedding des tokens liés à la classe, tout en gardant tous les modèles figés (source [Lin et al., 2024]). 112
3.78 Illustration du cadre EvoSeed pour optimiser le vecteur de graine initial z afin de générer un exemple adversarial naturel. La stratégie d'évolution par adaptation de la matrice de covariance (CMA-ES) affine itérativement le vecteur de graine initial z et trouve un vecteur de graine adversarial z'. Ce vecteur de graine adversarial z' peut ensuite être utilisé par le modèle de diffusion conditionnel G pour générer un exemple adversarial naturel x, capable de tromper le modèle de classification F (source [Kotyan et al., 2025]). 112

4.1 Distribution boostrap de l'estimateur θ. 121
4.2 Loi normale centré en x_i et d'écart-type $\sigma_{1-\beta}$ 122
4.3 Découpage de Ω en n partition disjointes A_i. 123
4.4 Système composé de modules d'IA. 123
4.5 Illustration de la méthode proposée par [Adedjouma et al., 2024]. 125
4.6 Illustration d'un ensemble de jeux de données regroupées par attributs de l'ODD. 126
4.7 Illustration d'un partitionnement des jeux de données. 129
4.8 Schéma du processus itératif de génération de données critiques en orientant celles-ci selon la direction du gradient. 135
4.9 Illustration du processus de génération : contraint à la variété des données, la génération produit des images maximisant une fonction d'erreur $\mathcal{L}$ grâce à l'orientation par gradient. 135
4.10 Illustration de la distribution réelle inconnue des données X et de la vraie probabilité d'erreur nommée ici P_{reel}. 136
4.11 Illustration de l'estimation biaisée de la probabilité d'erreur $\hat{P}$ à partir de la distribution de données échantillonnées. 136
4.12 Illustration de l'estimation de la probabilité d'erreur corrigée $\hat{P}_{our}$ par un système de pondération. 137

5.1 Illustration du système S (encadré en rouge) et du processus de calcul de la fonction d'erreur $e(x)$. 141
5.2 Illustration du système de coordonnées pour les boîtes 2D. 142
5.3 ODD partiel du système Field Monitoring. 144
5.4 Lieux géographiques d'acquisition des données Waymo : San Francisco à gauche, Mountain View au centre et Phoenix à droite. 146
5.5 Exemples d'images Waymo. 147

5.6 Histogramme de la quantité d'objets par image de Waymo pour les classes véhicule, piéton et cycliste. . . . 148
5.7 Histogramme 2D de la distribution des centroïdes des boîtes dans les images de Waymo pour les classes véhicule, piéton, cycliste. . . . 148
5.8 Histogramme 2D de la distribution de la largeur et hauteur des boites dans les images de Waymo pour les classes véhicule, piéton et cycliste. . . . 149
5.9 Lieux géographique d'acquisition des données BDD100K. . . . 149
5.10 Exemples d'images BDD100K. . . . 150
5.11 Exemples d'images BDD100K où le pare-brise impacte les images. . . . 151
5.12 Histogramme de la quantité d'objets par image sur BDD100K pour les classes véhicule, piéton et cycliste. . . . 151
5.13 Histogramme 2D de la distribution des centroïdes des boîtes dans les images sur BDD100K pour les classes véhicule, piéton, cycliste. . . . 152
5.14 Histogramme 2D de la distribution de la largeur et hauteur des boites dans les images sur BDD100K pour les classes véhicule, piéton et cycliste. . . . 152
5.15 Exemples d'images Stellantis. . . . 153
5.16 Histogramme de la quantité d'objets par image sur Stellantis pour les classes véhicule, piéton et cycliste. . . . 154
5.17 Histogramme 2D de la distribution des centroïdes des boîtes dans les images sur Stellantis pour les classes véhicule, piéton, cycliste. . . . 154
5.18 Histogramme 2D de la distribution de la largeur et hauteur des boites dans les images sur Stellantis pour les classes véhicule, piéton et cycliste. . . . 155
5.19 Exemples d'images Boxy. . . . 156
5.20 Exemples d'images Audi 157
5.21 Exemples d'images CarCrash. . . . 158
5.22 Exemples d'images DashCam. . . . 159
5.23 Exemples d'images NuImages. . . . 160
5.24 Exemples de remorques dans NuImagesLabels. . . . 160
5.25 Histogramme de la quantité d'objets par image sur NuImagesLabels(Front) pour les classes véhicule, piéton et cycliste. . . . 161
5.26 Histogramme 2D de la distribution des centroïdes des boîtes dans les images sur NuImagesLabels(Front) pour les classes véhicule, piéton, cycliste. . . . 161
5.27 Histogramme 2D de la distribution de la largeur et hauteur des boites dans les images sur NuImagesLabels(Front) pour les classes véhicule, piéton et cycliste. . . . 162
5.28 Lieux géographique d'acquisition des données Waterloo. . . . 163
5.29 Exemples d'images Waterloo. . . . 163
5.30 Exemples d'image PeSOTIF 165
5.31 Exemples d'image E&C 166
5.32 Histogramme de la quantité d'objets par image sur E&C pour les classes véhicule, piéton et cycliste. 166
5.33 Histogramme 2D de la distribution des centroïdes des boîtes dans les images sur E&C pour les classes véhicule, piéton, cycliste. . . . 167
5.34 Histogramme 2D de la distribution de la largeur et hauteur des boites dans les images sur E&C pour les classes véhicule, piéton et cycliste. . . . 168
5.35 Écran de démarrage et paramétrisation de la simulation. . . . 170
5.36 Visualisation des évènements avec des variations des paramètres de contexte (cf. Figure 5.35). . 171
5.37 Histogramme de la quantité d'objets par image des données synthétiques pour les classes véhicule, piéton et cycliste. . . . 172
5.38 Histogramme 2D de la distribution des centroïdes des boîtes dans les données synthétiques pour les classes véhicule, piéton, cycliste. . . . 173
5.39 Histogramme 2D de la distribution de la largeur et hauteur des boites dans les données synthétiques pour les classes véhicule, piéton et cycliste. . . . 173
5.40 Analyse macro : mAP@.5 sur tous les scénarios à 24 (cf. Table 5.4). . . . 175
5.41 Nombre moyen d'erreurs par image sur l'ensemble des 24 scénarios (cf. Tableau 5.4). E_i représente l'événement numéro i et B_i les phases intermédiaire sans évènement 176

5.42 Certains des événements du scénario n°2 (cf. Table 5.4) avec les erreurs rencontrées à différents moments (de gauche à droite) : Faux Négatif (FN) en rouge et Mauvaise Classification (MC) en jaune. . . . 176
5.43 Description des jeux de données d'entraînement augmentés avec des données synthétiques. . . . 177
5.44 Exemples d'images du jeu de données E&C avec des erreurs de détection avec YOLOv7 entraîné sur D_{train} sans données synthétiques (à gauche) et YOLOv7 entraîné sur $DU33_{train}$ avec 33% de données synthétiques (à droite). Code couleur : Faux Négatif (FN) en rouge et Mauvaise Classification (MC) en jaune. . . . 178

6.1 Analyse quantitative des performances des modèles de compression sur D_{test}. . . . 185
6.2 Relation entre le PSNR et le bpp pour chaque image de D_{test}, obtenue avec le modèle stf(mse) pour différentes valeurs de l'hyperparamètre λ. . . . 186
6.3 Comparaison qualitative d'une image compressée à l'aide de différentes méthodes de compression et à divers taux de compression (les annotations indiquent les valeurs de bpp/PSNR/MS-SSIM). . . . 186
6.4 Répartition des bits entre des taux de compression élevés et faibles avec le compresseur STF (résultats similaires observés avec QRes-VAE), accompagnée des valeurs correspondantes de bpp, PSNR et MS-SSIM annotées en vert. . . . 187
6.5 Observation d'artefacts sur une image Waymo fortement compressées à l'aide du modèle stf(mse)$_{\lambda_1}$ ($bpp = 0.036$). . . . 188
6.6 Exemples d'images issues du dataset Waymo (cf. Section 5.3.1) et de leurs cartes de qualité associées. Les zones correspondant aux véhicules, où une compression minimale est souhaitée, sont mises en évidence en blanc (λ^+), tandis que l'arrière-plan, où une compression plus forte est tolérée, est représenté en noir (λ^-). . . . 189
6.7 Image compressée et allocation de bits respective pour le modèle STF agnostique (a) et les modèles STF ROI avec deux ensembles d'hyperparamètres, λ_8 (b) et λ_9 (c). Les annotations en blanc indiquent respectivement les valeurs de bpp/PSNR/MS-SSIM pour chaque configuration. . . . 189
6.8 Illustration du modèle de diffusion sans le conditionnement (source [Rombach et al., 2022]). . . . 191
6.9 Exemples d'images générées avec le modèle diff_front. . . . 194
6.10 Exemples d'images générées avec le modèle diff_all. . . . 194
6.11 Processus d'analyse des erreurs de détection avec et sans la compression. . . . 196
6.12 Évolution des scores d'Average Precision pour les véhicules (APv), les piétons (APp), et les cyclistes (APc) en fonction du débit binaire. . . . 197
6.13 Analyse de l'impact du compresseur stf(mse) sur le détecteur YOLOv7 à l'aide de la méthode TIDE (cf. Figure 8.1 en annexes pour les hyperparamètres λ_2 et λ_3). . . . 197
6.14 Analyse de l'impact du compresseur stf(mse) sur le détecteur FasterRCNN à l'aide de la méthode TIDE (cf. Figure 8.2 en annexes pour les configurations avec λ_2 et λ_3). . . . 198
6.15 Analyse de l'effet du compresseur stf(mse) sur le détecteur RetinaNet à l'aide de la méthode TIDE (cf. Figure 8.3 en annexes pour les configurations avec λ_2 et λ_3). . . . 198
6.16 Comparison between the impact of our previous STF and QRes-VAE models and the new ROI-based models on YOLOv7 and FasterRCNN over all classes (mAP). The green arrows show the increase of mAP. . . . 199
6.17 Évaluation avec la méthode TIDE de l'effet du compresseur STF(mse) ROI sur le détecteur YOLOv7. . . . 199
6.18 Illustration du système S (encadré en rouge) et du processus de calcul de la fonction d'erreur $e(x)$. 200
6.19 Représentation idéalisée et abstraite de nos jeux de données dans l'espace défini par l'ODD et du partitionnage. . . . 201
6.20 Probabilité intra-dataset $\hat{\theta}_j$, $\hat{\theta}_j^{e,C}$ et $\hat{\theta}_j^{C}$ en fonction de ϵ, pour le système composé de stf(mse) et YOLOv7 pour les niveaux de compression λ_7 (0.349 bpp) et λ_1 (0.036 bpp) dans la configuration $C_3 = \{v, p, c\}$. . . . 203
6.21 Estimations $\hat{\theta}_j$ pour $\epsilon = 1$ pour chaque taux de compression de stf(mse). . . . 203
6.22 Probabilité intra-dataset $\hat{\theta}_j$, $\hat{\theta}_j^{e,C}$ et $\hat{\theta}_j^{C}$ en fonction de ϵ, pour le système composé de stf(mse) et YOLOv7 pour le niveau de compression λ_7 (0.349 bpp) dans la configuration $C_0 = \{v\}$. . . . 204
6.23 Probabilité intra-dataset $\hat{\theta}_j$, $\hat{\theta}_j^{e,C}$ et $\hat{\theta}_j^{C}$ en fonction de ϵ, pour le système composé de stf(mse) et YOLOv7 pour le niveau de compression λ_7 (0.349 bpp) dans la configuration $C_1 = \{p\}$. . . . 205

6.24 Probabilité intra-dataset $\hat{\theta}_j$, $\hat{\theta}_j^{e,C}$ et $\hat{\theta}_j^{C}$ en fonction de ϵ, pour le système composé de stf(mse) et YOLOv7 pour le niveau de compression λ_7 (0.349 bpp) dans la configurations $C_3 = \{c\}$. 205
6.25 Estimation inter-dataset avec une distribution uniforme appliquée au modèle de compression stf(mse)$_{\lambda_7}$ associé à YOLOv7, pour les configurations de classes C_3 et C_0. 207
6.26 Estimation inter-dataset avec une distribution uniforme appliquée au modèle de compression stf(mse)$_{\lambda_7}$ associé à YOLOv7, pour les configurations de classes C_1 et C_2. 208
6.27 Estimation des probabilité 209
6.28 Estimation inter-dataset avec une distribution pour le Field Monitoring appliquée au modèle de compression stf(mse)$_{\lambda_7}$ associé à YOLOv7, pour les configurations de classes C_3 et C_0. 211
6.29 Estimation inter-dataset avec une distribution pour le Field Monitoring appliquée au modèle de compression stf(mse)$_{\lambda_7}$ associé à YOLOv7, pour les configurations de classes C_1 et C_2. 211
6.30 Estimation des probabilité pour le *Field Monitoring* avec YOLOv7 et stf(mse)$_{\lambda_7}$. 212
6.31 Comparaison des estimations entre la méthodologie proposée (Uniform et Field Monitoring) et une estimation simplifiée basée sur l'ensemble des données (All Data). 213
6.32 Estimation de la probabilité d'erreur $\hat{\theta}$ pour chaque compresseur et pour chaque niveau de compression. 214
6.33 Estimation de la probabilité d'erreur $\hat{\theta}$ pour le modèle de compression stf(mse) agnostique avec les modèles de détection YOLOv7, *yolo_aug_front* et *yolo_aug*. 217
6.34 Estimation de la probabilité d'erreur $\hat{\theta}$ pour le modèle de compression stf(mse) agnostique avec les modèles de détection YOLOv7, *yolo_025unity*, *yolo_033unity* et *yolo_055unity*. 218
6.35 Estimation de la probabilité d'erreur $\hat{\theta}$ pour le modèle de compression stf(mse) agnostique avec les modèles de détection YOLOv7, yolostfmseq7 et yolostfmseq3. 220
6.36 Compression et allocation de bits pour une image "hors domaine" avec le compresseur stf(mse)$_{\lambda_7}$. 220
6.37 Estimation de la probabilité d'erreur $\hat{\theta}$ pour le modèle de compression stf(mse) agnostique et ROI augmenté avec les modèles de détection yolov7. 221
6.38 Processus de génération d'image orienté pour augmenter une fonction d'erreur. 222
6.39 Évolution de la fonction de perte l (gauche) et de la distance L_2 entre le latent d'origine $z_{origine}$ et le latent courant z (droite) à chaque étape de génération. 224
6.40 Images générées par diffusion avec orientation pour maximiser le MSE. 225
6.41 Évolution de la fonction de perte l (gauche) et de la distance L_2 entre le latent d'origine $z_{origine}$ et le latent courant z (droite) à chaque étape de génération. 226
6.42 Deux exemples d'images générées par diffusion avec orientation pour maximiser le bpp. 227
6.43 Évolution de la fonction de perte l (haut gauche), de la distance L_2 entre le latent d'origine z_{origine} et le latent courant z (haut droite), ainsi que de la quantité d'erreur (bas), à chaque étape de génération. 229
6.44 Détections sur images générées et compressées, orientées pour maximiser la fonction de perte de YOLOv7. Code couleur : véhicule (rouge), piéton (vert), cycliste (jaune). 230
6.45 Évolution de la fonction de perte l (haut gauche), de la distance L_2 entre le latent d'origine z_{origine} et le latent courant z (haut droite), ainsi que de la quantité d'erreur (bas), à chaque étape de génération. 231
6.46 Détections sur images générées et compressées, orientées pour maximiser la fonction de perte l_{obj}. Code couleur : véhicule (rouge), piéton (vert), cycliste (jaune). 232
6.47 Évolution de la fonction de perte l (haut gauche), de la distance L_2 entre le latent d'origine z_{origine} et le latent courant z (haut droite), ainsi que de la quantité d'erreur (bas), à chaque étape de génération. 233
6.48 Détections sur images générées et compressées, orientées pour maximiser la fonction de perte l_{cls}. Code couleur : véhicule (rouge), piéton (vert), cycliste (jaune). 234
6.49 Évolution de la fonction de perte l (haut gauche), de la distance L_2 entre le latent d'origine z_{origine} et le latent courant z (haut droite), ainsi que de la quantité d'erreur (bas), à chaque étape de génération. 235
6.50 Détections sur images générées et compressées, orientées pour maximiser la fonction de perte l_{box}. Code couleur : véhicule (rouge), piéton (vert), cycliste (jaune). 236

8.1 Évaluation TIDE sur D_{test} avec STF(mse) et YOLOv7. 270
8.2 Évaluation TIDE sur D_{test} avec STF(mse) et FasterRCNN. 271
8.3 Évaluation TIDE sur D_{test} avec STF(mse) et RetinaNet. 271

8.4 Évaluation TIDE sur D_{test} avec STF(ms-ssim) et YOLOv7. 272
8.5 Évaluation TIDE sur D_{test} avec STF(ms-ssim) et FasterRCNN. 272
8.6 Évaluation TIDE sur D_{test} avec STF(ms-ssim) et RetinaNet. 273
8.7 Évaluation TIDE sur D_{test} avec QRes-VAE(mse) et YOLOv7. 273
8.8 Évaluation TIDE sur D_{test} avec QRes-VAE(mse) et FasterRCNN. 274
8.9 Évaluation TIDE sur D_{test} avec QRes-VAE(mse) et RetinaNet. 274
8.10 Évaluation TIDE sur D_{test} avec QRes-VAE(ms-ssim) et YOLOv7. 275
8.11 Évaluation TIDE sur D_{test} avec QRes-VAE(ms-ssim) et FasterRCNN. 275
8.12 Évaluation TIDE sur D_{test} avec QRes-VAE(ms-ssim) et RetinaNet. 276

Liste des tableaux

2.1 Tableau récapitulatif de l'état de l'art des méthodes de Vérification Formelles. 44
2.2 Tableau récapitulatif de l'état de l'art sur la Certification Statistique. 45

3.1 Tableau récapitulatif de l'état de l'art des méthodes de Compression d'images apprise. 115
3.2 Tableau récapitulatif de l'état de l'art des méthodes de Détection d'objets. 116
3.3 Tableau récapitulatif de l'état de l'art des méthodes de Génération d'images. 117

5.1 Informations sur les scénarios de Waymo. 147
5.2 Informations sur les vidéos de CarCrash. 157
5.3 Conditions de déclenchement partielles des capteurs visuels provenant de [Peng et al., 2023]. . . 164
5.4 Combinaisons de paramètres de contexte par scénarios (cf. Figure 5.35). 172
5.5 Datasets naturelles et synthétiques simulés. 174
5.6 Estimation par bootstrap, avec $B = 100,000$ et avec un intervalle de confiance à 95%, du mAP@.5 de YOLOv7 entrainé et testé sur les datasets réel D et synthétique U. 175
5.7 Datasets augmentés avec des données synthétiques. 177
5.8 Estimation par bootstrap, avec $B = 100,000$ et avec un intervalle de confiance à 95%, du mAP@.5 de YOLOv7 entrainé sur le dataset de données réelles augmenté avec des données synthétiques. 177
5.9 Récapitulatif de l'ensemble des jeux de données. 179

6.1 Vitesse d'encodage et de décodage pour chaque compresseur pour une image de dimension 640×1280, mesurée sur un GPU NVIDIA RTX A5500. Les performances de nos modèles sont comparées à celles de deux modèles plus anciens, mais couramment mentionnés dans la littérature : mbt2018 [Minnen et al., 2018b] et cheng2020_attn [Cheng et al., 2020b]. 183
6.2 Datasets initiaux pour l'apprentissage de la Compression et de la Détection. 183
6.3 Modèles de compression agnostique entraînés avec différents hyperparamètres λ. 184
6.4 Modèles de compression appris ROI. 189
6.5 Datasets initiaux pour l'apprentissage de la Compression et de la Détection. 190
6.6 Average Precision (AP@.5) des détecteurs sur le dataset D_{test} avec intervalles de confiance à 95% estimés par méthode bootstrap ($B = 100,000$). 190
6.7 Datasets pour l'apprentissage du modèle de diffusion. 192
6.8 Performances des modules de diffusion évaluées sur 300 échantillons et comparées aux données du dataset $\text{Diffusion_Front}_{valid}$ et $\text{Diffusion_All}_{valid}$. 193
6.9 Tableau des poids w_j et des hyperparamètres p_j selon une distribution uniforme. 206
6.10 Tableau des poids w_j et des hyperparamètres p_j selon une distribution pour le *Field Monitoring*. 210
6.11 Partitions augmentées du dataset D. 216
6.12 Performances du detecteurs YOLOv7 entrainé en augmentant la base d'apprentissage avec les données NuImagesLabels. 217
6.13 Performances du detecteurs YOLOv7 entrainé en augmentant la base d'apprentissage avec des données comprimées. 219
6.14 Modèles de compression STF entrainé avec la metrique MSE appris ROI augmentés. 221

8.1 Récapitulatif des modèles de Compression. 276
8.2 Récapitulatif des modèles de détection et de leurs performances sur D_{test} : APv@.5, APp@.5 et APc@.5 correspondant respectivement aux AP@.5 pour les classes véhicules, piétons et cyclistes. 277

8.3 Récapitulatif des modèles de diffusion utilisés. 277