UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

OPTIMISATION DE L'EFFICACITÉ ÉNERGÉTIQUE ET DE LA DÉTECTION D'OBSTACLES D'UN VÉHICULE AUTONOME EN PÉRIODE HIVERNALE

THÈSE PRÉSENTÉE

COMME EXIGENCE PARTIELLE

DOCTORAT EN INGÉNIERIE (CONCENTRATION GÉNIE MÉCANIQUE) 2325

PAR

JONATHAN BOISCLAIR

OCTOBRE 2024

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

Doctorat en ingénierie (concentration génie mécanique) (2325)

Direction de recherche :

Sousso Kelouwani	Directeur de recherche
Kodjo Agbossou	Codirecteur de recherche

Jury d'évaluation

Miloud Bagaa	Président du Jury
Lama Seoud	Évaluateur externe
Usef Faghihi	Évaluateur interne
Sousso Kelouwani	Directeur de recherche
Kodjo Agbossou	Codirecteur de recherche

Thèse soutenue le 9 octobre 2024

Résumé

La détection d'obstacles est un aspect important des outils d'assistances à la conduite avancée (ADAS) ou d'autonomie complète, qui permettent au véhicule de se déplacer de manière autonome. Pour optimiser la distance pouvant être parcourue sans recharger la voiture, il faut mieux gérer la consommation de la voiture. Cependant, les systèmes de détection d'obstacles actuels ont beaucoup de lacunes, surtout en hiver, où la neige, le brouillard, la pluie ou le verglas peuvent affecter la qualité des images capturées par les capteurs. L'amélioration des détections passe par plusieurs aspects. Le premier aspect est l'utilisation de données de qualité, qui sont représentatives des conditions réelles et qui couvrent les différents scénarios possibles. Le deuxième aspect est la détection dans des conditions où la caméra visible ne peut rien percevoir telle que la nuit totale. Cet aspect est couvert par la fusion de données multispectrales et se compose de trois éléments principaux. Le premier élément est un ensemble de capteurs qui capturent des images dans différentes longueurs spectrales, telles que le visible et la vision thermique. Le deuxième élément est une étape de fusion des données, qui consiste à combiner les données provenant des différents capteurs, en tenant compte de leurs formats, de leurs résolutions, de leurs fréquences et de leurs angles de vue. Le troisième élément est une source de données contenant des données synchronisées, qui permettent de comparer les résultats de la fusion avec les données réelles. Le troisième aspect est l'intégration et l'évaluation énergétique de plusieurs méthodes de pointes afin d'obtenir un meilleur résultat que chaque méthode individuellement. L'intégration consiste à utiliser les résultats de la fusion des données pour adapter le comportement du véhicule en fonction de la situation, notamment en optimisant les trajectoires, les freinages et les accélérations. Pour cela, il faut tenir compte de la distance des objets et de leur type afin de les éviter.

Abstract

Obstacle detection is an important aspect of ADAS or full autonomy systems, which enable the vehicle to move autonomously. To optimize the distance that can be traveled without recharging the car, the car's consumption must be better managed. However, current obstacle detection systems have many shortcomings, especially in winter, where snow, fog, rain or ice can affect the quality of the images captured by the sensors. Improving detections involves several aspects. The first aspect is the use of quality data, which are representative of real conditions and cover the different possible scenarios. The second aspect is the detection in conditions where the visible camera cannot perceive anything, such as total night. This aspect is covered by the fusion of multispectral data and consists of three main elements. The first element is a set of sensors that capture images in different spectral lengths, such as visible and thermal vision. The second element is a data fusion step, which consists of combining the data from the different sensors, taking into account their formats, resolutions, frequencies and viewing angles. The third element is a data source containing synchronized data, which allow to compare the results of the fusion with the real data. The third aspect is the integration and energy evaluation of several state-of-the-art methods in order to obtain a better result than each method individually. The integration consists of using the results of the data fusion to adapt the behavior of the vehicle according to the situation, in particular by optimizing the trajectories, braking and accelerations. For this, it is necessary to take into account the distance of the objects and their type in order to avoid them.

Table des matières

Résumé iii

Abstract iv

Table des matières v

Liste des tableaux viii

Liste des figures ix

Liste des symboles xi

Chapitre 1 - Introduction 1

1.1 Question de recherche 3

1.2 Objectifs 3

1.3 Contribution 4

1.4 Organisation 4

Chapitre 2 - Revue de littérature 6

2.1 Systèmes de détection d'objets (SDO) 6

2.1.1 Données et entraînements 6

2.1.2 Architecture 9

2.1.3 Amélioration temps réel 12

2.2 Optimization énergétique 14

2.2.1 Freinage régénératif 14

2.2.2 Optimisation 15

Chapitre 3 - Amélioration des entraînements 17
3.1 Outil d'annotation de la saillance (OAS) 18
3.1.1 Cartes de saillance 21
3.1.2 Boîtes englobantes 24
3.2 Entraînement 26
3.3 Inférence 29
3.4 Résultats 29
3.5 Conclusion 38

Chapitre 4 - Amélioration du réseau de fusion 40
4.1 L'approche en branche 40
4.1.1 Taille des branches 42
4.2 Modifications aux réseaux existants 44
4.2.1 Modifications 44
4.2.2 Choix de l'hyperparamètre k 46
4.3 Préparation des données d'entrée 48
4.4 Implémentation proposée 49
4.5 Synchronisation 50
4.6 Entraînement 55
4.7 Résultats 55
4.8 Conclusion 62

Chapitre 5 - Intégration et énergétique 64
5.1 Modèle du véhicule 65
5.1.1 Forces du véhicule 65
5.1.2 Problème de contrôle optimal pour le freinage régénératif 66

5.1.3 Programmation dynamique 68
5.2 Système de détection 72
5.2.1 Blocs proposés 73
5.2.2 Métaréseau 75
5.3 Résultats expérimentaux et discussion 79
5.3.1 Résutats de la méthode par métaréseau 80
5.3.2 Résultats d'énergie 82
5.4 Conclusion 86

Chapitre 6 - Conclusion 88
6.1 Améliorations effectuées par cette thèse 88
6.2 Recommandations 89

Liste des tableaux

Tableau 3-1 Paramètres de Minkowski utilisé dans cette thèse. 25

Tableau 3-2 Les différents jeux de données 29

Tableau 3-3 Les différentes classes 32

Tableau 4-1 Statistiques sur le pourcentage des poids en fonction des longueurs d'onde pour les réseaux préentraînés de PyTorch. .. 43

Tableau 4-2 Différentes méthodes de synchronisation 52

Tableau 4-3 Résultats expérimentaux des capteurs. 58

Tableau 5-1 Paramètres dynamiques du véhicule 64

Tableau 5-2 L'énergie nécessaire au freinage et l'énergie récupérée par une stratégie de freinage optimale pour un véhicule de 1500 kg se déplaçant à 90 km/h (25 m/s). 69

Tableau 5-3 Distance de freinage optimale pour une vélocité initiale et une efficacité énergétique choisie ($\boldsymbol{\varepsilon}$). 70

Tableau 5-4 Jeux de données de conduite 76

Tableau 5-5 Évaluation des améliorations de perception 80

Liste des figures

Figure 3.1 Le procédé d'entraînement 19

Figure 3.2 Représentation visuelle du facteur de saillance selon l'équation 3.1 21

Figure 3.3 Exemple du trou de vision 22

Figure 3.4 Résultat de l'application de saillance sur une image 23

Figure 3.5 Exemple d'image de UQTR1. 31

Figure 3.6 Amélioration des annotations pour $T_U = 0.9, T_S = 0.9, T_3 = 0.25$. 33

Figure 3.7 Courbes de précision, rappel et F1 pour l'entraînement sur données standard vs données améliorées. Évalué sur le jeu d'entraînement. $\beta = 80\%$. 34

Figure 3.8 Métriques d'évaluation moyenne entre les réseaux. DSA est séparé égale entre saillance et régulière. 36

Figure 3.9 Métriques d'évaluation moyenne entre les réseaux (suite). DSA est séparé égale entre saillance et régulière. 37

Figure 4.1 Représentation des modifications au réseau. 45

Figure 4.2 Exemple de calibration imprécise. 49

Figure 4.3 Chronologie des différentes méthodes. 51

Figure 4.4 mAP comparée avec l'hyperparamètre k pour détection SSD. .. 56

Figure 4.5 Répartition du coût pour la fusion RGB-T. 58

Figure 4.6 Comparaison avec les réseaux de pointe. 60

Figure 5.1 Processus d'entraînement pour le métaréseau 75

Figure 5.2 Processus d'inférence en temps réel 78

Figure 5.3 Le véhicule et les capteurs utilisés pour les tests expérimentaux. 79

Figure 5.4 Énergie régénérée selon 6 profils environnementaux 84

Liste des symboles

ADAS outils d'assistances à la conduite avancée

ARE quantité d'énergie accumulé dans un freinage

BDD jeu de donnée "Berkeley Deep Drive"

CADC jeu de donnée "Canadian adverse condition dataser"

Edge AI système intelligent sur ordinateur de frontière

F_{aero} Force de friction aérodynamique

$F_{gravite}$ Force d'accélération selon la pente

F_{moteur} Force appliqué par le moteur

F_{route} Force de friction de la route

F_V Force d'accélération du véhicule

HoG histogrammes de gradient

IOU intersection sur union

IOS intersection sur plus petit

k numéro de la couche de fusion

mIOU moyenne d'intersection sur union

OAS outil d'annotation de saillance

P.E. Potentiel d'énergie récupérée

P.S. Potentiel de segmentation

R.M. Système de réduction des condition météorologiques

RNC réseaux de neurones convolutifs

RTSD jeu de donnée "Russian traffic sign dataset"

SDO systèmes de détection d'obstacles

$\alpha_{\Delta s}$ facteur de distance

Δs distance

ε efficacité énergétique d'un freinage

Ψ Portée de détection

$\Delta_{MINKOWSKI}$ distance de Minkowski

ω facteur d'importance

Chapitre 1 - Introduction

Dans le domaine du transport, peu de technologies ont eu autant d'impact que les voitures au cours du 20e siècle. La majorité des gens utilisent la voiture quotidiennement comme moyen de transport. En 2018, plus de 97 millions de voitures ont été produites dans le monde [1]. La tendance sur les routes est à l'électrification et à l'automatisation des véhicules. Avec les nouvelles lois sur l'électrification, les voitures seront bientôt intelligentes et électriques. Cette transition apporte plusieurs défis au niveau des voitures. Un défi majeur est la sécurité lors des déplacements effectués par ordinateur. En effet, un aspect important des systèmes ADAS ou d'autonomie complète est la détection d'obstacle qui doit être exécuté en temps réel. Afin d'arriver à un niveau d'autonomie de niveau 4 ou 5 selon le standard J3016 [1], les voitures doivent être capables de détecter tous les éléments de la route même dans les conditions difficiles, telles que l'hiver, la nuit et autres conditions à visibilité réduite. Un autre défi majeur est l'efficacité énergétique du véhicule lors de ses manœuvres. Deux sources d'énergie peuvent être optimisées à bord d'un véhicule électrique intelligent. La première source est l'énergie utilisée pour les déplacements du véhicule, soit l'énergie envoyée au moteur et regénérée lors des freinages. La deuxième source est l'énergie consommée par les capteurs rendant l'intelligence du véhicule possible.

L'optimisation de l'énergie de déplacement du véhicule est généralement effectuée en plusieurs temps. Pour l'amélioration à long terme, ou en mode hors ligne, l'environnement est considéré comme étant reconnu à 100%. Pour l'optimisation à court terme, l'optimisation passe par une meilleure détection de l'environnement autour de celui-ci. Cette optimisation peut concerner le maintien de la vitesse [2–4], les accélérations [5] et les freinages [6–9]. L'optimisation de l'énergie en accélération

peut être déterminée facilement en augmentant le temps pour atteindre la vitesse de croisière. L'optimisation du maintien de la vitesse et de la décélération dépendent fortement des détections de l'environnement à grande distance afin de pouvoir obtenir des profils de freinage optimaux. Cependant, la détection à grande distance est fortement diminuée lors des conditions hivernales, qui affecte plus de 200 millions de personnes habitant dans des régions ayant plus de 30 jours de neige par an [1]. Durant l'hiver, l'apparence de plusieurs obstacles change largement. Un exemple flagrant est les manteaux portés par les piétons, un autre exemple important est la neige recouvrant les toits des voitures [10]. Afin de détecter ces objets de la route, les systèmes d'assistance ou de conduite autonome utilisent généralement un mélange de caméras [11,12], de radar et de lidar [13]. Les systèmes de détection par caméras sont lourdement affectés par les produits aérosols, tels que la neige et la pluie, qui causent une distorsion de la lumière dans des angles difficiles à prédire [14]. Ces systèmes, incluant les systèmes thermiques, utilisant le spectre lumineux, sont des systèmes à très faible consommation énergétique, comparativement à une consommation plus élevée pour les radar et lidar.

Puisque l'optimisation de l'énergie de déplacement du véhicule par les différentes méthodes implique la détection de son environnement et que l'optimisation des freinages peut permettre une réduction jusqu'à 30% de la consommation électrique du véhicule, seule l'optimisation des freinages régénératifs seront abordés dans cette thèse. Cette optimisation peut prendre plusieurs formes. La première forme abordée dans cette thèse est l'assistance à la conduite avancée (ADAS). Dans cette forme, la voiture peut suggérer des freinages au conducteur afin de réduire sa consommation et d'augmenter la régénération. La deuxième forme est l'autonomie complète de la voiture. En mode de conduite 100% autonome, la voiture prend

l'ensemble des décisions et peut optimiser sa consommation par la prise d'actions sécuritaires sur la route. Cependant, afin d'optimiser, la portée de détection doit être améliorée. Une solution naïve serait d'utiliser les méthodes les plus performantes de détection disponibles actuellement [15–19] et d'installer de multiples capteurs sur les véhicules. Cependant, ces systèmes nécessitent une quantité d'énergie supérieure à ce que le véhicule pourra optimiser par leur détection. Étant donné l'importance de connaître l'environnement autour du véhicule lors des déplacements et de la capacité limitée de détection des approches actuelles en hiver, une nouvelle approche à faible consommation énergétique s'impose afin de pouvoir prendre les actions propices à l'optimisation des freinages et accélérations de la voiture. Une façon d'effectuer cette diminution de consommation est l'utilisation des systèmes V2X permettant la communication entre le véhicule et la route. Cependant, ces systèmes à faible consommation ont une grande latence et demandent l'installation de matériel spécifique sur les routes limitant son champ d'utilisation.

1.1 Question de recherche

Comment optimiser l'efficacité énergétique et la détection d'obstacles d'un véhicule autonome en conditions hivernales

1.2 Objectifs

L'objectif principal de cette thèse consiste à améliorer la sécurité et l'efficacité énergétique d'un véhicule intelligent grâce à l'anticipation des manœuvres de freinage dans un système à faible consommation. Les sous-objectifs sont:

1. détecter de manière robuste les obstacles en condition de conduite difficile comme l'hiver.

2. proposer au conducteur des indices de freinage et d'obstacle afin d'optimiser l'anticipation de freinage.

1.3 Contribution

Les systèmes de détection d'obstacles (SDO) autour d'un véhicule à faible consommation ont beaucoup de lacunes. Afin d'optimiser la consommation énergétique du véhicule, une perception à grande distance est requise [20–22]. La présente thèse propose un SDO temps réel à faible consommation énergétique offrant une meilleure performance en conditions difficiles, telles que l'hiver. Ce système utilise différents aspects des réseaux de neurones afin d'obtenir les résultats améliorés. Le premier aspect est les jeux de données utilisées durant l'entraînement, le deuxième aspect est l'architecture du réseau, et le troisième aspect est l'utilisation des métadonnées en sortie de plusieurs réseaux afin d'obtenir un résultat de meilleure qualité adapté aux conditions actuelles. En utilisant ces résultats de détection améliorés, le système peut appliquer l'optimisation du freinage en temps réel, avec une distance de freinage plus longue, tout en conservant la sécurité et la faible consommation énergétique du SDO.

1.4 Organisation

La suite de cette thèse est organisée comme suit: le chapitre 2 présente l'état de l'art, le chapitre 3 présente le travail réalisé par rapport aux entrées des entraînements afin d'améliorer la détection en hiver. Le chapitre 4 contient des améliorations par rapport au réseau ainsi qu'une fusion de capteur adaptative et robuste aux bris. Par la suite le chapitre 5 présente une méthode adaptée aux dispositifs à faible consommation énergétique pour produire des résultats en temps réel de bonne qualité

et prouvant l'amélioration de la quantité d'énergie régénérée durant le freinage. Cette thèse se conclut ensuite dans le chapitre 6.

Chapitre 2 - Revue de littérature

2.1 Systèmes de détection d'objets (SDO)

2.1.1 Données et entraînements

Pour le premier aspect de cette thèse, soit les données en entrée, un aspect important est de connaître la façon dont les réseaux s'entraînent. En ce qui concerne la détection de l'environnement, plusieurs générations de systèmes ont été présentées à travers les années. Leur méthode d'entraînement est restée relativement la même, soit des méthodes statistiques à rétropropagation. De ce fait, les méthodes d'entraînement satisfont la majorité des cas. Cependant, cela indique que, pour les cas mal représentés dans les jeux de données initiaux, la performance sera détériorée. Les excellents résultats obtenus par ces systèmes de détection sont attribuables à l'utilisation de jeux de données d'images massifs pour les initialiser. Un des tout premiers systèmes était basé sur les histogrammes de gradient (HoG). Ces histogrammes sont définis par leurs valeurs calibrées manuellement. Les réseaux de neurones convolutifs ont ensuite fait leur apparition afin de classifier les images dans différentes catégories [23, 24]. Ces réseaux de neurones utilisent une fenêtre déplaçable sur l'image et appliquent le même patron afin d'extraire celui-ci. Les poids de ces patrons sont définis par une étape d'entraînement dans laquelle les poids sont modifiés par un optimiseur avec rétropropagation. Ces réseaux convolutifs ont été à la tête des performances jusqu'à l'arrivée des réseaux de transformateurs. Les réseaux de transformateurs ont été popularisés par les méthodes de traitement du langage naturel et adapté aux images par sous-sections de l'image. Chaque mot à une valeur de corrélation nommé attention par rapport aux autres mots de la phrase. Le modèle comporte deux sections, l'encodeur et le décodeur. Ces deux sections utilisent

plusieurs blocs de transformateurs. L'encodeur crée une représentation des entrées et le décodeur formule une séquence de sortie appropriée. Trois tâches peuvent être effectuées avec ces réseaux, la tâche de classification/régression est la tâche la plus facile, car le résultat dépend de l'ensemble de l'image. La deuxième tâche est la détection des objets. Dans cette tâche, le réseau doit produire des boîtes et classifier chaque boîte dans une catégorie. La dernière tâche est la segmentation, dans cette tâche, le réseau classifie l'ensemble des points de l'image appartement à une classe et produit optionnellement les boîtes de détection en tâche connexe. Lorsque les éléments d'entraînement appartiennent à une population uniforme de conditions, les résultats sont généralement très performants dans les trois tâches. Lorsque ces réseaux sont confrontés à des environnements en dehors des données d'entraînements, ils réagissent de manière imprévisible selon les patrons identifiés dans l'image. Afin de régler cette situation, il faut entraîner les données sur l'ensemble des conditions météorologiques que le véhicule pourra rencontrer de manière équivalente afin d'éviter le biais vers certaines données. Afin d'apprendre l'ensemble des patrons nécessaire à cette détection indépendante du contexte, les réseaux doivent contenir plus de paramètres, rendant leur exécution plus lente ou incapable de réellement tout apprendre, réduisant le taux de détection [25, 26]. De plus l'annotation et la capture d'autant de données sans biais pour certaines conditions météorologiques ou pour certaines classes d'objets est très coûteux et parfois difficile dû au fait que l'hiver ne représente pas plus que quelque mois par an [27–40]. Les conditions météorologiques pouvant être considérées comme hiver représentent non la saison, mais plutôt la couverture de neige, les obstructions causées par la neige en tombée et le changement d'apparence de certains objets. L'utilisation de données augmentées artificiellement peut combler partiellement le manque de données dans une classe, cependant, il faut

s'assurer que l'augmentation choisie soit adaptée au domaine cible. L'utilisation de réseaux génératifs (GAN) ou de systèmes automatiques d'ajout de transformation d'image permet cette augmentation. Les ajouts de météo artificielle peuvent aussi être faits manuellement. Pour les tâches de classifications et détections, de multiples autres solutions ont été utilisées dans la littérature, telle que les réseaux récurrents à mémoire courte (LSTM) [41], DETR [42], des systèmes de détection manuelle et systèmes novateurs effectuant leurs entrées.

Afin de réduire la taille des jeux de données, les chercheurs à travers le monde ont travaillé sur le transfert d'apprentissage [43–45]. La méthode de transfert la plus courante est l'utilisation d'un préapprentissage dans lequel beaucoup de données sont utilisées pour initialiser des patrons génériques dans le modèle avant l'entraînement réel. L'utilisation des patrons augmente drastiquement l'apprentissage spécifique, car le modèle à un point initial proche de ce qu'il doit apprendre [43, 44]. Xi et al. [44] a pu énoncé qu'un choix approprié d'une sous-section d'un jeu de donnée peut être aussi efficace qu'un très grand jeu de donnée.

Le transfert d'apprentissage à aussi donnée naissance à l'annotation automatique d'image [46–49]. Plutôt qu'un humain identifie les images manuellement, ce qui est un procédé très long et coûteux, un ordinateur déjà entraîné sur un domaine similaire peut identifier les objets et l'humain a donc qu'a confirmer si la détection est réelle ou non afin d'obtenir des annotations beaucoup plus rapidement.

Afin de résoudre les problèmes précédents, une méthode rapide d'identification des données permettant d'avoir des annotations de qualité réduisant ainsi la taille des jeux de données est nécessaire. L'annotation de saillance dans l'image peut être utilisée afin d'identifier une importance à chaque point de l'image lors

de l'entraînement. Les premières itérations de saillance ont été présentées par Grad-CAM [50] afin de montrer l'importance de chaque pixel dans la décision du réseau par progression arrière.

La contribution majeure de cette thèse dans l'aspect des données se situe en la génération de donnée avec saillance selon un procédé simple et rapide. Les données avec saillance entraînent les réseaux plus efficacement, réduisant la taille des jeux de données. L'implémentation de la méthode sera présentée dans le chapitre 3.

2.1.2 Architecture

Deuxièmement, pour l'architecture du réseau, de nombreuses recherches ont été effectuées à partir de perceptron jusqu'aux réseaux transformateurs. Ces réseaux ont montré leur efficacité dans plusieurs tâches reliées à la vision par ordinateur [12]. Cependant, dans les conditions difficiles, les architectures actuelles ont de la difficulté à obtenir un niveau acceptable pour la conduite assistée [14, 51]. Ce niveau acceptable peut être défini de plusieurs façons et est suggestif. On pourrait d'abord définir ce stade comme étant celui où la portée de détection est adéquate pour ralentir le véhicule à partir de sa vitesse de croisière. Une deuxième définition suggestive est l'application directe de la métrique de précision. Cependant, cette version de la définition devient suggestive, puisqu'aucune valeur n'a été identifiée comme ligne directrice en détection. Le spectre visible est grandement affecté par les conditions atmosphériques, telles que la neige, la pluie, le brouillard et la nuit. Des capteurs utilisant différente longueur spectrale, telle que les caméras thermiques et caméras ultraviolettes ont fait leur apparition et sont de plus en plus accessibles. Contrairement au spectre visible, le spectre thermique n'est pas dégradé par les conditions de luminosité variables de l'environnement. Un aspect négatif du spectre

thermique est une faible résolution et un coup de capteur très élevé. L'utilisation simultanée du spectre visible et du spectre thermique permet dans bien des cas d'augmenter la détection des obstacles [51–56].

La vision thermique est reconnue dans les systèmes de vision nocturne [1, 27, 32, 57–67]. Elle est très peu altérée par les conditions, telles que la pluie et la neige, tout en n'étant aucunement altérée par le brouillard ou la nuit. L'inclusion simultanée des deux longueurs spectrale dans le réseau demande plusieurs éléments au niveau de l'architecture du réseau de détection. Le premier élément important est une méthode de fusion temporelle, car les capteurs n'arrivent pas tous simultanés. Le deuxième élément est une étape de fusion des données, puisque le format de chaque donnée est différent. Un troisième élément est une source de donnée contenant des données synchronisées contenant l'ensemble des longueurs spectrales utilisées.

Sachant que le succès des réseaux de neurones artificiels dans les tâches de vision a été causé par la taille imposante des jeux de données. Dû à leur arrivée récente, les jeux de données ayant les longueurs spectrales désirées ont dans l'ordre des dix milles donné plutôt que dans les millions.

Fusion multispectrale

Lors de l'utilisation en inférence des données, une synchronisation temporelle est nécessaire afin de pouvoir exécuter le fusion. Lorsque la situation le permet, la meilleure synchronisation possible est la synchronisation matérielle dans laquelle la capture des images est effectuée en réponse à un signal reçu. Cependant, ce ne sont pas tous les systèmes qui sont capables d'utiliser cette synchronisation. Une autre méthode de synchronisation populaire est l'attente de l'ensemble des

données. Cette méthode est facile à mettre en œuvre, car elle ne nécessite pas de compétence technique avancée. Cette méthode introduit beaucoup d'imprévisibilité dans la latence. Une autre méthode est l'exécution de la fusion à un pas de calcul fixe. Lors de l'utilisation d'un pas de calcul constant, il est possible d'observer un retard plus important que la méthode précédente. Un pas de calcul trop faible causera des fusions en échec dû à une donnée manquante. Lorsque la situation n'est pas en temps réel, beaucoup de méthodes existent afin d'essayer les différentes combinaisons de données afin de diminuer la latence au maximum. Cependant, en temps réel, tel qu'une voiture, une méthode plus simple est requise. Cette thèse propose dans le Chapitre 4 une combinaison novatrice de fusion.

Concernant la fusion des données, il existe plusieurs familles de fusion. La première grande famille est la fusion de bas niveau, dans cette fusion l'ensemble des données est fusionné à partir des données brutes des images. La fusion moyenne est effectuée sur des patrons détectés par intelligence artificielle [68]. La fusion de haut niveau est la fusion des résultats. Dans le cas de la détection d'obstacles, la fusion de haut niveau est effectuée sur les boîtes de détection. Lors de conditions météorologiques difficiles, la fusion de haut niveau donne de faibles résultats, car elle est incapable d'utiliser de l'information de plusieurs longueurs spectrale simultanément [69]. Les méthodes moyennes ont démontré les meilleurs résultats en fusion. Cependant, les méthodes actuelles sont indépendantes de la détection et ajoute de la latence et du coût de calcul supplémentaire [57]. Ces méthodes représentent de manière fiable les détails de position, un coût de calcul assez faible comparé aux deux autres familles et une certaine robustesse à une imprécision de calibration, tant que les données entraînées proviennent du même ensemble de capteurs. La calibration est une opération de modification du système de coordonnée des points formant l'image afin d'obtenir un

système de coordonnées uniforme entre les différentes images.

Cette thèse propose une méthode de synchronisation à faible coût afin de pouvoir être effectuée en temps réel ainsi qu'une architecture de réseau de détection combinant les fusions moyennes et détection en un seul réseau à faible coût pour pouvoir être sur un ordinateur de frontière (Edge AI). Le réseau est résistant aux coupures de signal en entrée et adapte les poids en temps réel afin de toujours obtenir un bon résultat lors de ces fusions. L'intégration de la fusion dans le réseau permet de réduire la latence d'un réseau externe de fusion, d'effectuer une fusion plus adaptée aux besoins de détection plutôt qu'aux besoins d'affichage humain ainsi que de réduire la quantité de calcul et de paramètres nécessaire pour l'exécution de l'opération.

2.1.3 Amélioration temps réel

Dernièrement, avec des données de qualité et une architecture améliorée, vient le temps d'intégrer l'énergie et la planification. Un des aspects manquant avec ces améliorations est l'aspect requis par les systèmes d'autonomie de niveau 4 afin d'indiquer que le véhicule a besoin d'une conduite humaine plutôt qu'assistée. Sans compter l'aspect sécurité sur la quantité d'énergie, un autre aspect important est l'optimisation énergétique permettant d'avoir ces capteurs sans perte d'autonomie. Les systèmes assistés par ordinateur peuvent effectuer simultanément les tâches de sécurité et efficacité énergétique [70–73]. Ces systèmes prennent en compte l'ensemble de l'énergie du véhicule, incluant celle des capteurs et du mouvement du véhicule afin d'optimiser les choix en restant sécuritaire. Beaucoup de réseaux de détection existent ayant des performances de détection et de segmentation de différent niveau.

Le niveau de performance du détecteur est en lien direct avec la quantité d'énergie pouvant être régénéré par freinage optimisé. Les recherches en optimisation des freinages démontrent qu'une plus grande distance de freinage permet de régénérer plus d'énergie [74]. Un problème majeur des méthodes d'entraînement et de détection des réseaux de détection actuels est l'absence d'importance à la distance de l'objet. Les objets à faible distance sont généralement très bien perçus, tandis que les objets à grande distance et ainsi petit dans l'image sont généralement ignorés [75]. Les métriques d'entraînement ne donnant pas d'importance à la distance sont la principale cause.

Concernant la détection de l'environnement, plusieurs générations de systèmes ont été présentées à travers les années telles qu'énoncées dans la section 2.1.1. Pour l'intégration, la tâche de segmentation est utilisée. Afin d'évaluer la performance de segmentation, la métrique de moyenne d'interception sur union (mIOU) est utilisée:

$$mIOU = \frac{\sum_{c=1}^{C} \frac{intersection}{union}}{C} \tag{2.1}$$

. Dans cette métrique, $intersection$ représente l'intersection entre les résultats prédis et les annotations, tandis que $union$ représente l'union entre ces informations. c représente un indice de classe pour l'ensemble des classes (C) reconnu par le réseau de segmentation.

L'ensemble de tous les modèles comme mentionné dans la section 2.1.1 souffrent d'un problème de généralisation. Puisque l'entraînement est fixé sur le jeu de donnée utilisée lors de l'entraînement et sur l'architecture utilisée, deux réseaux ayant la même métrique de performance peuvent en réalité avoir des résultats de performance différente sur certaines images. Un problème de toutes ces méthodes est l'absence

d'indicateur de performance intégré au réseau. Un résultat de segmentation modélisé de 90% n'indique rien si lors de conditions réelles ne peuvent pas indiquer les 10% d'échec. Le réseau se doit de pouvoir indiquer les moments dans lesquels la confiance envers le résultat est faible.

Afin d'optimiser les détections et l'énergie en toutes circonstances, un réseau ayant de bonnes détections à grande distance est requis. Mais un réseau fournissant un indice de confiance envers le résultat est requis pour la sécurité. Étant donnée l'absence de réseau pouvant fournir une bonne détection dans toutes les circonstances, plusieurs réseaux sont requis et entraînés sur les différentes conditions. Afin d'utiliser le bon réseau dans la bonne situation, un métaréseau sélecteur est nécessaire afin de bien choisir quel détecteur utiliser.

Dans cette thèse, un réseau classificateur sur lequel la couche de normalisation finale est retirée sert de métaréseau afin de modéliser le résultat de chaque option. Cette modélisation permet d'augmenter la performance de détection et d'identifier les cas où la conduite assistée est indisponible. Ce réseau produit simultanément un choix de réseau pour optimiser les résultats et un indice de performance pour désactiver le mode assisté. Ceci produit de meilleurs mIOU et une portée de détection accrue dans l'ensemble des contextes.

2.2 Optimization énergétique

2.2.1 Freinage régénératif

Le freinage régénératif est un freinage dans lequel le moteur du véhicule électrique agit comme générateur afin de récupérer l'énergie cinétique du véhicule et ainsi augmenter l'autonomie dudit véhicule [76]. Plusieurs facteurs affectent le freinage

régénératif, tel que l'efficacité du moteur à différentes vitesses rotationnelle [77]. L'efficacité de stockage du médium d'énergie, en l'occurrence, une batterie, est aussi nécessaire à des fins de freinage efficace. Il est important de noter qu'à faible vitesse le moteur, utilisé en mode régénératif, ne peut fournir un courant suffisant à la recherche de la batterie du véhicule. Yang et al. [78] ont fait l'analyse du freinage en séquence et du freinage en parallèle et ont conclu que le freinage en séquence, soit un freinage appliquant le freinage mécanique que pour les freinages demandant une valeur supérieure au maximum de régénération était la solution à utiliser.

2.2.2 Optimisation

L'optimisation du freinage peut se faire en plusieurs moments. Premièrement, le freinage peut être optimisé en mode hors ligne par des systèmes d'optimisation globale, tel que la programmation dynamique [79]. Les systèmes LSTM peuvent également être utilisés dans les déplacements répétés pour optimiser la consommation énergétique du véhicule pour un trajet donné. L'optimisation peut aussi être appliquée en temps réel de multiple façon. Les méthodes les plus courantes sont des systèmes à règles fixes, mais il en existe d'autres. Ces systèmes ultras simplifiés cherchent à trouver un effet de cause-conséquence directe entre les actions et l'énergie. En ce qui concerne les systèmes temps réel, on peut distinguer deux autres catégories : les systèmes recommandés et les systèmes autonomes. Les systèmes suggestifs et autonomes utilisent les mêmes algorithmes avec différence d'application directe ou simplement suggérés au conducteur. Parmi ces méthodes, la prédiction de contrôle optimale ou la minimisation de consommation de carburant équivalente (ECMS) [80] tendent vers l'optimalité tout en offrant des courbes en temps réel. Les systèmes de contrôle prédictif et systèmes d'efficacité énergétique utilisent des modèles simplifiés

afin de pouvoir utiliser des algorithmes d'optimisation plus complexe au détriment de la fidélité du modèle [81].

L'ensemble de ces systèmes utilisent la liberté supérieure obtenue par une détection à grande distance afin d'optimiser les déplacements du véhicule [81]. Afin d'obtenir des performances en temps réel et optimisé, la programmation dynamique approximative est généralement utilisée. Lorsque la technologie le permet, les systèmes de communication entre différents véhicules offrent les meilleures détections de l'environnement à grande distance ainsi qu'une approche collaborative pour l'optimisation de la vitesse. Cependant, ces systèmes requièrent une communication permanente entre différents acteurs qui ne peut être assurée en permanence due à des véhicules non autonomes ou à des routes trop éloignées pour posséder une architecture de communication.

Cette thèse propose dans le Chapitre 5 une méthode d'optimisation à court terme en temps réel basé sur une optimisation globale généralement appliqué en mode hors-ligne dû à son temps de calcul.

Chapitre 3 - Amélioration des entraînements

Afin d'améliorer les entraînements, les données doivent être améliorées. La première étape afin d'améliorer les entraînements est un outil d'annotation. L'utilisation de masques de segmentation a grandement amélioré les entraînements dans des réseaux, tels que MaskRCNN [82]. La saillance est une étape ultérieure aux masques, puisqu'elle permet d'associer un poids d'importance à chaque pixel de l'image et permet donc au réseau de prioriser certains patrons lors de l'entraînement. L'outil d'annotation de la saillance (OAS) sera utilisé pour mettre en évidence les données [1]. Par la suite, une méthode d'entraînement utilisant au mieux les nouvelles données sera présentée. Cette partie de mon étude a été antérieurement diffusée dans la référence [1].

La détection d'objet est généralement effectuée en identifiant des boîtes englobantes autour des dits objets afin de guider le réseau à qui en apprendra la forme. Cependant, cette approche nécessite beaucoup de données adverses afin de reconnaître les parties de la boîte appartenant à notre objet et ignorer les parties appartenant à l'arrière-plan. La limitation d'un masque de segmentation est qu'il ne donne pas d'importance aux éléments à l'intérieur de notre objet. La saillance ajoute ici cette information additionnelle d'importance. En segmentation, chaque pixel de l'image est assigné à une classe. En saillance, chaque pixel est plutôt associé à une valeur continue entre zéros et un représentant l'important du point. Cette valeur prioritaire guide le réseau à identifier les aspects les plus significatifs des objets et à ignorer les obstacles recouvrant l'objet. Cette méthode applique aussi une importance inférieure aux éléments tels que les logos de compagnie afin de guider le réseau à les ignorer. Je propose une méthode de transfert d'apprentissage afin d'utiliser cette information. Ces méthodes sont très connues en réseau de neurones

dû à leur utilisation fréquente. La méthode la plus utilisée est le préentraînement dans lequel le réseau est entraîné en deux temps. Une autre méthode est l'annotation automatique en utilisant des réseaux enseignant/étudiant. Dans cette thèse, les deux méthodes seront combinées. Un ensemble d'experts en réseau de neurones identifient la saillance dans l'image afin de guider, voire biaiser l'entraînement vers les pixels les plus importants des objets. Un entraînement multisource est requis attribuable aux multiples enseignants. Pour combler ce point, une moyenne de la saillance de chaque source est utilisée [83]. Avec les cartes de saillance créées, les étapes d'entraînement sont présentées dans la figure 3.1. On peut justifier l'utilisation d'experts humains pour concevoir des cartes de saillance en raison du fait que les méthodes de réseaux neuronaux cherchent à imiter l'être humain. Les patrons importants pour l'humain sont aussi ceux importants pour le réseau de neurones. Cependant, l'ajout d'humain dans l'entraînement empêche l'ajout dynamique de donnée popularisé par les systèmes enseignant/étudiant.

3.1 Outil d'annotation de la saillance (OAS)

L'utilisation de la saillance devient donc très adaptée à l'hiver. En hiver, les occlusions telles que la neige, ne doivent pas être apprises, car elles ne font pas partie de l'objet. La saillance applique donc une valeur d'importance proche de zéro pour ces valeurs. L'outil d'annotation se base sur le suivi du regard humain. L'être humain a tendance à porter une attention soutenue sur les éléments clés et à détourner rapidement son regard des détails secondaires.

Sachant que le suivi du regard est requis pour extraire la saillance à partir d'un humain, un outil est nécessaire. L'utilisation de caméra pour le suivi du regard est très courante, mais leur précision n'est pas suffisante. Cela dépend de la résolution

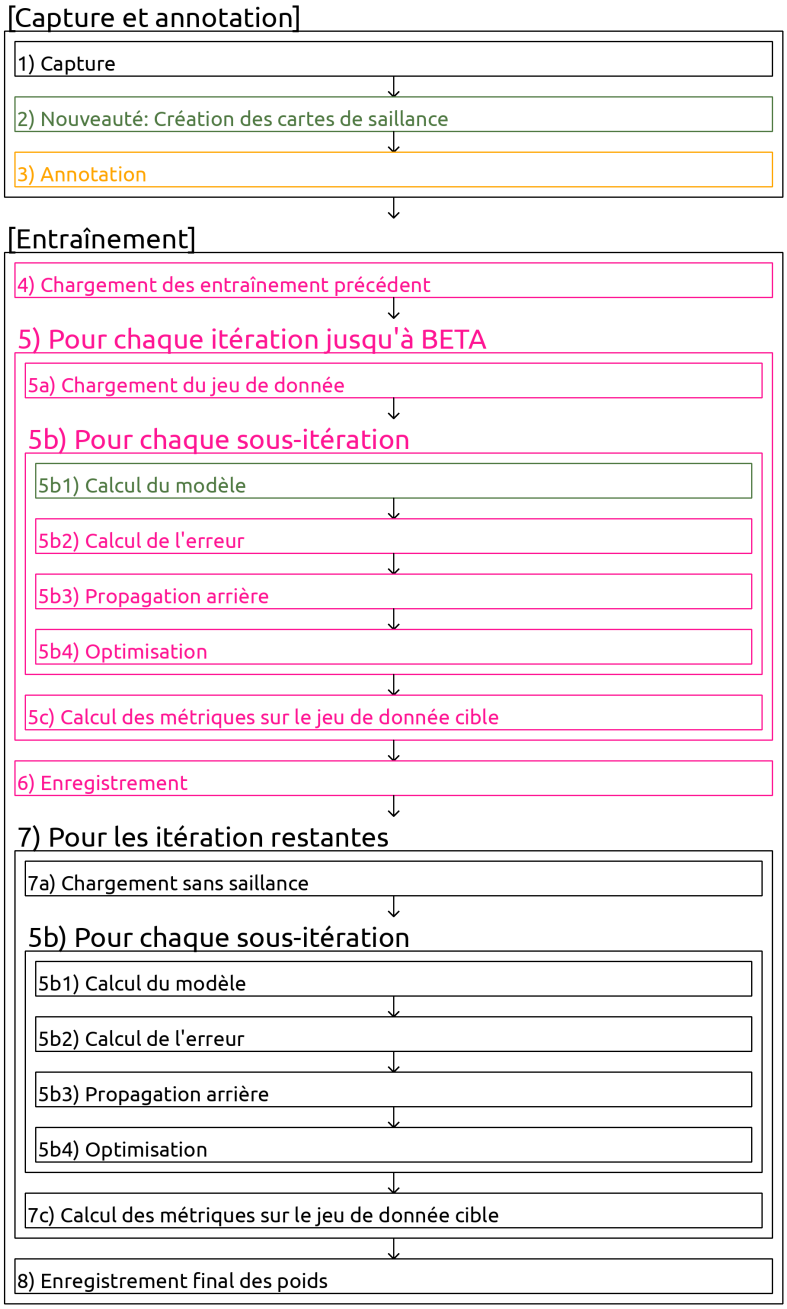


Figure 3.1 Le procédé d'entraînement

de la caméra, de l'illumination et de la position de l'environnement [84]. À cet égard, l'utilisation d'un logiciel et d'une souris est plus adaptée. L'outil utilise un voile noir sur l'image avec un trou de vision. Ce trou se déplace avec le curseur de la souris pour faciliter le suivi. En hiver, les éléments généralement ignorés sont les occlusions. En cachant les éléments non utiles à la détection, comme les plaques d'immatriculation, les publicités sur les autobus ou la neige couvrant les fenêtres, le réseau peut être guidé à apprendre les patrons les plus importants à l'identification de l'objet. Cela diminue donc le nombre de données requises par biais du jeu de donnée. Puisque le trou de vision est une surface plutôt qu'un seul point, une fonction d'importance en fonction de la distance est requise. Pour l'humain les récepteurs sont plus sensibles dans une région circulaire appelée la fovéa [85]. Il n'y a pas de formule standard pour décrire la densité des récepteurs, alors ceci est approximé par la formule empirique suivante:

$$\alpha_{\Delta s} = \begin{cases} 2 \times (1 - \frac{1.5^{(8\times\Delta s-1)}}{1.5^{(8\times\Delta s-1)}+1}) & \text{si } \Delta s < 1 \\ 0 & \text{si } \Delta s \geq 1 \end{cases} \tag{3.1}$$

Dans cette équation, $\alpha_{\Delta s}$ représente le facteur de distance par rapport au curseur et Δs représente la distance normalisée. $1.5^{(8\times\Delta s-1)}$ est une fonction sigmoïde avec valeur expérimentale. Δs est définie par:

$$\Delta s = \frac{\sqrt{(\Delta x)^2 + (\Delta y)^2}}{r} \tag{3.2}$$

où r est le rayon de vision, Δx représente la distance en x définie en pixels et Δy celle en y. Dans nos expérimentations, ceci a été fixé à 100 pixels sur un écran de 1920 pixels de large. Le facteur peut être vu dans la figure 3.2.

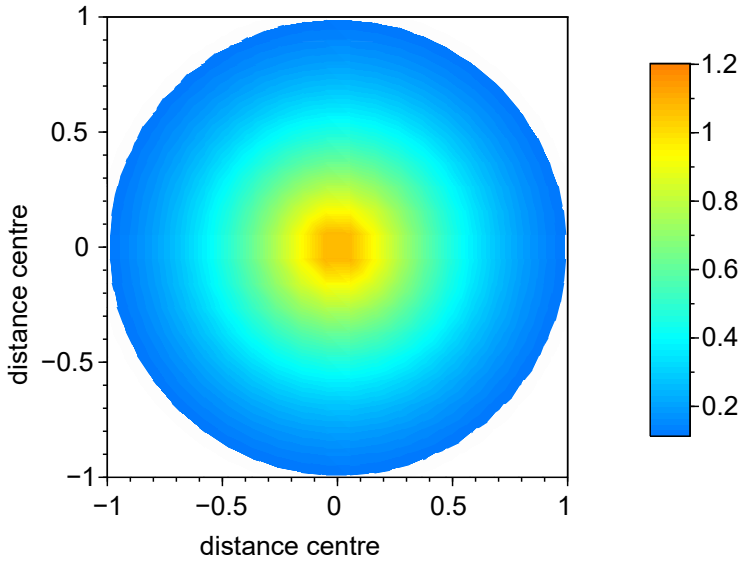


Figure 3.2 Représentation visuelle du facteur de saillance selon l'équation 3.1

3.1.1 Cartes de saillance

Lors de l'annotation, les images sont affichées en utilisant le voile et la fonction d'importance telle que visibles dans la figure 3.3. L'utilisation de ce trou de vision permet d'être précis dans le suivi du regard. L'outil enregistre la durée multipliée par le facteur de distance durant la durée de l'enregistrement. Une moyenne de plusieurs individus sera ensuite effectuée pour obtenir une carte de saillance. Un exemple de résultat est présenté dans la figure 3.4. Dans cette figure l'effet de la neige est démontré. De la neige partiellement fondue est visible au milieu de l'image, rendant la vision frontale floue. Les objets lointains sont recouverts d'une obstruction semblable à un brouillard. Les voitures sont partiellement recouvertes de neige et les lignes au sol sont recouvertes. Dans cette thèse, seuls les objets de la route sont considérés, car le travail requis pour détecter les lignes au sol dans de telle condition est un travail de thèse à part entière. La sommation de l'importance de chaque point est effectuée

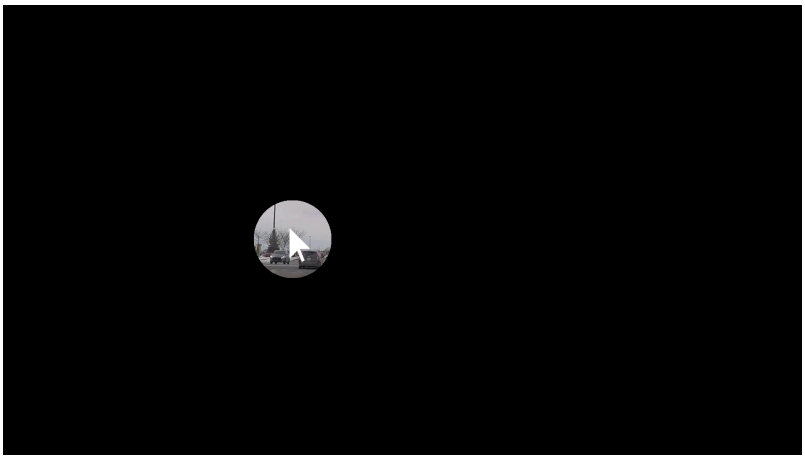


Figure 3.3 Exemple du trou de vision

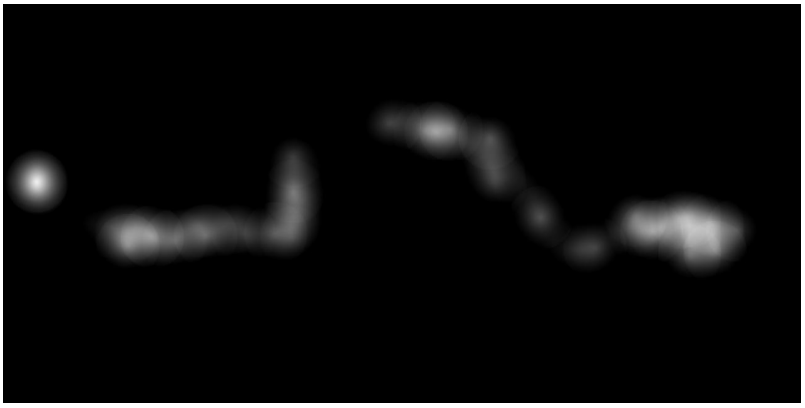
selon la formule suivante:

$$T_{x,y} = \begin{cases} \sum_k \Delta t_k \times \alpha_{\Delta s}(x, y, C_{x,k}, C_{y,k}) & si < A_{max} \\ A_{max} & sinon \end{cases} \tag{3.3}$$

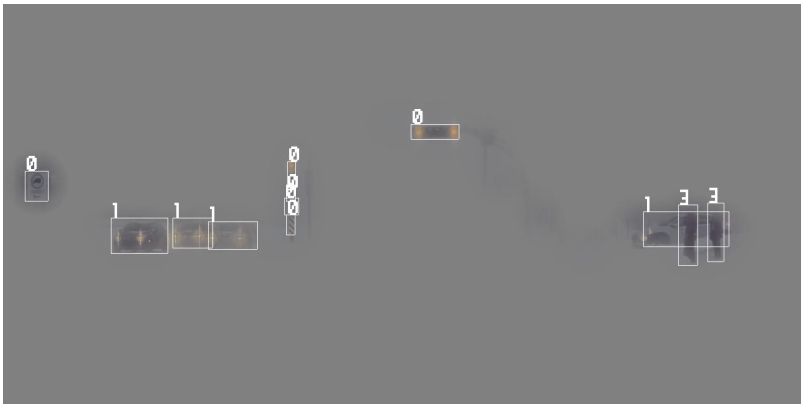
dans laquelle $T_{x,y}$ est le temps en seconde pour un point x, y, k est un mouvement du curseur, Δt_k est la durée ou le curseur est resté à un point précis, soit k multiplié par $\alpha_{\Delta s}$. $C_{x,k}$ et $C_{y,k}$ sont les centres du regard. Une valeur maximale de 5 secondes a été appliquée pour empêcher de mettre trop d'importance sur un point précis.

Une fois la carte de saillance créée, elle est normalisée entre zéros et un afin d'obtenir des pourcentages d'importance et que la durée totale d'annotation soit négligée. La normalisation est effectuée avant la moyenne des différents individus afin qu'ils aient une importance équivalente dans le résultat. Les valeurs d'importances sont ensuite converties en valeur de transparence et une couche grise de valeur 50% dans chaque couleur est appliquée en utilisant ce masque de transparence et l'image initiale.

(a) Image initiale.



(b) Saillance.



(c) Amelioration.

Figure 3.4 Résultat de l'application de saillance sur une image

3.1.2 Boîtes englobantes

Une deuxième partie du travail effectué par l'OAS est l'amélioration des boîtes englobantes des objets. Lors de l'ajout de la saillance à des données déjà annotées, tel que la base de données Berkeley Deep Drive (BDD), il est possible de recadrer ces boîtes par rapport à la saillance obtenue. Ceci est utile lorsqu'une partie du véhicule contenait un élément non essentiel à la détection, tel qu'un panneau publicitaire sur le toit, du style taxi. Dans cette thèse, la méthode X-Means [86] est utilisé pour obtenir des propositions de nouveaux cadres. Cette méthode, contrairement aux outils de proposition de cadre dans les réseaux de neurones, ne demande aucun entraînement, mais plutôt une fonction de distance. La fonction de distance utilisée doit prendre en compte les positions dans l'image et la saillance afin de créer des regroupements pouvant être des cadres. La fonction de distance utilisée est la distance de Minkowski:

$$\Delta_{MINKOWSKI} = \sqrt[p]{\omega_x \times |X_2 - X_1|^p + \omega_y \times |Y_2 - Y_1|^p + \omega_z \times |Z_2 - Z_1|^p} \tag{3.4}$$

où

$$0 \leq \omega_z \leq \ ({I_w}^2 + {I_h}^2) \tag{3.5}$$

$$|X_1 - X_2| \ \leq \ I_w \tag{3.6}$$

$$|Y_1 - Y_2| \ \leq \ I_h \tag{3.7}$$

$$|Z_1 - Z_2| \ \leq \ 1 \tag{3.8}$$

$$\omega_x \ = \ 1 \tag{3.9}$$

$$\omega_y \ \in \ \{1, I_w/I_h\} \tag{3.10}$$

Tableau 3-1 Paramètres de Minkowski utilisé dans cette thèse.

p	wx	wy	wz
1	1	1	3×255
1	1	I_w/I_h	$1 \times I_w$
4	1	1	3×255
4	1	1	4×255
4	1	I_w/I_h	$1 \times I_w$

où I_w est la largeur de l'image et I_h est sa hauteur.

Idéalement, chaque élément du vecteur de coordonnée pour Minkowski est d'échelle comparable. Dans le cas présent, un biais vers la saillance est nécessaire pour que x-Means puisse produire des cadres. Un ω_z plus grand que la largeur de l'image et inférieur à la diagonale est donc requis pour créer ce biais. L'opération est exécutée avec cinq combinaisons de paramètres telles qu'énoncées dans le tableau 3-1. Ces cinq combinaisons ont été choisies empiriquement pour leur similarité avec les boîtes d'origine.

Les améliorations aux cadres sont effectuées par le jumelage entre les boîtes existantes et celles proposées par X-Means. Le tout est effectué en quatre étapes:

1. Identification des cadres proposés ayant une intersection sur union (IOU eq.3.11) supérieure à un facteur TU et les accepter comme nouvelles boîtes.
2. Identification des cadres proposés ayant une intersection sûr plus petite (IOS eq.3.12) supérieure au facteur TS et agrandir la boîte initiale pour contenir la boîte proposée.
3. Identification des boîtes totalement incluses et diminuer la taille, effectuer seulement si la nouvelle taille est plus grande qu'un facteur TDiminution.
4. Conserver les boîtes n'ayant pas été affectés.

$$IoU\ (A,B) \quad = \quad \frac{A \cap B}{A \cup B} \tag{3.11}$$

$$IoS\ (A,B) \quad = \quad \frac{A \cap B}{\min\{A,B\}} \tag{3.12}$$

La méthode a aussi été adaptée pour les jeux de données non annotés, où l'humain doit proposer une boîte originale. L'être humain est capable de créer une boîte imprécise, car l'algorithme x-Means l'affinera.

D'autres approches ont été étudiées. Les propositions de boîte par réseau de neurones demandent un entraînement profond et de savoir la véritable boîte idéale, rendant le processus extrêmement long. Pour les approches en logique floue, leurs utilisations au moment de la thèse restaient des règles floues codées par l'humain, ce qui requiert des calibrations empiriques et un temps considérable pour son développement. Contrairement à ces approches, X-Means n'as pas d'assomption d'homogénéité des données et des changements requis aux boîtes, l'amélioration est effectuée uniquement pas regroupement des points saillants.

3.2 Entraînement

Avec les données annotées par la saillance, une méthode spéciale d'entraînement est requise. Les résultats sont validés sur plusieurs réseaux de détection ont été utilisés sous forme de boîte noire. Les réseaux YOLOv3 [87], SSD [88] et Faster-RCNN ont été sélectionné pour leur pertinence au moment où la recherche a été effectuée. L'utilisation de plusieurs réseaux provenant de plusieurs familles permet de démontrer la pertinence de la saillance dans l'entraînement.

Comme protocole d'entraînement, quatre types de donnée existent. En combinant ces données, nous arrivons à 15 combinaisons. Seul six de ces combinaisons sont aptes à l'entraînement. Elles sont définies dans le tableau 3-2 et sont utilisées pour l'entraînement des réseaux. Dans ce tableau, la colonne source représente la source des données contenues dans le jeu, soit Berkeley Deep Drive (BDD), UQTR ou Russian traffic sign dataset (RTSD). La colonne saillance représente si les données avaient de la saillance. La taille représente le nombre d'images et la colonne conditions contient les données météorologiques contenues. En entrée, il y a les données de BDD avec et sans saillance, ainsi que les données annotées à l'UQTR avec et sans saillance. Une combinaison supplémentaire est l'ensemble des données avec saillance (JDS) et données sans saillance (JDR).

L'entraînement est effectué pendant un taux β d'itérations sur une base de données avec saillance et puis $(1-\beta)$ itérations sur JDR. La première étape est le préentraînement dans lequel les patrons sont initialisés dans le modèle et où la saillance est le plus utile. Les étapes sur un jeu de donnée sans saillance par la suite permettent d'assurer son utilisation dans des conditions réelles. Cette méthode est similaire au préentraînement utilisé dans la littérature, à l'exception que le jeu de donnée de saillance n'est pas plus volumineux que le jeu sans saillance. L'importance du JDR permet de diminuer le biais induit par la saillance et présente des données adverses afin d'augmenter la généralisation de la base de connaissance. Pour ma thèse, j'ai fixé β à 80%. Pour l'évaluation des résultats, les métriques standard de précision, rappel et précision moyenne ont été conservées. La métrique F_1-score (3.15) a été ajouté afin de prendre la précision et le rappel en une seule formule.

$$precision = \frac{correct \cap identifies}{identifies} \tag{3.13}$$

$$rappel = \frac{correct \cap identifies}{correct} \tag{3.14}$$

$$F_1 - score = 2 \times \frac{precision \times rappel}{precision + rappel} \tag{3.15}$$

Dans eq.3.13, correct représente l'ensemble des données correctement identifiées, identifies est le nombre de résultats fournis par la méthode. La figure 3.1 montre l'ensemble des étapes de l'entraînement. Dans cette figure les parties en noir sont des parties régulières de l'entraînement, celles en vert sont des parties ajoutées et celle en rose provient du préentraînement. Il est donc possible d'améliorer les données en utilisant la saillance et d'améliorer les résultats de n'importe quel système de détection.

Le temps requis pour annoter une image en saillance est de 30 secondes pour l'humain et un temps de calcul de trois minutes pour l'ordinateur lors de l'utilisation de K-Means. L'annotation des boîtes sur une image annotée de saillance prend en moyenne 10 secondes, soit 20 secondes de moins que l'identification sur une image sans saillance. Ceci indique que pour 10 secondes supplémentaires par image, il est possible d'obtenir des images de meilleure qualité réduisant le nombre de données requis.

Tableau 3-2 Les différents jeux de données

ID	Source	Saillance	Taille	Précision	Rappel	Max F1	Conditions
DSA	all	R+P	14381	58	98	68	Sans obstruction / Été Hiver (Léger+Fort)
BDD	BDD	R+P	10999	–	–	–	Sans obstruction / Été Hiver (Léger)
BDDP	BDD	P	1000	60	98	75	Sans obstruction / Été Hiver (Léger)
DSP	all	P	2691	52	98	66	Sans obstruction / Été Hiver (Léger+Fort)
UQTR1P	UQTR	P	1691	60	98	72	Sans obstruction / Hiver (Fort) Sans obstruction / Hiver (Fort)
UQTR1R	UQTR	R	1691	43	78	53	Sans obstruction / Hiver (Fort) Sans obstruction / Hiver (Fort)
BDDR	BDD	R	9999	41	78	52	Sans obstruction / Été Hiver (Léger)
DSR	all	R	11690	–	–	–	Sans obstruction / Été Hiver (Léger+Fort)

3.3 Inférence

Lors de l'utilisation du système de détection en temps réel, la saillance n'est pas prise en compte. La saillance aura été utilisée principalement afin d'augmenter la qualité de la base de connaissance et n'est donc pas utile en temps réel.

3.4 Résultats

En suivant la méthode de saillance proposée, un jeu de donnée de 1691 images a été généré. 5160 images ont été capturées et une image sur 20 a été conservée pour le jeu de donnée. 500 images ont été ajoutées provenant de RTSD et 500 images provenant d'autres voitures. Seize experts ont été consultés pour créer les cartes de saillance. Cette base de données est orientée conduite et contient quatre classes d'objets. Les quatre classes sont les signalisations, les véhicules motorisés, les véhicules à deux roues et les humains. Les quatre catégories sont en ordre de danger allant du danger à l'égo-véhicule jusqu'a danger pour l'autre acteur. Dans le cas des humains, ce sont les humains qui sont en danger. Les catégories ont été choisies en fonction de l'action

qui doit être prise plutôt que l'apparence de l'objet.

Un exemple des données capturé et annoté est présent dans la figure 3.5. Cette image a été capturée et directement été annoté avec l'outil d'annotation de la saillance permettant une annotation en 40 secondes. Comme vu dans la figure, les boîtes sont similaires à celle de la figure 3.4. Ces similitudes indiquent une utilisation possible des deux types de données en coopération. Cette image montre que les fenêtres et les plaques d'immatriculation sont cachées afin de ne pas être utilisées dans l'entraînement, car la forme extérieure du véhicule est plus importante. Cette base de données est publiée sur notre GitHub à `https://github.com/irh-ca-team-car/attention-data`.

Afin d'évaluer la performance des nouvelles données, une démonstration des différences par rapport aux données originales est importante. Dans un premier temps les résultats sur les boîtes de détection sont présentés. L'outil d'annotation de saillance (OAS) permet de recalculer et d'améliorer les résultats de ces boîtes. L'amélioration est visible dans la figure 3.6. Dans cette figure il est possible d'apercevoir la carte de saillance associée à l'image et les améliorations des boîtes. Le changement dans les boîtes est de moins de 10% de leur taille et de leur position. Dans la figure 3.6(b), l'ensemble des propositions de boîte faites par X-Means sont présentées. Les boîtes sont colorées pour les différencier indépendamment de leur classe d'objets. X-Means a proposé beaucoup de boîte dans le coin supérieur droit, mais l'absence de données en entrée permet de les ignorer. La camionnette de police de cette figure a le plus grand changement dans sa boîte, car l'humain a porté plus attention à la camionnette qu'a la ventilation sur le toit et les roues. Les petits véhicules ne sont pas améliorés, car les boîtes proposées ne correspondaient pas

(a) Image initiale

(b) Saillance + annotation

Figure 3.5 Exemple d'image de UQTR1.

Tableau 3-3 Les différentes classes

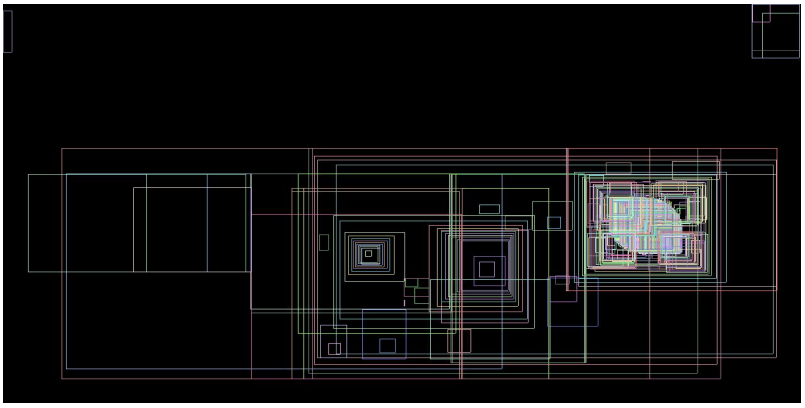
Indice	Descriptipn	Détail
0	Signalisation	Panneaux, feu de signalisation, cône
1	Véhicule motorisé	Voiture, camion, bus
2	Deux routes	Vélo, Moto
3	Humain	Piéton, mannequin

aux critères. Les numéros sur les boîtes sont les classes telles que définies dans le tableau 3-3.

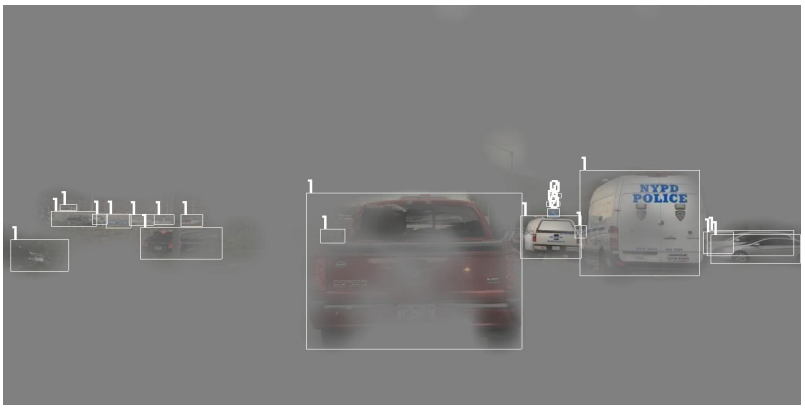
Le réseau est ensuite entraîné sur ces données en utilisant le procédé en deux temps énumérés précédemment. La figure 3.7 montre plusieurs métriques en fonction des itérations d'entraînement (Epoch). YOLOv3 entraîné sur les données sans amélioration sature à 59% précisions et 87% rappels. Celui entraîné sur la saillance sature plutôt à 74% précisions et 98% rappels. Il est à noter que les deux jeux de données contiennent les mêmes images avec et sans l'amélioration due à la saillance.

Les entraînements ont été effectués sur plusieurs configurations dans Pytorch. L'ensemble des optimiseurs ont été utilisés ainsi que plusieurs taux d'apprentissage. Les figure 3.8-3.9 montrent les résultats moyens obtenus. Figure 3.8(a) contient les résultats de précision et figure 3.8(b) et les résultats de rappel pour l'entraînement. Finalement, la figure. 3.9(a) représente les résultats F_1-score (3.15) et la figure. 3.9(b) contient une courbe précision/rappel des résultats. Ces figures sont une moyenne des trois réseaux YOLOv3, SSD, et Faster-RCNN ainsi que l'ensemble des optimiseurs. Les courbes bleu (BDDR) et rose (UQTR1R) montrent les données sans saillance. Les courbes noires (BDDP) et rouge (UQTR1P) sont les données avec saillance. Et finalement les courbes turquoise (DSP) représentent l'ensemble des données contenant de la saillance et les vertes (DSA) représentent tous les jeux de données combinés. Il est important de noter que les données de BDD contiennent quelques

(a) Image initiale.



(b) Saillance.



(c) Amelioration.

Figure 3.6 Amélioration des annotations pour $T_U = 0.9, T_S = 0.9, T_3 = 0.25$.

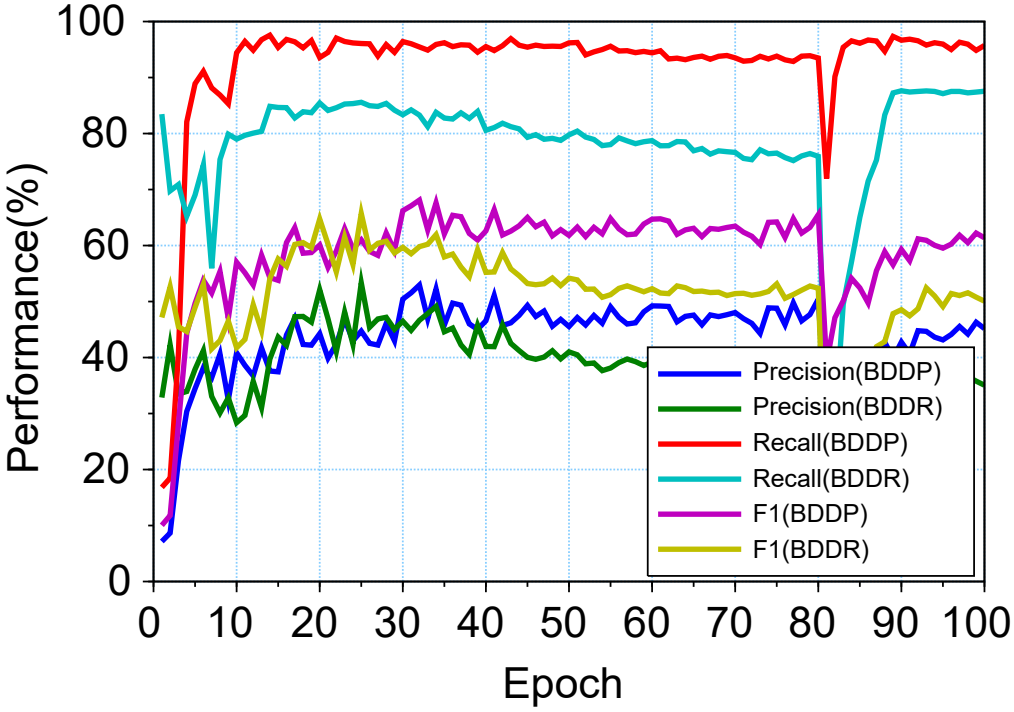
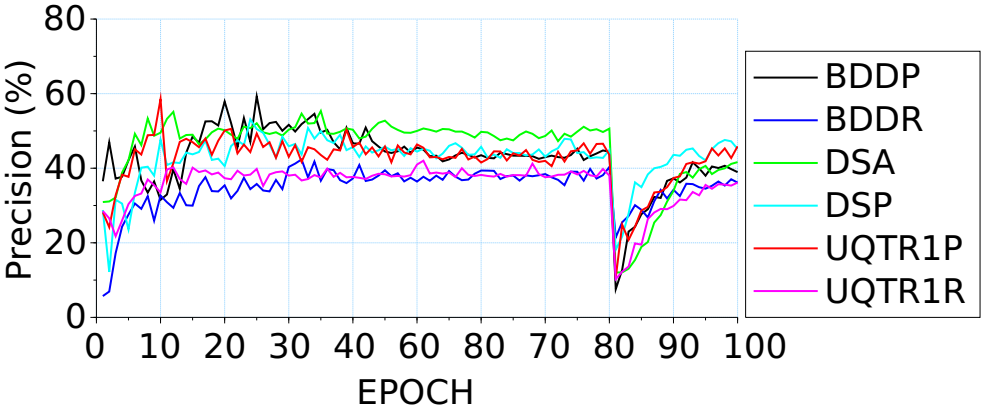
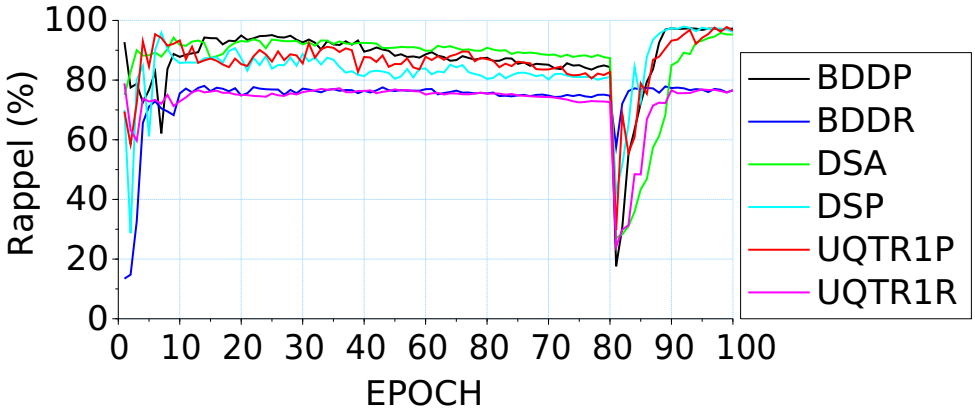


Figure 3.7 Courbes de précision, rappel et F1 pour l'entraînement sur données standard vs données améliorées. Évalué sur le jeu d'entraînement. $\beta = 80\%$.

rares images d'hiver alors que UQTR1 ne contient que des données hivernales. Les données de BDD ont été prises par une équipe de l'université de Berkeley dans la ville de New York où l'importance de l'hiver est moindre qu'au Canada. Ceci rend difficile la détection en hiver canadien et cause des performances moindre lorsqu'évalué sur ces conditions. Dû à la quantité d'inconsistance de donnée dans les différents jeux de données, une quantité de mille images a été utilisée lors des entraînements afin que tous les jeux de données soient de taille comparable. Puisque le jeu DSA contient des données avec et sans saillance, une proportion de 50% de chaque type est utilisé. Ceci indique aussi que seul 50% des données est requise pour faire un entraînement avec ce type de donnée. Cette diminution de moitié permet d'alléger le coût d'annotation et valorise grandement une quantité de données inférieures. La saillance est une méthode utilisable à grande échelle. Elle montre des bénéfices lorsque les jeux de données sont trop petits pour converger vers un modèle efficace. Les jeux de données contenant de la saillance, soit DSP, DSA, BDDP et UQTR1P ont obtenu 1 meilleur F_1-score (3.15). L'utilisation de données d'hiver tel que UQTR1P a fourni une amélioration du nombre d'itérations nécessaire passant de 25 à 9 itérations pour atteindre le maximum. DSA, qui contient les deux types de données ainsi que des données d'été et d'hiver, démontre le meilleur F_1-score (3.15) en un petit nombre d'itérations. Il est important de noter que le changement radical à l'itération 80 est dû au changement de donnée d'entraînement pour diminuer le surentraînement. Toutes les valeurs de validation ont été calculées sur le jeu de donnée UQTR1R afin d'en évaluer les performances en hiver.

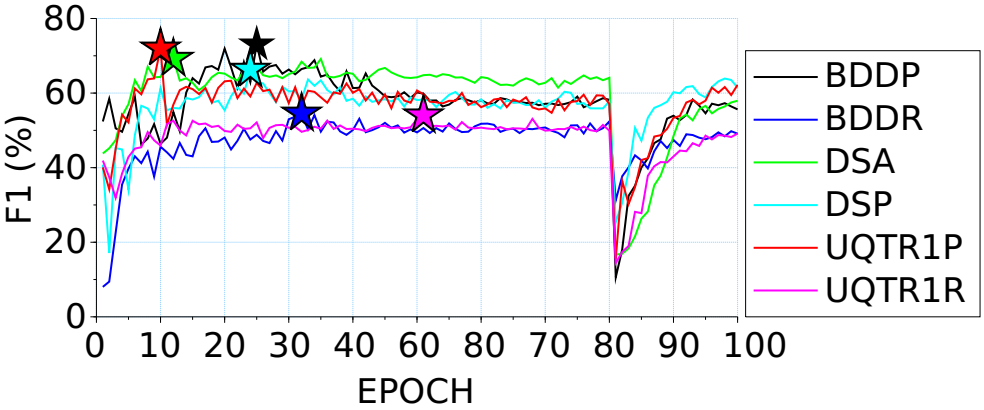


(a) Precision

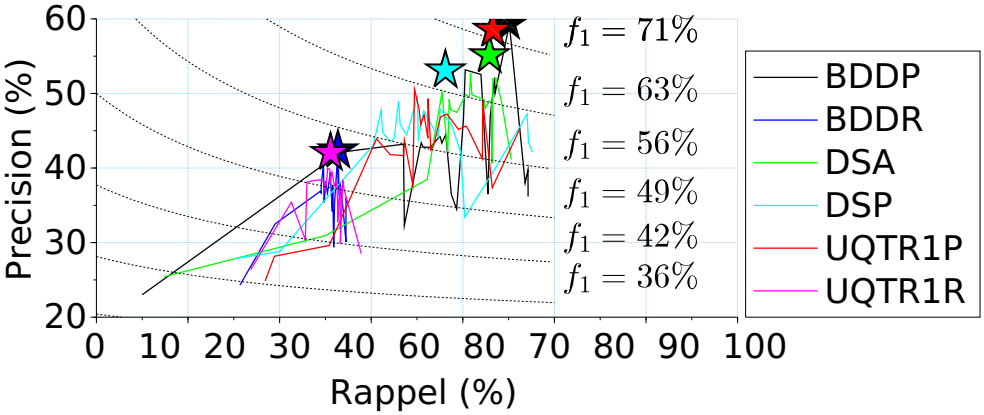


(b) Recall

Figure 3.8 Métriques d'évaluation moyenne entre les réseaux. DSA est séparé égale entre saillance et régulière.



(a) F1-score



(b) rappel, precision

Figure 3.9 Métriques d'évaluation moyenne entre les réseaux (suite). DSA est séparé égale entre saillance et régulière.

3.5 Conclusion

Ce chapitre présente une nouvelle approche pour améliorer la détection d'objets dans les véhicules autonomes en utilisant le mécanisme de saillance. L'outil proposé, nommé OAS, exploite le regard humain pour extraire la saillance du foyer visuel d'une personne et l'intègre aux images régulières. Les images résultantes sont utilisées pour générer des boîtes englobantes de haute qualité pour entraîner des modèles de détection d'objets. La motivation derrière cette approche est de reproduire la capacité humaine à conduire avec un haut niveau de conscience environnementale, qui est essentielle pour une navigation sûre et efficace. Le chapitre introduit également un nouveau jeu de données de 1691 images orientées vers la conduite avec des annotations de saillance.

Ce chapitre a démontré que le mécanisme de saillance peut améliorer la précision des modèles de détection d'objets pour les véhicules autonomes, en particulier dans des scénarios difficiles, comme la conduite hivernale. Le chapitre montre également que la méthode proposée peut réduire le temps et le coût de la collecte et de l'étiquetage des données, ainsi que le temps d'entraînement des modèles. Grâce à la saillance, le chapitre soutient que la taille du jeu de données peut être réduite de moitié sans affecter les performances, ce qui entraîne une réduction de 25% du temps total de marquage. De plus, le chapitre rapporte que les modèles entraînés avec la saillance atteignent le score F_1-maximum (3.15) en environ la moitié du nombre d'époques par rapport aux modèles entraînés sans saillance. Le chapitre évalue plusieurs réseaux neuronaux avec et sans saillance et rapporte une meilleure précision de détection pour les premiers. Le chapitre contribue au domaine de la conduite autonome en proposant une nouvelle et efficace façon d'utiliser la saillance

pour la détection d'objets.

Chapitre 4 - Amélioration du réseau de fusion

Suite aux excellents résultats obtenus avec la saillance, d'autres problèmes restent à être résolus. Bien que la saillance améliore la détection dans les conditions difficiles telle que l'hiver, elle n'est pas suffisante pour la conduite autonome en hiver. Les résultats de l'ordre du 60% F1-Score nécessite du travail supplémentaire afin de pouvoir l'adapter aux voitures. Comme mentionné plutôt, l'imagerie thermique permet d'améliorer considérablement la vision nocturne et hivernale. Afin d'y arriver, l'ajout d'une couche de fusion est nécessaire. Cette section présentera la méthode de fusion par branche suivie d'une explication de comment optimiser les résultats. Ceci est suivi par des instructions pour modifier les réseaux actuels et une méthode pour choisir la profondeur de la fusion. Finalement, la synchronisation est présentée en détail. Cette partie est en partie tirée de [89]. Une autre partie de cette section est publiée dans [90].

4.1 L'approche en branche

Inspiré par YOLO [91–94] et SSD [88] qui combinent la proposition de région avec le classificateur afin d'obtenir un réseau bout à bout, la méthode proposée combine la détection et la fusion en un seul réseau afin d'obtenir des résultats avec une quantité inférieure d'opérations permettant l'utilisation d'ordinateur moins performant. Le fait de combiner ces deux réseaux permet d'enlever la demande d'une information intermédiaire adaptée à l'humain, mais pouvant ne pas être optimale pour la tâche de détection. Ceci a cependant pour inconvénient qu'il est impossible de voir les résultats de la fusion directement. La méthode suggérée consiste à diviser les couches les moins profondes du réseau en branches et à ajouter une couche de fusion au centre du réseau. Ces branches peuvent traiter l'information à des moments différents puisque

les capteurs sont asynchrones afin d'optimiser les performances. La couche de fusion étant après quelques couches représente alors une abstraction de l'image. L'ajout des branches ajoute des opérations de calcul au réseau, cependant, la quantité de calcul ajouté est moindre qu'un réseau de fusion totalement séparé. En comparaison aux méthodes de fusion hâtive, la méthode proposée offre une utilisation plus efficace des données spécifiques à chaque longueur d'onde. La méthode possède aussi l'avantage d'un coût de calcul inférieur à une fusion tardive, effectivement, créant un compromis entre la précision de calibration requise des fusions hâtives et le coût de calcul élevé des méthodes tardives.

L'utilisation des branches permet donc de réduire le coût de calcul comparé aux fusions tardives. Le coût pour séparer le réseau en branche est le suivant:

$$C_{FO}(k) = \sum_{i=1}^{k} (\sum_{c=1}^{q} FO_{ic}) + \sum_{i=k+1}^{K} FO_i \tag{4.1}$$

où FO_{ic} est le nombre d'opérations pour la couche i dans la modalité c. $C_{FO}(\cdot)$ est le coût en nombre d'opérations, FO_i est le nombre d'opérations pour la couche i. k est le nombre de couches avant la fusion. K est le nombre total de couche du réseau et q est le nombre de modalités.

Le coût pour la couche convolutive FO_i est défini en tant que:

$$FO_i = d_i \times w_i \times h_i \times c_{xi} \times c_{yi} \times d_{i-1} \tag{4.2}$$

où w_i et h_i sont les dimensions de l'information à la couche i, c_{xi} et c_{yi} sont les dimensions du filtre convolutif et d_i est le nombre de patrons du filtre.

En réalité, les branches n'ont pas à être de taille identique. L'utilisation de branche

de différente taille peut être bénéfique en réduisant le biais vers certaines modalités. Une taille de branche, en termes de nombre de paramètres, réduisant le biais vers l'information présente en grande quantité serait inversement proportionnelle à la quantité de données pour la longueur d'onde.

Une version simplifiée de l'équation de coût est disponible si toutes les branches sont de taille identique:

$$C_{FO}(k) = \sum_{i=1}^{k} (n \times FO_i) + \sum_{i=k+1}^{K} FO_i \tag{4.3}$$

où n est le nombre de différentes longueurs d'onde.

4.1.1 Taille des branches

Les réseaux neuronaux ont tendance à répartir le poids sur toutes les entrées de manière presque égale, comme le montre le tableau 4-1, qui indique les poids appliqués à chaque canal d'entrée. Le canal rouge représente en moyenne 33.13% des poids des réseaux préentraînés, tandis que les canaux vert et bleu représentent respectivement 41.79% et 25.06%. On observe un léger biais en faveur du canal vert, qui est plus déterminant que le bleu dans une image. Toutefois, cette valeur reste dans la plage pour suivre statistiquement la tendance à séparer le poids également sur les entrées. En utilisant la distribution des poids mentionnée ci-dessus, l'ajout d'une modalité telle que la thermique à une image en couleur représenterait environ 25% des poids sur cette modalité. Toutefois, en raison des maxima locaux du poids dans le réseau, il est possible que ce quatrième canal ait tendance à avoir une proportion déficiente lors d'entraînement de pointe. Les informations supplémentaires ne suivraient pas cette distribution en utilisant un

Tableau 4-1 Statistiques sur le pourcentage des poids en fonction des longueurs d'onde pour les réseaux préentraînés de PyTorch.

Modèle	Rouge	Vert	Bleu
fasterrcnn_mnet_v3_lg_fpn	28.83	56.11	15.05
fasterrcnn_resnet50_fpn	34.04	38.21	27.74
ssd300_vgg16 [88]	29.70	36.58	33.70
ssdlite320_mnet_v3_large	33.95	46.43	19.61
resnet50 [95]	34.04	38.21	27.74
alexnet [96]	35.56	35.73	28.70
densenet121 [97]	34.14	40.36	25.49
efficientnet_b0 [98]	30.74	45.20	24.04
inception_v3 [99]	39.56	31.09	29.34
mnasnet0_5 [100]	31.16	46.49	22.34
regnet_x_16gf	32.91	42.81	24.27
shufflenet_v2_x0_5	33.09	45.22	21.67
vgg11 [101]	34.59	37.33	28.07
Moyenne	33.13	41.79	25.06

réseau préentraîné qui a déjà atteint un maximum sur les branches régulières. Cette mauvaise intégration des données supplémentaires ne permettrait pas suffisamment au réseau d'utiliser convenablement les données supplémentaires afin d'augmenter la précision globale. Plusieurs solutions existent pour cette mauvaise intégration des informations. Puisque chaque branche est indépendante des autres, il est possible d'entraîner partiellement le réseau ou de réutiliser des branches.

Le nombre de neurones et de filtres associés à une modalité spécifique peut ensuite être forcé à la proportion souhaitée en ajustant la taille des branches. La taille des branches est cruciale, car elle oblige le réseau à prendre en compte les entrées, quelle que soit la proportion de ces données dans l'information. La taille du réseau peut alors être adaptée à un point de calcul inférieur à celui du vaste réseau pour une précision similaire. Il n'y a pas de restriction sur la taille de la branche tant que la

sortie de ces branches est d'une taille spatiale compatible pour être concaténée. Elles doivent avoir la même résolution spatiale pour concaténer la sortie des branches, mais elles peuvent avoir n'importe quel nombre de caractéristiques. Une taille appropriée doit être fournie pour atténuer le nombre supplémentaire d'opérations en virgule flottante causées par les branches. Les réseaux de neurones plus petits ont prouvé qu'ils atteignaient une précision plus élevée dans des jeux de données de plus petite taille, car ils ont besoin de moins d'images pour être entraînés. La taille appropriée pour chaque branche doit être définie en fonction du problème souhaité, de la taille du jeu de données et de la qualité du jeu de données. La taille proposée pour la branche serait la proportion inverse du nombre d'images pour chaque branche. La proportion de la quantité de données contenant une longueur spectrale multipliée par la proportion du nombre de paramètres d'une branche sur l'ensemble des branches devrait égaler un afin d'éliminer le biais vers les branches ayant beaucoup de données. Cela permettrait de réduire la taille des branches, réduisant ainsi le coût. Toutefois, il convient de noter que la taille de la branche peut être ajustée en fonction de la taille des données, et dans le pire des cas (ensemble de données rare), un surapprentissage peut se produire dans la branche de données correspondante uniquement.

4.2 Modifications aux réseaux existants

4.2.1 Modifications

Pour appliquer l'approche, les réseaux doivent subir des modifications minimales. Un réseau est d'abord choisi à partir de l'état de l'art comme réseau de base. La première modification à apporter au réseau est de dupliquer les premières couches, jusqu'à la couche k, comme on le voit sur la figure 4.1. Pour appliquer les poids des bases de connaissances existantes, la duplication ne modifie pas la taille, ce

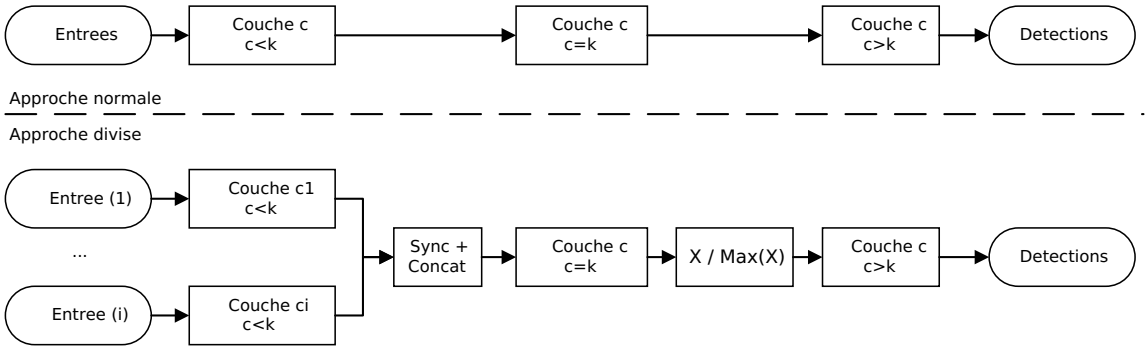


Figure 4.1 Représentation des modifications au réseau.

qui rend les poids compatibles sauf pour les couches de fusion et de première. La première couche aura une taille d'entrée de 1 caractéristique par branche. La couche de fusion à k possède n fois les caractéristiques où n est le nombre de branches. Pour que cette couche reçoive toutes les entrées requises, la sortie de toutes les branches est concaténée. Toutefois, la taille de la convolution et le nombre de filtres seront modifiés pour spécialiser les branches du réseau. Ceci à pour effect néfaste d'empêcher l'utilisation des noyaux existants. Chaque branche étant indépendante, elles peuvent être traitées sur différentes unités de traitement ou réparties à l'aide d'un algorithme de remplissage de vallée pour être exécutées pendant les temps morts entre les captures.

Le choix de k a un impact sur la performance de détection et le coût de traitement. Si k est choisi proche du début du réseau, il agit comme une fusion de bas niveau. Le principal problème de ces approches est le besoin de précision de calibration. En revanche, une valeur de k proche de la fin du réseau agit comme une fusion tardive, augmentant considérablement le coût de traitement. Les inconvénients de cette méthode sont que le réseau entier doit être calculé même sur des canaux sans information, comme le canal de la couleur verte, lors de la détection d'un panneau d'arrêt. Il est essentiel de prendre en compte à la fois le coût de calcul

et l'information significative. Les réseaux neuronaux fonctionnent en créant une abstraction plus détaillée à chaque niveau jusqu'à la détection. Par exemple, un tableau de pixels colorés en entrée peut devenir plus tard une abstraction d'un carré. Cette abstraction sera le meilleur point pour diviser le réseau, car chaque canal abstraira des informations similaires pour le même objet. Cependant, les distributions d'entrée varieront.

Le point d'abstraction d'un réseau neuronal ne peut pas être facilement déterminé. Chaque architecture de réseau et chaque ensemble de données impliqueront un niveau d'abstraction différent pour les couches convolutives. Un autre inconvénient de ne pas connaître la couche d'abstraction est que les motifs spécifiques aux canaux peuvent être abstraits après la couche de fusion, réduisant l'efficacité pour tous les canaux si la fusion est réalisée trop tôt dans le réseau. La fusion tardive ou la présence de caractéristiques de taille incorrectement choisie dans les branches entraînera un calcul plus élevé et nécessitera une plus grande étendue du jeu de données.

4.2.2 Choix de l'hyperparamètre k

Pour proposer une valeur pour k, le réseau doit d'abord être entraîné avec $k = K$, c'est-à-dire à la fin. Une fois les données de l'entraînement acquises, il y a deux options. Dans la première option, un visualiseur de couches est utilisé pour afficher les poids de chaque couche à un expert en réseaux neuronaux. Cet expert identifie le numéro de la couche ayant les poids les plus similaires et utilise cette valeur comme k. Il est à noter que la valeur de k devra alors être choisi avant l'entraînement.

La deuxième option approxime automatiquement la décision de l'expert en se basant sur l'entropie. Il faut quatre étapes pour extraire les informations

de l'entraînement précédent. Dans la première étape, l'ordinateur évalue trois statistiques pour chaque canal de toutes les couches. La première statistique est l'entropie calculée pour une variable X avec:

$$H(X) = -\left(\frac{\sum_{n=1}^{N} \text{logsoftmax}(X)_n \times e^{\text{logsoftmax}(X)_n}}{N}\right) \tag{4.4}$$

La deuxième statistique est la moyenne calculée avec:

$$\bar{X} = \frac{\sum_{n}^{N} X_n}{N} \tag{4.5}$$

où X_n représente une valeur individuelle de la variable X.Finalement, la variance calculée avec:

$$Var(X) = E\left[(X - \bar{X})^2\right] \tag{4.6}$$

La différence entre ces valeurs est ensuite calculée:

$$\sum_{i=2}^{C} \|S_i - S_{i-1}\| \tag{4.7}$$

Ces trois statistiques sont ensuite concaténées par une fonction de distance de Minkowski:

$$\Delta_{MINKOWSKI} = \sqrt[p]{\omega_x \times |X_2 - X_1|^p + \omega_y \times |Y_2 - Y_1|^p + \omega_z \times |Z_2 - Z_1|^p} \tag{4.8}$$

dans laquelle X, Y et Z sont nos 3 statistiques pour les couches en indice. Il suffit alors de trouver la valeur de k ayant la différence la moindre.

4.3 Préparation des données d'entrée

Quant aux entrées, le réseau modifié nécessite toujours une calibration partielle des capteurs. La déformation des objectifs dans les caméras est connue sous le nom de distorsion optique [69]. La calibration est nécessaire si les différents capteurs ont des distorsions différentes. La calibration de la caméra est une partie essentielle de la fusion des caméras. Une telle calibration permet d'obtenir la transformation nécessaire pour que tous les capteurs soient dans le même référentiel mondial, par exemple les paramètres intrinsèques et extrinsèques. Comme la calibration peut être une partie fastidieuse du processus de fusion et coûter cher, il faut réduire le besoin de calibration. Comme les réseaux neuronaux obtiennent des caractéristiques par des statistiques spatiales, la calibration spatiale est obligatoire. Cependant, les réseaux neuronaux sont tolérants à l'imprécision. Une calibration faite de manière approximative serait suffisante pour que les images soient fusionnées. Tant que les deux images se chevauchent dans un monde, le réseau pourrait apprendre les caractéristiques spatiales nécessaires à la fusion. En raison de leurs convolutions, les réseaux sont intrinsèquement invariants par rapport à la translation [102]. Lors d'une entrée à haute fréquence et d'un mouvement non soudain, les objets ne se déplacent que légèrement sur une courte période, ce qui rend ces réseaux tolérants à une légère variation dans le temps. La méthode de calibration proposée consiste à identifier la distance optimale de détection, placer un objet à cette distance et le juxtaposer dans plusieurs images. Un exemple d'une telle calibration est présenté dans la figure4.2. Dans cette figure, la première ligne représente une image visible, la deuxième l'image thermique et la troisième la juxtaposition montrant la majorité des objets bien alignés.

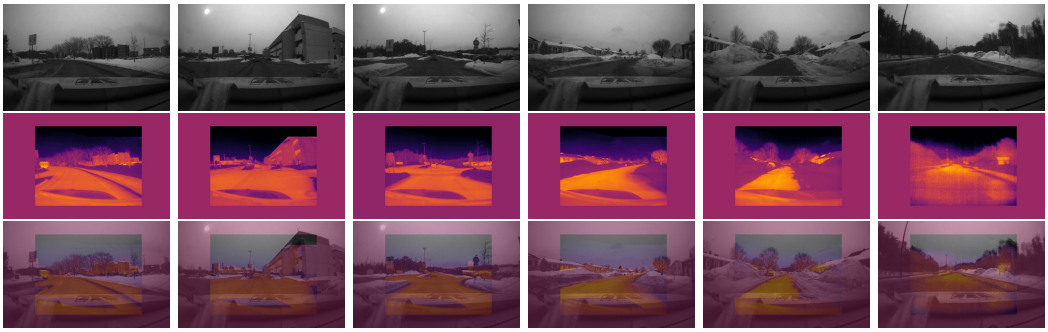


Figure 4.2 Exemple de calibration imprécise.
1

4.4 Implémentation proposée

La plateforme PyTorch est utilisée pour mettre en œuvre la recherche actuelle. Chaque réseau proposé a été converti en un format standard prenant le paramètre k en hyperparamètre. Cependant, en raison de la nature automatique de la séparation, les réseaux contenant des connexions de saut comme ResNet [95] ne peuvent pas être utilisés. La conversion manuelle de ces réseaux est possible, mais il faut ajouter le paramètre supplémentaire de la façon de séparer une connexion de saut dans la recherche actuelle. Pour qu'une connexion de saut fonctionne de manière semi-automatique, il suffit de s'assurer que la source et la cible de la connexion de saut sont sur la même branche. Pour l'architecture de réseau proposée, il existe deux branches principales. La première branche est la branche couleur et se divise en trois sous-branches. L'image thermique est insérée pour la reconnaissance dans des conditions météorologiques plus difficiles sur la deuxième branche.

À la couche de fusion, une modification supplémentaire est appliquée à la convolution. Le fait que les entrées sont optionnelles pour la robustesse cause une possibilité d'information manquante. Les ordinateurs exécutent les convolutions efficacement par des multiplications de points matriciels. Ainsi, les poids sont

regroupés par leur entrée et peuvent être supprimés sans changer la taille de la sortie. Ils sont adaptés en temps réel et traités pour un canal manquant. Comme la suppression de ces poids de la somme pondérée de la multiplication par points réduira la plage de la sortie, une normalisation est ajoutée sous la forme de l'équation suivante afin de conserver les poids (X) dans la plage zéro à un:

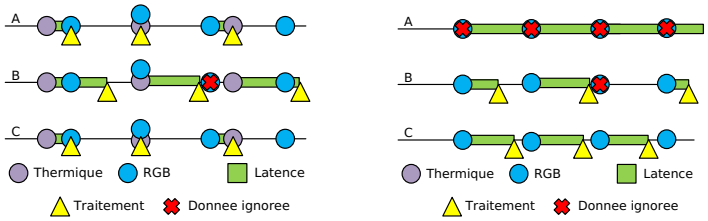
$$X^{'} = \frac{X}{X_{\max}} \tag{4.9}$$

Cette étape de normalisation est visible sur la figure 4.1, où elle est appliquée directement après la couche convolutive responsable de la fusion.

La méthode proposée a été testée avec trois réseaux de base comme extracteurs de caractéristiques. Le premier réseau de base est le Alexnet [96], le deuxième est le VGG-16 [101], et le troisième réseau de base a été créé sur l'architecture encodeur-décodeur. Ces réseaux de base ont été choisis pour leur absence de couches à avance rapide, ce qui les rend adaptés pour la fusion de réseaux. Le réseau basé sur l'encodeur se compose de cinq couches avec un noyau de taille trois, suivies de cinq convolutions avec un suréchantillonnage de deux. Le détecteur SSD [88] assume la partie détection du réseau, car il peut fonctionner avec n'importe quel réseau de base.

4.5 Synchronisation

La fusion inhérente des couches dans le réseau permet le calcul parallèle ou partiel du réseau, ce qui rend possible le traitement séparé des capteurs avec des fréquences différentes, uniformisant la charge de calcul. Cependant, une synchronisation est importante sur les données en entrée afin de diminuer le désalignement potentiel des



(a) Conditions normales (b) Capteur défectueux

A: Attente — B: Fréquence fixe — C: Delai maximal

Figure 4.3 Chronologie des différentes méthodes.
1

données. Le traitement peut être uniformisé en remplissage de vallées à condition que les données d'entrée soient appliquées sur des données collectées au même moment. Les algorithmes de remplissage de vallée visent à répartir et à réduire les pics de calcul [103, 104]. Il est possible de calculer les couches 1 à k pour chaque entrée avec des entrées à des fréquences variables. Par conséquent, une technique de synchronisation doit être utilisée [105].

Comme le montre la figure 4.3, une méthode de synchronisation régulière qui agit avant le réseau introduit beaucoup de latence et traite toutes les données simultanément, ce qui nécessite un meilleur équipement de calcul pour maintenir la latence et la fréquence à des niveaux en temps réel.

Plusieurs possibilités sont étudiées pour réduire le coût de traitement, principalement trois méthodes, chacune avec trois variantes. Le tableau 4-2 montre les spécificités de chaque technique.

La figure 4.3 illustre les méthodes proposées avec une représentation visuelle. Le cercle violet signifie l'entrée de la caméra FLIR. Le cercle bleu représente l'entrée

Tableau 4-2 Différentes méthodes de synchronisation

Méthode	Variante	Coût	Latence	Donnée manquante
Attente (A)	Parallèle	Haut	Faible	Échec
	File	Faible	Haute	Échec
	Premier arrivé	-	-	Échec
Fréquence fixe (B)	Parallèle	Haut	Modérée	Aucun changement
	File	Faible	Haute	Aucun changement
	Premier arrivé	Faible*	Modérée	Aucun changement
Attente + delai maximal	Parallèle	Haut	Modérée	Latence
	File	Faible	Haute (C)	Latence
	Premier arrivé	Faible*	Faible	Latence

*Possiblement modéré

de la caméra Mynteye. La ligne verte indique la latence entre les données entrantes et le traitement, représenté par le triangle jaune. Les cercles barrés indiquent les données qui ont été ignorées, car des données plus récentes sur le même canal ont été reçues avant le traitement. L'utilisation de l'image la plus récente permet de réduire la différence de temps entre la capture et le traitement. La figure 4.3b montre un événement théorique où la caméra FLIR a cessé de fonctionner.

La première méthode consiste à attendre que toutes les données soient disponibles avant de commencer le traitement. Comme dans le cas optimal de la figure 4.3a, chronologie A, la latence est courte, mais cette première méthode échoue si une défaillance du capteur se produit comme dans la figure 4.3b:

- Sa première variante consiste à traiter toutes les données simultanément une fois que tout est arrivé.

- La deuxième variante consiste à traiter chaque branche du réseau une à une, mais à attendre quand même toutes les informations. Cette méthode réduit le coût maximal au détriment de la latence.
- La troisième variante n'est pas disponible pour cette méthode, car nous ne pouvons pas traiter l'information au fur et à mesure qu'elle arrive et attendre quand même que toutes les informations soient disponibles.

La deuxième méthode vise à résoudre le problème de défaillance du capteur en utilisant une fréquence fixe pour le traitement des données, tel qu'illustré dans la chronologie B de la figure 4.3a. Avec un intervalle de traitement fixe, la perte d'un capteur n'a pas d'impact sur le système, car la période entre les traitements de données reste constante:

- Cette première variante de cette méthode vise à modérer la latence comme on le voit sur la figure 4.3a. Cependant, la fréquence fixe peut créer de faux positifs pour les défaillances de capteurs et induire des retards si elle doit être correctement synchronisée avec les capteurs. Elle a également un coût de traitement élevé.
- Par la deuxième variante, toutes les données sont traitées une à une une fois que la fréquence a été atteinte, ce qui crée une latence élevée, mais le coût reste faible.
- La troisième méthode traite les informations entrantes immédiatement, et seule la partie partagée du réseau est traitée à une fréquence fixe. Cependant, cette approche du premier arrivé, premier traité, peut conduire à un traitement redondant pour un capteur. Si la fréquence d'un capteur lui permet d'arriver deux fois avant le déclenchement de la fréquence fixe, deux événements de traitement se produiront, même si seule la sortie la plus récente est utilisée.

La présente thèse propose une troisième méthode pour résoudre les problèmes de latence modérée et de double traitement tout en corrigeant les défaillances des capteurs. Cette troisième méthode de synchronisation offre les avantages des deux méthodes précédentes. Au lieu d'utiliser une fréquence fixe, cette méthode attend l'information comme la première. Toutefois, un délai maximal commence à être calculé une fois que le premier capteur est lu. L'information est traitée si ce délai est atteint avant de récupérer les données de tous les capteurs. Dans la figure 4.3a, cette méthode est représentée par la chronologie C, dans laquelle les lignes de latence sont à leur durée maximale:

- La première variation traite toutes les informations après que le déclencheur a été activé, ce qui entraîne un traitement parallèle et un coût élevé. Cependant, la latence peut être modérée dans le pire des cas et faible dans le meilleur des cas, car le délai est une fréquence fixe si un capteur est manquant.
- La deuxième variante met en file d'attente chaque branche une fois que le déclencheur est activé. Comme pour les autres méthodes, le coût devient faible et la latence élevée.
- Pour la troisième variante, chaque branche est traitée lorsque les données sont disponibles, ce qui réduit le coût maximal et la sortie anticipée induite si toutes les données des capteurs sont disponibles, réduisant ainsi la latence au minimum. L'aspect négatif de cette méthode est qu'un capteur manquant provoque une latence.

Toutes les méthodes de synchronisation et leurs variantes sont présentées dans le tableau 4-2. Il est important de noter que chaque branche est séparée des méthodes précédentes. Sauf pour la première variante, dans laquelle toutes les données sont après le déclenchement de la méthode de synchronisation, seul le tronc serait traité

après le déclenchement de la procédure de synchronisation. Il devrait être attaché à un mécanisme automatique qui éteindrait le capteur manquant après quelques itérations pour réduire ce retard et le réactiver automatiquement s'il envoie de nouvelles données pour corriger la latence potentielle de la dernière option. Cette étape est présente dans l'étape sync + concat du réseau comme on le voit sur la figure 4.1.

4.6 Entraînement

Enfin, cette architecture unique nécessite un entraînement spécial. Dans la première étape, le réseau est entraîné sur le jeu de données de fusion, qui contient des images synchronisées de toutes les branches. Cette étape initialise la couche de fusion de manière à supporter toutes les entrées. Pendant la première étape, le taux d'apprentissage du réseau est réglé bas pour apprendre des patrons appropriés. Dans la deuxième étape, le réseau est entraîné sur tous les jeux de données disponibles. Cela est possible en sélectionnant un sous-ensemble des poids dans la couche de fusion. Toutes les branches qui ne sont pas contenues dans le jeu de données sont gelées pour éviter le changement. Enfin, le réseau est réentraîné sur le jeu de données de fusion pour s'assurer que le résultat est optimisé en utilisant plusieurs capteurs.

4.7 Résultats

Pour utiliser l'architecture proposée, il faut fixer l'hyperparamètre k. Plusieurs valeurs de k ont été évaluées et sont présentées dans la figure 4.4. La figure montre le résultat du réseau sur le jeu de données FLIR ADAS avec différentes valeurs de k et la colonne auto représentant la valeur de k trouvé par entropie. Il est visible sur la figure 4.4 une tendance selon laquelle plus k est élevé, plus la mAP est élevé sans changer aucun autre hyperparamètre. Les exceptions à cela sont $k = 8$ et $k = 14$ pour

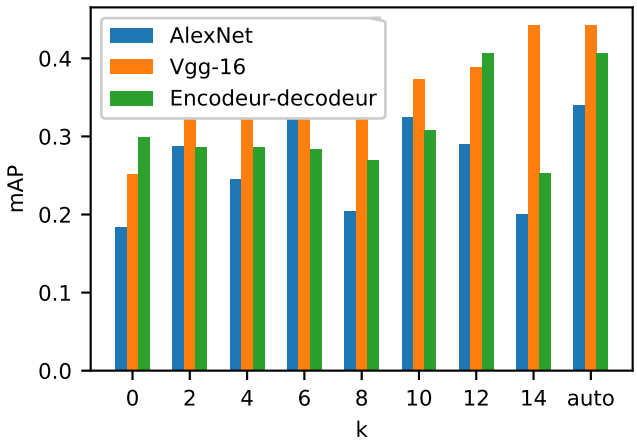


Figure 4.4 mAP comparée avec l'hyperparamètre k pour détection SSD.

le réseau de base AlexNet, dans lesquels la mAP baisse drastiquement. Ces chutes peuvent s'expliquer par une taille de branche sous-optimale sur le réseau de base, car tout a été automatisé au lieu d'être optimisé manuellement pour une comparaison plus équitable (figure 4.4). Cette figure présente également la performance du réseau proposé sur le jeu de données FLIR avec 34.1% pour le réseau de base Alexnet, 40.7% pour l'autoencodeur, et 44.3% pour le réseau de base VGG (figure 4.4). Dans tous ces réseaux de base, l'entropie a pu récupérer la mAP complet. Il est possible d'apercevoir dans cette figure que VGG obtient de très bon résultats pour les couches $k > 6$ et qu'il bénéficie grandement d'une fusion plus tardive.

Selon l'amélioration de la méthode de synchronisation, ces méthodes affectent fortement le coût de traitement et la latence du réseau sans affecter le résultat. Le tableau 4-3 présente un résumé du processus de synchronisation. Suite à la capture des données pour la fusion hivernale, les capteurs fonctionnent souvent plus lentement que prévu. La caméra RGB, utilisée pour la partie RGB du jeu de données, a capturé ses informations entre 20 et 62 hz. La fréquence de sortie de la caméra est en moyenne

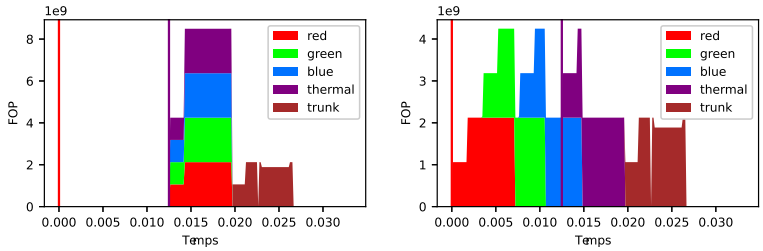
de 40 hz. La caméra thermique a fourni une fréquence d'images de 14 à 34 hz avec une moyenne de 24 hz. Suite à ces distributions de fréquences, plusieurs simulations de 30 minutes ont été effectuées avec les trois méthodes de synchronisation mentionnées ci-dessus. Ces simulations ont fourni les résultats expérimentaux du tableau 4-3 pour tous les états possibles de défaillance des capteurs. La première méthode a montré une bonne fréquence, une faible latence et aucune défaillance dans des conditions optimales. Cependant, cette méthode n'a pas réussi à fusionner les données lorsqu'un capteur était manquant, comme le montrent les 2e et 3e lignes du tableau 4-3. La deuxième méthode a montré une fréquence stable indépendante de la défaillance du capteur, mais a échoué 2479 fois à traiter les informations, car le déclenchement s'est produit sans recevoir les lectures de la caméra. Ces fusions ratées ont mal utilisé le temps de traitement, gaspillant 6.8% du temps de traitement (tableau 4-3). Cette méthode a également une latence moyenne élevée par rapport à la première. Enfin, le processus de synchronisation proposé a fourni une faible latence et une fréquence de fusion élevée pendant une étape optimale ; il n'a pas gaspillé de traitement sur des fusions ratées ni ignoré beaucoup de lectures tout en ne ralentissant que pour les capteurs manquants. La méthode proposée est un algorithme de traitement plus robuste que les méthodes actuellement utilisées.

Dans la figure 4.5, les courbes rouge, vertes et bleues représentent le traitement de l'image en couleur. La courbe violette représente le traitement de l'image thermique. La ligne verticale rouge représente la capture de l'image RGB, et la ligne verticale violette représente la capture de l'image thermique. Le coût de traitement pourrait être réduit de 8 GFOP à 4 GFOP, comme le montre la figure 4.5, en utilisant l'algorithme de synchronisation sélectionné et un algorithme de remplissage de vallée. Cette figure montre parfaitement l'algorithme de remplissage de vallée pour présenter

Tableau 4-3 Résultats expérimentaux des capteurs.

Méthode	RGB	T	Traitement	Partiel	Échoué	Ignoré	Latence (ms) Minimum	Moyenne	Maximum	Fréquence fréquence (Hz)
Attente	✓	✓	40310	0	0	24837	0.0003	13.96	49.82	22.39
Attente		✓	0	0	0	40260	∞	∞	∞	0
Attente	✓		0	0	0	65289	∞	∞	∞	0
Fréquence fixe de 20Hz	✓	✓	36001	2605	0	36009	0.01974	24.78	50	20
Fréquence fixe de 20Hz		✓	36001	36001	2479	6842	0.0028	21.37	50	18.62
Fréquence fixe de 20Hz	✓		36001	36001	2479	29178	0.0011	15.38	50	18.62
Délai maximal de 50ms	✓	✓	43909	3635	0	21411	0.0003	13.44	49.95	24.39
Délai maximal de 50ms		✓	23501	23501	0	16726	2.38	22.88	50	13.06
Délai maximal de 50ms	✓		27192	27192	0	38022	2.84	13.89	50	15.11

La colonne RGB indique si le capteur RGB était présent pendant la simulation. La colonne T indique si le thermique était présent pendant cette simulation. La colonne Traitement est le nombre de la fusion effectué. La colonne Partiel indique le nombre de fois où la fusion a traité des valeurs de capteur incomplètes. La colonne Échoué compte le nombre de fusions effectuées sans aucune donnée de capteur. La colonne Ignoré représente la quantité de données ignorées pour avoir reçu plusieurs des mêmes capteurs capturés entre les fusions. Les colonnes de latence enregistrent le temps écoulé entre la première donnée et le processus de fusion.



(a) Coût pour VGG-19 en parallèle (b) Coût pour VGG-19 avec la méthode proposé et optimisation ValleyField

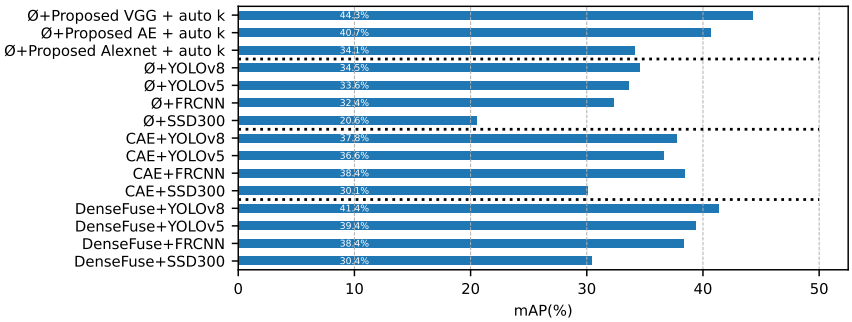
Figure 4.5 Répartition du coût pour la fusion RGB-T.

1

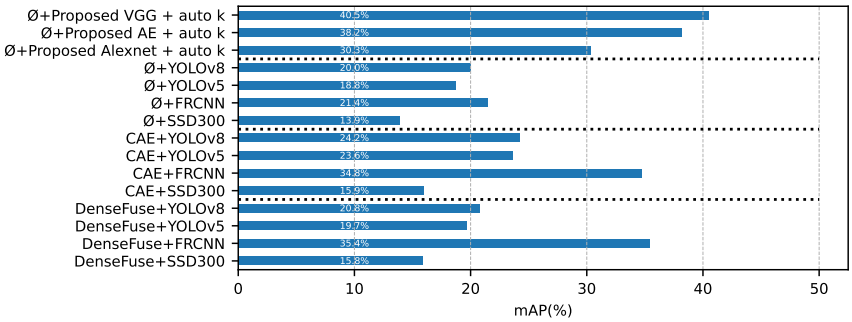
le meilleur résultat. Cependant, le coût de traitement sera légèrement plus élevé selon l'implémentation choisie [103,104]. La latence ne sera pas affectée dans des conditions normales, car les capteurs sont désynchronisés. Dans la figure 4.5b, en utilisant la latence gaspillée entre la capture de l'image visible et de l'image thermique, aucune latence supplémentaire n'a été utilisée, et le coût a été réduit de moitié. Le tronc (couches $k+1$ à K) n'est affecté par aucune des deux méthodes.

Le réseau de fusion-détection de bout en bout proposé est ensuite comparé à des combinaisons de méthodes de fusion et de détecteurs de pointe pour offrir une comparaison équitable avec les méthodes de pointe. La nouvelle architecture a été entraînée en utilisant un ensemble de données de fusion de 256 entrées combiné avec le MS COCO 2017 et le jeu de données FLIR ADAS [106]. Deux fusions de pointe sont présentées. Premièrement, la fusion DenseFuse [107], qui a été initialement entraînée sur le jeu de données FLIR ADAS [106]. Deuxièmement, l'autoencodeur convolutif variationnel [108] est utilisé pour ses résultats convaincants avec une taille de jeu de données minimale. L'aspect génératif de cette méthode permet de générer des images fusionnées de meilleure qualité par rapport aux autres approches basées sur l'apprentissage profond. Comme l'approche proposée fournit à la fois la fusion et la détection sans fournir l'image intermédiaire, il est impossible de la comparer avec des méthodes de fusion seules. Trois détecteurs de pointe ont été utilisés. Le Faster-RCNN [109] avec le Resnet50 [95] comme réseaux de base sont des détecteurs historiques. Le SSD300 [88] fournit un réseau récent avec des aspects convaincants tels que le temps réel et la mAP élevée. Enfin, le YoloV8 [92, 93] est utilisé pour une implémentation moderne avec des performances élevées et une faible latence. Les détecteurs examinés étaient équipés de modèles préentraînés sur le jeu de données Common Objects in Context (COCO). Un affinage a été effectué pendant 100 itérations sur les nouveaux jeux de données pour évaluer la performance de ces détecteurs sur les jeux de données de fusion.

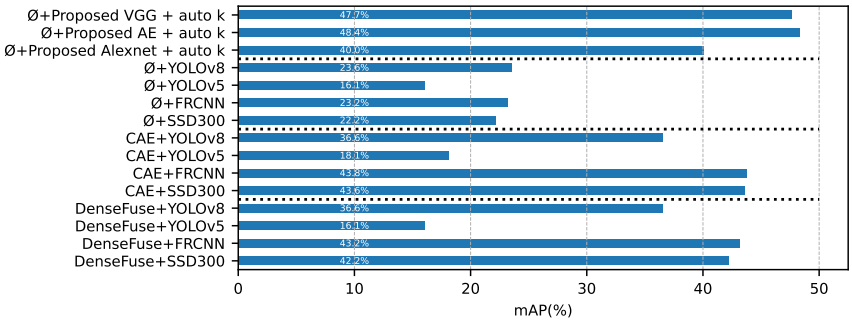
Le choix de k basé sur l'entropie proposée a été utilisé pour comparer l'état de l'art présenté au début de cette section. Les résultats des combinaisons de fusion et de détecteurs de pointe sont ensuite comparés à la méthode basée sur l'entropie proposée dans la figure 4.6. La figure présente le réseau évalué sur le jeu de données



(a) FLIR ADAS dataset



(b) Harsh winter dataset



(c) PST900 Dataset

Toutes les méthodes proposées utilisent l'entête SSD300 header pour la détection
Ø = Seul RGB est utilisé, sans fusion, CAE = auto-encodeur à variation convolutive,
AE = auto-encodeur

Figure 4.6 Comparaison avec les réseaux de pointe.

FLIR dans la figure 4.6(a), le jeu de données basé sur l'hiver de 256 images fourni par notre équipe dans la figure 4.6(b), et le jeu de données PST-900 [?] dans la

figure 4.6(c). Le jeu de données basé sur l'hiver sévère est nettement plus difficile à détecter les objets en raison de la neige et de la nuit. Cet ensemble de données de 256 images annoté par notre équipe contient des tempêtes de neige et d'autres données sévères, qui sont difficiles à analyser par les réseaux neuronaux. Le jeu de données PST-900 est pris dans des conditions souterraines pour le sauvetage des survivants. Le jeu de données PST900 est souvent pris dans l'obscurité, où l'image thermique contient l'information principale, contrairement à la conduite hivernale, où les phares des voitures rendent l'image visible moins difficile à comprendre. DenseFuse obtient des résultats légèrement meilleurs que la méthode VCAE d'un demi-pour cent. La méthode SSD300 utilisant une fusion de pointe obtient le pire résultat avec 15.8% dans des conditions hivernales et 30.1% pour le jeu de données FLIR ADAS (figure 4.6). Le SSD est connu dans la littérature pour fournir une mAP plus faible que les autres méthodes de pointe ; cependant, il donne des résultats en temps réel adapté aux appareils embarqués. Cette mAP très faible indique que le SSD, contrairement aux autres méthodes de pointe, est plus sensible aux variations météorologiques. Le détecteur FasterRCNN est plus lent et introduit de la latence, ce qui pourrait être fatal dans le cas de la conduite autonome, mais il fournit 38.4% sur le jeu de données FLIR, 43.8% sur PST-900, et 34.8% sur le jeu de données hivernal (figure 4.6). Pour YoloV5, la mAP atteint 39.4% en utilisant la fusion DenseFuse et 23.6% sur le jeu de données hivernal. YoloV8, étant plus précis qu'YOLOv5, obtient 41.36% sur le jeu de données FLIR, 36.6% sur PST-900, et 24.21% sur le jeu de données hivernal (figure 4.6). Le réseau de bout en bout utilisant des auto-encodeurs ou VGG comme réseaux dorsaux obtient de meilleurs résultats que les détecteurs de pointe dans toutes les conditions en utilisant l'information thermique sans l'image intermédiaire.

4.8 Conclusion

Ce chapitre propose un réseau de fusion de bout en bout avec synchronisation pour détecter les obstacles routiers en utilisant la fusion d'images thermiques et visibles. En suivant le chemin de SSD, notre méthode intègre deux réseaux séquentiels en un seul de bout en bout pour réduire les coûts de traitement et augmenter les performances de détection. La fusion est appliquée par une approche branche et tronc avec des branches basées sur la longueur spectrale et des poids adaptatifs au niveau de la couche de fusion. La méthode proposée suggère également d'utiliser une méthode de synchronisation à retard maximum qui ajoute de la robustesse pour les capteurs manquants tout en gardant la latence faible. L'approche modulaire branche et tronc permet une adaptation de l'entrée sans réapprentissage. En utilisant la modularité du réseau, un entraînement partiel peut réduire le biais créé par la taille du jeu de données lors de l'entraînement. La façon proposée de diviser le réseau permet une ramification automatique pour les réseaux de détection sans aucune avance rapide dans les tenseurs. Le point de fusion est automatiquement sélectionné par la méthode proposée basée sur l'entropie. Le réseau proposé a atteint 40.5% de mAP sur le jeu de données hivernal fortement dégradé de 256 paires d'images capturées par notre équipe et 44.3% sur le jeu de données public FLIR ADAS, qui contient des conditions légèrement dégradées, en utilisant des images thermiques combinées à des images du spectre visible. Il convient de noter que la mAP de 44.3% obtenus par la méthode proposée est un bond significatif de performance dans les conditions défavorables mentionnées ci-dessus, et que l'inclusion d'images thermiques diurnes dans le jeu de données a montré une amélioration drastique de la mAP rapportée par les méthodes SOTA, comme dans le jeu de données FLIR ADAS. Ce chapitre propose une nouvelle approche de fusion de bout en bout pour l'imagerie thermique

et visible afin d'améliorer les performances de la conduite autonome pendant les conditions météorologiques difficiles, comme l'hiver. Elle propose également une nouvelle méthode de synchronisation pour une synchronisation en temps réel à faible latence qui est essentielle dans la conduite autonome. Cette nouvelle méthode de synchronisation est résistante aux défaillances des capteurs, comme la déconnexion de la caméra pendant le fonctionnement du véhicule. L'architecture proposée peut être facilement appliquée avec une longueur spectrale supplémentaire, comme l'ultraviolet, en ajoutant des branches et des données tout en conservant les poids des branches existantes. L'architecture s'applique également à d'autres tâches utilisant des réseaux neuronaux avec des entrées de types compatibles et une couche de convolution agissant comme couche de fusion.

Chapitre 5 - Intégration et énergétique

Pour donner suite à l'amélioration des données et l'amélioration des réseaux par fusion visible et thermique, une intégration avec une évaluation des effets de ces détections sur l'énergétique de la voiture est requise.

Ce chapitre proposera un réseau de sélection en temps réel qui modélise la performance et la distance de chaque méthode disponible. La sécurité du véhicule est améliorée par la détection améliorée résultant de cette sélection. Des seuils de sécurité sont également établis pour augmenter la sécurité et déterminer la faisabilité de la conduite autonome au niveau 4 d'autonomie. Ce chapitre propose ensuite une application qui tire parti de cette détection accrue à distance pour réaliser des économies d'énergie potentielles lors du freinage. L'organisation du reste de cette section est la suivante: Le modèle énergétique et son optimisation sont présentés dans la Section 5.1, un aperçu de la méthode proposée est fourni dans la Section 5.2, les différents résultats énergétiques et de segmentation sont présentés dans la Section 5.3, et la section se conclut dans la section 5.4.

Tableau 5-1 Paramètres dynamiques du véhicule

Parametre	Valeur
ρ	1.2
Cd	0.3
A	2.2 m^2
V_{masse}	1500 kg
$Max_{deceleration}$	8 m/s^2
$Max_{acceleration}$	2.65 m/s^2
μ	0.02
$Roue_{rayon}$	0.57 m

5.1 Modèle du véhicule

Un système de masse ponctuelle longitudinale composé de la somme des forces est considéré pour le contrôle optimal basé sur le modèle dans ce chapitre. Les paramètres du véhicule utilisé ont été obtenus à partir d'un modèle Kia Soul EV 2017 et sont présentés dans le tableau 5-1. Les valeurs ont été arrondies à deux chiffres significatifs.

5.1.1 Forces du véhicule

L'équation suivante résume la somme des forces appliquées au véhicule:

$$F_V = F_{moteur} - F_{aero} - F_{Fr} - F_{Br} + F_{gravite} \tag{5.1}$$

où F_V est la force appliquée à l'objet de masse ponctuelle, F_{moteur} est la force d'accélération appliquée par le moteur, F_{aero} est la traînée aérodynamique du véhicule, F_{Fr} est la friction de la route, F_{Br} est la force de freinage appliquée par le freinage mécanique et régénératif, et $F_{gravite}$ est la force appliquée sur le véhicule par la gravité en raison de la pente. La deuxième loi du mouvement de Newton définit l'accélération du véhicule comme:

$$a = \frac{F_V}{V_{masse}} \tag{5.2}$$

dans laquelle V_{masse} est la masse du véhicule à déplacer, F_V est la force appliquée sur ce véhicule et a est l'accélération ou décélération obtenue. L'équation suivante définit la traînée aérodynamique du véhicule:

$$F_{aero} = 0.5 \times C_d \times A \times \rho \times v^2 \tag{5.3}$$

où C_d est le coefficient de traînée, A est la surface frontale du véhicule, ρ est la densité de l'air, et v est la vitesse actuelle du véhicule. L'équation suivante définit la friction de la route:

$$F_f = \mu \times V_{masse} \times g \times \cos\theta \tag{5.4}$$

où μ est le coefficient de friction, g est la gravité, et θ est la pente de la route. L'équation suivante définit la force appliquée par la gravité:

$$F_{gravite} = -V_{masse} \times g \times \sin\theta \tag{5.5}$$

Les forces sont divisées en trois catégories: les forces environnementales (F_{env}), la force d'accélération (F_{moteur}), et la force de décélération (F_{Br}). La force environnementale est définie par l'équation suivante:

$$F_{env} = F_{aero} + F_{Fr} + F_{gravite} \tag{5.6}$$

Une formule simplifiée est obtenue en combinant toutes les forces:

$$F_V = F_{moteur} - F_{Br} - F_{env} \tag{5.7}$$

où F_{moteur} et F_{Br} sont des forces qui peuvent être contrôlées par le véhicule, tandis que F_{env} est une force qui ne peut pas être contrôlée par le véhicule.

5.1.2 Problème de contrôle optimal pour le freinage régénératif

Le freinage régénératif est une méthode dans laquelle le moteur agit comme un générateur électrique, convertissant l'énergie cinétique du véhicule en énergie électrique qui recharge la batterie. Cette méthode est considérée comme un moyen

prometteur de récupération d'énergie [110]. L'énergie cinétique du véhicule, K_i, est donnée par

$$K_i = \frac{1}{2} \times V_{masse} \times v_i^2 \tag{5.8}$$

où V_{masse} est la masse du véhicule et v_i est la vitesse en m/s au moment i. Une session de freinage est définie comme

$$K_0 > K_b \tag{5.9}$$

Les forces appliquées sur le véhicule ont causé la réduction de l'énergie du véhicule

$$\Delta K = K_b - K_0 = \int_{t=0}^{b} F_V(t) \times v(t) \times dt \tag{5.10}$$

Trois forces peuvent diviser F_V, comme le définit (5.7). Cependant, la force d'accélération appliquée par le moteur (F_{moteur}) est de 0 pendant le freinage, donc elle est exclue, ce qui donne le suivant:

$$\Delta K = \int_{t=0}^{b} F_{Br}(t) \times v(t) \times dt + \int_{t=0}^{b} F_{env}(t) \times v(t) \times dt \tag{5.11}$$

F_{Br} est la seule force qui peut être contrôlée. Deux forces peuvent subdiviser F_{Br}:

$$F_{Br} = F_R + F_M \tag{5.12}$$

où F_R est la force appliquée par le freinage régénératif à maximiser, et F_M est la force appliquée par le freinage mécanique. La commande de contrôle est séparée comme suit:

$$\int_{t=0}^{b} F_R(t) \times v(t) \times dt + \int_{t=0}^{b} F_M(t) \times v(t) \times dt \tag{5.13}$$

La fonction à maximiser est définie comme:

$$J = \int_{t=0}^{b} P_{RGN}(v(t), a(t), \Theta(d(t)))dt \tag{5.14}$$

pour une formulation de problème de contrôle optimal, où $a(t)$ est l'entrée de décélération à concevoir, $\Theta(d(t))$ est l'état du système au moment t en tenant compte des entrées précédentes, y compris la pente, la vitesse, le déplacement. Enfin, P_{RGN} est la puissance régénérative systématisée donnée par:

$$\int_{t=0}^{b} F_R(t) \times v(t) \times dt \tag{5.15}$$

Une trajectoire de vitesse optimale $v^*(t)$ est générée par le problème de planification de la décélération pour maximiser la régénération d'énergie, tandis qu'une trajectoire d'entrée d'accélération $a^*d(t)$ est trouvée sur un intervalle de temps de planification de vitesse $0 \leq t \leq b$ qui maximise le critère de performance de récupération d'énergie donnée dans (5.14).

5.1.3 Programmation dynamique

Pour maximiser le freinage régénératif, un problème de contrôle optimal en temps discret (OCP) est formulé comme suit:

$$\begin{aligned} \max_{a_d(k)\in\xi} \sum_{k=0}^{N-1} & P_{RGN}(v(k), a_d(k), \Theta(d(k)))\Delta t \\ \text{s.t.} \quad & v(k+1) = v(k) + a_d(k) \times \Delta t \\ & d(k+1) = d(k) + v(k) \times \Delta t \end{aligned} \tag{5.16}$$

Tableau 5-2 L'énergie nécessaire au freinage et l'énergie récupérée par une stratégie de freinage optimale pour un véhicule de 1500 kg se déplaçant à 90 km/h (25 m/s).

Distance	Énergie requise	Énergie régénérée	ε
	F_{Br}	F_R	
0m	-	-	-
40m	451438.45J	98272.31J	21.76%
50m	447264.14J	122328.65J	27.35%
60m	443121.33J	146184.35J	32.98%
70m	440311.23J	161953.55J	36.78%
80m	434942.47J	193307.76J	44.44%
90m	432602.06J	206615.91J	47.76%
100m	426896.46J	239661.17J	56.14%
110m	423293.46J	260488.16J	61.53%
120m	418971.16J	285256.96J	68.08%
140m	413798.35J	315274.73J	76.19%
160m	407376.99J	352326.04J	86.48%
170m+	399277.32J	399277.32J	100.00%

où Δt est l'intervalle de temps d'échantillonnage et $d(k)$ est la distance mise à jour par la valeur de $v(k)$. L'ensemble de contraintes ξ est donné par

$$v_b \leq v(k) \leq v_{k-1} \tag{5.17}$$

$$d_{\min} \leq d(k) \leq d_{\max} \tag{5.18}$$

$$d_{\max} \leq \beta \tag{5.19}$$

$$-a_{\max} \leq a_d(k) < 0 \tag{5.20}$$

$$\Delta a_d(k) < 3m/s^3 \tag{5.21}$$

$$0 \leq P_{RGN}(.) \leq \mathrm{MRL} \tag{5.22}$$

Tableau 5-3 Distance de freinage optimale pour une vélocité initiale et une efficacité énergétique choisie (ε).

Initial Velocity	Distance de freinage require pour une efficacité énergétique: $\varepsilon =$ 0%	25%	50%	75%	90%
10 km/h	2 m	2 m	2 m	2 m	2 m
20 km/h	4 m	4 m	4 m	4 m	4 m
30 km/h	6 m	6 m	6 m	6 m	8 m
40 km/h	10 m	10 m	10 m	14 m	16 m
50 km/h	16 m	16 m	18 m	28 m	32 m
60 km/h	22 m	22 m	28 m	46 m	54 m
70 km/h	30 m	30 m	46 m	66 m	80 m
80 km/h	40 m	40 m	68 m	104 m	116 m
90 km/h	50 m	50 m	92 m	146 m	166 m
100 km/h	60 m	60 m	122 m	180 m	234 m

La contrainte de vitesse dans l'équation 5.17 garantit que le véhicule maintient une décélération constante, ce qui est crucial pour un comportement de freinage prévisible et sûr. Ici, v_b représente la vitesse du véhicule à la fin de la sous-section du profil de freinage. Pour garantir que $d(k)$ reste dans ses seuils, l'équation 5.18 est appliquée. De plus, l'équation 5.19 assure que le véhicule maintient une distance de sécurité, par rapport à la cible la plus proche observée, β, empêchant ainsi les collisions potentielles. Pour renforcer encore la sécurité, l'équation 5.20 limite la décélération à des valeurs acceptables, garantissant que la force de freinage n'est ni trop brutale ni trop faible. Par ailleurs, les limites d'accélération sont indirectement régies par la contrainte de jerk dans l'équation 5.21, qui est définie pour éviter l'inconfort du conducteur et simplifier la génération de profils de freinage possibles. Enfin, la contrainte dans l'équation 5.22 garantit que la puissance générée lors du freinage ne dépasse pas la limite de régénération du moteur (MRL), protégeant ainsi la batterie et le moteur de tout dommage potentiel.

L'équation du système est calculée et mise à jour dans le domaine temporel, tandis que les entrées sont évaluées pour le même domaine. Cependant, les changements environnementaux, tels que la pente sont imposés dans le domaine spatial. Par conséquent, des calculs supplémentaires sont nécessaires pour l'optimisation dans le domaine temporel pour les systèmes nécessitant des informations sur la pente. Le nombre de nœuds à traverser dans l'algorithme de programmation dynamique (DP) est fixé à trois. Pour suivre l'environnement de manière plus réaliste, le Δt est fixé à $1/30$ s et la commande contrôlée par DP est répétée jusqu'à ce que l'étape de temps DP soit atteinte. L'algorithme DP est ensuite appliqué de manière récursive à chaque sous-chemin généré jusqu'à ce qu'un chemin complet soit généré en utilisant l'étape de temps souhaitée. L'étape de temps initiale est déterminée à l'aide d'une table de correspondance, fixée à 75% du temps optimal nécessaire pour atteindre la cible sur une route de niveau. Ce tableau de correspondance est construit à l'aide de profils de décélération optimaux générés par une version non récursive de la DP. Une table de correspondance raccourcie est présentée dans Tableau 5-3. Elle affiche la relation entre la vitesse initiale et la distance requise pour une certaine efficacité énergétique (ε). Plusieurs valeurs d'efficacité énergétique allant de 0 à 90% sont présentées. Les valeurs de distance sont discrétisées à 2 m, donc les profils nécessitant une courte portée de freinage, comme le 10 km/h affichent la même distance pour plusieurs efficacités. La variation des valeurs de ε est déterminée par la variabilité de la distance optimale et dépend du processus de perception sélectionnée, qui est définie dans la section suivante. Ce DP récursif permet une estimation rapide en temps réel du profil de freinage optimal. La vitesse optimale $v^(k)$ déterminée à chaque nœud est utilisée pour mettre à jour le déplacement:

$$d(k) = d(k+1) - v^(k)\Delta t \tag{5.23}$$

où est mise à jour par la vitesse optimale à l'instant k. L'efficacité énergétique (ε) de différents profils est évaluée en fonction de l'énergie totale de freinage nécessaire pour arrêter le véhicule:

$$\varepsilon = \frac{\int_{t_a}^{t_b} F_{rg}(t) \times v^{(}t) \times dt}{\int_{t_a}^{t_b} F_b(t) \times v^{(}t) \times dt} \tag{5.24}$$

Tableau 5-2 présente l'énergie de freinage requise et l'énergie récupérée grâce à une stratégie de freinage optimale pour un véhicule de 1500 kg se déplaçant à 90 km/h (25 m/s). Le résultat montre qu'une distance plus longue est nécessaire pour atteindre une efficacité énergétique plus élevée. Ce tableau révèle également que la force de freinage totale pour arrêter le véhicule diminue avec la distance à mesure que les forces environnementales augmentent. Cela limite la distance de freinage, car une décélération au-delà du point de récupération d'énergie maximale entraîne une régénération d'énergie moindre par rapport aux périodes de freinage plus courtes. Un paramètre crucial affectant l'efficacité de l'OCP pour le système de freinage régénératif est β. L'utilisation d'une valeur constante pour β est inefficace et peut entraîner une récupération d'énergie incorrecte, affectant à la fois l'efficacité énergétique et la sécurité. Pour remédier à cela, la présente thèse se concentre sur la modélisation de β en temps réel grâce au processus de perception. Cet ajustement dynamique garantit que le système de freinage fonctionne de manière optimale, maximisant la récupération d'énergie tout en maintenant des conditions de conduite sûres. La méthode pour réaliser cette modélisation est détaillée dans la section suivante.

5.2 Système de détection

Plusieurs blocs de perception sont utilisés pour cette détection à longue portée en temps réel. Ces blocs contiennent tout, de l'amélioration de l'image à la segmentation,

la fusion et la détection d'objets.

5.2.1 Blocs proposés

Amélioration de l'image

Un premier type de bloc disponible est les blocs d'amélioration d'image. Bien que plusieurs types existent, seulement deux types d'amélioration de l'image sont utilisés dans cette thèse. Le premier type d'amélioration est la manipulation des données lors de l'entraînement. Des données artificielles sont utilisées pour les données manquantes ou de faible qualité. Telle que vue dans le chapitre 3, l'amélioration des données lors de l'entraînement corrèle directement avec les résultats obtenus lors de la détection. Un exemple d'amélioration nécessaire est de combler les données manquantes. Certains modèles de reconnaissance nécessitent des images thermiques, mais certains jeux de données n'en contiennent pas. Par conséquent, la génération de données artificielles est utilisée. Un réseau U-Net [111], qui est un réseau efficace principalement appliqué pour la segmentation sémantique en milieu médical, peut également convertir une image en une autre. Ainsi, des images thermiques sont créées à partir d'images couleur par un réseau U-Net. Ces images thermiques ne contiennent pas de caractéristiques spécifiques au spectre thermique, cependant, elles permettent aux réseaux nécessitant des images thermiques d'être traités malgré ce manque. La génération de données artificielles est aussi utilisée pour compléter les jeux de données dans le cas où l'image thermique est dégradée au-delà de l'utilisation.

La deuxième amélioration de l'image utilisée est l'amélioration de la qualité. L'effet des conditions météorologiques immédiates telles que les gouttes de pluie et les flocons de neige est réduit par une autre amélioration pour les images. La méthode image à image de U-Net est appliquée pour réduire l'effet des conditions

météorologiques et recréer une image de qualité suffisante artificiellement à partir de l'image dégradée. L'entraînement de ce réseau U-Net passe par l'ajout de météo artificielle sur les images d'origines afin de créer une paire dégradé/qualité.

Segmentation

Afin d'arriver rapidement à un système de perception, les réseaux de segmentations ont été choisis pour l'entraînement. Bien que la tâche de segmentation soit différente de la tâche de détection, la quantité de données disponibles pour la segmentation et l'utilisation de détecteur faible performance sur les images présegmentées obtiennent de bons résultats pour un faible coût. Les réseaux de segmentation basés sur les transformateurs sont inclus dans certains des réseaux de perception les plus efficaces actuellement publiés. Le système de perception embarqué dans un véhicule nécessite un temps réel et un coût de traitement faible. Les réseaux de pyramides de caractéristiques [112] et les systèmes basés sur U-Net [111] fonctionnent efficacement avec des extracteurs de caractéristiques pour fournir des images de segmentation en utilisant des dispositifs à faible puissance. Plus de 200 réseaux de segmentation ont été évalués, y compris U-net combiné avec cinq variantes de Resnet [113], ResNext [114], Res2Net [115], 24 variantes de RegNetX [116], EfficientNet [117], RepVGG [118], MobileNet [119], Mix Vision Transformer (MiT) [120], et d'autres. Les architectures U-Net++ [121], FPN [112] et DeepLabV3 [122] ont également été évaluées avec les mêmes extracteurs de caractéristiques. Un sous ensemble de modèles a pu fonctionner avec un minimum de 30hz sur le Nvidia Jetson AGX Xavier parmi tous les modèles évalués. Les trois meilleurs extracteurs de caractéristiques et deux architectures ont été choisis parmi les performances annoncées de ces réseaux. Les segmenteurs ont l'avantage de pouvoir

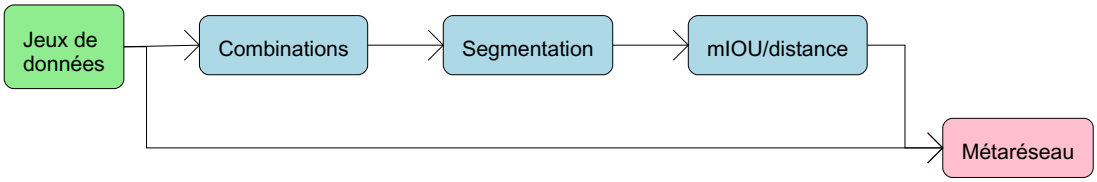


Figure 5.1 Processus d'entraînement pour le métaréseau

effectuer différentes tâches liées au véhicule autonome simultanément.

Évaluation de la distance

Des boîtes englobantes d'obstacles sont nécessaires pour évaluer la distance de chaque objet. Des détecteurs à faible coût de traitement sont utilisés pour une détection haute performance, car l'image de segmentation n'a pas de motif complexe et les classes sont déjà classifiées. Le Tinier-YOLO [123] est appliqué pour convertir les images de segmentation en boîtes utilisables. Les boîtes englobantes sont ensuite utilisées dans des estimateurs de localisation 3d, tels que l'estimateur de distance multiobjets [124]. Ces systèmes d'estimation de positions sont basés sur la déformation géométrique de la lentille de caméra et la connaissance de la taille des objets permettant d'approximer leur position en 3D dans le monde à partir d'une seule image.

5.2.2 Métaréseau

La nature statistique des poids et des métriques préfère toujours certaines caractéristiques à d'autres [133]. De ce fait, certains réseaux avec de meilleures métriques d'évaluation pourraient être moins performants sur certaines images où des réseaux avec des métriques inférieures pourraient identifier correctement l'image

Tableau 5-4 Jeux de données de conduite

Jeu de données	Conduite	Visible	Thermique	Lidar	Été	Hiver
ImageNet [125]	Sous-ensemble	✓	✗	✗	✓	✗
VoC [126]	Sous-ensemble	✓	✗	✗	✓	✗
COCO [127]	Sous-ensemble	✓	✗	✗	✓	✗
Citiscapes [128]	✓	✓	✗	✓	✓	✗
Kitti [129]	✓	✓	✗	✓	✓	✗
FLIR [106]	✓	✓	✓	✗	✓	✗
CADC [130]	✓	✓	✓	✓	✓	✓
RADIATE [131]	✓	✓	✓	✓	✓	✓
Argoverse2 [132]	✓	✓	✓	✓	✓	✓
UQTR V1 [1]	✓	✓	✓	✓	✓	✓
UQTR V2 [89]	✓	✓	✓	✓	✓	✓
UQTR V3	✓	✓	✓	✓	✓	✓

spécifique. Une solution par image est nécessaire pour identifier quelle combinaison d'amélioration de l'image, d'architectures, d'extracteurs de caractéristiques et de l'entraînement surpasserait les autres. Un classificateur pourrait être utilisé à l'aveugle pour identifier quelles combinaisons de blocs seraient les plus performantes comme première solution. Cependant, plusieurs réseaux pourraient avoir des performances similaires sur une seule image, et en tant que telles, plusieurs classes seraient activées à la fois. La performance de segmentation et la distance à laquelle les objets peuvent être détectés de manière fiable sont modélisées simultanément par un réseau de régression. La métrique standard mIoU pourraient définir la performance de segmentation:

$$mIoU = \sum_{n=0}^{N} \frac{\text{Intersection}}{\text{Union}} \times \frac{1}{N} \tag{5.25}$$

où N est le nombre de classes de segmentation. Cette métrique exclut les classes non présentes dans une image. La distance maximale où la précision de détection est supérieure à un seuil de 90% est utilisée pour l'évaluation de la portée. La distance est

empiriquement plafonnée à 150m dans l'évaluation, puisqu'un maximum est requis pour les images où tous les objets sont correctement détectés. Ces métriques de performance servent également de mesures de sécurité utilisables pour définir le seuil de reprise par l'humain. Un classificateur simple tel que AlexNet [96] est utilisé comme réseau de régression pour un traitement rapide et économe en énergie des images. L'utilisation des métriques plutôt qu'un classificateur permet à l'approche de se généraliser et de fournir des métriques de performances sur des images en dehors du champ d'application initialement entraîné. Cela permet l'évaluation de la performance des réseaux en temps réel et d'assurer un seuil minimum de performance ainsi que de connaître les environnements dans lequel aucune combinaison n'as une performance acceptable.

Entraînement

Chaque combinaison est appliquée sur chaque image de l'ensemble de données et la performance est recueillie pour entraîner ce métaréseau. Les étapes pour obtenir les informations et entraîner les réseaux méta sont présentées dans la figure 5.1. Une image de segmentation pour chaque combinaison de blocs est obtenue en passant l'image originale à travers celle-ci. Les métriques mIoU et de distance pour ladite combinaison sur l'image sont recueillies en passant cette image de segmentation à travers le détecteur et l'évaluateur de distance. Les métriques de performance de chaque réseau sont ensuite concaténées et agissent comme vérité terrain lors de l'entraînement du métaréseau. Les images sont rapidement transmises au métaréseau en tant qu'entrées.

Une quantité colossale de données est nécessaire pour l'entraînement et l'évaluation [134]. Certains des ensembles de données les plus importants pour

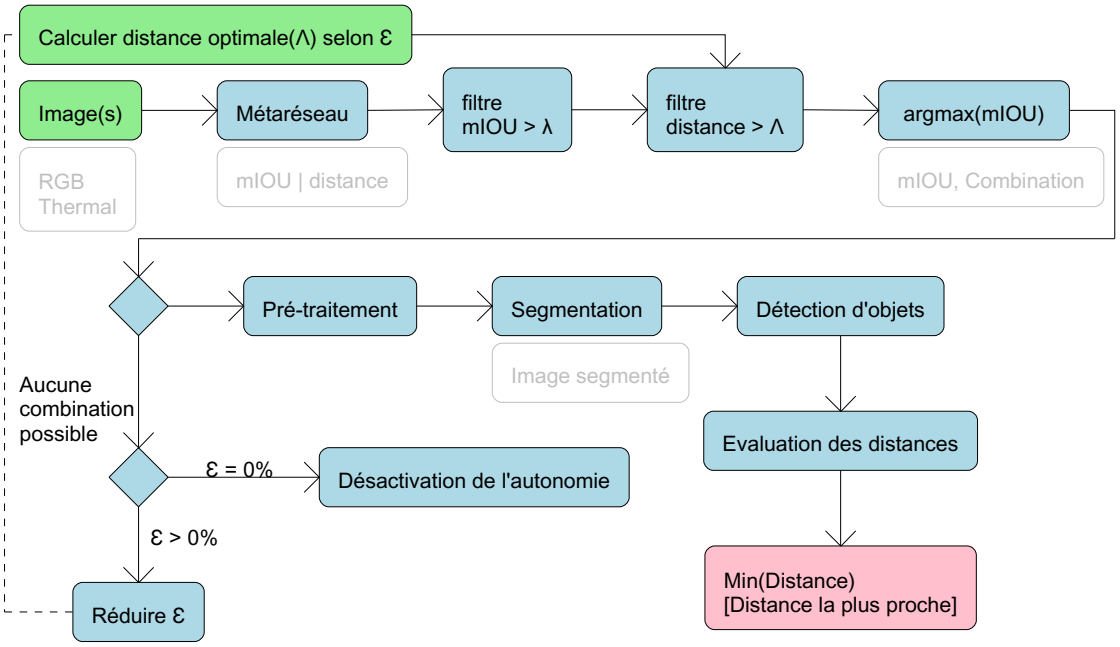


Figure 5.2 Processus d'inférence en temps réel

les voitures autonomes sont affichés dans le tableau 5-4. Tous les ensembles de données présentés sont fusionnés pour former l'ensemble d'entraînement pour la méthode proposée, puisqu'aucun ensemble de données à grande échelle ne contient d'informations de toutes les conditions météorologiques et capteurs.

Inférence en temps réel

Un processus multiétapes est utilisé pour l'inférence en temps réel des objets et l'évaluation de la distance. Ce processus complexe est démontré sous forme de diagramme de flux dans la figure 5.2. Tout d'abord, pour des raisons de sécurité, une valeur Λ est calculée en utilisant le DP pour la distance de détection sûre pour l'efficacité énergétique souhaitée (ε). Les images sont ensuite envoyées au métaréseau pour modéliser la performance de chaque combinaison disponible. Lors de l'évaluation, le méta réseau produit la métrique de performance sans l'exécution

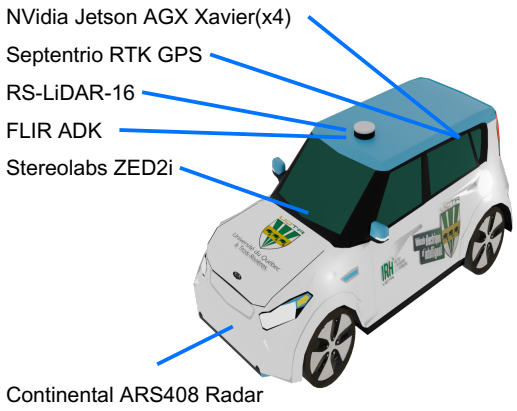


Figure 5.3 Le véhicule et les capteurs utilisés pour les tests expérimentaux.

de ces combinaisons. Les performances de chaque combinaison sont ensuite filtrées d'abord sur la performance de perception minimale (λ). Les résultats sont ensuite davantage filtrés en fonction de la distance dans laquelle la détection est jugée fiable avec le Λ de l'efficacité énergétique souhaitée. La meilleure combinaison est ensuite sélectionnée en fonction de la métrique de segmentation (mIoU). L'efficacité énergétique est réduite jusqu'à ce qu'une combinaison soit disponible dans l'événement où aucune combinaison n'est disponible.

5.3 Résultats expérimentaux et discussion

Les tests expérimentaux de l'architecture proposée sont effectués sur une Kia Soul EV 2017 qui est contrôlée par un dispositif de conduite par fil, en l'occurrence, un dispositif DriveKit de la compagnie PolySync, aujourd'hui défunte. Les dispositifs utilisés pour la perception du véhicule sont montrés dans la figure 5.3. Une caméra Stereolabs Zed2i fournit le flux vidéo du spectre visible, tandis qu'un Flir ADK monté sur le toit fournit le flux vidéo thermique. Le radar Continental ARS408 et le RS-LiDAR-16 fournissent une distance de secours au planificateur d'énergie

Tableau 5-5 Évaluation des améliorations de perception

Méthode \ Condition	Amélioration d'image	Ensoleillé	Neigeux	Pluvieux	Ensoleillé	Neigeux	Pluvieux	Ensoleillé	Neigeux	Pluvieux	Ensoleillé	Neigeux	Pluvieux
		SP			mIoU			distance (m)			Énergie récupérable (J)		
FPN + MiT-b1	✓	0.9707	0.9226	0.9591	0.8733	0.7272	0.8311	40.91	40.20	42.28			
FPN + MiT-b1	✗	0.9707	0.8971	0.9061	0.8733	0.5899	0.6136	40.63	40.86	39.79			
FPN + MiT-b3	✓	0.9171	0.8377	0.9067	0.8287	0.6642	0.7890	44.07	40.15	41.40			
FPN + MiT-b3	✗	0.9171	0.8099	0.8206	0.8287	0.5371	0.5615	37.06	36.07	35.58			
FPN + MiT-b5	✓	0.8687	0.8227	0.8569	0.7871	0.6520	0.7481	53.88	46.29	44.08			
FPN + MiT-b5	✗	0.8687	0.7602	0.7757	0.7871	0.5024	0.5298	32.19	31.54	28.79			
UNet + MiT-b1	✓	0.9775	0.8474	0.9358	0.8794	0.6721	0.8123	51.99	54.07	56.35			
UNet + MiT-b1	✗	0.9775	0.8177	0.8394	0.8794	0.5409	0.5726	44.73	49.44	54.30			
UNet + MiT-b3	✓	0.8754	0.7535	0.8455	0.7934	0.6022	0.7391	61.55	59.02	60.44			
UNet + MiT-b3	✗	0.8754	0.6974	0.7321	0.7934	0.4657	0.5053	52.30	51.74	49.85			
UNet + MiT-b5	✓	0.8279	0.7271	0.8020	0.7517	0.5818	0.7026	70.73	55.64	65.77			
UNet + MiT-b5	✗	0.8279	0.7253	0.7476	0.7517	0.4828	0.5143	49.26	41.29	45.81			
exfuse [135]	✓	0.9372	0.8981	0.8052	0.8414	0.7079	0.6977	71.29	57.28	63.27			
exfuse [135]*	✗	0.9625	0.9033	0.8288	0.8641	0.7121	0.7181	73.91	52.11	44.61			
Sélecteur proposé	Conditionnel	0.9971	0.9425	0.9672	0.8954	0.7429	0.8381	74.65	62.94	65.99			
Sélecteur parfait	Conditionnel	1.0000	1.0000	1.0000	0.8978	0.7883	0.8665	75.51	65.39	69.03			

* Le réseau ne peut opérer avec des données incomplètes, puisqu'il requière des données thermiques. Seul un sous-ensemble du jeu de donnée est utilisé.

indépendamment de la perception visuelle. Le flux vidéo et des capteurs sont traités en temps réel par quatre unités Nvidia Jetson AGX Xavier, qui génèrent le profil de freinage optimal lorsque nécessaire. Les profils sont ensuite envoyés à l'unité de contrôle qui convertit le profil généré et la trajectoire en déplacement du véhicule en utilisant la technologie de conduite par fil.

5.3.1 Résutats de la méthode par métaréseau

Les résultats de segmentation sont résumés dans Tableau 5-5. La colonne 'méthode' détaille les combinaisons de perception utilisées pour obtenir ces résultats, avec le réseau ExFuse [135] inclus à des fins de comparaison. De plus, le métaréseau proposé fait référence à la méthode de perception introduite dans ce chapitre et le sélecteur parfait représente un modèle idéalisé qui identifie systématiquement la méthode avec les meilleures métriques de performance pour chaque image. Il

sert également de référence pour l'entraînement du sélecteur proposé. La colonne d'Amélioration d'image spécifie si le bloc d'amélioration d'image a été intégré à la combinaison de perception. Les conditions météorologiques (Ensoleillé, Neigeux et Pluvieux) sont listées séparément pour chaque métrique dans différentes colonnes. Le tableau comprend quatre métriques: mIOU, potentiel de segmentation (PS), $/Psi$. Pour permettre une comparaison plus complète de tous les réseaux et fournir des valeurs au-delà de celles offertes par mIOU, la métrique SP est également utilisée. Cette métrique est définie comme le ratio de la mIOU d'un modèle spécifique à la mIOU maximale atteinte par n'importe quel réseau pour un ensemble donné d'images:

$$SP = \frac{\text{mIOU}}{\text{optimal mIOU}} \tag{5.26}$$

Le meilleur résultat de chaque colonne est mis en évidence en vert, tandis que le deuxième meilleur est mis en évidence en rouge. Selon les conditions ,ensoleillées, neigeuses ou pluvieuses,différentes méthodes peuvent atteindre les deuxièmes meilleurs résultats. Par temps ensoleillé, U-Net + MiT-b1, avec et sans l'amélioration de l'image, surpasse toutes les autres piles de perception, tandis que l'architecture FPN + MiT-b1 donne de meilleurs résultats pour les conditions neigeuses et pluvieuses. Le système proposé basé sur le métaréseau surpasse significativement les autres méthodes en offrant une sélection en temps réel des combinaisons optimales. En conditions ensoleillées, il atteint un SP impressionnant de 99,71%. Même dans des environnements dégradés, il maintient de bonnes performances, atteignant des SP de 94,25% et 96,72%. Concernant la métrique mIOU, le métaréseau proposé atteint des valeurs de 0,8954 en conditions ensoleillées, 0,7429 en conditions neigeuses et 0,8381 en conditions pluvieuses, marquant une amélioration substantielle par rapport aux méthodes existantes.

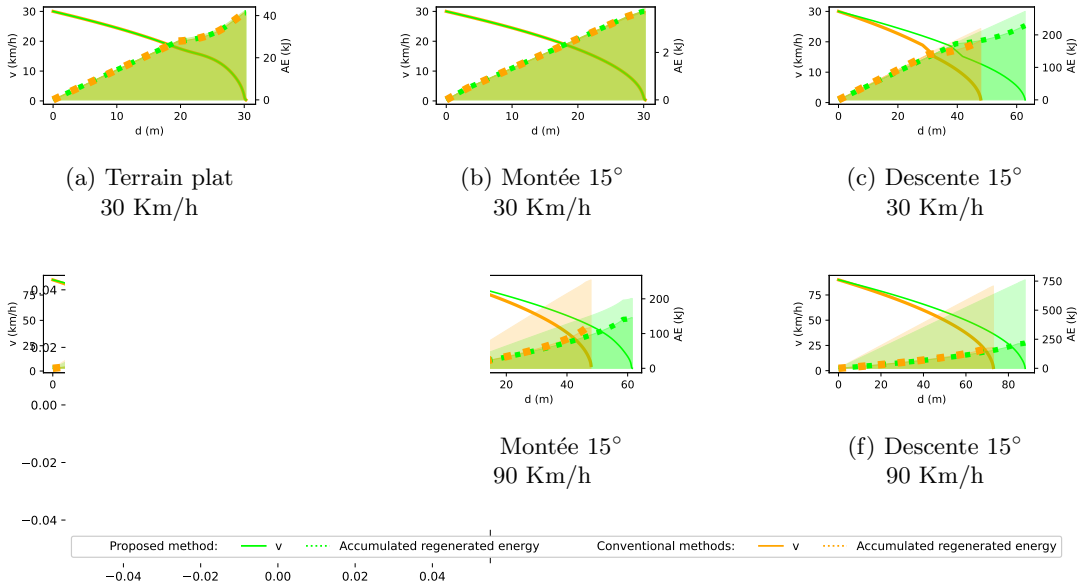
Dans le cadre de cette thèse, le non-biais de la métrique a été prouvé sur plusieurs centaines de sous-sections du jeu de données. Pour souci de clarté, seules les sections ayant un sens explicable tel que les conditions météo sont présentées.

5.3.2 Résultats d'énergie

Le tableau 5-5 présente aussi les résultats expérimentaux en termes de distance et d'énergie. Il inclut spécifiquement les métriques de $\mathbf{\Psi}$ (m) et d'énergie régénérée (J). Ici, $\mathbf{\Psi}$ désigne la distance à laquelle les réseaux peuvent détecter 90% des boîtes englobantes avec une grande confiance. Pour les images qui sont entièrement classées correctement, une distance de 150 m est assignée comme valeur empirique. De plus, l'énergie régénérée représente la récupération potentielle d'énergie réalisée sur la distance de freinage depuis 90 km/h. Les résultats révèlent que le sélecteur proposé démontre de solides performances dans diverses conditions météorologiques, approchant de près la référence idéalisée établie par le Sélecteur parfait. La méthode proposée atteint les valeurs de $\mathbf{\Psi}$ les plus élevés dans toutes les conditions testées : 74,65 m par temps ensoleillé, 62,94 m par temps neigeux et 65,99 m par temps pluvieux. Ces résultats indiquent que le sélecteur proposé détecte efficacement 90% des boîtes englobantes avec une grande confiance, atteignant presque la performance du Sélecteur parfait, qui obtient des valeurs légèrement plus élevées, mais nécessite des conditions idéales. En termes de régénération d'énergie, le sélecteur proposé montre également des performances remarquables, enregistrant les valeurs d'énergie régénérée les plus élevées —181 599 J par temps ensoleillé, et atteignant systématiquement 153 663 J par temps neigeux et pluvieux. Ces résultats sont proches de ceux du Sélecteur parfait, qui surpasse légèrement la méthode proposée dans des conditions pluvieuses. Le réseau ExFuse, inclus à des fins de comparaison,

fonctionne bien par temps ensoleillé avec un Ψ de 71,29 m et une énergie régénérée de 171 212 J. Cependant, ses performances diminuent par temps neigeux et pluvieux, où il est surpassé par les sélecteurs proposés. Les autres méthodes qui ont obtenu les deuxièmes meilleurs résultats, mises en évidence en orange, incluent le FPN + MiT-b3 (avec amélioration d'image) et l'UNet + MiT-b5 (avec amélioration d'image) dans diverses conditions. Ces méthodes de deuxième niveau mettent en évidence la variabilité des performances en fonction des conditions météorologiques et de la présence de l'amélioration de l'image. Cependant, elles ne surpassent pas le sélecteur proposé. Dans l'ensemble, la méthode de perception basée sur le métaréseau proposé rivalise de près avec le Sélecteur parfait, offrant des gains substantiels en termes de distance de détection et de régénération d'énergie dans diverses conditions météorologiques. Ses performances en font une solution robuste et pratique pour les systèmes autonomes réels, offrant des résultats qui sont presque au même niveau que le modèle idéalisé. Les méthodes de deuxième niveau offrent des points de référence importants et démontrent l'influence de différentes configurations sur les performances, soulignant l'importance de sélectionner la pile de perception optimale pour des conditions spécifiques.

Il est important de prendre en compte que la quantité d'énergie accumulée présenté dans cette section de résultat est l'intégrale de P_{RGN} et non une variation de l'état de charge. Ceci est causé par l'utilisation d'un modèle en boîte noire de la batterie et du moteur. Concernant les résultats énergétiques, en tenant compte des résultats présentés dans le Tableau 5-3 et le Tableau 5-5, atteindre 90% d'efficacité énergétique lors du freinage à partir de 100 km/h est actuellement inatteignable. Pour atteindre un tel niveau d'efficacité, un Ψ de 234 mètres serait nécessaire, une capacité que la technologie existante manque, à l'exception des systèmes V2X. Les conditions



(a) Terrain plat 30 Km/h

(b) Montée 15° 30 Km/h

(c) Descente 15° 30 Km/h

Montée 15° 90 Km/h

(f) Descente 15° 90 Km/h

Figure 5.4 Énergie régénérée selon 6 profils environnementaux

1

météorologiques défavorables, telles que la neige, réduisent le $\mathbf{\Psi}$, entraînant des distances plus courtes pour optimiser le profil de freinage. Dans de telles conditions, la vitesse maximale du véhicule doit être réduite pour garantir qu'elle reste dans la plage de $\mathbf{\Psi}$ réalisable. Sans un profil de freinage approprié pour le $\mathbf{\Psi}$ fourni, le véhicule ne peut pas fonctionner de manière autonome, priorisant ainsi la sécurité des usagers de la route.

La figure 5.4 illustre l'efficacité de la méthode proposée en montrant l'énergie régénérée accumulée (ARE) sous forme de lignes pointillées, et la vitesse (v) sous forme de lignes solides, en fonction du déplacement du véhicule (d) sur six scénarios de freinage différents. La zone colorée représente l'énergie totale de freinage fournie par le véhicule. Les sous-figures (a-c) représentent des profils de freinage à une vitesse de 30 km/h, tandis que les sous-figures (d-f) représentent un freinage à une vitesse d'autoroute de 90 km/h. Dans chaque sous-figure, la ligne orange montre le profil de

freinage en utilisant la moyenne de Ψ des piles de perception étudiées, tandis que la ligne verte représente le profil de freinage obtenu avec la détection basée sur le métaréseau proposé. Les types de terrain sont les suivants : les sous-figures a et d montrent un terrain plat, les sous-figures (b,e) montrent un terrain en montée à 15°, et les sous-figures (c,f) illustrent un terrain en descente à 15°. La valeur de 15° a été sélectionnée car elle représente l'une des pentes les plus abruptes autorisées sur les routes publiques, ce qui a produit des différences notables dans les graphiques résultants. Pour des raisons de sécurité, les profils doivent décélérer sur au moins 30 mètres avant de s'arrêter complètement. Dans le scénario de descente à 15°, 25 mètres supplémentaires sont fournis par rapport aux autres scénarios, car un freinage sur une distance plus courte n'était pas faisable et les objets sont plus faciles à détecter à de plus grandes distances en descendant.

Les résultats des sous-figures (a-b) pour des vitesses plus basses montrent que la méthode proposée n'améliore pas significativement la récupération d'énergie par rapport aux méthodes conventionnelles. Ainsi, les profils de récupération d'énergie sont presque identiques. Cela peut s'expliquer facilement par le fait que leur efficacité ($\boldsymbol{\varepsilon}$) est déjà à 100%. Par conséquent, prolonger la distance de freinage n'offre aucune amélioration supplémentaire de la récupération d'énergie. La sous-figure (c) montre que sur une pente descendante à basse vitesse, augmenter la distance de freinage améliore légèrement la récupération d'énergie, avec une efficacité ($\boldsymbol{\varepsilon}$) proche de 100%. Cependant, les sous-figures (d-f) pour des vitesses plus élevées montrent une augmentation significative de l'énergie régénérée en utilisant la méthode proposée, qui bénéficie de 20 mètres supplémentaires de déplacement. La valeur de $\boldsymbol{\varepsilon}$ est indirectement montrée comme la proportion de la ligne pointillée par rapport à l'énergie totale. $\boldsymbol{\varepsilon}$ varie considérablement selon les scénarios : à basse vitesse, elle

approche 100%, car la distance de freinage optimale est facilement atteinte. En revanche, à des vitesses plus élevées ou sur des pentes descendantes où une force de freinage plus grande est nécessaire, $\boldsymbol{\varepsilon}$ tend à être plus faible. En revanche, dans des cas comme les pentes montantes, où une force de freinage moindre est requise, $\boldsymbol{\varepsilon}$ atteint souvent 100% même sur des distances plus courtes. Cela suggère que bien que l'augmentation de la distance de freinage ($\boldsymbol{\Psi}$) ne puisse pas toujours améliorer la récupération d'énergie à basse vitesse, elle devient essentielle pour maximiser la récupération d'énergie à des vitesses plus élevées.

Pour conclure les résultats expérimentaux, chaque combinaison de perception s'est avérée efficace pour des conditions météorologiques spécifiques. Cependant, la méthode basée sur le métaréseau proposé a montré une adaptabilité aux changements environnementaux, permettant de sélectionner dynamiquement la combinaison de perception optimale pour chaque condition. Cette adaptabilité a permis à la méthode proposée d'atteindre des résultats qui correspondent de près à ceux du Sélecteur parfait idéalisé. L'importance de cette approche était particulièrement évidente à des vitesses de véhicule élevées, où les considérations de sécurité deviennent primordiales. La capacité de la méthode proposée à améliorer la distance de détection et la régénération d'énergie dans ces conditions souligne son importance pour les systèmes autonomes réels, offrant une solution fiable et pratique qui excelle même dans des environnements difficiles.

5.4 Conclusion

Ce chapitre propose un système de perception pour les appareils de bord qui peut augmenter la distance de détection, la sécurité et l'efficacité énergétique. Le système maximise la récupération d'énergie grâce au freinage régénératif, qui dépend de la

distance de freinage. Le système utilise un métaréseau qui modélise la performance de diverses combinaisons de techniques de pré-traitement, de traitement et de post-traitement. Le système sélectionne la meilleure combinaison en temps réel pour traiter l'image. Le système limite également la vitesse et assure la sécurité lorsque la portée de détection n'est pas suffisante pour fournir un profil de freinage dans la limite de performance énergétique définie par l'utilisateur. Le problème de contrôle optimal pour le freinage est formulé en utilisant un modèle de véhicule longitudinal à masse ponctuelle avec somme des forces. Un algorithme DP récursif fournit la stratégie de freinage optimale basée sur les contraintes du véhicule et de l'environnement pour un freinage régénératif optimal en énergie. La distance de décélération est le facteur clé pour augmenter l'énergie totale sous la contrainte des limites du moteur du véhicule et du confort de l'utilisateur. Les profils de décélération sont testés en utilisant un véhicule à commande par fil avec mesure en temps réel de la puissance de la batterie pour validation et vérification. L'approche est évolutive pour les nouveaux réseaux de segmentation en temps réel.

Chapitre 6 - Conclusion

Nous avons vu qu'avec l'arrivée des véhicules autonomes, viennent de multiples problèmes. L'automatisation des véhicules est un défi majeur pour les chercheurs du monde entier. Ces véhicules doivent non seulement être capables de se déplacer de manière autonome, mais aussi de réduire leur consommation d'énergie, notamment en utilisant l'électricité comme source d'énergie. Pour cela, ils doivent optimiser leurs trajectoires, leurs freinages et leur anticipation des objets. Cependant, l'anticipation est rendue difficile par les conditions météorologiques, qui affectent la qualité des détections. La thèse actuelle vise à améliorer ces détections en hiver, afin d'augmenter la sécurité et les performances énergétiques des véhicules. En effet, en optimisant les freinages, on peut économiser jusqu'à 30% d'énergie électrique.

6.1 Améliorations effectuées par cette thèse

Quant à l'objectif de cette thèse d'améliorer la sécurité et l'efficacité énergétique d'un véhicule intelligent, ceci a été séparé en sous-objectif. Le premier étant de détecter de manière robuste les obstacles en condition difficile. Cet objectif est couvert dans le chapitre 3, le chapitre 4 et le chapitre 5. Concernant le deuxième sous-objectif de fournir des indices de freinage optimisé au conducteur, cette tâche est une section du chapitre 5.

Le chapitre 3 de cette thèse démontre que l'utilisation de la saillance permet d'améliorer les données d'entraînement des systèmes d'apprentissage profonds et d'ainsi améliorer les résultats pour les conditions difficiles. Ce travail s'accompagne d'un outil d'annotation et d'un jeu de donnée.

Le chapitre 4 montre que l'utilisation des données thermique lorsque bien utilisé

permet d'augmenter la qualité des détections la nuit, en conditions de brouillard et de neige. La méthode proposée augmente aussi la résilience aux bris de capteur par une méthode adaptative.

Le chapitre 5, quant à lui, intègre les résultats précédents et propose une solution au deuxième sous-objectif. Ce chapitre propose une méthode d'intégration de multiples réseaux avec sélection intelligente afin d'obtenir les meilleurs résultats possibles selon les conditions météorologiques. L'utilisation de plusieurs combinaisons de réseaux de détection et de méthodes à la pointe de la recherche permet lorsque bien sélectionné d'augmenter la détection sans nouvel entraînement. Concernant l'énergie, l'amélioration de la détection est directement responsable d'une détection des objets à 5m de distance de plus, récupérant en moyenne 5Wh par freinage.

6.2 Recommandations

Cette thèse a démontré que l'utilisation de meilleures données ainsi que la fusion de plusieurs capteurs fournissaient un meilleur résultat de détection et qu'il existe des combinaison de méthode et capteur meilleur selon les conditions météorologiques. Plusieurs aspects de recherche peuvent être considérés comme axe de recherche futur par rapport à la recherche de cette thèse. Premièrement, des recherches approfondies doivent être faites afin de faire la fusion de données n'étant pas sous le format de matrice 2D tel qu'une image. Concernant l'aspect d'optimisation énergétique, des expérimentations doivent être faites pour tenir compte de l'état de charge ainsi que l'efficacité énergétique de la batterie et du moteur du véhicule. En troisième lieu, un aspect important à rechercher est la possibilité d'évaluer des méthodes d'optimisation des trajectoires prenant en compte la vélocité et trajectoire d'objets dynamique ainsi que de vérifier que l'amélioration de la détection est applicable aux objets

dynamiques. En dernier lieu, l'utilisation de la méthode de métaréseau par apport à la détection des voies de circulation afin de prédire des trajectoires non linéaires pour le freinage.

Bibliographie

[1] J. Boisclair, S. Kelouwani, F. K. Ayevide, A. Amamou, M. Alam, and K. Agbossou, "Attention transfer from human to neural networks for road object detection in winter," IET Image processing, 2022.

[2] P. G. Anselma, "Optimization-driven powertrain-oriented adaptive cruise control to improve energy saving and passenger comfort," Energies, vol. 14, no. 10, p. 2897, 2021.

[3] M. Shen, C. R. He, T. G. Molnar, A. H. Bell, and G. Orosz, "Energy-efficient connected cruise control with lean penetration of connected vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 4, pp. 4320–4332, 2023.

[4] C. Pan, A. Huang, J. Wang, L. Chen, J. Liang, W. Zhou, L. Wang, and J. Yang, "Energy-optimal adaptive cruise control strategy for electric vehicles based on model predictive control," Energy, vol. 241, p. 122793, 2022.

[5] A. K. Podder, O. Chakraborty, S. Islam, N. M. Kumar, and H. H. Alhelou, "Control strategies of different hybrid energy storage systems for electric vehicles applications," IEEE Access, vol. 9, pp. 51 865–51 895, 2021.

[6] D. Kim, J. S. Eo, and K.-K. K. Kim, "Parameterized energy-optimal regenerative braking strategy for connected and autonomous electrified vehicles: A real-time dynamic programming approach," IEEE Access, vol. 9, pp. 103 167–103 183, 2021.

[7] J. Zhang, Y. Yang, D. Qin, C. Fu, and Z. Cong, "Regenerative braking control method based on predictive optimization for four-wheel drive pure electric vehicle," IEEE Access, vol. 9, pp. 1394–1406, 2020.

[8] P. Polverino, I. Arsie, and C. Pianese, "Optimal energy management for hybrid electric vehicles based on dynamic programming and receding horizon," Energies, vol. 14, no. 12, p. 3502, 2021.

[9] A. T. Hamada and M. F. Orhan, "An overview of regenerative braking systems," Journal of Energy Storage, vol. 52, p. 105033, 2022.

[10] R. Sato, K. Domany, D. Deguchi, Y. Mekada, I. Ide, H. Murase, and Y. Tamatsu, "Visibility estimation of traffic signals under rainy weather conditions for smart driving support," in 2012 15th International IEEE Conference on Intelligent Transportation Systems, Conference Proceedings, pp. 1321–1326.

[11] T. Zhe, L. Huang, Q. Wu, J. Zhang, C. Pei, and L. Li, "Inter-vehicle distance estimation method based on monocular vision using 3d detection," IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 4907–4919, 2020, cited By :3 Export Date: 1 December 2020.

[12] Z. Zhao, P. Zheng, S. Xu, and X. Wu, "Object detection with deep learning: A review," IEEE Transactions on Neural Networks and Learning Systems, vol. 30, no. 11, pp. 3212–3232, 2019.

[13] C. Lundquist, "Sensor fusion for automotive applications," Thesis, 2011.

[14] M. Malik and S. Majumder, "An integrated computer vision based approach for driving assistance to enhance visibility in all weather conditions," in International and National Conference on Machines and Mechanisms, Conference Proceedings.

[15] J. Wu, H. Xu, Y. Tian, R. Pi, and R. Yue, "Vehicle detection under adverse weather from roadside lidar data," Sensors, vol. 20, no. 12, p. 3433, 2020.

[16] J. Wu, H. Xu, J. Zheng, and J. Zhao, "Automatic vehicle detection with roadside lidar data under rainy and snowy conditions," IEEE Intelligent Transportation Systems Magazine, vol. 13, no. 1, pp. 197–209, 2020.

[17] A. Palffy, J. Dong, J. F. Kooij, and D. M. Gavrila, "Cnn based road user detection using the 3d radar cube," IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 1263–1270, 2020.

[18] Y. Wang, Z. Han, Y. Xing, S. Xu, and J. Wang, "A survey on datasets for the decision making of autonomous vehicles," IEEE Intelligent Transportation Systems Magazine, 2024.

[19] K. Tan, J. Wu, H. Zhou, Y. Wang, and J. Chen, "Integrating advanced computer vision and ai algorithms for autonomous driving systems," Journal of Theory and Practice of Engineering Science, vol. 4, no. 01, pp. 41–48, 2024.

[20] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," IEEE Transactions on Vehicular Technology, vol. 66, no. 2, pp. 952–964, 2017.

[21] A. Nakamura, Y. C. Liu, and B. Kim, "Short-term multi-vehicle trajectory planning for collision avoidance," IEEE Transactions on Vehicular Technology, vol. 69, no. 9, pp. 9253–9264, 2020.

[22] C. Yuan, H. Liu, J. Shen, L. Chen, and H. Jiang, "Design and analysis of an autobraking system controller for autonomous vehicles under the influence of perturbation," IEEE Transactions on Vehicular Technology, vol. 67, no. 3, pp. 1923–1931, 2018.

[23] M.-T. Leung, W. Engeler, and P. Frank, "Fingerprint image processing using neural networks," in IEEE TENCON'90: 1990 IEEE Region 10 Conference on Computer and Communication Systems. Conference Proceedings. IEEE, 1990, pp. 582–586.

[24] C. Nebauer, "Evaluation of convolutional neural networks for visual recognition," IEEE transactions on neural networks, vol. 9, no. 4, pp. 685–696, 1998.

[25] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel, "Benchmarking robustness in object detection: Autonomous driving when winter is coming," arXiv:1907.07484v1 [cs.CV], 2019.

[26] D. R. Musicant, V. Kumar, and A. Ozgur, "Optimizing f-measure with support vector machines," in FLAIRS conference, Conference Proceedings, pp. 356–360.

[27] A. Singha and M. K. Bhowmik, "Tu-vdn: Tripura university video dataset at night time in degraded atmospheric outdoor conditions for moving object detection," in 2019 IEEE International Conference on Image Processing (ICIP). IEEE, Conference Proceedings, pp. 2936–2940.

[28] Q.-H. Pham, P. Sevestre, R. S. Pahwa, H. Zhan, C. H. Pang, Y. Chen, A. Mustafa, V. Chandrasekhar, and J. Lin, "A*3d dataset: Towards autonomous driving in challenging environments," arXiv:1909.07541v1 [cs.CV], 2019.

[29] A. L. Pavlov, P. A. Karpyshev, G. V. Ovchinnikov, I. V. Oseledets, and D. Tsetserukou, "Icevisionset: lossless video dataset collected on russian winter roads with traffic sign annotations," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, Conference Proceedings, pp. 9597–9602.

[30] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000km: The oxford robotcar dataset," The International Journal of Robotics Research (IJRR), 2016.

[31] X. Li, F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, K. Li, and D. M. Gavrila, "A new benchmark for vision-based cyclist detection," in 2016 IEEE Intelligent Vehicles Symposium (IV). IEEE, Conference Proceedings, pp. 1028–1033.

[32] S. Hwang, J. Park, N. Kim, Y. Choi, and I. So Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in Proceedings of the IEEE conference on computer vision and pattern recognition, Conference Proceedings, pp. 1037–1045.

[33] Y. Xu, H. Wang, X. Liu, H. R. He, Q. Gu, and W. Sun, "Learning to see the hidden part of the vehicle in the autopilot scene," Electronics, vol. 8, no. 3, p. 331, 2019.

[34] Y. Li, G. Tong, H. Gao, Y. Wang, L. Zhang, and H. Chen, "Pano-rsod: A dataset and benchmark for panoramic road scene object detection," Electronics, vol. 8, no. 3, p. 329, 2019.

[35] A. Dominguez-Sanchez, M. Cazorla, and S. Orts-Escolano, "A new dataset and performance evaluation of a region-based cnn for urban object detection," Electronics, vol. 7, no. 11, p. 301, 2018.

[36] M. B. Jensen, M. P. Philipsen, A. Møgelmose, T. B. Moeslund, and M. M. Trivedi, "Vision for looking at traffic lights: Issues, survey, and perspectives,"

IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 7, pp. 1800–1815, 2016.

[37] T. Hodan, P. Haluza, O. Š, J. Matas, M. Lourakis, and X. Zabulis, "T-less: An rgb-d dataset for 6d pose estimation of texture-less objects," in 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Conference Proceedings, pp. 880–888.

[38] S. Chowdhuri, T. Pankaj, and K. Zipser, "Multinet: Multi-modal multi-task learning for autonomous driving," in 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), Conference Proceedings, pp. 1496–1504.

[39] Y. Fisher, X. Wenqi, C. Yingying, L. Fangchen, L. Mike, M. Vashisht, and D. Trevor, "Bdd100k: A diverse driving video database with scalable annotation tooling," arXiv:1805.04687 [cs.CV], 2018, export Year:2019.

[40] P. Matthew, G. Danson, R. Jason, S. Michael, W. Carlos, C. Krzysztof, and W. Steven, "Canadian adverse driving conditions dataset," arXiv:2001.10117 [cs.CV], 2020, export Year:2019.

[41] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, "Two stream lstm: A deep fusion framework for human action recognition," in 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Conference Proceedings, pp. 177–186.

[42] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," arXiv preprint arXiv:2010.04159, 2020.

[43] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, 2010.

[44] Y. Xi, A. David, and F. Sanja, "Neural data server: A large-scale search engine for transfer learning data," arXiv:2001.02799 [cs.CV], 2020, export Year:2019.

[45] K. Wang, X. Gao, Y. Zhao, X. Li, D. Dou, and C.-Z. Xu, "Pay attention to features, transfer learn faster cnns," in International Conference on Learning Representations, Conference Proceedings.

[46] D. Zhang, M. M. Islam, and G. Lu, "A review on automatic image annotation techniques," Pattern Recognition, vol. 45, no. 1, pp. 346–362, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0031320311002391

[47] J. Jeon, V. Lavrenko, and R. Manmatha, "Automatic image annotation and retrieval using cross-media relevance models," in Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval, ser. SIGIR '03. New York, NY, USA: Association for Computing Machinery, 2003, p. 119–126. [Online]. Available: https://doi.org/10.1145/860435.860459

[48] Y. Chen, X. Zeng, X. Chen, and W. Guo, "A survey on automatic image annotation," Applied Intelligence, vol. 50, no. 10, pp. 3412–3428, 2020.

[49] Y. Ma, Y. Liu, Q. Xie, and L. Li, "Cnn-feature based automatic image annotation method," Multimedia Tools and Applications, vol. 78, no. 3, pp. 3767–3780, 2019.

[50] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Conference Proceedings, pp. 839–847.

[51] K. Ji, W. Lei, and W. Zhang, "A deep retinex network for underwater low-light image enhancement," Machine Vision and Applications, vol. 34, no. 6, p. 122, 2023.

[52] H. Du, X. Hao, Y. Ye, L. He, and J. Guo, "A camera style-invariant learning and channel interaction enhancement fusion network for visible-infrared person re-identification," Machine Vision and Applications, vol. 34, no. 6, p. 117, 2023.

[53] N. Watt and M. C. du Plessis, "Neuro-augmented vision for evolutionary robotics," Machine Vision and Applications, vol. 34, no. 6, p. 95, 2023.

[54] J. Yao, B. Huang, S. Yang, X. Xiang, and Z. Lu, "Traffic sign detection and recognition under low illumination," Machine Vision and Applications, vol. 34, no. 5, p. 75, 2023.

[55] M. Coenen, T. Schack, D. Beyer, C. Heipke, and M. Haist, "Consinstancy: learning instance representations for semi-supervised panoptic segmentation of concrete aggregate particles," Machine Vision and Applications, vol. 33, no. 4, p. 57, 2022.

[56] C. B. Murthy, M. F. Hashmi, and A. G. Keskar, "Efficientlitedet: a real-time pedestrian and vehicle detection algorithm," Machine Vision and Applications, vol. 33, no. 3, p. 47, 2022.

[57] Z. Kandylakis, K. Vasili, and K. Karantzalos, "Fusing multimodal video data for detecting moving objects/targets in challenging indoor and outdoor scenes," Remote Sensing, vol. 11, no. 4, p. 446, 2019.

[58] Q. Liu, X. Lu, Z. He, C. Zhang, and W.-S. Chen, "Deep convolutional neural networks for thermal infrared object tracking," Knowledge-Based Systems, vol. 134, pp. 189–198, 2017.

[59] P. Jonsson, "Remote sensor for winter road surface status detection," in SENSORS, 2011 IEEE. IEEE, Conference Proceedings, pp. 1285–1288.

[60] J. Light, S. Parthasarathy, and W. McIver, "Monitoring winter ice conditions using thermal imaging cameras equipped with infrared microbolometer sensors," Procedia Computer Science, vol. 10, pp. 1158–1165, 2012.

[61] G. J. Fetzer, D. N. Sitter Jr, D. Gugler, W. L. Ryder, A. J. Griffis, D. Miller, A. Gelbart, and S. Bybee-Driscoll, "Ultraviolet, infrared, and near-infrared lidar system and method," 2010.

[62] I. Shopovska, L. Jovanov, and W. Philips, "Deep visible and thermal image fusion for enhanced pedestrian visibility," Sensors (Switzerland), vol. 19, no. 17, 2019, export Date: 23 October 2019.

[63] K. N. R. Chebrolu and P. N. Kumar, "Deep learning based pedestrian detection at all light conditions," in Proceedings of the 2019 IEEE International Conference on Communication and Signal Processing, ICCSP 2019, Conference Proceedings, pp. 838–842, export Date: 23 October 2019.

[64] E. Bercier, B. Louvat, O. Harant, E. Balit, J. Bouvattier, and L. Nacsa, "Far-infrared thermal camera: An effortless solution for improving adas detection robustness," in Proceedings of SPIE - The International Society for Optical Engineering, vol. 11009, Conference Proceedings, fIND PAPER.

[65] R. Yang, Y. Zhu, X. Wang, C. Li, and J. Tang, "Learning target-oriented dual attention for robust rgb-t tracking," in 2019 IEEE International Conference on Image Processing (ICIP). IEEE, Conference Proceedings, pp. 3975–3979.

[66] H. Li and X. Wu, "Densefuse: A fusion approach to infrared and visible images," IEEE Transactions on Image Processing, vol. 28, no. 5, pp. 2614–2623, 2019.

[67] P. Ghamisi, B. Rasti, N. Yokoya, Q. Wang, B. Hofle, L. Bruzzone, F. Bovolo, M. Chi, K. Anders, R. Gloaguen, P. M. Atkinson, and J. A. Benediktsson, "Multisource and multitemporal data fusion in remote sensing: A comprehensive review of the state of the art," IEEE Geoscience and Remote Sensing Magazine, vol. 7, no. 1, pp. 6–39, 2019.

[68] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "Mid-fusion: Octree-based object-level multi-instance dynamic slam," in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 5231–5237.

[69] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," Sensors, vol. 21, no. 6, p. 2140, 2021.

[70] S. Geraee, H. Mohammadbagherpoor, M. Shafiei, M. Valizadeh, F. Montazeri, and M. R. Feyzi, "Regenerative braking of electric vehicle using a modified direct torque control and adaptive control theory," Computers and Electrical Engineering, vol. 69, pp. 85–97, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S004579061733272X

[71] M. Ziadia, S. Kelouwani, A. Amammou, and K. Agbossou, "An adaptive regenerative braking strategy design based on naturalistic regeneration performance for intelligent vehicles," IEEE Access, 2023.

[72] Y. Yin, F. Kang, X. Yan, T. Li, and F. Duan, "A novel penta-jet vtol aerial robot: Modeling, control, and simulation verification," Computers and Electrical Engineering, vol. 111, p. 108906, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790623003300

[73] A. V. Malawade, T. Mortlock, and M. A. A. Faruque, "Ecofusion: Energy-aware adaptive sensor fusion for efficient autonomous vehicle perception," arXiv:2202.11330 [cs.CV], 2022, export Year:2019.

[74] A. T. Hamada and M. F. Orhan, "An overview of regenerative braking systems," Journal of Energy Storage, vol. 52, p. 105033, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X22010350

[75] K. Zeng, Q. Ma, J. Wu, S. Xiang, T. Shen, and L. Zhang, "Nlfftnet: A non-local feature fusion transformer network for multi-scale object detection," Neurocomputing, vol. 493, pp. 15–27, 2022.

[76] A. T. Hamada and M. F. Orhan, "An overview of regenerative braking systems," Journal of Energy Storage, vol. 52, p. 105033, 2022.

[77] M. Ziadia, S. Kelouwani, A. Amamou, Y. Dubé, and K. Agbossou, "Using an intelligent vision system for obstacle detection in winter condition," 2019.

[78] Y. Yang, Q. He, Y. Chen, and C. Fu, "Efficiency optimization and control strategy of regenerative braking system with dual motor," Energies, vol. 13, no. 3, p. 711, 2020.

[79] A. T. Hamada and M. F. Orhan, "An overview of regenerative braking systems," Journal of Energy Storage, vol. 52, p. 105033, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352152X22010350

[80] L. Serrao, S. Onori, and G. Rizzoni, "Ecms as a realization of pontryagin's minimum principle for hev control," in 2009 American control conference. IEEE, 2009, pp. 3964–3969.

[81] S. S. Bhurse and A. Bhole, "A review of regenerative braking in electric vehicles," in 2018 international conference on computation of power, energy, information and communication (ICCPEIC). IEEE, 2018, pp. 363–367.

[82] P. Bharati and A. Pramanik, "Deep learning techniques—r-cnn to mask r-cnn: a survey," Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019, pp. 657–668, 2020.

[83] R. Monroy, S. Lutz, T. Chalasani, and A. Smolic, "Salnet360: Saliency maps for omni-directional images with cnn," Signal Processing: Image Communication, vol. 69, pp. 26–34, 2018.

[84] X. Pingmei, A. E. Krista, Z. Yinda, F. Adam, R. K. Sanjeev, and X. Jianxiong, "Turkergaze: Crowdsourcing saliency with webcam based eye tracking," arXiv:1504.06755 [cs.CV], 2015, export Year:2019.

[85] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," IEEE Transactions on Pattern Analysis AND Machine Intelligence, no. 11, pp. 1254–1259, 1998.

[86] D. Pelleg and A. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in Proceedings of the Seventeenth International Conference on Machine Learning, Conference Proceedings, p. 727–734.

[87] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[88] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in European conference on computer vision. Springer, Conference Proceedings, pp. 21–37.

[89] J. Boisclair, S. Kelouwani, A. Amamou, M. Alam, L. Zeghmi, and K. Agbossou, "Image fusion by considering multimodal partial deep neural networks for self driving during winter," in 2021 IEEE Vehicle Power and Propulsion Conference (VPPC), 2021, pp. 1–6.

[90] J. Boisclair, A. Amamou, S. Kelouwani, M. Z. Alam, H. Oueslati, L. Zeghmi, and K. Agbossou, "A tree-based approach for visible and thermal sensor fusion in winter autonomous driving," Machine Vision and Applications, vol. 35, no. 4, pp. 1–17, 2024.

[91] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in Proceedings of the IEEE conference on computer vision and pattern recognition, Conference Proceedings, pp. 7263–7271.

[92] J. Terven and D. Cordova-Esparza, "A comprehensive review of yolo: From yolov1 to yolov8 and beyond," arXiv preprint arXiv:2304.00501, 2023.

[93] J. Q. G. Jocher, A. Chaurasia, "Yolo by ultralytics," 2023.

[94] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022.

[95] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Deep residual learning for image recognition," arXiv:1512.03385 [cs.CV], 2015, export Year:2019.

[96] K. Alex, "One weird trick for parallelizing convolutional neural networks," arXiv:1404.5997 [cs.NE], 2014, export Year:2019.

[97] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[98] B. Koonce and B. Koonce, "Efficientnet," Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization, pp. 109–123, 2021.

[99] X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," in 2017 2nd international conference on image, vision and computing (ICIVC). IEEE, 2017, pp. 783–787.

[100] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "Mnasnet: Platform-aware neural architecture search for mobile," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 2820–2828.

[101] S. Karen and Z. Andrew, "Very deep convolutional networks for large-scale image recognition," arXiv:1409.1556 [cs.CV], 2014, export Year:2019.

[102] R. Blything, V. Biscione, I. I. Vankov, C. J. Ludwig, and J. S. Bowers, "The human visual system and cnns can both support robust online translation tolerance following extreme displacements," Journal of vision, vol. 21, no. 2, pp. 9–9, 2021.

[103] K. Valentine, W. G. Temple, and K. M. Zhang, "Intelligent electric vehicle charging: Rethinking the valley-fill," Journal of Power Sources, vol. 196, no. 24, pp. 10 717–10 726, 2011.

[104] Z. Ma, "Decentralized valley-fill charging control of large-population plug-in electric vehicles," in 2012 24th Chinese Control and Decision Conference (CCDC). IEEE, 2012, pp. 821–826.

[105] N. Kaempchen and K. Dietmayer, "Data synchronization strategies for multi-sensor fusion," in Proceedings of the IEEE Conference on Intelligent Transportation Systems, vol. 85, Conference Proceedings, pp. 1–9.

[106] T. F. LLC, "Flir adas dataset," Online, 2019. [Online]. Available: https://www.flir.ca/oem/adas/

[107] H. Li and X.-J. Wu, "Densefuse: A fusion approach to infrared and visible images," IEEE Transactions on Image Processing, vol. 28, no. 5, pp. 2614–2623, 2018.

[108] L. Ren, Z. Pan, J. Cao, and J. Liao, "Infrared and visible image fusion based on variational auto-encoder and infrared feature compensation," Infrared Physics and Technology, vol. 117, p. 103839, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1350449521002115

[109] X. Zhao, W. Li, Y. Zhang, T. A. Gulliver, S. Chang, and Z. Feng, "A faster rcnn-based pedestrian detection system," 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), pp. 1–5, 2016.

[110] R. K. Chidambaram, D. Chatterjee, B. Barman, P. P. Das, D. Taler, J. Taler, and T. Sobota, "Effect of regenerative braking on battery life," Energies, vol. 16, no. 14, 2023. [Online]. Available: https://www.mdpi.com/1996-1073/16/14/5303

[111] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications," Ieee Access, vol. 9, pp. 82 031–82 057, 2021.

[112] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2117–2125.

[113] M. Lobanov and D. Sholomov, "On the acceleration of the convolutional neural network architecture based on resnet in the task of road scene objects recognition," ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, vol. 69, pp. 57–65, 2019.

[114] F. Wenfeng, Z. Xin, and Z. Guangpeng, "Resnetx: a more disordered and deeper network architecture," arXiv:1912.12165 [cs.CV], 2019, export Year:2019.

[115] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang, M.-H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," IEEE transactions on pattern analysis and machine intelligence, vol. 43, no. 2, pp. 652–662, 2019.

[116] J. Li, L. Tu, X. Gao, X. Li, Z. Zhong, and X. Feng, "Automatic classification of galaxy morphology based on the regnetx-cbam3 algorithm," Monthly Notices of the Royal Astronomical Society, vol. 517, no. 1, pp. 808–824, 2022.

[117] T. Mingxing and V. L. Quoc, "Efficientnet: Rethinking model scaling for convolutional neural networks," arXiv:1905.11946 [cs.LG], 2019, export Year:2019.

[118] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, "Repvgg: Making vgg-style convnets great again," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 13 733–13 742.

[119] B. Koonce and B. Koonce, "Mobilenetv3," Convolutional Neural Networks with Swift for Tensorflow: Image Recognition and Dataset Categorization, pp. 125–144, 2021.

[120] X. Yu, J. Wang, Y. Zhao, and Y. Gao, "Mix-vit: Mixing attentive vision transformer for ultra-fine-grained visual categorization," Pattern Recognition, vol. 135, p. 109131, 2023.

[121] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4. Springer, 2018, pp. 3–11.

[122] B. Baheti, S. Innani, S. Gajre, and S. Talbar, "Semantic scene segmentation in unstructured environment with modified deeplabv3+," Pattern Recognition Letters, vol. 138, pp. 223–229, 2020.

[123] W. Fang, L. Wang, and P. Ren, "Tinier-yolo: A real-time object detection method for constrained environments," IEEE Access, vol. 8, pp. 1935–1944, 2019.

[124] H. Yu and J. Oh, "Anchor distance for 3d multi-object distance estimation from 2d single shot," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 3405–3412, 2021.

[125] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. FeiFei, "Imagenet: A large-scale hierarchical image database," in IEEE conference on computer vision and pattern recognition, IEEE, Ed. IEEE, Conference Proceedings, p. 248–255.

[126] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," International Journal of Computer Vision, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[127] L. Tsung-Yi, M. Michael, B. Serge, B. Lubomir, G. Ross, H. James, P. Pietro, R. Deva, C. L. Zitnick, and D. Piotr, "Microsoft coco: Common objects in context," arXiv:1405.0312 [cs.CV], 2014, export Year:2019.

[128] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[129] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," International Journal of Robotics Research (IJRR), 2013.

[130] M. Pitropov, D. E. Garcia, J. Rebello, M. Smart, C. Wang, K. Czarnecki, and S. Waslander, "Canadian adverse driving conditions dataset," The International Journal of Robotics Research, vol. 40, no. 4-5, pp. 681–690, dec 2020. [Online]. Available: https://doi.org/10.1177%2F0278364920979368

[131] M. Sheeny, E. De Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang, and A. Wallace, "Radiate: A radar dataset for automotive perception," arXiv preprint arXiv:2010.09076, 2020.

[132] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, D. Ramanan, P. Carr, and J. Hays, "Argoverse 2: Next generation datasets for self-driving perception and forecasting," in Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021), 2021.

[133] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

[134] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, "Why does unsupervised pre-training help deep learning ?" in Proceedings of the thirteenth international

conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2010, pp. 201–208.

[135] Z. Zhang, X. Zhang, C. Peng, X. Xue, and J. Sun, "Exfuse: Enhancing feature fusion for semantic segmentation," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 269–284.