Contribution to the improvement of the robustness of perception systems based on multimodal neural networks

Robin Condat

Normandie Université

THÈSE

Pour obtenir le grade de Docteur de Normandie Université

Spécialité Informatique

l'École Doctorale Mathématiques, Information, Ingéniérie des Systèmes

Contribution to the improvement of the robustness of perception systems based on multimodal neural networks

Contribution à l'amélioration de la robustesse de systèmes de perception fondés sur des réseaux de neurones profonds multimodaux

Présentée et soutenue par

Robin CONDAT

Dirigée par Abdelaziz BENSRHAIR et encadrée par Alexandrina ROGOZAN

Thèse soutenue publiquement le 08/07/2022 devant le jury composé de

Mr Fabien MOUTARDE	Professeur, Ecole Nationale Supérieure des Mines de Paris	Rapporteur
Mr Olivier ORFILA	Chargé de Recherche HDR, Université Gustave Eiffel	Rapporteur
Mr Fabrice MERIAUDEAU	Professeur, Université de Bourgogne	Examinateur
Mr Fawzi NASHASHIBI	Directeur de recherche, INRIA Paris-Rocquencourt	Examinateur
Mme Samia AINOUZ	Professeure, INSA de Rouen Normandie	Examinatrice
Mr Abdelaziz BENSRHAIR	Professeur, INSA de Rouen Normandie	Directeur
Mme Alexandrina ROGOZAN	Maître de Conférence, INSA de Rouen Normandie	Encadrante

INSA ROUEN NORMANDIE

MI/IΣ

litis

Abstract

In order to guarantee an optimal perception for the autonomous vehicle, the analysis of road scenes has been widely explored in the last years in the field of ADAS. More recently, deep learning via neural networks has allowed to overcome a significant performance gap. However, the developed methods are rarely robust in degraded conditions, which can cause a bad interpretation of the road scene and lead to severe accidents. To remedy this, the use of multimodality, taking advantage of sensors from various sources, allows a better analysis of the situation. Nonetheless, this implies a possible cause of degradation: the malfunction or failure of one or more sensors.

In this thesis, we address the problem of robustness of multimodal neural networks for the analysis of road scenes in case of sensor malfunctioning. We study the impact of this damaged data on the performance of multimodal convolutional neural networks, and propose techniques to limit the performance losses.

First, we review the different proposals using multimodality for road scene analysis in the literature. The different neural networks for object detection, as well as the fusion strategies chosen to support multimodal data are also presented. A focus is made on strategies for improving the robustness of neural networks in degraded conditions.

Afterwards, we focus on the 2D detection of road traffic actors. We introduce several convolutional neural network architectures taking as input multimodal image-based data. Several fusion strategies are explored in order to extract and combine information from the modalities in the best possible way while reducing the size of the networks and thus ensuring real-time operation. An analysis of the impact of each input modality on our networks is performed to understand their role in the detection process.

Then, we study the problem of robustness of multimodal convolutional neural networks when one or more modalities are missing, i.e. replaced by a null modality containing no information. Several data augmentation techniques are presented to improve the performance of multimodal convolutional neural networks in degraded conditions with partial input data, while maintaining maximum accuracy in ideal conditions without malfunction. An in-depth analysis is also performed on a robust neural network to identify the consequences of one or more missing modalities on its detection process.

Finally, we address a more complex case of malfunction, where some of the input modalities are strongly noisy and therefore unusable. We introduce several methods to improve the robustness of neural networks under these conditions, in order to limit the perturbation of noisy modalities on their functioning. We show that our approaches can significantly mitigate the loss of accuracy of our networks in case of noisy modalities, but that a trade-off must be made between performance, robustness and processing time.

Résumé

Afin de garantir une perception optimale pour le véhicule autonome, l'analyse de scènes routières a largement été explorée durant ces dernières années dans le domaine de l'ADAS. Plus récemment, l'apprentissage profond via des réseaux de neurones a permis de passer un cap significatif en termes de performances. Cependant, les méthodes développées sont rarement robustes en conditions dégradées, pouvant ainsi causer une mauvaise interprétation de la scène routière et engendrer des accidents sévères. Pour remédier à cela, l'utilisation de la multimodalité, profitant de capteurs de sources diverses, permet une meilleure analyse de la situation. Cela implique, malgré tout, une cause possible de dégradation : le dysfonctionnement ou la panne d'un ou plusieurs capteurs.

Dans cette thèse, nous abordons la problématique de robustesse des réseaux de neurones multimodaux pour l'analyse de scènes routières en cas de dysfonctionnement de capteurs. Nous étudions l'impact de données endommagées sur le fonctionnement de réseaux de neurones convolutionnels multimodaux, et proposons des techniques permettant de limiter les pertes de performances engendrées.

Tout d'abord, nous passons en revue les différentes propositions utilisant la multimodalité pour l'analyse des scènes routières dans la littérature. Les différents réseaux de neurones pour la détection d'objets, ainsi que les stratégies de fusion choisies pour prendre en charge les données multimodales sont également présentés. Un accent est mis sur les stratégies permettant d'améliorer la robustesse des réseaux de neurones dans des conditions dégradées.

Dans la suite de la thèse, nous nous concentrons sur la détection 2D des usagers de la route. Nous introduisons plusieurs architectures de réseaux de neurones convolutionnels prenant en entrée des données multimodales basées sur l'image. Plusieurs stratégies de fusion sont explorées afin d'extraire et de combiner au mieux l'information des modalités tout en réduisant la taille des réseaux et ainsi assurer un fonctionnement en temps réel. Une analyse de l'impact de chaque modalité d'entrée sur nos réseaux est réalisée afin de comprendre leur rôle dans le processus de détection.

Ensuite, nous étudions la problématique de robustesse des réseaux de neurones convolutionnels multimodaux lorsqu'une ou plusieurs modalités sont manquantes, c'est-à-dire remplacée par une modalité nulle, ne contenant aucune information. Plusieurs techniques d'augmentation de données sont présentées pour l'amélioration des performances des réseaux de neurones convolutionnels multimodaux en conditions dégradées avec des données d'entrées partielles, tout en conservant une précision maximum en conditions idéales sans dysfonctionnement. Une analyse en profondeur est également effectuée sur un réseau de neurones robuste pour identifier les conséquences de l'absence d'une ou plusieurs modalités en entrée sur son processus de détection.

Enfin, nous abordons un cas plus complexe de dysfonctionnement, où une partie des modalités d'entrées sont fortement bruitées et donc inexploitables. Nous introduisons plusieurs méthodes pour améliorer la robustesse des réseaux de neurones dans ces conditions, afin de limiter la perturbation de ces modalités bruitées sur leur fonctionnement. Nous montrons que nos approches permettent d'atténuer significativement les pertes de précisions de nos réseaux face à des modalité bruitées, mais qu'un compromis est à définir entre performance, robustesse et temps d'exécution.

Remerciements

3 pages d'expression libre. La section que la majorité des gens vont lire uniquement dans ce manuscrit. Un discours solennel posé à l'écrit, là où rien n'était préparé durant la soutenance de thèse. Mesdames et messieurs, voici mes remerciements.

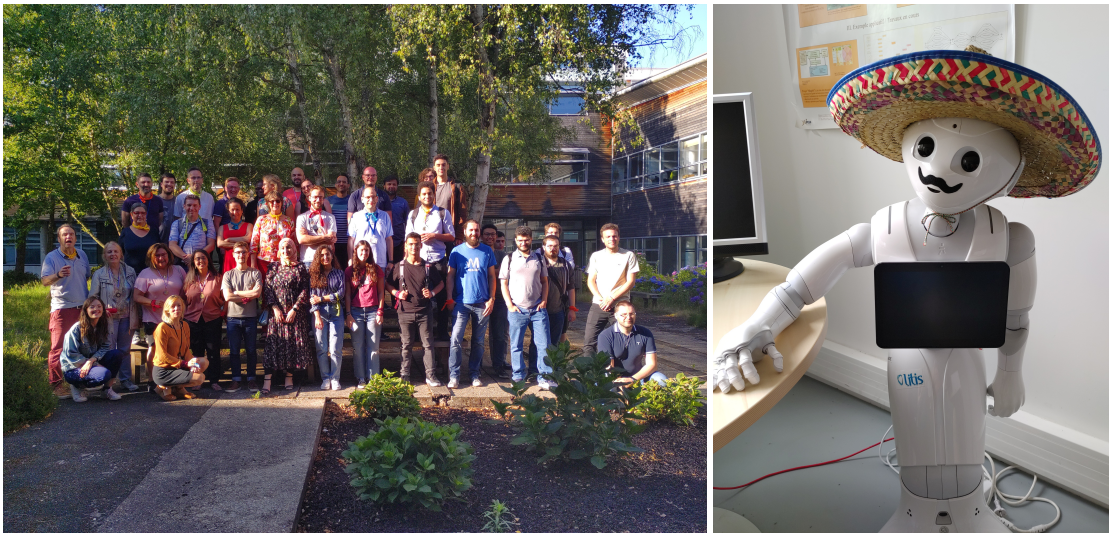
Pour commencer, je tiens à remercier les membres du jury pour leurs retours sur mes travaux des dernières années, à travers ce rapport et la soutenance. Cela m'a permis de me rendre compte notamment de la valeur de mes recherches et d'avoir plus confiance en moi dans mon futur académique.

Je remercie ensuite Aziz pour la direction de la thèse. Cela n'a pas été facile durant ces 4 années (je ne suis pas quelqu'un de facile, je le reconnais), mais on a tenu jusqu'au bout, et c'est en partie grâce à ta diplomatie. C'est durant mes études via tes cours que j'ai pu déceler une étincelle personnelle dans la filière de la vision par ordinateur (et non vision informatique, désolé), et une envie débordante d'aller voir Avatar 2. Je n'oublierai pas tes "C'est pas important, mais rappelle-moi vite", tes "Tout va comme tu veux ?" ou encore tes "On ne lâche rien camarade !".

Je remercie également Alexandrina pour son encadrement durant la thèse. Tu m'as permis de prendre confiance en mes expérimentations, et à ne pas me sentir tout penaud face "aux grandes universités mondiales". Malgré des désaccords, tu m'as soutenu coûte que coûte, et j'en suis très reconnaissant. J'espère te revoir prochainement.

Je tiens à remercier particulièrement Samia. Tout d'abord, pour avoir pris la suite de l'encadrement durant ces derniers mois, alors que tu n'avais vraiment pas le temps de caler ça dans ton agenda. Ensuite, pour la gestion de l'équipe STI également, qui reste selon moi, en toute objectivité, la meilleure équipe du labo, et de loin. Et enfin, pour m'avoir donné confiance en mes travaux et pour m'avoir aidé à traverser certains moments assez compliqués durant ces dernières années.

Ensuite, je voudrais remercier plus généralement tous mes collègues du LITIS. J'ai passé 4 merveilleuses années (bon plutôt 2 ans 1/2 à cause d'un certain virus), et j'ai adoré travailler parmi vous. Brigitte, Sandra, je vous remercie pour votre aide et pour votre amabilité alors que je m'y prenais toujours au dernier moment, voire beaucoup trop tardivement. La team pause café post-déjeuner, avec de nombreuses conversations passionnantes plus ou moins scientifiques. Les professeurs avec qui j'ai pu effectuer mes missions enseignement, toujours de bon conseil. Un énorme merci à tous ceux surnommés "les jeunes" au labo (doctorants, post-doc, stagiaires, etc.), pour ces nombreux moments de vie passés en grande partie dans les locaux, mais également au laser game (Imane, je veux ma revanche), à l'Antre du Malt ou au sous-sol du Makao à manger un gâteau licorne vraiment pas terrible. Merci d'avoir toléré mon exubérance à travers mes propositions débiles, qui se sont parfois concrétisées (Figure 1).



(a) (b)

Figure 1 – 2 projets utiles et concrets réalisés durant la thèse : a) le jeu de piste du Pr. Séraphin BULEAU et b) le projet Pepper-Sombrero-Moustache

Merci également au personnel de l'INSA dans sa globalité. Après 5 ans d'étude et 4 ans de thèse, il est temps que je parte. Je garderai d'excellents souvenirs de ces 9 années qui m'ont forgé en tant que scientifique, mais aussi en tant qu'humain.

Éloignons-nous du laboratoire maintenant. Je remercie ensuite tous mes potes musiciens de l'AMIR. Cela a été un plaisir énorme de partager autant de moments avec vous, sur scène, en répétition, mais également durant de nombreuses soirées à expérimenter le Captain Charrette. Spéciale dédicace à Pink Fluffy et aux nombreux kebabs post répétitions. Merci à tous les animateurs et animatrices de colo avec qui j'ai pu partir lors des 4 derniers étés, pour m'avoir permis de décompresser mentalement et ses nombreux 5èmes avant tout pédagogiques et bienveillants.

Rentrons dans la sphère privée. Je remercie évidemment tous mes amis, qu'on se soit vus de près ou de loin, et qui m'ont aidé à décrocher de mes travaux. Merci Gaétan pour toutes nos performances musicales et pour ces nombreuses discussions Deep Learning ultra techniques, même si je reste persuadé que ton travail se rapproche plus de la magie que de la science. Merci Manon, pour nos débats essentiels sur des passions peu communes : le biathlon et Fort Boyard. Merci Sylvain de m'avoir supporté en colocation durant ces 4 dernières années, de m'avoir remonté le moral lorsque ça n'allait vraiment pas et pour tes prédictions exactes sur le gagnant de Top Chef (à savoir le candidat que tu détestes dès la 1ère émission). Merci Matthieu pour ta franchise, ta capacité à me canaliser lorsque je vais un peu trop loin, mais aussi pour ton énergie créative, nous permettant de concrétiser beaucoup d'idées douteuses sur le papier [1] [2]. Un énorme merci également à ma famille, et en particulier à papa, maman, Léo et Matéo. Même si vous n'étiez pas totalement emballés par le projet au départ, vous m'avez considérablement soutenu durant ces 4 années.

1. https://youtu.be/z51NmYJasqY
2. https://youtu.be/s3CrM8tlW9c

Pour finir, quelques remerciements pêle-mêle sans ordre d'importance, mais qui au final ont contribué à l'élaboration de mes travaux : les nombreux podcasteurs qui m'ont accompagné durant les longues sessions de code (Floodcast, 2 heures de perdues, Un bon moment, 4 comiques dans le vent), la personne qui a inventé le houmous, l'équipe de Trashtalk qui m'a évité de nombreuses nuits blanches tout en suivant la NBA, les concepteurs de deepl.com, les gens au labo qui se dévouent pour faire le café, l'Orient Express pour les meilleurs kebabs rouennais après les répétitions intenses, les protagonistes du teaser EUREKA, les probables personnes que j'ai oublié de mentionner dans ce long pavé (je m'en excuse d'avance), et toi qui va peut-être lire le reste du manuscrit et qui donnera du sens à mon travail réalisé durant ce long périple tumultueux, appelé plus communément "une thèse".

Bravo aux remerciements.

Contents

List of Figures

List of Tables

List of Acronyms

AP Average Precision.
mAP Mean Average Precision.

ADAS Advanced Driver-Assistance Systems.
AMP Automatic Mixed Precision.

BL Blue Channel from RGB image.
BLUR Gaussian Blur algorithm.

CNN Convolutional Neural Network.
CST Constant value algorithm.

DLP Dead Leaves Pattern algorithm.
DNN Deep Neural Network.
DOL Depth from stereovision, Optical flow and LIDAR..
DP Depth Map Modality from Stereo vision.

FPN Feature Pyramid Network.

GA-Net Guided Aggregation Net.
GFU Gated Fusion Unit.
GPS Global Positioning System.
GPU Graphics Processing Unit.
GR Green Channel from RGB image.

IMU Inertial Measurement Unit.
IoU Intersection over Union.

LD Depth Map Modality from LiDAR.
LRGD Local Random Gaussian Distribution algorithm.

ModAM Modality Activator Model.

NAR Noise Augmentation Rate.

NLSPN Non Local Spatial Propagation Network.
NMS Non-Maximum Suppresion.

OF Optical Flow Modality.

RCC Random Channel Cut.
RD Red Channel from RGB image.
RGB Color image (stands for Red-Green-Blue).
RGD Random Gaussian Distribution algorithm.
RGPN Random Gaussian Pixel Noise algorithm.
RMC Random Modality Cut.
RoI Region of Interest.
RPN Region Proposal Network.
RSC Random Signal Cut.

SFU Stack Fusion Unit.
SGBM Semi-Global Block Matching.
SHUF Shuffle algorithm.
SSD Single Shot Multibox Detector.
SVM Support Vector Machine.

VCN Volumetric Correspondence Networks.

YOLO You Only Look Once.

Introduction

Context

The development of driver assistance systems (DAS) began in the 1950s with the introduction of anti-lock braking systems into the vehicle market. Subsequently, these different driving aids have diversified to always offer safer driving and thus limit the number of road accidents caused mainly by human error. Among the precursor DAS are electronic stability control, blind spot information systems, lane departure warning, adaptive cruise control and traction control. All of these aids have evolved to understand and communicate with the surrounding environment, marking the development of Advanced Driver Assistance Systems (ADAS). This required the use of more advanced sensors, such as cameras, radar or LiDAR. In order to make the best use of these data, one of the challenges in the development of ADAS is the understanding of road scenes. From data in different forms, the objective is to analyze the surrounding environment, so that the systems decide the best action to take, then either inform the driver or take control of the vehicle. In the challenge of road scene understanding, there are many different tasks to master such as road traffic actors detection and tracking, road scenes semantic segmentation, road and lanes estimation or disparity estimation among others. These tasks, mostly computer vision challenges, have seen great improvements during the last decade with the emergence of deep learning based on neural networks, allowing to obtain high performances on many benchmarks on this subject.

However, the vast majority of the research work carried out in this field focuses on operation under normal conditions, i.e. in a classical environment with a fully operational sensor system. On the other hand, the robustness of these perception systems is much less studied in degraded conditions. This observation is explained by the challenge of road scene understanding, which is already complex in normal conditions. The state-of-the-art perception systems obtain excellent performances on different benchmarks on the subject, but a step is still to be taken for a real time operation in real-world environments. In the same perspective, a large part of the work centered on the robustness of perception systems is focused on road scene understanding in adverse weather conditions, which is equivalent to a complex environment but still with a fully operational sensor system. On the other hand, the robustness of these perception systems in case of sensor malfunctions is a topic that is not sufficiently studied. This is all the more the case with deep neural networks, which have become an essential tool in the field of computer vision thanks to their performance. The developed methods are therefore rarely robust in case of sensor malfunctions, which can be caused by an incident inherent to the sensor itself, but also by an external event: hilly road, sudden braking, an object altering a sensor

or severe weather. All of this can cause a misinterpretation of the road scene and thus lead to severe accidents. To avoid these risks, the robustness of perception systems is a key issue for road scene understanding.

In this thesis, we address the problem of robustness of multimodal neural networks for road scene analysis in case of sensor malfunction. We focus on the malfunction of one or more sensors, generating partial multimodal data. We study the impact of damaged data on the performance of multimodal convolutional neural networks, and propose techniques to limit the performance losses.

Motivations

From this problem, we have identified two axes of progress, which have subsequently motivated our work: the proposal of multimodal neural networks, in particular for object detection, as well as the improvement of their robustness in degraded conditions in case of sensor malfunction.

Multimodal neural networks for 2D object detection

Object detection is a widely studied research topic in the literature, especially with the emergence of convolutional neural networks. Many architectures have been proposed for 2D object detection from a single RGB image. In the ADAS context, limiting oneself to the use of a single color camera for the analysis of road scenes is inconceivable, since the slightest failure of this sensor has major consequences on the understanding of the situation. Therefore, multimodality appeared as one of the safest solutions to not only ensure robustness of detection, but also to better understand complex situations, especially in adverse weather conditions.

However, the handling of multiple modalities coming from different sensors inevitably leads to a higher computation time. It is therefore necessary to select the best modalities to be processed, in their best form, in order to exploit the maximum amount of information useful for the good analysis of road scenes. Improvements are possible on the design of the architecture of these networks, in order to be able to take in charge a maximum of various modalities, while keeping a correct execution speed for a real time operation. For this, the strategy used for data fusion is crucial. Several fusion pipelines (early, intermediate or late), as well as multiple fusion methods (addition, concatenation, ensemble fusion, specific fusion) have already been used in the literature. All these proposals have advantages as well as drawbacks, on the performance of perception systems and on their execution speed. In the context of robustness, it is also necessary to be able to support several modalities, whereas the majority of the works carried out on this subject have only two modalities to fuse. This additional constraint implies a reflection on the design of neural network architectures, in order to bring more scalability in data fusion.

Robustness improvement of multimodal neural networks

Despite its necessity in the context of ADAS, the robustness of multimodal neural networks in the event of sensor malfunction is little studied. By widening the spectrum of possible applications, we find strategies opting for new neural network architectures. This can range from a simple module to be implemented in an existing architecture, to complete new models designed for robustness. For the understanding of road scenes, there are already many multimodal neural networks performing well under normal conditions. From this observation, the best strategy is to adapt these existing networks to make them robust to failure. This potentially requires changes in their architecture or training process. It is also important to ensure that these modifications do not generate offsets in the performance of the network under normal conditions, as well as in its execution speed.

Whatever the robustness improvements made, a sensor malfunction will have consequences in the detection process of the perception systems. Depending on the type of malfunction and the nature of the affected sensor, these incidents alter the detection process in different ways. It is therefore important to identify the nature of the consequences of these different malfunctions. Among the possible degraded conditions, some will be much more critical than others, making the perception systems vulnerable, and it is essential to identify these critical conditions, to then focus the research of improvements on these cases.

Outline

Since the robustness of perception systems is a major issue in the context of ADAS, we propose in this thesis to study the robustness of multimodal neural networks for road scene understanding, and to contribute to their improvement. The thesis is structured in four chapters, whose contents are summarized below, and a conclusion.

Chapter 1 : Literature review

In this first chapter, we review the different uses of multimodality for road scene understanding. We start by listing the usual image-based modalities used in the ADAS context, as well as their advantages and drawbacks. The available multimodal datasets, essential for the development of multimodal perception systems, are also presented. In a second step, we focus on multimodal solutions based on deep learning for computer vision tasks. Convolutional neural networks for 2D object detection are first presented. Then, we discuss different data fusion strategies within a neural network for computer vision tasks. The three fusion pipelines, namely early fusion, middle fusion and late fusion, are presented, along with their respective advantages and drawbacks on neural network performance and speed. The fusion methods to best combine the information of the input modalities are also listed. Finally, we introduce strategies to improve the robustness of multimodal neural networks under degraded conditions. We have classified the proposed solutions in 3 main categories: multimodal datasets focusing on robustness via the proposal of partially degraded data, specific neural network architectures for improving their robustness, and the generation of degraded data by training neural networks.

Chapter 2 : Multimodal convolutional neural networks for 2D object detection

In this chapter, we propose new architectures of multimodal convolutional neural networks for 2D object detection. In order to improve their robustness, we try to make them take as many modalities as possible as input. However, the accumulation of modalities implies a heavier architecture as well as a higher processing time. The designs of our neural network architectures have therefore been primarily thought to answer this problem, while limiting their complexity, in order to find an acceptable compromise between performance and processing time. We first propose a scalable architecture of convolutional neural networks using middle fusion to take as input any number of modalities image-based as desired , without considerably increasing the number of parameters. We show the limitations of this strategy, from a neural network performance point of view, as well as on the impact of each modality on the neural network decisions. We then introduce 2 architectures of multimodal convolutional neural networks, using early and middle fusion, for 2D object detection. Evaluations of our 2 networks show the contribution of multimodality on their performances despite a lower execution speed. Experiments on the robustness of our contributions are carried out in order to reveal the main flaws of the latter in degraded conditions. Finally, we compare our contributions with state-of-the-art perception systems for 2D detection of road traffic actors, and highlight the possible improvements.

Chapter 3 : Robustness improvement of multimodal neural networks with missing modalities

In this chapter, we address the problem of robustness of multimodal neural networks when one or more input modalities are missing. The experiments performed in the previous chapter have shown the vulnerability of our neural networks in several degraded configurations. To remedy this, we propose 3 data augmentation techniques, aiming at generating, during neural network training, partial multimodal data where one or more modalities are disabled and therefore null. These three techniques can easily be applied on a multimodal neural network, whatever the input modalities and does not require any modification of its architecture. We show through various experiments on multimodal convolutional neural networks that our approaches clearly improve their robustness in degraded conditions, but that a decrease in performance in normal conditions without failure can sometimes be observed. A good parameterization of our techniques is thus necessary to limit these losses, while maximizing the performances in degraded conditions. We also observe different behaviors depending on the missing modalities, and thus analyze the impact of our modalities in the detection process of our multimodal networks. Finally, we evaluate a robust multimodal convolutional neural network in various degraded conditions with complementary metrics in order to better understand the changes in its detection process, and thus the nature of the performance losses observed.

Chapter 4 : Robustness improvement of multimodal neural networks with noisy modalities

In this last chapter, we address a more complex issue, namely the robustness of multimodal neural networks in case of strongly noisy input modalities. Unlike a missing modality, a strongly noisy modality is unusable for a neural network and contains false information which can disturb it, and thus cause strong performance losses. We first propose a data augmentation technique, aiming at generating data with randomly noisy modalities during the training of the neural network. Through our experiments on several multimodal convolutional neural networks, we show the vulnerability of the latter to different noises, and the significant contribution of our method on their robustness, whatever the modalities concerned and the type of noise applied. We highlight the limits of this technique, which can cause reduced performances in normal conditions, limited improvements against some noises, as well as its inability to make our perception systems robust to all possible types of noise. We introduce in a second step a light preprocessing neural network for the identification and deactivation of unusable modalities. The objective with this contribution is to simplify these disturbing noisy modalities by removing the false information they contain, so that they become null, which is easier to handle for our networks. Our experiments show that this preprocessing network, combined with our data augmentation technique, significantly improves the performance of our neural networks under normal conditions, as well as under most degraded conditions. We also highlight the vulnerabilities caused by this strategy, when faced with unknown noises not learned in training.

Publications

This thesis work has resulted in the following publications:

— Robin Condat, Alexandrina Rogozan, Abdelaziz Bensrhair. *"Random Signal Cut for Improving CNN Robustness of 2D Road Object Detection"*, **European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning**, 2020.

— Robin Condat, Alexandrina Rogozan, Abdelaziz Bensrhair. *"GFD-Retina: Gated Fusion Double RetinaNet for Multimodal 2D Road Object Detection"*, **The 23rd IEEE International Conference on Intelligent Transportation Systems (ITSC)**, 2020.

— Robin Condat, Alexandrina Rogozan, Samia Ainouz, Abdelaziz Bensrhair. *"Identifying and Deactivating Unusable Modalities to Improve Multimodal CNN Robustness for Road Scene Understanding"*, submitted to **The 25th IEEE International Conference on Intelligent Transportation Systems (ITSC)**, 2022.

— Robin Condat, Alexandrina Rogozan, Samia Ainouz, Abdelaziz Bensrhair. *"Amélioration de la robustesse des réseaux de neurones multimodaux par identification et désactivation des modalités endommagées"*, submitted to **GRETSI, XXV IIIème Colloque francophone de traitement du signal et des images**, 2022.

Introduction

Contexte

Le début du développement des systèmes d'aides à la conduite (DAS) remonte aux années 1950 avec l'apparition des systèmes de freinage antiblocage dans le marché du véhicule. Par la suite, ces différentes aides à la conduite se sont diversifiées pour toujours proposer une conduite plus sécurisée et ainsi limiter le nombre d'accidents de la route causés en majorité par des erreurs humaines. Parmi les DAS précurseurs, on retrouve le contrôle électronique de la stabilité, les systèmes d'information sur les angles morts, l'alerte de franchissement de ligne, le régulateur de vitesse adaptatif et le contrôle de la traction. Toutes ces aides ont évolué afin de comprendre le milieu environnant et de communiquer avec, marquant ainsi le développement des systèmes avancés d'aide à la conduite (ADAS). Cela a nécessité l'utilisation de capteurs plus avancés, tels que des caméras, radars ou des LiDAR. Afin d'exploiter au mieux ces données, un des défis du développement des ADAS est la compréhension de scènes routières. A partir de données sous différentes formes, l'objectif est d'analyser le milieu environnant, afin que les systèmes décident de la meilleure action à prendre, pour ensuite informer le conducteur ou prendre le contrôle du véhicule. Dans le défi de la compréhension des scènes routières, il y a beaucoup de tâches différentes à maîtriser telles que la détection et le suivi des usagers de la route, la segmentation sémantique des scènes routières, l'estimation des routes et des voies ou l'estimation de la disparité, entre autres. Ces tâches, en majorité des défis de vision informatique, ont connu de grandes améliorations durant la dernière décennie avec l'émergence de l'apprentissage profond basé sur des réseaux de neurones, permettant d'obtenir des performances élevées sur de nombreux benchmarks sur le sujet.

Cependant, la grande majorité des travaux de recherche effectués dans ce domaine se concentre sur le fonctionnement en conditions normales, c'est-à-dire dans un environnement classique et avec un système de perception totalement opérationnel. En revanche, la robustesse de ces systèmes de perception est nettement moins étudiée en conditions dégradées. Ce constat s'explique par la difficulté de la compréhension des scènes routières, qui est déjà complexe en conditions normales. Les systèmes de perception de l'état de l'art obtiennent d'excellentes performances sur différents benchmarks sur le sujet, mais un cap est encore à franchir pour un fonctionnement en environnement réel et en temps réel. Dans la même perspective, une grosse partie des travaux axés sur la robustesse des systèmes de perception se concentre sur la compréhension de scènes routières en conditions météorologiques adverses, ce qui équivaut à un environnement complexe mais toujours avec un système de perception entièrement opérationnel. En revanche, la robustesse de ces systèmes de perception en cas de dysfonctionnements de

capteurs est un sujet trop peu étudié. C'est d'autant plus le cas avec les réseaux de neurones profonds, devenus ces dernières années un outil incontournable dans le domaine de la vision informatique grâce à ses performances. Les méthodes développées sont donc rarement robustes en cas de dysfonctionnements de capteurs, pouvant être causés par un incident inhérent au capteur lui-même, mais également par un événement extérieur : une route vallonnée, une météo sévère, un freinage brusque ou un objet altérant un capteur. Tout cela peut provoquer une mauvaise interprétation de la scène routière et ainsi engendrer des accidents sévères. Afin d'éviter ces risques, la robustesse des systèmes de perception est un enjeu clé pour la compréhension des scènes routières.

Dans cette thèse, nous abordons la problématique de robustesse des réseaux de neurones multimodaux pour l'analyse de scènes routières en cas de dysfonctionnement de capteurs. Nous nous concentrons sur le dysfonctionnement d'un ou plusieurs capteurs, générant ainsi des données multimodales partielles. Nous étudions l'impact de données endommagées sur le fonctionnement de réseaux de neurones convolutionnels multimodaux, et proposons des techniques permettant de limiter les pertes de performances engendrées.

Motivations

De cette problématique, nous avons identifié deux axes de progressions, qui ont par la suite motivé nos travaux : la proposition de réseaux de neurones multimodaux, en particulier pour la détection d'objets, ainsi que l'amélioration de leur robustesse en conditions dégradées en cas de dysfonctionnement de capteur.

Réseaux de neurones multimodaux pour la détection d'objets 2D

La détection d'objet est un sujet de recherche largement étudié dans la littérature, qui plus est avec l'apparition des réseaux de neurones convolutionnels. De nombreuses architectures ont été proposées pour de la détection d'objet 2D à partir d'une seule image RGB. Dans le contexte ADAS, se limiter à l'utilisation d'une seule caméra couleur pour l'analyse de scènes routières est inconcevable, puisque la moindre défaillance de ce capteur a des conséquences majeures sur la compréhension de la situation. De ce fait, la multimodalité est apparue comme l'une des solutions les plus sûres pour non seulement assurer une robustesse de détection, mais également pour mieux appréhender des situations complexes, notamment en conditions météorologiques adverses.

Toutefois, la prise en charge de multiples modalités provenant de différents capteurs entraîne inévitablement un temps de calcul plus conséquent. Il est donc nécessaire de sélectionner au mieux les modalités à traiter, sous leur meilleure forme, afin d'exploiter le maximum d'information utile à la bonne analyse de la scène routière. Des améliorations sont possibles sur le design de l'architecture de ces réseaux, afin de pouvoir prendre en charge un maximum de modalités diverses, tout en conservant une vitesse d'exécution correcte pour un fonctionnement en temps réel. Pour cela, la stratégie employée pour la fusion de données est déterminante. Plusieurs pipelines de fusion (précoce, intermédiaire ou tardive), ainsi que de multiples méthodes de fusion (addition, concaténation, fusion d'ensembles, fusion spécifique) ont déjà été utilisées dans la littérature. Toutes

ces propositions présentent des avantages ainsi que des inconvénients, sur les performances des systèmes de perception et sur leur vitesse d'exécution. Dans le contexte de la robustesse, il faut également pouvoir prendre en charge plusieurs modalités, là où la majorité des travaux réalisés sur ce sujet n'ont que deux modalités à fusionner. Cette contrainte supplémentaire implique une réflexion sur la conception des architectures de réseaux de neurones, afin d'apporter une meilleure modularité dans la fusion de données.

Amélioration de la robustesse des réseaux de neurones multimodaux

Malgré sa nécessité dans le contexte ADAS, la robustesse des réseaux de neurones multimodaux en cas de dysfonctionnement de capteurs est peu étudiée. En élargissant le spectre des applications possibles, on trouve des stratégies optant pour de nouvelles architectures de réseaux de neurones. Cela peut aller d'un simple module à implanter dans une architecture existante, jusqu'à des nouveaux modèles complets conçus pour la robustesse. Pour la compréhension de scènes routières, il existe déjà de nombreux réseaux de neurones multimodaux performants en conditions normales. De ce constat, la meilleure stratégie est d'adapter ces réseaux existants pour les rendre robustes à la panne. Cela nécessite potentiellement des changements dans leur architecture ou dans leur processus d'entraînement. Il faut également veiller à ce que ces modifications apportées ne génèrent pas de contreparties dans les performances en conditions normales du réseau, ainsi que dans sa vitesse d'exécution.

Quelques soient les améliorations de robustesse apportées, un dysfonctionnement de capteur aura des conséquences dans le processus de détection des systèmes de perception. En fonction du type de dysfonctionnement, de la nature du capteur impacté, ces incidents altèrent le processus de détection différement. Il est ainsi important d'identifier la nature des conséquences de ces différents dysfonctionnements. Parmi les conditions dégradées possibles, certaines seront beaucoup plus critiques que d'autres, rendant les systèmes de perception vulnérables, et il est essentiel d'identifier ces conditions critiques, pour ensuite axer les recherches d'améliorations sur ces cas.

Structure de la thèse

Puisque la robustesse des systèmes de perception est un enjeu majeur dans le contexte ADAS, nous proposons dans cette thèse d'étudier la robustesse des réseaux de neurones multimodaux pour la compréhension des scènes routières, et de contribuer à leur amélioration. La thèse est structurée en quatre chapitres, dont les contenus sont résumés ci-dessous, et une conclusion.

Chapitre 1 : Revue de l'état de l'art

Dans ce premier chapitre, nous étudions les différentes utilisations de la multimodalité pour la compréhension des scènes routières. Nous commençons par lister les modalités usuelles basées sur l'image utilisées dans le contexte ADAS, ainsi que leurs avantages et inconvénients. Les jeux de données multimodales disponibles, essentiels pour le développement des systèmes de perception multimodaux, sont également présentés. Dans un second temps, nous nous penchons sur les solutions multimodales basées sur l'apprentissage profond pour la vision informatique. Les réseaux de neurones convolutionnels pour la détection d'objets 2D sont d'abord présentés. Ensuite, nous abordons les différentes stratégies de fusion de données au sein d'un réseau de neurones pour des tâches de vision informatique. Les trois schémas de fusion, à savoir la fusion précoce, la fusion intermédiaire et la fusion tardive, sont présentés, ainsi que leurs avantages et inconvénients respectifs sur les performances et rapidité d'exécution des réseaux de neurones. Les méthodes de fusion pour combiner au mieux les informations des modalités d'entrées sont également listées. Enfin, nous introduisons les stratégies d'amélioration de la robustesse de réseaux de neurones multimodaux en conditions dégradées. Nous avons classé les solutions proposées dans 3 catégories principales : les jeux de données multimodales axés sur la robustesse via la proposition de données partiellement dégradées, les architectures de réseaux de neurones spécifiques pour l'amélioration de leur robustesse, et la génération de données dégradées en entraînement des réseaux de neurones.

Chapitre 2 : Réseaux de neurones convolutionnels multimodaux pour la détection d'objets 2D

Dans ce chapitre, nous proposons de nouvelles architectures de réseaux de neurones convolutionnels multimodaux pour la détection d'objets 2D. Dans l'objectif d'améliorer à posteriori leur robustesse, nous cherchons à ce que ces derniers prennent en entrée un maximum de modalités. Cependant, l'accumulation de modalités implique une architecture plus lourde ainsi qu'un temps de calcul plus conséquent. Les designs des architectures de nos réseaux de neurones ont donc été pensé en premier lieu pour répondre à cette problématique, tout en limitant leur complexité, afin de trouver un compromis acceptable entre performance et temps de calcul. Nous proposons dans un premier temps une architecture modulable de réseaux de neurones convolutionnels utilisant la fusion intermédiaire pour prendre en entrée n'importe quel nombre de modalités basés sur l'image voulu, sans augmenter considérablement le nombre de paramètres. Nous montrons les limites de cette stratégie, d'un point de vue des performances générales des réseaux de neurones, ainsi que sur l'impact de chaque modalité sur les décisions du réseau de neurones. Nous introduisons ensuite 2 architectures de réseaux de neurones multimodaux convolutionnels, utilisant les fusions précoce et intermédiaire pour la détection 2D d'objets. Les évaluations de nos 2 réseaux montrent l'apport de la multimodalité sur leurs performances malgré une vitesse d'exécution plus réduite. Des expérimentations sur la robustesse de nos contributions sont effectuées afin de révéler des failles principales en conditions dégradées. Nous comparons pour finir nos contributions avec les systèmes de perception de l'état de l'art pour la détection 2D des usagers de la route, et mettons en exergue les améliorations possibles à apporter.

Chapitre 3 : Amélioration de la robustesse des réseaux de neurones multimodaux face à des modalités manquantes

Dans ce chapitre, nous adressons la problématique de robustesse des réseaux de neurones multimodaux lorsqu'une ou plusieurs modalités d'entrées sont manquantes. Les expérimentations réalisées dans le chapitre précédent ont pu montrer la vulnérabilité de nos réseaux de neurones dans plusieurs configurations dégradées. Pour remédier à cela, nous proposons 3 techniques d'augmentation de données, visant à générer durant l'entraînement du réseau de neurones des données multimodales partielles où une ou plusieurs modalités d'entre elles sont désactivées et donc nulles. Ces trois techniques peuvent facilement être appliquées sur un réseau de neurone multimodal, quelles que soient les modalités d'entrée et ne nécessite aucune modification de son architecture. Nous montrons à travers différentes expérimentations sur des réseaux de neurones convolutionnels multimodaux que nos approches améliorent nettement la robustesse de ces derniers en conditions dégradées, mais qu'une baisse de performances en conditions normales sans panne peut parfois être constatée. Un bon paramétrage de nos techniques est donc nécessaire pour limiter ces pertes, tout en maximisant les performances en conditions dégradées. Nous observons également des comportements différents en fonction des modalités manquantes, et analysons ainsi l'impact de nos modalités dans le processus de détection de nos réseaux multimodaux. Nous évaluons ensuite un réseau de neurones convolutionnel multimodal robuste dans diverses conditions dégradées avec des métriques complémentaires afin de mieux comprendre les changements occasionnés sur son processus de détection, et ainsi la nature des pertes de performances constatées.

Chapitre 4 : Amélioration de la robustesse des réseaux de neurones multimodaux face à des modalités bruitées

Dans ce dernier chapitre, nous abordons un cas plus complexe, à savoir la robustesse des réseaux de neurones multimodaux en cas de modalités fortement bruitées en entrée. Contrairement à une modalité manquante, une modalité fortement bruitée est inutilisable pour le réseau de neurones et contient de la fausse information pouvant le perturber, et ainsi provoquer de fortes pertes de performances. Nous proposons dans un premier temps une technique d'augmentation de données, visant à générer aléatoirement des données avec des bruitées modalités fortement durant l'entraînement du réseau de neurones. A travers nos expérimentations sur plusieurs réseaux de neurones convolutionnels multimodaux, nous montrons la vulnérabilité de ces derniers face à différents bruits, et l'apport significatif de notre contribution sur leur robustesse, quelles que soient les modalités concernées et le type de bruit appliqué. Nous mettons en exergue les limites de cette technique, pouvant causer des performances réduites en conditions normales, des améliorations limitées face à certains bruits, ainsi que son incapacité à elle-seule à pouvoir rendre nos systèmes de perception robustes face à la diversité des bruits. Nous introduisons dans un second temps un réseau de neurones de prétraitement léger pour l'identification et la désactivation de ces modalités inutilisables. L'objectif avec cette contribution est de simplifier ces modalités bruitées perturbantes en enlevant la fausse information qu'elles contiennent, pour revenir sur une modalité nulle, plus facile à prendre

en charge pour nos réseaux. Nos expérimentations montrent que ce réseau de prétraitement, combiné avec notre technique d'augmentation de données, améliore significativement les performances en conditions normales de nos réseaux de neurones, ainsi que dans la plupart des conditions dégradées. Nous mettons également en évidence les vulnérabilités causées par cette stratégie, face à des bruits inconnus non appris en entraînement.

Publications

Ce travail de thèse a donné lieu aux publications suivantes:

— Robin Condat, Alexandrina Rogozan, Abdelaziz Bensrhair. *"Random Signal Cut for Improving CNN Robustness of 2D Road Object Detection"*, **European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning**, 2020.

— Robin Condat, Alexandrina Rogozan, Abdelaziz Bensrhair. *"GFD-Retina: Gated Fusion Double RetinaNet for Multimodal 2D Road Object Detection"*, **The 23rd IEEE International Conference on Intelligent Transportation Systems (ITSC)**, 2020.

— Robin Condat, Alexandrina Rogozan, Samia Ainouz, Abdelaziz Bensrhair. *"Identifying and Deactivating Unusable Modalities to Improve Multimodal CNN Robustness for Road Scene Understanding"*, soumis à **The 25th IEEE International Conference on Intelligent Transportation Systems (ITSC)**, 2022.

— Robin Condat, Alexandrina Rogozan, Samia Ainouz, Abdelaziz Bensrhair. *"Amélioration de la robustesse des réseaux de neurones multimodaux par identification et désactivation des modalités endommagées"*, soumis à **GRETSI, XXV IIIème Colloque francophone de traitement du signal et des images**, 2022.

Chapter 1

Literature Review

Contents

1.1 Introduction

Road scene analysis is a discipline where many improvements are still required. In spite of many works on the subject, there are still needs to be satisfied to guarantee a safe autonomous driving. The road environments to be treated are complex, where many situations are difficult to predict in advance, and the slightest failure can lead to fatal accidents. The robustness of the analysis is therefore a key issue. Among the many problematic situations, sensor failures or malfunctions are possible causes of analysis errors. Even if the sensors used are becoming increasingly reliable, a partial or total malfunction of the sensors is not inconceivable. The methods developed for road scene understanding must therefore take this possibility into account, in order to ensure a correct analysis with the still functional sensors thanks to the essential use of multimodality.

In this chapter, we first go through the use of multimodality for road scene understanding in the literature. The different image-based modalities used in ADAS context, as well as their usefulness in facing of certain problems, are detailed with the applications using this modality among others in the literature. The multimodal datasets for road scene understanding are listed, allowing us to understand the major issues that ADAS community seeks to master.

For object detection, multimodality can be combined with convolutional neural networks (CNN), achieving the best results in many computer vision tasks. In a second step, the main CNN architectures for object detection with their different pipelines are described. These reference networks are all unimodal, taking only one color image as input. Different deep sensor fusion strategies for computer vision tasks are then reviewed, in order to combine multimodality and deep neural networks for computer vision tasks.

Finally, the main part of the thesis being on the robustness of multimodal CNNs, the strategies used for its improvement in the literature are described, in degraded conditions (sensor failure, adverse weather conditions).

1.2 Multimodality in Advanced Driver-Assistance Systems

The field of ADAS requires to master several challenges, including object detection, but also object tracking, depth estimation, motion estimation, semantic segmentation, road lane detection or visual odometry for instance. The diversity of these tasks involves the autonomous vehicle algorithms to acquire specific skills depending on the challenge. We believe that the use of a single color camera is not appropriate for all these different challenges, not to mention the issue of robustness. Therefore, multimodality is an essential point in order to respond to these different issues. However, multimodality use implies an increase in computational resources and embedding multiple sensors on a vehicle drastically increases its price. The input modalities must thus be chosen widely. In this section, we first reviewed the multimodal approaches for road scene understanding, and list the different modalities used for computer vision challenges in ADAS context. We will then detailed the different multimodal datasets available in this field, which are an essential basis for the development of multimodal solutions.

1.2.1 Multimodal approaches for road scene understanding

Despite the great diversity of tasks for road scene understanding, we will focus only on computer vision challenges in this task. Most of the time, a grayscale or a color image from a single camera is part of the input data. Acquiring this input is quite easy, and this signal is very suitable for object classification, hence for object detection or semantic segmentation. However, using only one grayscale or color image for depth estimation for instance makes this task more complicated. The addition of other modalities from diverse sensors makes it possible to add relevant information according to the needs in the differents tasks.

One of the most used sensor for road scene understanding is the LiDAR. This signal is often combined with color or grayscale images thanks to their complementarity. Indeed, LiDAR signal allows to estimate the depth easily, filing one of the color image gaps. It is all the more beneficial for tasks requiring to estimates object's distance to the ego-vehicle, like 3D object detection, to the point where a lot of work for developing methods using only a LiDAR signal for road scene understanding has been done. As related in [1], LiDAR signal could be represented in different forms like as a volume [58, 59, 119], as a projected 2D image [18, 25, 28, 52, 80] or as a set of points [110]. Moreover, these representations are sometimes projected in the camera plane [25, 28, 52, 80], in bird's eye view [57, 58], or both [18]. Finally, this signal being in the form of a point cloud intially, it can also be processed to be seen as a dense depth map [21, 22, 25]. In [18], Chen et al. proposed Multi-View 3D Networks (or MV3D), a multimodal framework for 3D object detection taking as input RGBimages and LiDAR point cloud projected in both camera plane and bird's eye view. MV3D is composed of a 3D Proposal Network that utilizes bird's eye view representation of LiDAR point cloud to generate 3D candidates boxes, and a Region-based Fusion Network that extracts region-wise features by projecting these proposals from multiple views. Liang et al. introduced in [59] a multi-task multi-sensor detector taking LiDAR point cloud as a volume and RGB images. This network jointly solve multiple perception tasks, namely 2D and 3D object detection, ground estimation and depth completion, in order to learn better feature representations which results in better detection performance. However, LiDAR remains a expensive sensor for autonomous vehicle and can't deliver accurate measurements of surroundings in fog, dust, snow or night for instance. A very close alternative is the use of radar signal [14, 16, 71] instead of LiDAR one. Comparing to the previous sensor, radar can easily operate at night and in cloudy weather conditions, but it has low accuracy and resolution, complicating the distinction of objects on the road.

Another possibility to have depth information is to compute it with stereo vision images. A stereo camera setup can provide dense depth maps which can be processed by computer vision methods using RGB or grayscale images. Therefore, we can observe some work using stereo vision in addition of RGB images [17, 19, 32, 56, 78, 82, 99, 118, 120]. In [17], Chen et al. introduced a 3D object proposal method taking RGB images and depth maps from stereo cameras as input for multi-task prediction as 2D and 3D object detection and orientation estimation. Their method exploits stereo imagery to place proposals in the form of 3D bounding boxes that are then run through a CNN to obtain high-quality 3D object detection. Chen et al. presented in [19] a bi-directional cross-modality guided encoder that fuses feature maps extracted from RGB images and Depth maps from stereo

vision at different steps, for road scene semantic segmentation. They also introduced SA-Gate module, a gate unit that recalibrates and fuses the complementary information from the 2 modalities, and Bi-direction Multi-Step Propagation module, that propagates the fused multimodal features along with modality-specific features. Nonetheless, the majority of proposed methods using depth with RGB images for road scene understanding adopted LiDAR signal. Indeed, stereo vision configuration uses matching algorithm to find correspondences in both images and calculate the depth of each point relative to the camera, increasing processing power required. Then, stereo camera setup requires calibration, but the system is intended to be installed on moving vehicles, being able to decalibrate it quite easily.

For tasks like object tracking, orientation estimation or motion estimation, it is relevant to use temporal information. This can be extracted from several recordings from one sensor (camera or LiDAR). Each recording can be processed independently from the other, to obtain outputs that will be then fused to obtain temporal informations in post-processing. Recordings could also be processed with feedback of outputs from the previous ones, like in a recurrent neural network. Finally, recordings can be processed in order to create a new modality. Thereby, some methods were proposed using optical flow [25, 32, 78, 82, 83, 111] as input, computed from 2 temporal adjacent images from the same camera, or LiDAR flow [83], following the same processus but with lidar scans. Enzweiler and Gavrila presented in [32] a multilevel Mixture-of-Experts approach to combine information from grayscales images, depth maps from stereo vision and flow images for pedestrian classification. Modalities are transformed into Histogram of Gradients and Local Binary Patterns features which are then used by a multilayer perceptron and a linear Support Vector Machine for classification. In [83], Rashed et al. introduced FuseMODNet, a robust and real-time CNN using RGB image optical flow and LiDAR flow for moving object detection. Both optical flow and LiDAR flow extractions are made by two subnetworks, then these outputs, with RGB image, are processed for extracting features that are all fused for finally create an output mask. Nevertheless, flow signal, like stereo vision signal, uses matching algorithm to find correspondences in both recordings (images or LiDAR scans), in order to determine their displacement in their environment. Theses algorithms require processing power, increasing processing time and making real time operation more difficult. Moreover, in case of fixed sensors, informations containted in flow signal are easily readable and in high quality, which is not the case when the sensors are in motion. Therefore, a high-quality flow signal with relevant information will be more difficult to produce from sensors on board an autonomous vehicle.

The use of non conventional imaging for road scene understanding can also be very interesting. Opting for non conventional sensors in addition of classic cameras aims autonomous vehicle algorithms to be more efficient in case of adverse weather conditions, like fog and rain, or during the night for instance. Furthermore, non conventional images are very close to RGB images and thus can be processed by classical computer vision algorithms. Several multimodal methods using non conventional sensors have been developed, like with infrared camera [56], polarized camera [7, 99] or multispectral camera [20, 105, 123] for instance. Chen and Huang presented in [20] a multimodal machine learning method that combines data from stereo vision cameras and a multispectral camera for pedestrian detection. Disparity map from stereo images are computed into the

framework and used for thermal image alignment on the color image plane. Then feature extraction and classification are made separately on color images, disparity map and thermal images, and all these outputs are finally fused for the final decision. 4 multimodal proposals were proposed in [7] fusing color and polarimetric images for object detection in road scenes in adverse weather conditions. Experiments were made with different modalities configuration as input, which contains color and polarimetric images in different color spaces in order to find the optimal combination between color spaces and polarimetric features for enhancing object detection in road scenes under fog. Despite all these advantages mentioned before, non conventional sensors are very expensive and are therefore still little used for autonomous vehicles. In addition, there are few databases offering these modalities, even if we can cite KAIST [51] or Elektra [41] for respectively thermal and infrared images.

Regarding the fusion of more than 2 modalities for road scene understanding, there are some propositions, but they are getting rarer. We can observe different modality configuration, like RGB image + optical flow + LiDAR [25], RGB image + optical flow + depth from stereovision [32, 82], RGB image + depth from stereovision + infrared image [56] or panoramic RGB image + depth from stereovision + polarized image [99]. In [25], Daniel Costea et al. introduced a boosting-based sliding window solution using RGB images, disparity map computed from LiDAR point cloud and optical flow for object detection. Multimodal multiresolution filtered features are used for constructing discriminating features from these modalities, then a multiscale sliding window with a boosting-based classifier detect objects from these features with 2D and 3D context. Krotosky and Trivedi presented in [56] a multimodal trifocal framework consisting of stereo RGB cameras and an infrared camera for pedestrian detection. Disparity map from stereo images is computed on top of the framework, and infrared image alignment is made with disparity map with trifocal tensor. One of the reasons for this low number of proposals is that databases for road scene understanding do not offer an unlimited number of modalities. Moreover, having more modalities in input data implies more processing time, while one of the major needs for road scene understanding is real-time operation. For these reasons, authors first seek to take as input data the most useful modalities to solve their task. Thus, adding a modality serves above all to fill a gap in the input data.

1.2.2 Datasets for road scene understanding

During the last decade, several datasets for road scene understanding have been released and have laid the foundation for many applications, including those presented in the previous section. These datasets offer a lot of data, in different forms and captured in multiple environments around the world, in order to get as close as possible to real driving conditions. In this section, we reviewed the multimodal datasets for road scene analysis, proposing data recorded from several sensors and labelled for different computer vision tasks, such as object detection, semantic segmentation, depth estimation, among many others. The datasets' properties are summarized up in table 1.1.

TME Motorway Caraffi et al. released the Toyota Motor Europe Motorway Dataset [12] in 2012, for vehicle detection. TME Motorway is composed by 28 clips corresponding to approximately 30,000 frames. Image sequences were selected from acquisition made in North Italian motorways, with variable traffic situations, number of lanes, road curvature, and lighting. This dataset comprises color images from stereo cameras.

KITTI In 2012, Geiger et al. introduced KITTI dataset [37], a multimodal dataset which marked an important milestone in the ADAS field. Its data was recorded by driving around Karlsruhe city (Germany) with a sensor system consisting of two high-resolution stereo camera systems (grayscale and color), a 64-beam LiDAR and inertial sensors. KITTI 2D Object Detection Benchmark is composed of 14,999 recording scenes, with 80,256 objects labeled on its training images from 8 classes : *Car*, *Pedestrian*, *Cyclist*, *Truck*, *Van*, *Tram*, *Person Sitting* and *Misc* (for Miscellaneous, designating other road traffic objects like Trailers, Motorcycle or Segways).

KAIST KAIST MultiSpectral Pedestrian Dataset [51] was released in 2015. It consists of 95,328 pairs of images recorded with RGB and thermal cameras for 2D pedestrian detection. A total of 103,128 labelled 2D objects are given from 3 classes including *person*, *people* and *cyclist*.

Elektra In 2016, González et al. introduce the Elektra Visible-FIR Day-Night Pedestrian Sequence Dataset [41], for pedestrian detection. It consists of nearly 7000 pairs of images with 3000 pedestrians annotated for training, and around 1500 for testing with 3500 labelled pedestrians. This dataset includes 2 sequences, one recorded at daytime and the other at night, with a RGB and a Far Infrared camera.

Cityscapes Cordts et al. presented in 2016 Cityscapes Dataset [23], a large-scale dataset that contains a diverse set of stereo video sequences recorded in street scenes from 50 different cities, resulting in approximately 25,000 images for semantic urban scene understanding. They defined 30 visual classes for annotation, grouped into eight categories : *flat*, *construction*, *nature*, *vehicle*, *sky*, *object*, *human*, and *void*. 5000 of Cityscapes images have high quality pixel-level annotations and 20,000 additional images have coarse annotations to enable methods that leverage large volumes of weakly-labeled data. A subset of its dataset was also used to build the CityPersons dataset [122] for pedestrian detection.

ApolloScape ApolloScape Open Dataset [50], realased in 2018, is a large dataset for various tasks including 3D reconstruction, instance segmentation, stereo estimation and self-localization among others. It consists of RGB videos and corresponding dense 3D point clouds acquired in 4 regions in China with a sensor system consisting of 6 RGB cameras and 2 LiDAR. This results in a total of more than 144,000 images pixel-wise labelled in 35 classes.

Argoverse Chang et al. introduced Argoverse [15] in 2019, two datasets for respectively 3D object tracking and motion forecasting. Argoverse was collected in Pittsburgh and Miami by a fleet of autonomous vehicles. The first dataset includes 113 sequences recorded

with 7 cameras and 3D point clouds from long range LiDAR. A total of 11,052 objects are labelled from 15 classes.

nuScenes In 2020, Caesar et al. introduced nuScenes [9], a public large-scale dataset for 3D object detection. nuScenes consists of 1000 driving scenes of 20 second collected in Boston and Singapore, resulting in approximately 40,000 images, with 1.4 million 3D objects labelled from 23 classes. Its acquisitions were made with a sensor system including 6 RGB cameras, 1 LiDAR, 5 Radars and inertial sensors. For 2D object detection and instance segmentation, nuImages, a subset of nuScenes, was created, resulting in 93,000 RGB images more challenging and approximately 700,000 2D bounding boxes.

Waymo Waymo Open Dataset [100] was launched in 2020. This dataset consists of 1150 scenes that each span 20 seconds, resulting in 390,000 images collected in several places using 5 RGB cameras and 5 LiDAR. These images are labelled for object detection with 11.8 millions 2D bounding boxes and 12.6 millions 3D bounding boxes from 4 classes : *vehicle*, *pedestrian*, *cyclist* and *sign*.

A2D2 Geyer et al. published in 2020 the Audi Autonomous Driving Dataset (or A2D2), a dataset for semantic segmentation and 3D object detection featuring respectively 41,280 frames pixel-wise annotated in 38 classes, and 12,499 frames labelled with 3D bounding boxes from 14 classes relevant to driving (e.g. cars, pedestrian, buses, etc.). These data were collected in 3 cities in the south of Germany, with a sensor suite consists of six cameras and five LiDAR units, providing full 360° coverage.

ADUULM Pfeuffer et al. released in 2020 ADUULM-Dataset [76], a semantic segmentation dataset which consists of fine-annotated camera data and pixel-wise labeled LiDAR data recorded in good weather conditions, but also in adverse weather conditions such as darkness, fog or heavy rain. This dataset includes 3893 images recorded with 2 RGB stereo cameras, 1 LiDAR and inertial sensors. These images were pixel-annotated in 12 classes including road traffic actors such as *Car*, *Truck*, *Pedestrian*, *Bicyclist* among others.

PolarLITIS In [6], Blin et al. introduced PolarLITIS, a 2D object detection dataset containing pairwise RGB and polarimetric version of the same road scene under three conditions : foggy, sunny and cloudy scenes. This dataset contains 2569 pairs of RGB and polarimetric images, collected in Rouen, France, annotated with more than 18,000 2D bounding boxes from 4 classes : *Car*, *Person*, *Bike* and *Motorbike*.

CADC In 2021, Pitropov et al. proposed in [77] the Canadian Adverse Driving Conditions Dataset (or CADC) for 3D road object detection. It consists of sequences collected in the region of Waterloo, Canada, and focus on real world driving data in snowy weather conditions. CADC was recorded with a sensor system consisting of 8 wide angle cameras, 1 LiDAR and inertial sensors. This results in approximately 32,000 images annotated with 3D bounding boxes from 10 classes including *Car*, *Pedestrian*, *Truck*, *Bus* and *Bicycle* among others.

PandaSet Xiao et al. released in 2021 PandaSet [113], a multimodal dataset for semantic segmentation which provides a complete kit of high-precision sensors covering a 360° field of view. The dataset was collected using 6 RGB cameras, one 360° mechanical spinning LiDAR and one forwardfacing, long-range LiDAR. The dataset contains more than 100 scenes, resulting in 8240 frames, and provides 28 types of labels for 3D object detection and 37 types of labels for semantic segmentation.

Dataset	Year	Sensors	Computer vision Task(s)	Number of Samples	Locations	Time of the day	Weather	Classes
TME Mortorway [12]	2012	2 RGB cameras	2D Object Detection	30K images	Highways	Day	Clear	2
KITTI [37]	2012	2 RGB cameras 2 Grayscales cameras 1 LiDAR	2D/3D Object Detection Orientation Estimation Optical Flow Estimation Stereo Matching Visual Odometry	15K images	Urban areas Rural areas Highways Residential areas	Day	Clear	8
KAIST [51]	2015	1 RGB camera 1 Thermal camera	2D Object Detection	95K images	Urban areas	Day Night	Clear	3
Elektra [41]	2016	1 RGB camera 1 Infrared camera	2D Object Detection	8.5K images	Urban areas	Day Night	Clear	1
Cityscapes [23]	2016	2 RGB cameras	Semantic Segmentation	25K images	Urban areas	Day	Clear	30
ApolloScape [50]	2018	6 RGB cameras 2 LiDAR	Instance Segmentation 3D Reconstruction Stereo estimation etc.	144K images	Urban areas Rural areas Highways Residential areas	Day Night	Clear Rainy	35
Argoverse [15]	2019	7 RGB cameras 1 LiDAR	3D Object Tracking	113 sequences	Urban areas	Day Night	Clear Rainy	15
nuScenes [9]	2020	6 RGB cameras 1 LiDAR 5 Radars	3D Object Detection	40K images	Urban areas	Day Night	Clear Rainy	23
Waymo [100]	2020	5 RGB cameras 5 LiDAR	2D/3D Object Detection	390K images	Urban areas Rural areas Highways Residential areas	Day Night	Clear Rainy	4
A2D2 [38]	2020	6 RGB cameras 5 LiDAR	3D Object Detection Semantic Segmentation	41K (segmentation) 12.5K (detection)	Urban areas Rural areas Highways	Day	Clear Rainy	38 (segmentation) 14 (detection)
ADUULM [76]	2020	2 RGB cameras 1 LiDAR	Semantic Segmentation	3.9K images	Urban areas Rural areas	Day Night	Clear Rainy Foggy	12
PolarLITIS [6]	2020	1 RGB camera 1 Polarimetric camera	2D Object Detection	2.5K images	Urban areas Highways Residential areas	Day	Clear Foggy	4
CADC [77]	2021	8 RGB cameras 1 LiDAR	3D Object Detection	32K images	Urban areas Rural areas Highways Residential areas	Day	Snow	10
PandaSet [113]	2021	6 RGB cameras 2 LiDAR	3D Object Detection Semantic Segmentation	8.2K images	Urban areas Rural areas	Day Night	Clear	28 (detection) 37 (segmentation)

Table 1.1 – Comparison of multimodal datasets for road scene understanding.

1.3 Multimodal deep 2D object detection

The emergence of CNNs in recent years has enabled great advances in many computer vision tasks. These methods generally obtain the best performances on the various benchmarks available nowadays. Among these tasks, object detection is one of the most challenging issues. This task consists in locating objects in an image, symbolised by a bounding box, and classifying them. The vast majority of the many proposed approaches are unimodal, taking as input only a color image. However, there are many works on deep sensor fusion, combining data fusion with a deep learning strategy. This section first focuses on the most competitive deep architectures for common object detection. Then, the deep fusion architectures for computer vision tasks such as classification, object detection or semantic segmentation are listed.

1.3.1 CNN for 2D object detection

At the start of the 21st century, several non-neural approaches for object detection were proposed, initially intended to detect one specific type of object, like faces with Viola-Jones algorithm [108], based on Haar features, or pedestrians with the Histogram of Oriented Gradients [24]. With the appearance of CNN [55], many advances have been made during the last decade, as related in [65], and many challenges appeared with large databases for multi-class object detection, like COCO [61] and ImageNet [27]. DetectorNet [102], DeepMultiBox [33] and OverFeat [92] are the precursors. However, the main difficulties associated with this task are the variable number of objects present in the image, with the possibility of having zero, as well as their size and their aspect ratios. This requires the algorithms to select a huge number of regions of interests, with different locations, sizes and aspect ratios, and classify the presence of the object within each of them, which can computationally blow up.

In order to bypass the problem of regions selection, Girshick et al. proposed **R-CNN** (Regions with CNN features) in 2014, a 2D object detector that selects a maximum of 2000 regions, called region proposals, with a selective search algorithm. Then, a CNN computes each of these region proposals, warped into a square, in order to extract their features. Finally, a Support Vector Machine (SVM) classifies the presence of the object in the features of each candidate region proposal features, and refine the coordinates of its bounding box. Even if the selective search reduces the number of regions of interests (RoIs) to analyze, R-CNN takes several dozens of seconds to analyze a single large image, and hence cannot perform in real-time.

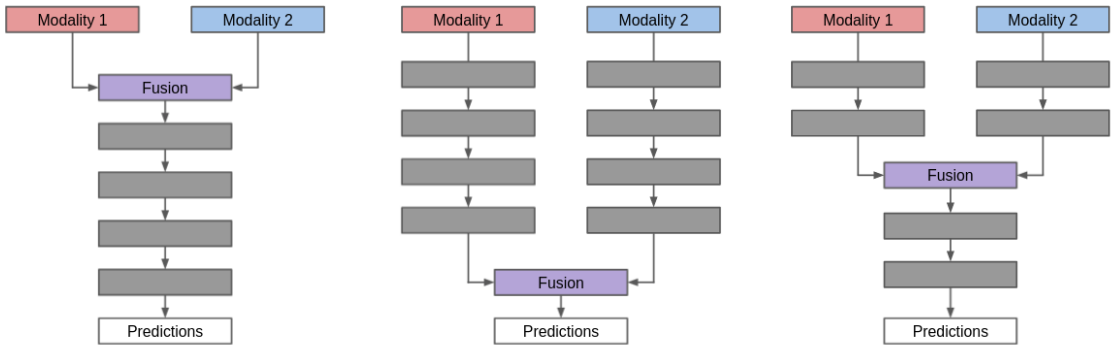
The same author updated his previous network to **Fast R-CNN** [39], a faster object detection algorithm that performs feature extraction over the image with a CNN to generate a convolutional feature map, and then uses selective search to identify region proposals. These are warped into squares and reshaped into a fixed size by a RoI pooling layer. Finally, fully connected layers and a softmax layer predict the class of each region proposal and the offset values for the bounding box. Fast-RCNN is considerably faster than R-CNN because we don't have to feed its CNN with 2000 region proposals when analyzing an image. However, selective search is slow and affects the performance of the network.

This network bottleneck was removed with **Faster R-CNN** [88], an update of Fast R-CNN that eliminates the selective search algorithm. First, similar to Fast R-CNN, a CNN extracts features from the input image. Then, a Region Proposal Network (RPN) predict the region proposals, instead of the selective search algorithm. Finally, the predicted region proposals are reshaped with a RoI pooling layer, and a subnetwork consisting of fully connected layers and a softmax layer is used to classify them and predict the offset values for their bounding boxes. The withdrawal of the selective search algorithm, replaced by the RPN, allows Faster R-CNN to be applied for real-time object detection. Nonetheless, like Fast R-CNN and R-CNN, Faster R-CNN is a two-stage object detector, that is to say, an object detector that, first, generates region proposals, and then sends them down the pipeline for object classification and bounding box regression. These models reach the highest accuracy rates but are typically slower.

On the other hand, one-stage object detectors require only a single pass through the neural network by making a fixed number of prediction on grid. This make these networks faster, but potentially less accurate than the two-stage object detectors. **YOLO** (You Only Look Once) [86] is one of the most common example of this category of object detectors. It consists of a single CNN that processes an image split into a $N \times N$ grid with M bounding boxes in each part of the grid, and predicts the class probabilities and offset values for each bounding box. Other updates of YOLO were proposed, such as YOLO 9000 [84], YOLOv3 [85] and YOLOv4 [8], faster and more accurate than the first one while keeping the same strategy. YOLO networks are among the fastest object detectors but struggle with small objects within the image.

In the same category, Liu et al. proposed Single Shot Multibox Detector (**SSD**) [66], a one-stage object detector as fast as YOLO, but with an accuracy competitive with two-stage object detectors like Faster R-CNN. SSD is composed of a CNN that extract feature maps at different scales, and specific convolutional layers, called extra feature layers, to produce a set of bounding boxes with vectors of probabilities corresponding to the confidence over each class of object. This network is followed by the Non-Maximum Suppression (NMS) method, to keep the most relevant bounding boxes. Despite its high scores on different detection challenges, SSD, like other one-stage object detectors, faces the class imbalance issue between the positive boxes, which contain an object, and the negative ones. During its training, SSD evaluate roughly between ten to hundred thousand candidate locations per image, with a large majority of negative boxes, which dominates the loss. Therefore, the few interesting cases, that is to say the positive boxes, go undetected by the network.

To avoid this problem, Lin et al. introduced **RetinaNet** [62], another one-stage object detection model that uses Focal Loss, a loss function to address class imbalance during training. This CNN is composed of a backbone as a feature extractor, combined with a Feature Pyramid Network (FPN) for constructing a rich multi-scale feature pyramid from one single resolution input image. Two subnetworks complete its architecture, one for classification which predict the probability of object presence at each spatial position, and the other one for bounding box regression, that outputs the offset values for each bounding box. During its training, RetinaNet uses Focal Loss which is an enhancement over Cross-Entropy Loss and is designed to down-weight easy examples and thus focus training on hard negatives.



(a) (b) (c)

Figure 1.1 – Illustration of the different fusion schemes : a) Early fusion ; b) Late fusion ; and c) Middle fusion.

For the construction of our multimodal networks, we decided to start from an existing object detector, and adapted its architecture to our problematic. We are looking primarily to create multimodal CNNs with high accuracy, positioning speed as a secondary objective. However, these multimodal CNNs have to be able to handle multiple signals, thus having a larger and more complex architecture which would increase its learning time. In view of these issues, we select RetinaNet as a basis, a good compromise between speed and high object detection accuracy in our opinion.

1.3.2 Deep sensor fusion

As previously mentionned in this chapter introduction, sensor fusion is vital for autonomous vehicle. Therefore, deep fusion task has been widely explored during the last decade [36]. This section summarizes the different schemes and fusion operations in deep fusion for computer vision tasks. We restrict our literature review to methods fusing images or volumes. The proposed deep fusion methods can be classified in three different fusion schemes : early fusion, late fusion and middle fusion, illustrated in figure 1.1.

Early fusion

Early fusion consists in fusing raw or preprocessed data before passing it through the network. This fusion scheme enables light architectures to learn the joint feature of multiple sensors without requiring a lot of processing power. Its simplicity means that several methods opt for this fusion scheme [96, 105, 109, 112, 115].

In [96], Sindagi et al. proposed Multimodal VoxelNet (or MVX-Net), a CNN extended from Voxelnet [124] for 3D Object detection fusing RGB images and LiDAR maps projected on front view plane at an early stage. They experimented two feature fusion techniques, PointFusion and VoxelFusion, projecting respectively 3D points and 3D voxels from LiDAR point cloud on RGB image, followed by a single stage detector. Their experimental results show significant improvements over state-of-the-art methods on the KITTI dataset, with better accuracy for MVX-Net combined with PointFusion. Yang et al. fused

LiDAR point cloud in bird's eye view with high-definition maps for 3D Object detection with HDNET [115]. These modalities are concatenated before being processed by a single stage detector. They also proposed an online map prediction module that estimates bird eye's view maps from LiDAR point cloud when offline high definition (HD) maps are not available. This network, using online or offline HD maps, shows better performances that detectors using only LiDAR on KITTI bird eye's view object detection benchmark.

However, early fusion involves a lot of drawbacks. First of all, early fusion comes with the cost of model inflexibility : replacing a modality by a new one, or extending the input channels implies networks to be completely retrained. Then, networks using early fusion are very sensitive to data misalignment caused by sensor defect, calibration error or different sampling rate. Finally, comparing to late and middle fusion, early fusion usually make networks less accurate, as related in [11, 18, 64, 75, 91].

Late fusion

Contrary to early fusion, late fusion consists in first processing each modality separately with a domain specific network and then fuse the decision output of them. This fusion scheme aims to design networks with high flexibility, as replacing or adding a new modality only requires to its domain specific network to be trained. Moreover, since fusion is made at the end of the pipeline and that each decision output is provided independently of the other, networks are usually more robust to sensor breakdown. Thus, late fusion scheme was investigated in different networks [30, 68, 72, 73, 103].

In [73], Pang et al. introduced Camera-LiDAR Object Candidates Fusion (CLOCs) that process separately RGB images and LiDAR point cloud and then fuse the detection candidates for 3D object detection. They obtained the best result on KITTI Benchmark by using Cascade R-CNN [10] for 2D object detection on RGB images, and PV-RCNN [94] for 3D object detection on LiDAR point cloud. Takumi et al. [103] opted for fusing detection from multispectral images, composed of RGB images, near-, middle- and far-infrared images for 2D object detection. Modalities are processed by four YOLO [86] networks, before being combined with an ensemble method. Oh and Kang [72] proposed an object detection and classification method using decision-level fusion from RGB images and LiDAR point cloud. The classification outputs from each modality are provided by unary classifiers and then combined using a CNN.

Nonetheless, late fusion scheme main drawback is that it is computationally expensive because of its architecture including one specific network for each modality. Therefore, late fusion are usually slower, which complicates networks to process in real time.

Middle fusion

Middle fusion is a good compromise between early and late fusion. It consists in combining the data from multiple sensors inside the processing pipeline, usually after feature extraction. This scheme offers multiple possibilities of fusion and therefore make deep networks highly flexible as they learn different cross modalities with different feature representations and at different levels. With a custom architecture, deep mid-fused networks are more accurate that early-fused ones and requires less processing power than late-fused networks. However, choosing middle fusion implies a lot of work in order to find

the optimal way to fuse intermediate layers given a specific network architecture. Most of the work on deep multimodal fusion is based on this scheme [5, 13, 19, 29, 44, 52, 58, 59, 101, 107, 114].

Dou et al. proposed SEG-VoxelNet [29], which fuses RGB image and LiDAR point cloud for 3D object detection. In Seg-VoxelNet, RGB image is first segmented with a MobileNet network [48]. Then, the LiDAR point cloud is projected on the segmented map, and this output is finally processed by a variant of VoxelNet [124]. Chadwick et al. [13] fuse RGB image and radar raw data during feature extraction for 2D object detection. These modalities are first processed separately with ResNet18 blocks [46] and are then concatenated before being processed with others ResNet18 blocks. Xu et al. introduced PointFusion in [114] for 3D object detection using RGB image and LiDAR point cloud. Feature extraction is made on these two modalities with respectively a ResNet and a PointNet [79], and are then combined with a dense fusion subnetwork. Valada et al. [107] proposed a multimodal CNN based on AdapNet [106] for semantic segmentation using RGB images with another modality, among depth and thermal images. This framework uses two ResNet backbones for encoding. Then, Self-Supervised Model Adaptation units recalibrate and fuse modality-specific feature maps. Finally, fused feature maps are incorporated at different levels of framework decoder. Sun et al. fuse RGB and thermal images for semantic segmentation with RTFNet [101], where fusion is made between ResNet blocks via element-wise summation during signals encoding. Ha et al. fuse also RGB and thermal images for the same task with MFNet [44]. Modalities are processed by two encoders proposed in [45]. Then, feature maps are combined via mini-inception blocks after encoding, and via shortcut blocks during maps decoding, where convolutions, concatenation and element-wise summation are made inside these blocks.

Fusion operations

When designing a multimodal deep neural network, it is important to determine what fusion operations should be used to best combine the information contained in each fusion input. As related in [36], we find some typical fusion operations in the literature. Feature maps could be fused via element-wise summation, in order to create a fused feature map that has the same size as the ones in input. These input could also be combined with average mean. These representations condense input information quite simply and require little calculation. However, these fusion strategies cause a significant loss of information. Therefore, concatenation operation could be used in order to keep all input information. Usually, feature maps are stacked along their depth axis before being processed by a convolution layer. When opting for late fusion, networks usually combines their output with ensemble fusion, where the combined output is the union of the input. All these operations are quite simple but provide little help to generate optimal joint feature representation. Indeed, they do not consider informativeness of a sensing modality, hoping that the multimodal neural network can implicitly learn to weight the feature maps. Therefore, when facing redundancy or contradiction in multimodal input, model's performances may be dramatically affected. To remedy these problems, Mixture of Experts approach is often used, by explicitly modeling the weight of a feature map. The principle of such a fusion is first to process each modality by domain-specific networks, producing feature maps outputs which will be then averaged and combined by a gating

network. This fusion strategy makes remarkable progress in terms of accuracy, as the learning model takes into account the contribution of multimodal features at multiple stages.

1.4 Robustness improvement of multimodal deep neural networks

Few works focus on the robustness of multimodal neural networks, as related in [3], although it remains a major issue for the autonomous vehicle domain. Among those concerned, we can distinguish 3 categories of methods for improving robustness.

First of all, some work focuses on proposing multimodal databases for robustness enhancement, proposing data in more diverse conditions to obtain robust and reliable results in detection or segmentation. Among the datasets presented in the section 1.2.2, we can mention ADUULM-dataset, CADC or PolarLITIS which have been particularly established for the robustness problem in adverse weather conditions. The main advantage of these databases is that they contain real data of degraded conditions, and thus to develop adapted methods on real disturbances that a vehicle can meet. To complement these rather sparse data, there are also synthetic databases simulating adverse weather conditions [75, 90]. However, all these databases are only focused on this problem, and not on other robustness issues such as technical failures or non-calibration of sensors. Moreover, they are an essential support for the creation of robust multimodal neural networks, but they alone cannot solve the robustness problem without an explicit training strategy for these models.

Several works have opted for the design of neural networks with a specific architecture in order to support deteriorated or missing data. Liu et al. proposed in [63] Sensor Dropout, a specific customization of Dropout layer applied on feature map from multiple sensors (camera images, laser scans, GPS and odometry), to improve multisensory policy robustness and handle partial failure in the sensor system. In [70], Neverova et al. introduced ModDrop, a multimodal Dropout Layer for learning cross-modality correlations while preserving uniqueness of each modality-specific representation by randomly dropping modalities during network training. Kim et al. presented in [53] Robust Deep Multimodal Learning, a CNN for 2D detection taking RGB and LiDAR as input. Its robustness learning is achieved through Gated Information Fusion Networks, by weighting the input feature maps before fusing them. The various results obtained on synthetic degraded data have shown that this approach significantly reduces performance losses in case of sensor failure. All the above mentioned networks have shown that this approach remains one of the best options for improving performance with imperfect sensor data, but this result often implies a loss of scalability in these networks, requiring very specific data fusion strategies that are not applicable to all types of multimodal networks.

Finally, a good part of the work on multimodal neural networks robustness enhancement focuses on the data augmentation technique, by simulating noise on real data during training. In [31], Eitel et al. presented a data augmentation scheme for robust learning with depth images by corrupting them with realistic noise patterns for RGB-D object recognition. de Blois et al. introduced in [26] Input Dropout, a data augmentation tech-

nique that consists in stochastic hiding of one or many input modalities at training time. This technique, experimented with multiple computer vision tasks (classification, image dehazing, object detection) and different modalities configurations (RGB + Depth, RGB + Thermal image), improved performances at test time, even if the additional modality is unavailable. Bijelic et al. used in [4] a data augmentation method mixing object and background images for pedestrian classification. Experiments on several classification databases show that a neural network may be able to ignore data from underperforming sensors even though it has never seen that data during training.

1.5 Summary

In this chapter, we first reviewed multimodality in ADAS context. The main modalities used for road scene understanding are first listed with their application, on different computer vision tasks for autonomous driving field. Then, the different multimodal datasets for road scene understanding are reviewed. Through this section, we have seen that the most used combination of modalities are RGB images from color camera with LiDAR point cloud. This can be explained with the profusion of datasets offering data from these 2 sensors. Moreover, we can see 2 trends in the datasets recently released. Some opt for several color cameras and several LiDAR in order to cover a 360° point of view [9, 15, 38, 50, 100], thus proposing a large amount of data in rather classical driving conditions. On the other hand, some datasets focus more on data diversity, using more specific sensors (thermal camera [41], multispectral camera [51], polarimetric camera [6]), in more difficult conditions, and in particular in adverse weather conditions [6, 76, 77]. Finally, as mentioned in the section 1.2.1, the vast majority of the works using multimodality for road scene understanding focus only on the improvement of the general performances or the computation time. Through this perspective, the modalities are carefully chosen and are only used as an added value. In our case, we have a different approach for the improvement of CNN robustness in case of sensor failure. Indeed, instead of selecting modalities only to improve algorithm performances, we prefer to opt for modalities that will ensure the best possible detection in the absence of other modalities. Therefore, information redundancy becomes an asset for us.

In a second step, we reviewed the main CNNs for object detection. Then, we listed the different fusion strategies used for multimodality in computer vision tasks with neural networks. These strategies are divided into 3 categories: early fusion, late fusion and middle fusion. Early fusion, despite its simplicity and reduced computational requirements, generally gives the worst results, and therefore remains a little chosen scheme. Regarding the late fusion, even if its flexibility and robustness are strong points, its scheme is computationally expensive, and is a strategy to avoid when one wants to fuse a lot of modalities. Finally, the middle fusion remains the best option, combining good performances and reduced computational needs. However, this strategy has one major flaw: its flexibility. When we want to add an input modality, we have to change CNN architecture and sometimes its fusion operations. Moreover, for each input modality, there is almost all the time a part of CNN architecture dedicated only to this one (in particular during the feature extraction), which increase its number of parameters. The fusion of more than 2 modalities thus requires a lot of computing power, and makes its real time operation

much more complicated. In our problem, where the use of several modalities allows to guarantee a better robustness, we explore different possibilities to reduce the size of the network without losing performance.

Lastly, the different strategies for improving the robustness of CNNs have been listed. We found that there is very little work on the subject, despite the need for this guarantee on ADAS context. Moreover, a large part of this work focuses on adverse weather conditions, which remains an important problematic, but which is far from ours: the malfunction or failure of one or more sensors.

Chapter 2

Multimodal convolutional neural networks for 2D object detection

Contents

2.1 Introduction

Road Traffic Actors Detection plays an important role in the field of Advanced Driver-Assistance Systems (ADAS). In this context, it's a crucial step to ensure safe autonomous driving. Indeed, the result produced by the detection will have a very strong influence on the vehicle's action decisions. Therefore, the slightest detection or classification error can cause fatal accidents. A lot of work has been done during the last decades on this task in order to get a very high accuracy in real time and in real world environment. However, because of the variety of road traffic actors (e.g. Car, Pedestrian, Cyclist), their visibility, and other parameters like weather conditions, road traffic actors detection is still an open challenge. In recent years, the emergence of convolutional neural networks has made it possible to pass a milestone in terms of performance and speed. Designed first for image classification, they were quickly adopted for computer vision tasks, including object detection. Even if state-of-the-art CNNs have very large architectures, requiring a lot of computing power, these detectors remain the best option at the moment. In order to get these accurate CNNs, many large databases for road scene understanding have emerged during this last decade, proposing real world data in several modalities, diverse conditions around the world, and labelled for different tasks.

However, many state-of-the-art methods using CNN for road traffic actors detection use only one RGB image from color camera as input data. It can be explained by the fact that most of the computer vision challenges, where CNNs can be used, are based on analyzing and extracting information from one color image. The most popular CNNs are designed for taking only one 3-channel image as input, whatever the task they perform. Starting from this work, one can easily achieve remarkable performances for road traffic actor detection by fine-tuning a CNN on a dataset designed for the same task. Moreover, the databases for road scene understanding differ among themselves with regard to their input data, but they often have at least one color image in common. Therefore, CNNs taking one color image as input could learn and be evaluated on most of these databases. The problem is that basing the detection on a single modality, meaning on a single camera, in the context of autonomous driving, is very unsafe. A single color camera may be sufficient for road traffic actors detection in ideal conditions (bright sun on a clear road), but in real world environment, it is less suitable for nighttime, bad weather conditions (e.g. rain, dazzling sun, snow, fog), or locations that are more difficult to analyze (crowded urban environment, highway with fast vehicles, countryside road without line markings). Moreover, in case of sensor breakdown, these methods become completely obsolete, which can cause fatal accidents.

In view of these different issues, it is better to use multimodal CNNs for road traffic actors detection. Choosing the multimodality allows us to take advantage of the redundant information underlying in the data, and benefit from the rich and diverse knowledge of the surroundings providing by the different sensors, allowing to improve CNN accuracy and robustness. In the literature, the majority of authors using multimodality for road scene understanding primarily aim to improve CNNs performances, usually combining two modalities only. However, to have the most information in the input data in order to minimize the consequences in the event of a sensor breakdown, it is preferable to feed the CNN with as much diverse data as possible, coming from sensors of different types.

Unfortunately, this implies having slower CNNs with large architectures, requiring more learning time. A trade-off between input data size and network size must be found, involving to a judicious choice of input data as well as data fusion strategies.

We introduce in this chapter several multimodal networks in this perspective, seeking to fuse as best as possible multiple modalities. Our objective being the improvement of CNN robustness in case of sensor failure, we seek to fuse the maximum of modalities while avoiding too complex CNNs, slow and therefore useless for real-time object detection. We will also analyze the impact of each modality in multimodal CNN input data, in order to better understand what they can bring to object detection.

This chapter is organized as follows. Section 2.2 introduces our first multimodal CNN for road traffic actors detection, Multimodal Retina, designed for taking as input as many modalities as desired without exploding its number of parameters. In section 2.3, we present two multimodal approaches performing early and middle fusion : Stacked Fusion Double RetinaNet and Gated Fusion Double RetinaNet. Finally, section 2.4 presents the performances of our multimodal approaches comparing to the state-of-the-art algorithms on KITTI Benchmark Object Detection.

2.2 Multimodal RetinaNet

The architecture of a multimodal CNN is often not very scalable. Usually, adding a modality in input data implies to change its architecture and considerably increase the number of its parameters, requiring more processing power. Therefore, these restrictions cause these networks to take a limited number of input modalities. To tackle this problem, we designed a multimodal mid-fusion CNN for 2D object detection : Multimodal RetinaNet (or MM-Retina). With this proposed network, our primary intention was to create a unique CNN for the fusion of at least 2 modalities, taking as input as many modalities as desired without exploding its number of parameters.

In this section, Multimodal RetinaNet architecture is first detailed. Then, we introduce our input data processing, with a more in-depth presentation of the KITTI [37] dataset, as well as our methodology for modalities extraction. Next, our experimental setup and our evaluation metrics used are explained. Finally, we evaluate our proposed CNN with different modalities configuration and we analyze the impact of each input data modality on MM-Retina performances.

2.2.1 Network architecture

MM-Retina architecture is based on RetinaNet [62], which offers a good compromise between computation speed up and high object detection accuracy as related in 1.3.1. The main objective through this network is to mutualize the tasks realized before the data fusion, usually performed separately for the input modalities of a mid-fusion network. For this purpose, a common backbone is used for all the modalities, since it has a role of image feature extractor. Therefore, each modality is represented as a 3-channel image in order to be processed by the same backbone. We opted for mid-fusion into MM-Retina, faster and requiring less power processing than late fusion and more efficient than early fusion, as related in section 1.3.2.

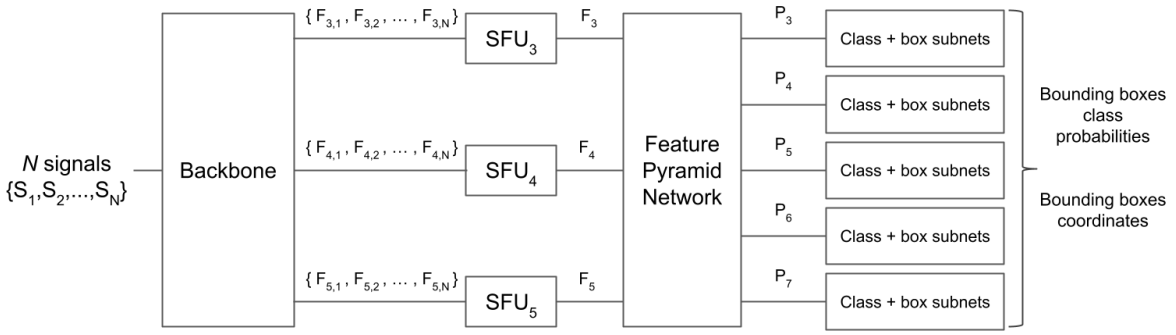


Figure 2.1 – Multimodal RetinaNet architecture

Figure 2.1 shows MM-Retina architecture. Considering N modalities as input, the proposed network is first composed of a common backbone for feature extraction of the N modalities. For each modality S_k, with $k \in \{1:N\}$, MM-Retina backbone provides, from its 3rd, 4th and 5th blocks, 3 feature maps called respectively $F_{3,k}$, $F_{4,k}$ and $F_{5,k}$. Stack fusion is performed on these feature maps via 3 Stack Fusion Units (SFU). With $i \in \{3,4,5\}$ and for each modality S_k, each fusion unit SFU_i takes as input all the feature maps $F_{i,k}$, and outputs F_i feature map of the same size as the previous ones.

Stack Fusion Unit architecture is shown in figure 2.2. In this figure, ® denotes ReLU activation function and © denotes concatenation. It is first composed of a concatenation layer, taking $F_{i,k}$ feature maps as input and outputting a fused feature map $F_{i,fused}$. Following this concatenation layer, a convolution layer with 1×1 kernels $(w_{i,reduce}, b_{i,reduce})$ reduces the number of features of the fused feature maps, with a Rectified Linear Unit (ReLU) activation function, outputting F_i. The operations of Stack Fusion Unit are summarized in the following equations :

$$\begin{aligned} F_{i,fused} &= \bigodot_{k=1}^{N} F_{i,k} \\ F_i &= ReLU(w_{i,reduce} * F_{i,fused} + b_{i,reduce}) \end{aligned} \tag{2.1}$$

where

— $\bigodot$: concatenation

— $F_{i,k}$, $F_{i,fused}$, F_i : feature maps

— $w_{i,reduce}$: kernel weights

— $b_{i,reduce}$: kernel biases

Next, a Feature Pyramid Network extracts a multi-scale feature pyramid from F_3, F_4 and F_5 feature maps. Then, a first subnetwork classifies backbone outputs by predicting the probability of object presence at each spatial position, for each defined anchor and for each object class. Finally, in parallel to the classification subnetwork, a second subnetwork performs convolution bounding box regression for predicting the object location with respect to anchor box if an object is detected. The classification subnetwork is regulated by a focal loss function, which is a scaled cross entropy loss function that helps the learning to focus on interesting examples which are difficult to classify before all easy

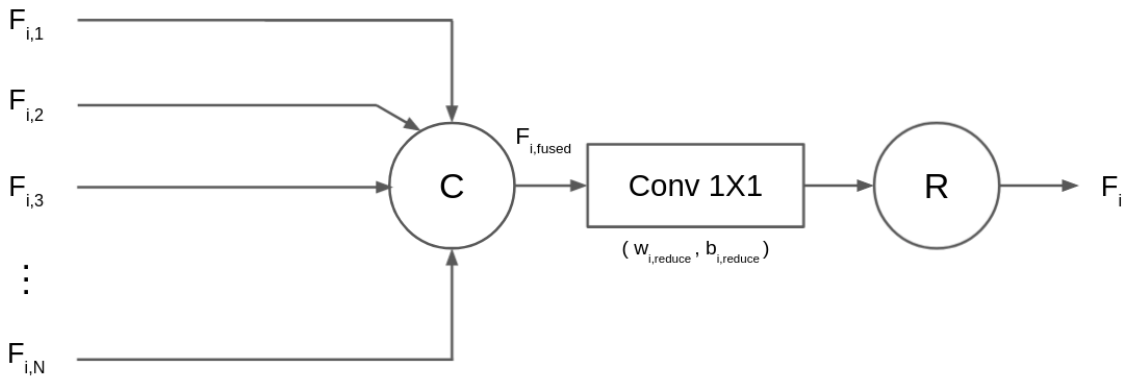


Figure 2.2 – Stack Fusion Unit architecture

ones. The regression subnetwork is regulated by as Smooth L1 loss. Therefore, as for RetinaNet network, MM-Retina is regulated by the sum of these two loss functions.

The main advantage of Multimodal RetinaNet is its size when increasing its number of input modalities. Usually, adding an input modality requires a restructuring of the neural network architecture, sometimes leading to a considerable increase of its number of parameters. In the case where a neural network uses middle-fusion, its sub-networks for feature extraction and for feature map fusion will increase its size. For its feature extraction, the addition of a modality leads to the addition of an extra backbone for this new input. This backbone, in the context of a simple network without fusion, represents a considerable part of the number of parameters of the whole network. If we want to take a maximum of modalities as input to ensure a maximum robustness of the neural network, the addition of several backbones leads to an explosion of the number of parameters, making the network difficult to train because it requires a lot of computing resources, and with a considerably increased computation speed.

The architecture of MM-Retina, thanks to its common backbone for all the modalities, requires fewer parameters and allows us to have more modalities as input. Its Stacked Fusion Units easily adapt to the number of input modalities. Concerning their architecture, SFU concatenation layer are not limited to a specific number of input modalities and do not contain any parameters. However, their 1×1 convolution layers for feature reduction, $(w_{i,reduce}, b_{i,reduce})$ with $i \in \{3,4,5\}$, increase when adding a new modality. Considering N input modalities and $F_{i,k}$ feature maps of size $A_i \times B_i \times D_i$, the concatenation output $F_{i,fused}$ feature map will have a size of $A_i \times B_i \times (N \times D_i)$. SFU convolution layer will therefore output the feature map F_i with a size similar to the initial feature maps, $A_i \times B_i \times D_i$. Thus, each SFU convolution layer have $N \times D_i^2$ parameters. Therefore, adding a new modality in input data increase the size of MM-Retina with $\sum_{i=3}^{5} D_i^2$ parameters, with the values D_i depending on the backbone used.

Backbone	RetinaNet Size	Backbone Size	D_3, D_4, D_5	MM-Retina Size with 2 modalities		MM-Retina Size with 4 modalities	
				Separate Backbones	Common Backbone	Separate Backbones	Common Backbone
ResNet18 [46]	20.7 M	11.2 M (54 %)	128, 256, 512	32.5 M (+ 57 %)	21.4 M (+ 3 %)	55.6 M (+ 169 %)	22.0 M (+ 7 %)
ResNet34 [46]	30.8 M	21.3 M (69 %)	128, 256, 512	52.8 M (+ 71 %)	31.5 M (+ 2 %)	96.0 M (+ 212 %)	32.2 M (+ 4 %)
ResNet50 [46]	37.2 M	23.5 M (63 %)	512, 1024, 2048	71.8 M (+ 93 %)	48.2 M (+ 30 %)	129.8 M (+ 249 %)	59.3 M (+ 59 %)
ResNet101 [46]	56.2 M	42.5 M (76 %)	512, 1024, 2048	109.7 M (+ 95 %)	67.2 M (+ 20 %)	205.7 M (+ 266 %)	78.2 M (+ 39 %)
ResNet152 [46]	71.9 M	58.1 M (81 %)	512, 1024, 2048	141.0 M (+ 96 %)	82.9 M (+ 15 %)	268.3 M (+ 273 %)	93.9 M (+ 31 %)
VGG16 [95]	24.3 M	14.7 M (61 %)	256, 512, 512	40.2 M (+ 65 %)	25.5 M (+ 5 %)	70.8 M (+ 191 %)	26.7 M (+ 9 %)
VGG19 [95]	29.6 M	20.0 M (68 %)	256, 512, 512	50.8 M (+ 72 %)	30.8 M (+ 4 %)	92.1 M (+ 211 %)	32.0 M (+ 8 %)
DenseNet-121 [49]	18.1 M	7.0 M (39 %)	512, 1024, 1024	29.7 M (+ 65 %)	22.8 M (+ 26 %)	48.4 M (+ 168 %)	27.5 M (+ 52 %)
DenseNet-169 [49]	25.3 M	12.5 M (49 %)	512, 1280, 1664	47.1 M (+ 86 %)	34.6 M (+ 37 %)	81.4 M (+ 222 %)	44.0 M (+ 74 %)
DenseNet-201 [49]	31.7 M	18.1 M (57 %)	512, 1792, 1920	64.1 M (+ 102 %)	46.0 M (+ 45 %)	114.6 M (+ 262 %)	60.3 M (+ 90 %)
EfficientNet-B0 [104]	15.6 M	4.0 M (26 %)	240, 672, 1280	23.9 M (+ 53 %)	19.9 M (+ 28 %)	36.2 M (+ 132 %)	24.2 M (+ 55 %)
EfficientNet-B1 [104]	18.1 M	6.5 M (36 %)	240, 672, 1280	28.9 M (+ 57 %)	22.4 M (+ 24 %)	46.2 M (+ 155 %)	26.7 M (+ 47 %)
EfficientNet-B2 [104]	19.7 M	7.7 M (39 %)	288, 720, 1408	32.5 M (+ 65 %)	24.8 M (+ 26 %)	53.1 M (+ 170 %)	30.0 M (+ 53 %)
EfficientNet-B3 [104]	23.0 M	10.7 M (47 %)	288, 816, 1536	39.9 M (+ 74 %)	29.2 M (+ 27 %)	67.5 M (+ 194 %)	35.4 M (+ 54 %)
EfficientNet-B4 [104]	30.6 M	17.5 M (57 %)	336, 960, 1792	56.6 M (+ 85 %)	39.0 M (+ 28 %)	100.2 M (+ 228 %)	47.5 M (+ 56 %)
EfficientNet-B5 [104]	42.0 M	28.3 M (67 %)	384, 1056, 2048	81.3 M (+ 93 %)	53.0 M (+ 26 %)	148.9 M (+ 254 %)	63.9 M (+ 52 %)
EfficientNet-B6 [104]	55.1 M	40.7 M (74 %)	432, 1200, 2304	109.7 M (+ 99 %)	69.0 M (+ 25 %)	205.1 M (+ 272 %)	82.9 M (+ 50 %)
EfficientNet-B7 [104]	78.9 M	63.8 M (81 %)	480, 1344, 2560	159.9 M (+ 103 %)	96.1 M (+ 22 %)	304.6 M (+ 286 %)	113.3 M (+ 44 %)

Table 2.1 – Number of parameters of RetinaNet and Multimodal Retina (in millions) depending on its number of modalities input, for several CNN as backbones.

To show the size reduction of MM-Retina with the common backbone, we compare it with a similar network but using one backbone per input modality. We use the different versions of ResNet [46], VGG [95], DenseNet [49] and EfficientNet [104] backbones. Table 2.1 shows the number of parameters of MM-Retina depending on the backbone used, the number of modalities in input data and the use of a common backbone for feature extraction. First, we can see that the backbone represents an important part of the number of parameters of a simple RetinaNet, ranging from 26 % for EfficientNet-B0, to 81 % for ResNet-152. With 2 input modalities, the size of MM-Retina with a common backbone is much smaller than with separate backbones, with a maximum size increase of 45 % for DenseNet-201. These differences between these two configurations are even more visible with 4 input modalities, where the number of parameters is divided by 3 with a common backbone for ResNet152 or EfficientNet-B7 compared to the same network with separate backbones. Finally, the size increases of MM-Retina are due to the depths D_3, D_4 and D_5 of the feature maps F_3, F_4 and F_5: the larger they are, the larger the size increase of the networks will be. For the Resnet18, ResNet34, VGG16 or VGG19 backbones, their D_i depths result in a size increase below 10 % for an MM-Retina with 4 input modalities compared to a simple RetinaNet. This allows us to take many modalities as input and thus ensure maximum detection robustness in case of sensor failure.

2.2.2 Input data processing

KITTI 2D Object Detection Dataset

The KITTI Vision Benchmark Suite [37] was released in 2012 and remains nowadays a reference dataset for perception for autonomous driving. It consists of multiple benchmarks for multiple challenges : depth estimation from stereo, odometry estimation, 2D and 3D object detection, optical flow estimation, depth completion, road and lane detection, object orientation estimation among others. It was one of the first dataset providing real-world data recorded from diverse sensors, and their benchmarks are still relevant for recent work on real-world computer vision. The data was recorded by driving around Karlsruhe city (Germany), in rural areas and on highways, in good weather conditions (sunny daytime) and from a sensor system consisting of two high-resolution stereo camera systems (grayscale and color), a 64-beam LiDAR and a GPS/IMU localization unit, as shown in figure 2.3.

KITTI 2D Object Detection Benchmark is one of the first benchmark released with KITTI dataset in 2012. This benchmark is composed of 14,999 recording scenes (7481 for training and 7518 for test), each of them including 2 color images from stereo color cameras, 2 grayscale images from stereo grayscale cameras, LiDAR point cloud, and camera calibration matrices. Moreover, the 3 temporally preceding frames from color and grayscale cameras are available. 80,256 objects are labeled on training images from 8 classes : *Car*, *Pedestrian*, *Cyclist*, *Truck*, *Van*, *Tram*, *Person Sitting* and *Misc* (for Miscellaneous, designating other road traffic objects like Trailers, Motorcycle or Segways). Annotations on areas to ignore are also given in a ninth class called *DontCare*. For each labeled object, coordinates of 2D and 3D bounding box, rotation and orientation angles as well as truncation percentage and occlusion state are provided. These information allows to class these objects in 3 different classes of difficulty :

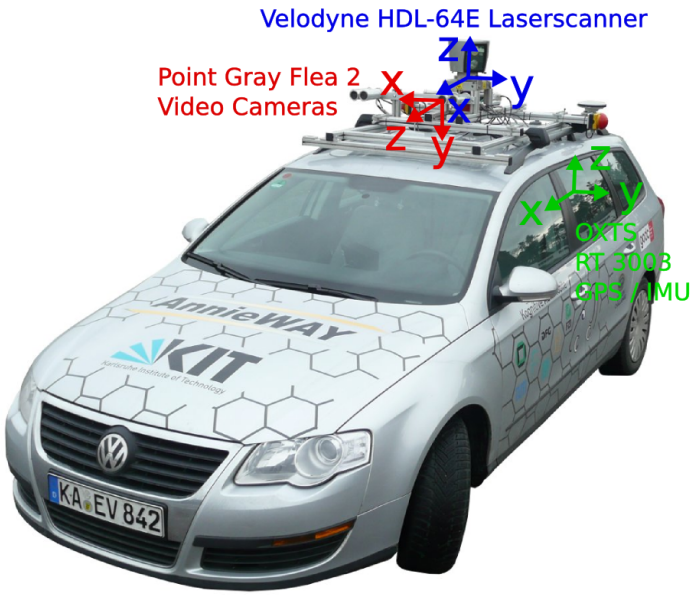


Figure 2.3 – KITTI's recording platform

— **Easy** : Objects with a minimum bounding box height of 40 pixels, fully visible and with a maximum of truncation of 15 %.

— **Moderate** : Objects with a minimum bounding box height of 25 pixels, partially occluded and with a maximum of truncation of 30 %.

— **Hard** : Objects with a minimum bounding box height of 25 pixels, hardly visible and with a maximum of truncation of 50 %.

KITTI dataset advantages are numerous. First, for each recorded scene, we have a lot of diverse data, offering many possibilities for multimodality experimentations. Then, even if KITTI 2D Object detection Benchmark was released in 2012, which makes it one of the oldest benchmark for autonomous driving, many algorithms are still evaluated on it nowadays. Furthermore, getting started with this dataset is relatively easy. Finally, various challenges are available using the same data, or relatively similar data, allowing researchers to easily evaluate their algorithms or methods for other tasks.

Modalities extraction

Starting from KITTI 2D Object Detection Benchmark, we were able to extract 4 different modalities from these data. We used image from left color camera as the RGB image reference, as these images were used for labeling. The other modalities with their different extraction methods used are presented in this section. For each modality, we used classic image processing algorithms that do not require a lot of processing power. We decided to use modalities as processed images for 2D object detection, since their format is well suitable for CNN.

Figure 2.4 – Examples of DP modality with : a) RGB image from left camera ; b) RGB image from right camera ; c) Depth map from the stereo images, extracted with SGBM algorithm.

Depth from stereovision From stereo color cameras outputs, we computed a depth map we called **DP**. This representation is a 1-channel image of the same size as the reference image. Each pixel in this map contains information relating to the distance of the surfaces of scene objects from the reference camera (here the left color camera). Matching algorithm are needed for depth map creation, in order to find correspondences in both images and therefore estimate their distance from the stereo cameras.

In order to create dense depth map from stereo images, we used one of the most popular algorithm for this task, Semi-Global Block Matching algorithm (SGBM), which is a variant of Semi-Global Matching algorithm (SGM) proposed by Hirschmuller [47]. This algorithm has 3 major modules: Matching Cost Calculation, Directional Cost Calculation and Post-processing. In the first one, Center-Symmetric Census Transform (CSCT) [97] is applied on left and right image, using a 9-by-7 pixel sliding window, followed by Hamming distance computing between both CSCT outputs. Then, Directional Cost Estimation module rectifies ambiguous matching costs in first module output by adding directional constraints penalizing changes of neighboring disparities. Finally, minimum cost index calculation, interpolation and uniqueness function are used in Post-processing module. Examples of DP modality extracted from several samples are shown in Figure 2.4.

Optical flow Having two images from the same camera, taken at a short time interval, we are able to extract optical flow information. We constructed a dense optical flow map for each scene, called **OF**. This 2-channels representation, comparing to sparse optical flow which tracks only some interesting pixels, estimates the flow of all pixels in the reference image. Therefore each pixel contains two values corresponding to the magnitude and the direction of a flow vector. In order to create these maps, matching algorithms are also required, like for depth map creation. However, while stereo algorithms constrain the search space to a horizontal scanline to estimate pixel distance to the reference camera, flow algorithms constrain the search space to a limited window, since object displace-

Figure 2.5 – Examples of OF modality with : a) RGB image from left camera ; b) RGB preceding image from left camera ; c) Optical flow map, from the 2 images, extracted with Farneback algorithm.

ments are usually small when the two images are temporally close. The major difficulty of optical flow extraction in our case is that the camera used is placed on a moving vehicle. It is then necessary to compensate for the movement of the vehicle in order to identify the displacement of the objects only.

One of the most popular algorithm for optical flow estimation is Farneback [35] algorithm. This method, derived from Lucas-Kanade algorithm [67], creates dense optical flow map from two temporal images. First, Farneback algorithm generates an image pyramid, where each level has a lower resolution compared to the previous one, in order to detect larger displacements between two lowest resolution images. Then, comparing to Lucas-Kanade method, which estimates motion from features detected by Shi-Tomasi Corner Detector [93], Farneback method approximates each neighborhood of both frames by quadratic polynomials through expansion transform. Furthermore, by observing how the polynomial transforms under translation, a method to estimate displacement field is defined from polynomial expansion coefficients. Finally, dense optical flow is computed after a series of refinements. Examples of OF modality extracted from several samples are shown in Figure 2.5.

Disparity from LiDAR point cloud KITTI's recorded platform includes a 64-beam LiDAR, producing for each recording scene 3D point clouds with more than one million points per second around the vehicle. We used only images for 2D object detection, thus we projected LiDAR point cloud on left camera plane using sensor calibration matrices provided by KITTI Benchmark. We obtained sparse depth map, as shown in figure 2.6. In this point cloud projected on RGB image, each positive color point corresponds to a distance measurement between object in camera plane and LiDAR sensor. We can see in this representation that top part of sparse depth maps don't have any measurements, since this area is not on LiDAR's detection area. From this sparse map, we opted to make depth completion so as to create a dense map.

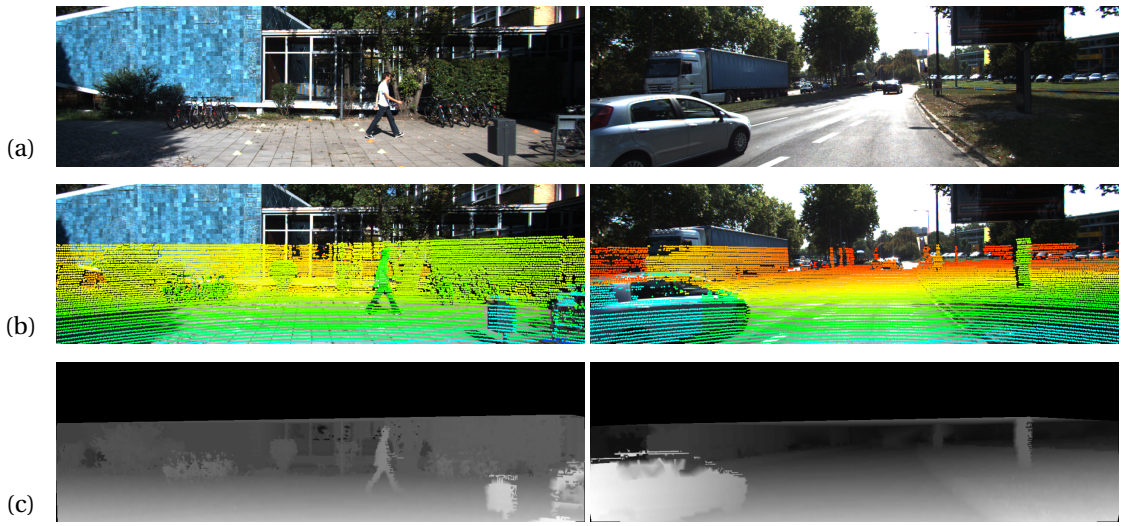


Figure 2.6 – Examples of LD modality with : a) RGB image from left camera ; b) RGB image with projected LiDAR point cloud ; c) Dense depth map, from LiDAR point cloud, constructed with barycentric interpolation.

We used linear barycentric interpolation for depth completion of LiDAR sparse map. First, a triangulation is built with sparse map points in order to divide the convex hull of the grid into triangular tiles. Then, given a point x inside a triangle formed by 3 projected LiDAR points x_1,x_2 and x_3, we can express x as a weighted average of these 3 points, following this formula :

$$x = \sum_{i=1}^{3} \alpha_i x_i \tag{2.2}$$

where $\alpha_i = 0$ and $\sum \alpha_i = 1$. Therefore, the point x is filled depending on all x_i and α_i, with $i \in [1,2,3]$ with this equation :

$$f(x) = \sum_{i=1}^{3} \alpha_i f(x_i) \tag{2.3}$$

Figure 2.6 shows examples of completed dense map with 2D bilinear interpolation. It is to be noticed that we transformed interpolation outputs so as to highlight objects close to the camera (disparity map), whereas LiDAR point clouds usually have pixels with high values where objects are far from the sensor (depth map). This method, quite simple, gives a first estimation of completed disparity map. However, some improvements are possible in order to have smoother disparity map. Furthermore, this method works properly here because KITTI's LiDAR projection has enough points for this, enabling the algorithm to construct small triangular tiles, and therefore fill the empty areas with neighbouring grid data values quite close to it. Is is to be noticed that this algorithm will have more difficulties for other datasets using cheaper LiDAR with less laser beams, and thus less points in sparse LiDAR projected depth map.

2.2.3 Experimental setup

For all the CNNs developed for our experiments, we used samples of KITTI 2D object detection training dataset, with 8 different classes (Car, Van, Truck, Pedestrian, Person Sitting, Cyclist, Tram and Misc). We handled ignored regions proposed by KITTI dataset during training, validation and evaluation, so that they don't penalize the loss when the CNN predicts an object in these areas. We divide the dataset into 5 disjoint folders for a 5-fold cross validation, with four for training and one for validation, resulting in nearly 6000 images for training and 1500 for validation and evaluation. We rescale the input data such that their shorter size is fixed to 600 pixels, therefore the average size for each image of input data is $2000 \times 600 \times 3$ pixels. All CNNs developed are trained with Adam [54] optimizer with a learning rate of 10^{-4} and a batchsize of 16. We also used automatic mixed precision (AMP) [69] in order to accelerate their training. Because of computational resource constraint, we chose ResNet18 [46] for each CNN backbone.

We evaluated Multimodal RetinaNet on different modalities configuration in order to study the impact of each modality. Therefore, for each possible input data configuration including at least one modality and up to all of our 4 modalities (RGB, DP, OF and LD), we trained a network depending on the number of modalities in it : a Multimodal RetinaNet if there is at least two modalities in our input data configuration, a RetinaNet otherwise. For all our models, we used transfer learning for RetinaNet backbone with a ResNet18 pretrained on ImageNet [27] dataset. These networks were fine-tuned on KITTI 2D Object Detection Dataset for 100 epochs and the optimal weights are selected according to the lowest value of the validation loss. We used image horizontal flip as only data augmentation technique.

2.2.4 Evaluation metrics

Once our object detectors are trained, we used specific metrics for evaluating their predictions. When an object detector predict one object, it outputs its class label, its localisation as a bounding box and a score, which is the percentage of confidence that the network has on its prediction. We consider a prediction as a True Positive regarding on a ground truth label when the following conditions are valid:

— Prediction class label corresponds to ground truth class label ;

— Intersection over Union (IoU) of prediction and ground truth bounding boxes, i.e. the division of their intersection area by their union area, is above a given threshold.

In the contrary case, the prediction is classified as a False Positive. After processing all the predictions of a network, all the remaining ground truth label where the network failed to predict an object are considered as False Negatives.

Therefore, during evaluation, we must consider the number of good predictions (True Positives), which we seek to increase, as well as the number of bad predictions (False Positives) and the number of not predicted ground truth objects (False Negatives), which we want to decrease both. For that, Precision and Recall metrics are used. The first one measures the percentage of predictions which are correct among all the predictions. The second one measures the percentage of predictions which are correct among all the ground truth objects. Their equations are the followings :

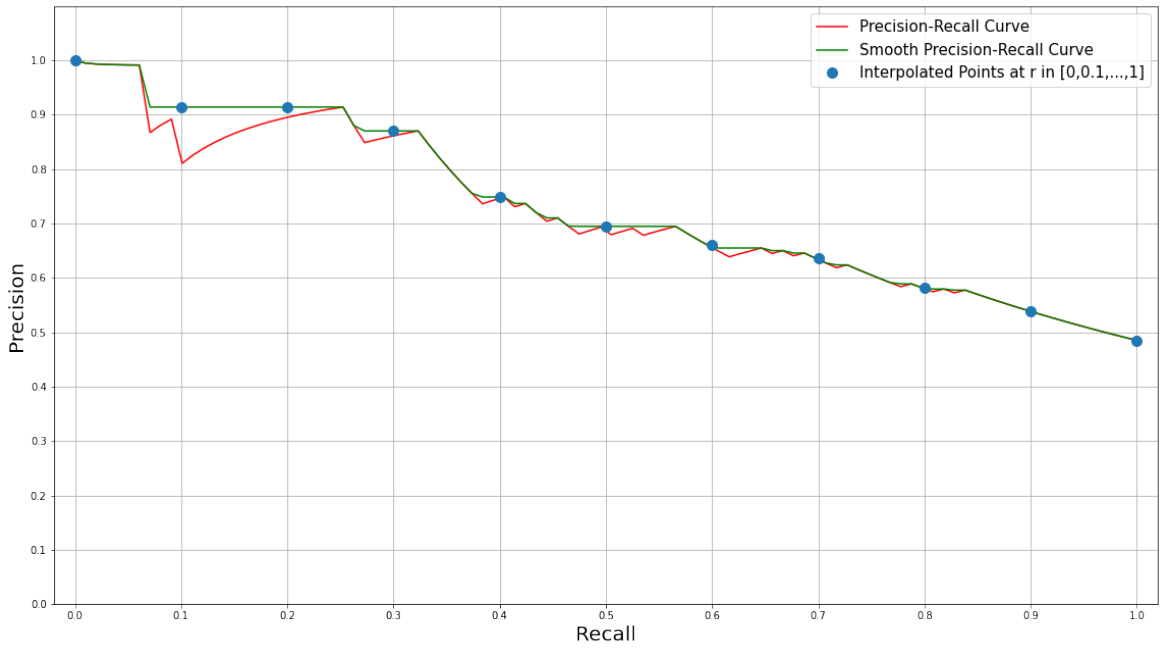


Figure 2.7 – Illustration of precision-recall curve (in red) and its smooth version (in green). Interpolated Average Precision is computed from the interpolated points in blue.

$$precision = \frac{\text{TP}}{\text{TP}+\text{FP}}, \quad recall = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{2.4}$$

where TP, FP and FN stand respectively for True Positives, False Positives and False Negatives.

For each class, all the predictions of this class are sorted decreasingly depending on their score, and then, prediction and recall are computed after each processed prediction. While the precision values vary as we go down the prediction ranking, the recall values increase since a False Positive does not affect this metric. Therefore, we are able to plot a curve of the precision depending on the recall. Figure 2.7 shows an example of this precision-recall curve in red.

Average Precision (*AP*) metric is defined as the area under ther precision-recall curve. This area is calculated on a smooth precision-recall curve, where precision value for each recall value is replaced by the maximum precision value obtained with a greater or equal recall, (green curve in figure 2.7). Their equation is the following :

$$p_{interp}(r) = \max_{\tilde{r} \geq r} p(\tilde{r}) \tag{2.5}$$

where $\tilde{r}$ corresponds to all the recall values between r and 1, and $p(r)$ is the precision value at the corresponding recall r.

Interpolated Average Precision metric is most commonly used since it has been widely popularized by PASCAL VOC challenge [34]. This metric has the following formula :

$$\text{AP} = \frac{1}{11} \sum_{r \in [0,0.1,\ldots,1]} p_{interp}(r) \tag{2.6}$$

where $r \in [0, 0.1, ..., 1]$ corresponds to a set of 11 recall values equally spaced from 0 to 1 (blue points on figure 2.7). *AP* is computed on each class separately.

Finally, Mean Average Precision (*mAP*) metrics is the mean of Average Precision on each class. *mAP* equation is the following :

$$m\text{AP} = \frac{1}{c} \sum_{i=1}^{c} \text{AP}_i \tag{2.7}$$

where c corresponds to the number of classes and AP_i the average precision of the class i.

AP and *mAP* metrics are configurable according to the following two thresholds:

— The IoU threshold, defining the minimum IoU that a prediction must have with a ground truth to be considered as True Positive. The higher this threshold, the closer the bounding boxes of the network predictions must be to those of the ground truth;

— The score threshold, defining the minimum confidence that the network must have in its prediction to be taken into account in the calculation of the metrics. The predictions with a confidence lower than this threshold are then ignored. The higher this threshold, the more confident the neural network must be in its predictions.

For a better comparison of our CNNs, we will use 3 mean average precision and 3 average precision metrics :

— mAP_{50} corresponds to the *mAP* with an IoU threshold of 50 % and a score threshold of 5 % ;

— mAP_{75} corresponds to the *mAP* with an IoU threshold of 75 % and a score threshold of 5 % ;

— *mAP* corresponds to the average *mAP* with an IoU threshold from 50 to 95 % with a step size of 5 %, and a score threshold of 5 % ;

— Car_{75} corresponds to the *AP* of *Car* class with an IoU threshold of 75 % and a score threshold of 5 % ;

— Ped_{50} corresponds to the *AP* of *Pedestrian* class with an IoU threshold of 50 % and a score threshold of 5 % ;

— Cyc_{50} corresponds to the *AP* of *Cyclist* class with an IoU threshold of 50 % and a score threshold of 5 % ;

We use the metrics Car_{75} , Ped_{50} and Cyc_{50} in order to get closer to the metrics used during the evaluation of perception systems on the KITTI 2D Object Detection benchmark. The higher threshold of Car_{75} metric is explained by the important number of *Car* objects in this dataset, as well as their larger size on average than *Pedestrian* and *Cyclist* objects. In the rest of the thesis, we will use the mean average precision by varying IoU and score thresholds to better analyze the detection process of our CNNs.

Number of modalities	Inference time	Input modalities	mAP	mAP $_{50}$	mAP $_{75}$	Car $_{75}$	Ped $_{50}$	Cyc $_{50}$
1	28 ms	RGB	62.25 ± 1.34	87.85 ± 0.79	69.85 ± 2.46	86.71 ± 0.74	82.72 ± 1.49	87.31 ± 2.10
		DP	46.45 ± 2.25	72.54 ± 2.55	50.33 ± 3.13	75.27 ± 2.00	65.47 ± 2.34	66.48 ± 1.28
		OF	22.25 ± 0.88	45.84 ± 1.55	18.82 ± 1.01	37.53 ± 0.70	42.33 ± 1.29	29.33 ± 2.73
		LD	46.01 ± 2.41	74.63 ± 2.60	49.06 ± 3.58	73.12 ± 0.61	75.38 ± 1.33	77.05 ± 3.22
2	35 ms	RGB, DP	63.05 ± 1.92	88.34 ± 1.08	71.11 ± 2.76	86.78 ± 0.51	83.50 ± 1.48	88.37 ± 1.40
		RGB, OF	61.95 ± 1.52	87.45 ± 0.99	69.79 ± 2.36	86.89 ± 0.53	83.43 ± 1.53	87.00 ± 1.78
		RGB, LD	61.80 ± 2.32	88.05 ± 1.23	69.14 ± 3.73	86.20 ± 1.10	84.48 ± 1.38	88.58 ± 2.25
		DP, OF	46.27 ± 1.55	72.42 ± 1.52	50.23 ± 2.38	75.63 ± 0.96	67.55 ± 1.96	66.91 ± 3.01
		DP, LD	51.87 ± 1.32	78.97 ± 1.64	57.73 ± 2.36	79.60 ± 0.37	75.66 ± 0.92	78.60 ± 2.13
		OF, LD	47.50 ± 1.50	76.87 ± 1.38	50.94 ± 2.93	74.74 ± 0.95	76.05 ± 2.25	77.45 ± 3.26
3	42 ms	RGB, DP, OF	61.61 ± 1.86	87.88 ± 0.87	69.42 ± 3.35	86.57 ± 0.83	83.18 ± 1.56	87.45 ± 1.65
		RGB, DP, LD	62.92 ± 1.41	88.28 ± 0.84	70.98 ± 1.85	87.14 ± 0.40	83.69 ± 1.15	88.70 ± 1.71
		RGB, OF, LD	63.18 ± 1.97	88.45 ± 0.99	71.55 ± 3.37	87.33 ± 0.41	84.29 ± 1.18	87.99 ± 2.67
		DP, OF, LD	54.22 ± 2.13	80.85 ± 1.71	60.32 ± 2.87	80.71 ± 0.60	77.54 ± 1.88	80.59 ± 1.27
4	49 ms	RGB, DP, OF, LD	63.06 ± 1.94	88.30 ± 1.33	71.01 ± 3.12	87.07 ± 1.13	84.33 ± 1.54	88.28 ± 2.41

Table 2.2 – Comparison of the detection score of RetinaNet and Multimodal RetinaNet on their validation sets. The rows in red and in blue correspond respectively to networks using RGB modality as input and the other ones. The best detection score for each metric is in bold.

2.2.5 MM-Retina performances depending on modality configuration

We compared our networks performances with mAP_{50} , mAP_{75} , mAP, Car_{75} , Ped_{50} and Cyc_{50} metrics. Table 2.2 shows the scores obtained for each input data configuration. We also recorded for each CNN the inference time of one input data, with modalities with an average size of $2000 \times 600 \times 3$ and with one Nvidia Tesla V100 graphic card.

First, with the scores obtained with RetinaNet networks (the first 4 lines of the table), we can notice that RGB modality, as expected, outperforms the other ones for object detection. Next, RetinaNet using DP or LD modalities are more or less as precise as the other. OF modality shows the worst scores for RetinaNet comparing to the others. This can be explained by the fact that OF modality seeks to highlight the moving objects, which is not the case for all of our objects labeled in our data. Moreover, DP, OF and LD are not suitable for object classification since we can only see the shape and size of objects in these modalities. It is therefore logical to observe these results for RetinaNet.

Multimodal RetinaNet performances demonstrate here that its fusion of modalities does not increase its accuracy. When using RGB modality in its inputs (red lines of the table), Multimodal RetinaNet obtain similar performances as RetinaNet using this modality. Therefore, networks detection is mainly based on RGB modality comparing to the others. Conversely, OF modality seems to bring nothing in terms of performances to Multimodal RetinaNet. Surprisingly, DP and LD modalities combined (9th line of the table) show better mean average precision than the two RetinaNet trained with each of these modalities, whereas these two contained the same information (they are respectively depth and disparity maps) and are therefore redundant.

Concerning CNN speed, MM-Retina is obviously slower than RetinaNet using only one modality as input. Moreover, MM-Retina is getting slower as the number of modalities grows. These inference times are acceptable for real-time object detection, but we must consider that our modalities are already extracted. Therefore, these values are only useful for comparing our network with each other.

Number of modalities	Input modalities	mAP $_{50}$	w/o RGB	w/o DP	w/o OF	w/o LD
2	RGB, DP	88.34	0.19	87.22	-	-
	RGB, OF	87.45	0.04	-	86.67	-
	RGB, LD	88.05	4.46	-	-	80.76
	DP, OF	72.42	-	3.32	65.48	-
	DP, LD	78.97	-	39.65	-	30.71
	OF, LD	76.87	-	-	64.30	0.53
3	RGB, DP, OF	87.88	0.64	87.18	87.52	-
	RGB, DP, LD	88.28	7.38	87.54	-	82.95
	RGB, OF, LD	88.45	6.17	-	87.72	83.44
	DP, OF, LD	80.85	-	45.62	75.92	35.75
4	RGB, DP, OF, LD	88.30	4.55	87.85	88.04	84.73

Table 2.3 – mAP_{50} of Multimodal RetinaNet depending on input data configuration and missing modalities during evaluation. Higher losses comparing to normal conditions (third column) are highlighted with a darker blue.

From these results, we notice that Multimodal Retinanet architecture is not suitable for sensor fusion in terms of accuracy. Indeed, they are all slower than a single RetinaNet, and the best multimodal networks trained obtained similar performances as a single RetinaNet using RGB modality as input. The use of the common backbone for feature extraction does not make a compromise between all the modalities in input, but converge to base his weights to extract the best features for the modality which obtained the more suitable information for object detection (RGB in priority, DP and LD next).

2.2.6 Impact of each modality on CNN performances

Now that we saw the overall performances of Multimodal Retinanet, it is interesting to understand the role of each modality in CNN detection process. For that, we performed an ablation study of MM-Retina by dropping input modalities, replaced by a black 3-channel image of the same size. These experiments allow us to better show the importance of each modality, and better understand how MM-Retina choose their information inside of its fusion units from the feature maps processed by the common backbone. However, these results are not to be taken as they are : if a modality caused a greater loss of accuracy than another one, this does not necessarily mean that this modality is more useful than the other. Nevertheless, we could detect from these results some useful information about MM-Retina detection process.

Table 2.3 shows the performances of MM-Retina depending on its input data configuration and the missing modality during evaluation. As we can see on this table, all our multimodal networks using RGB modality as input get almost zero performance when this modality is missing. Moreover, when DP, OF or LD is missing, those networks accuracy do not decrease much, and most of these losses are ignorable (less than 1 % in terms of accuracy loss). RGB modality takes over all the other ones during network training. Therefore, most of the decisions made by the network are only based on information from RGB modality, which explains the similar performances that obtain all these networks, presented in table 2.2.

Input	mAP		mAP $_{50}$		mAP $_{75}$	
modality	NORMAL	FREEZE	NORMAL	FREEZE	NORMAL	FREEZE
RGB	62.25	59.39	87.85	85.47	69.85	67.20
DP	46.45	34.88	72.54	58.22	50.33	36.07
OF	22.25	12.07	45.84	27.65	18.82	8.67
LD	46.01	34.66	74.63	60.67	49.06	34.96

Table 2.4 – Comparison of RetinaNet performances using RGB, DP, OF or LD modality as input in 2 configurations : *NORMAL*, which corresponds to RetinaNet with a classical learning and *FREEZE* ; which corresponds to Retinanet initialized with a frozen MM-Retina backbone previously trained.

Concerning Multimodal RetinaNet networks that does not use RGB modality as input, we notice that when using OF modality, its absence does not affect much their accuracy. These results support our hypothesis, stated in section 2.2.5, that OF modality has no impact in MM-Retina detection process. Finally, for MM-Retina with DP and LD modalities as input (6th row), both modalities are important since their cutting involves a significant accuracy loss, respectively 42,67 % and 71,53 %. This is also the only input data configuration where all input modalities dramatically decrease MM-Retina performances when they are missing.

2.2.7 Limitations of Multimodal RetinaNet

As noticed in sections 2.2.5 and 2.2.6, Multimodal RetinaNet fusion process does not work as expected. Indeed, except for DP/LD input data configuration, there is a strong imbalance in the use of modality. Therefore, it is necessary to better understand the cause of this imbalance.

First of all, one can wonder about the quality of our modalities extraction, presented in Section 2.2.2, in particular for DP modality with SGBM algorithm, and even more for OF one with Farneback algorithm. Visually, these extractions produced modalities that are quite difficult to analyze. In DP input data, we manage to detect the bigger objects easily and some of the small ones, but there is a lot of noise due to matching errors during the processing. Moreover, on particular scenes, like where there is some vegetation, SGBM struggles to match stereo images points of interest, and thus produce low quality depth maps. Regarding Farneback algorithm, this latter works efficiently when the camera does not move, which is rarely the case in autonomous driving. Therefore, OF input data sometimes succeeds in highlighting moving objects, but they are almost invisible most of the time. It is even surprising that a RetinaNet using only OF modality as input reaches on average a *mAP* of 45,84 % with an IoU of 50 % (see table 2.2).

Next, LD modality format (a 2D grayscale image) is potentially not well suited for our purpose. When projecting LiDAR point cloud and interpolate its points for creating a disparity map, there is a loss of information compared to its initial format. Therefore, it would be interesting to adapt MM-Retina architecture in order to take a point cloud as input, but we would move away from the initial objective which is processing all the modalities by a common backbone.

Then, the use of a common backbone in Multimodal RetinaNet is not satisfactory. Our goal was to create a light architecture and we believed that sharing the same backbone will converge to a state where this feature extractor made a compromise between its modalities as input in order to benefit from all the information available. However, in view of the results previously presented, network learning focus on one modality to the detriment of others. To show this phenomenon, we trained 4 RetinaNet taking respectively RGB, DP, OF and LD modality as input. During their initialisation, we transfer the weights of our trained MM-Retina backbone (using the 4 modalities as input) to RetinaNet one, and freeze it. We then trained each RetinaNet on the same training and validation sets as MM-Retina during 100 epochs with a learning rate of 10^{-5} and Adam optimizer. We finally evaluated these networks on the validation set and compared them to the previous RetinaNet trained (in section 2.2.5). Table 2.4 shows the results obtained with the previous RetinaNet and the new ones with transfer learning and backbone freezing. These performances show that MM-Retina backbone work well with RGB modality, with slightly decreased performances as a classical RetinaNet learning. Nonetheless, MM-Retina backbone can not properly perform with DP, OF and LD modalities, resulting in a significant loss of performances against classical RetinaNet. Therefore, these experiments highlight this problem : the common backbone favors RGB modality over DP, OF and LD.

Moreover, even if the common backbone reduce the number of MM-Retina parameters when adding a modality, during its training, in order to backpropagate the gradient, all the feature maps produced when passing an input must be kept in graphic card memory, increasing the power processing needed. That is the reason why we had to choose ResNet18 as MM-Retina backbone, instead of more efficient one like ResNet50 or VGG16 [95] for example. Moreover, each modality have to pass though the backbone one by one, and causing a slowdown of our CNN. It would be possible to clone the backbone so as each modality passes through one backbone at the same time, but then it would be more interesting to have a separate backbone for each modality.

Finally, 2D object detection task consists in detecting and classifying objects in the 2D input data. On the other hand, DP and LD modalities illustrate the distance between object and the camera, while OF modalities show the movement of objects, which are relevant information, but not the most relevant for detection and even more for classification. These modalities would surely be more useful for other tasks like depth estimation or orientation estimation. Multi-task learning, which is the use of one network for several tasks, is therefore interesting to explore.

2.3 Stacked and Gated Fusion Double Retina

Through the performances obtained with MM-Retina, we see that the use of a common backbone for feature extraction within a multimodal CNN has limitations. Its backbone privileged RGB modality over the others, and the metrics show no improvement over a simple RetinaNet using this modality. However, assigning a unique backbone for each modality is complicated when there are numerous, and it would explode the number of multimodal CNN parameters. To remedy this problem, we opted for the early fusion of the three 1-channel modalities (DP, OF and LD) to create a 3-channel image called DOL. From this, our input modalities are contained in two 3-channel images: RGB and DOL. Their feature extraction then requires 2 backbones for this purpose, which is much more feasible. However, a fusion step of these 2 images is necessary, and for that, the middle fusion remains the best option.

In this section, we present Stacked Fusion Double RetinaNet (SFD-Retina) and Gated Fusion Double RetinaNet (GFD-Retina), two multimodal one-stage CNN for 2D object detection. They are designed to take two 3-channel images as input, and perform middle fusion of them. We first detail their architecture, and their fusion units that differ from each other. We then describe our new methods for modalities extraction, and compare them to those used for DP, OF and LD modalities in MM-Retina input data processing. Finally, we evaluate our two models on KITTI [37] 2D Object Detection dataset, and analyze the impact of each input modality on their performances.

2.3.1 Network architecture

SFD-Retina and GFD-Retina have the same global architecture we called Double RetinaNet, based on RetinaNet [62]. Its architecture, shown in figure 2.8 is first composed of 2 backbones. For each signal $k \in \{1,2\}$, 3rd, 4th and 5th blocks of its corresponding backbone provides 3 feature maps : $F_{3,k}$, $F_{4,k}$ and $F_{5,k}$. Middle fusion is performed on these feature maps via a Fusion Unit in order to obtain 3 fused feature maps : F_3, F_4 and F_5. Like RetinaNet and MM-Retina, Double RetinaNet is then followed by a Feature Pyramid Network, that extracts a multi-scale feature pyramid from the fused feature maps, and two subnetworks for classification and bounding box regression. Double RetinaNet is also regulated by the sum of a focal loss function for its classification subnetwork, and a smooth L1 loss function for its regression subnetwork.

Our two multimodal CNNs differ in their fusion operations, performed with Double RetinaNet Fusion Units. For a given $i \in \{3,4,5\}$ each fusion unit FU_i has a role of combining two feature maps, called $F_{i,1}$ and $F_{i,2}$, produced by the corresponding backbone, to provide an unique feature map called F_i which has the same size as $F_{i,1}$ and $F_{i,2}$. We use 2 versions of fusion unit, for each version of our network architectures, SFD-Retina and GFD-Retina.

SFD-Retina uses Stack Fusion Unit (SFU), which is the same fusion unit we used for MM-Retina, presented in section 2.2.1. GFD-Retina fusion unit is Gated Fusion Unit (GFU), first introduced as GFU v2 in [123]. Figure 2.9 show GFU architecture. In this figure, ® denotes ReLU activation function, © denotes concatenation, and ⊕ denotes element-wise summation. On $F_{i,1}$ and $F_{i,2}$, we apply respectively the two kernels 3×3

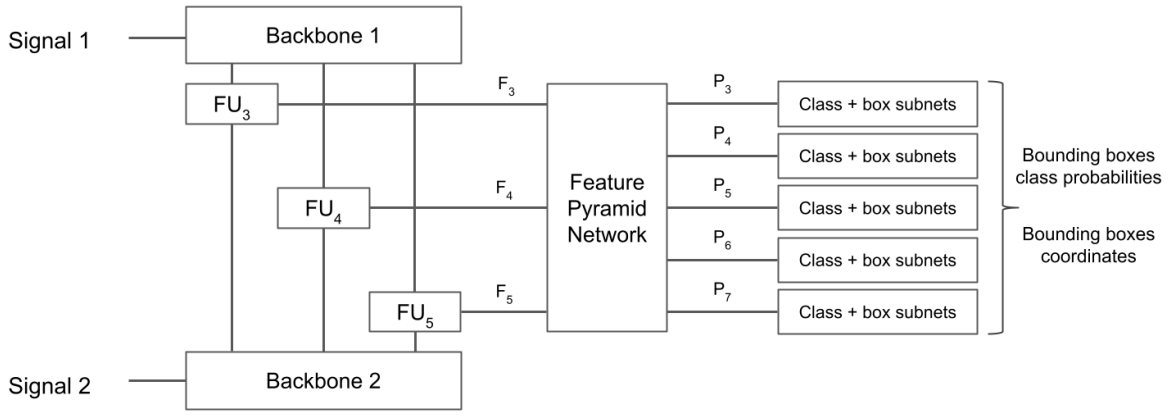


Figure 2.8 – Double RetinaNet Architecture

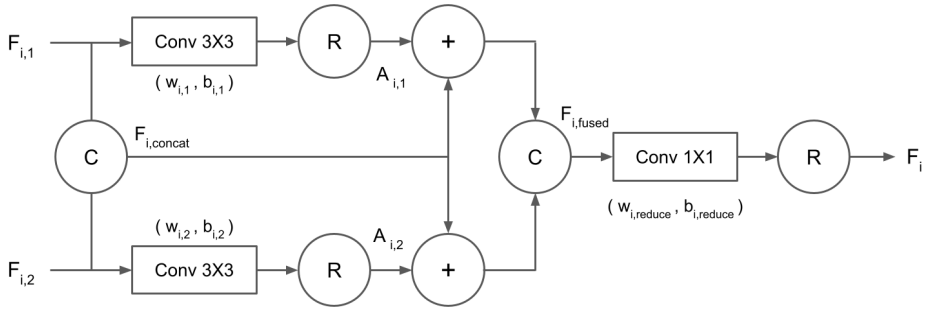


Figure 2.9 – Gated Fusion Unit Architecture

$(w_{i,1},b_{i,1})$ and $(w_{i,2},b_{i,2})$, followed by a Rectified Linear Unit (ReLU) activation function, outputting respectively $\mathrm{A}_{i,1}$ and $\mathrm{A}_{i,2}$. We then perform an element-wise summation on $\mathrm{A}_{i,1}$ and $\mathrm{A}_{i,2}$ with $\mathrm{F}_{i,concat}$, the concatenation of $\mathrm{F}_{i,1}$ and $\mathrm{F}_{i,2}$. The results of these two operations are concatenated in order to create $\mathrm{F}_{i,fused}$, which is then passed to a 1×1 kernel $(w_{i,reduce}, b_{i,reduce})$ to generate the joint feature output F_i. The operations of this GFU are summarized in the following equations :

$$\begin{aligned}
\mathrm{F}_{i,concat} &= \mathrm{F}_{i,1} \odot \mathrm{F}_{i,2} \\
\mathrm{A}_{i,1} &= \mathrm{ReLU}(w_{i,1} * \mathrm{F}_{i,1} + b_{i,2}) \\
\mathrm{A}_{i,2} &= \mathrm{ReLU}(w_{i,2} * \mathrm{F}_{i,2} + b_{i,2}) \\
\mathrm{F}_{i,fused} &= (\mathrm{F}_{i,concat} \oplus \mathrm{A}_{i,1}) \odot (\mathrm{F}_{i,concat} \oplus \mathrm{A}_{i,2}) \\
\mathrm{F}_i &= \mathrm{ReLU}(w_{i,reduce} * \mathrm{F}_{i,fused} + b_{i,reduce})
\end{aligned} \tag{2.8}$$

where

— $\odot$: concatenation

— $\oplus$: element-wise summation

— $\mathrm{F}_{i,1}$, $\mathrm{F}_{i,2}$, $\mathrm{F}_{i,concat}$, $\mathrm{F}_{i,reduce}$, F_i : feature maps

— $\mathrm{A}_{i,1}$,$\mathrm{A}_{i,2}$: ReLU activation outputs

— $w_{i,1}$, $w_{i,2}$, $w_{i,reduce}$: kernel weights

— $b_{i,1}$, $b_{i,2}$, $b_{i,reduce}$: kernel biases

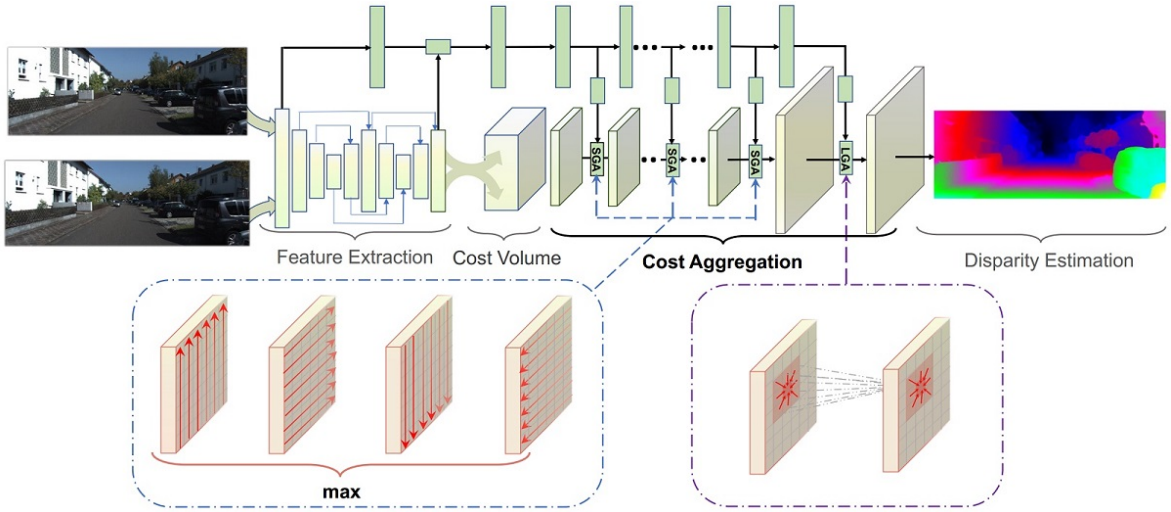


Figure 2.10 – GA-Net Architecture. Figure extracted from [121]

2.3.2 Input data processing

One of our hypotheses for the non satisfactory results obtained with MultiModal RetinaNet, explained in section 2.2.7, is that our input modalities were not well extracted. Therefore, we investigated to use more efficient methods based on deep learning to extract these modalities with a better quality. We then compared the extracted modalities for 2D object detection task with the previous one used with MM-Retina.

Modalities extraction

Depth from stereovision For Multimodal RetinaNet, we used Semi-Global Block Matching for extracting depth map from stereo images. We noticed that this algorithm produced maps with a lot of noise due to matching errors and struggles to match stereo images points of interest on particular scenes or objects (especially where there is vegetation).

Therefore we explored depth map creation from stereo images with Guided Aggregation Net (or GA-Net) [121]. This real-time deep neural network takes 2 stereo images as input for stereo matching and is designed to solve matching cost aggregation problem by using traditional geometric and optimization inside of his architecture. For this, GA-Net includes Semi-Global Aggregation (SGA) and Local Guided Aggregation (LGA) layers in order to replace 3D convolutional layers, generally used for feature extraction in other stereo matching methods based on DNN, allowing it to be more accurate while decreasing both memory and computation costs. GA-Net architecture, shown in figure 2.10 can be divided into 4 parts. First, both images passed through a shared Feature Extraction block, consisting of a stacked hourglass network densely connected by concatenations between different layers. Next, the extracted features from these stereo images are used to form a 4D cost volume, which is fed into a cost aggregation block including SGA and LGA layers. Then, a guidance subnetwork, consisting of 2D convolutions layers, generates the weight matrices for the guided cost aggregations in SGA and LGA layers. Finally, a disparity regression is used to output the disparity estimation.

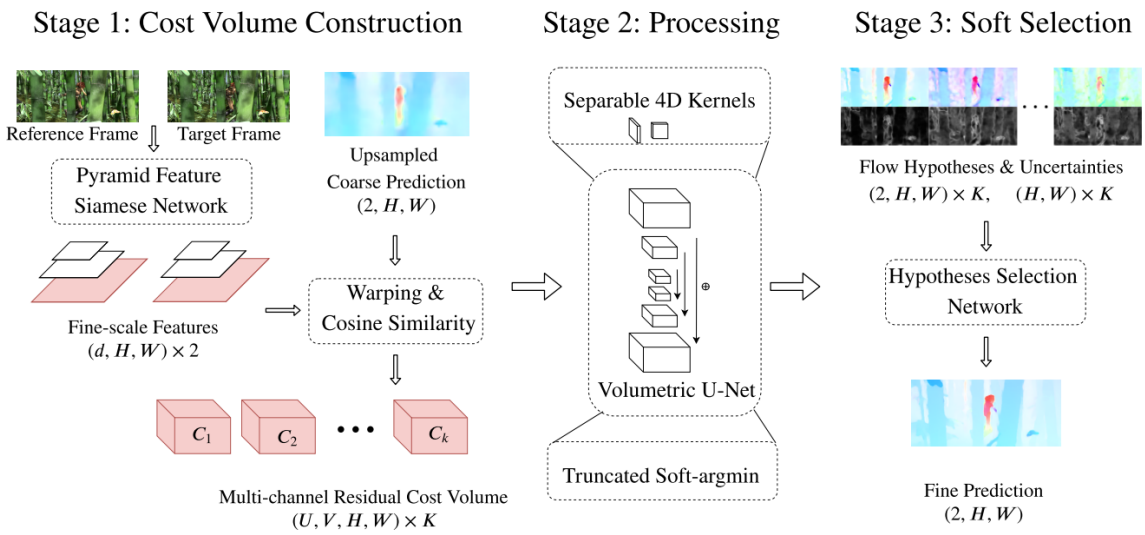


Figure 2.11 – VCN Architecture. Figure extracted from [117]

Examples of depth map generated with SGBM and GA-Net are shown in figure 2.13. We can observe that GA-Net produces by more accurate and smoother depth map than SGBM. This can be explained by the fact that GA-Net is more recent than SGBM (2019 for GA-Net and 2007 for SGBM). Moreover, we used a GA-Net pretrained on KITTI Stereo Benchmark, contrary to SGBM, with parameters optimization, but not designed specifically for road scenes. However, GA-Net requires more processing power, with GPUs, making our all framework for road traffic actors detection slower.

Optical flow We previously used Farneback method for extracting optical flow from two temporal images taken from the same sensor. We remarked that its outputs were really difficult to analyze. This method is especially not suitable for motion compensation. Thus, it is necessary to find an alternative.

Among state-of-the-art methods for optical flow estimation, we opted for Volumetric Correspondence Networks (VCN) [117]. Figure 2.11 shows its architecture. Proposed in 2019, VCN produces fine prediction of optical flow from two frames with a 4D volumetric counterpart of 2D encoder-decoder U-Net architecture [89]. Before passing through this subnet, cost volume construction is made from the two images via pyramid feature siamese network, feature warping of the reference image with upsampled coarse flow and multi-channel cost volume computing. Then, the multi-channel cost volume is filtered with separable 4D convolutions included in the volumetric U-Net subnet, outputting multiple flow hypotheses. Finally, these flow hypotheses are linearly combined considering their uncertainties and the appearance feature, for soft selection, so as to produce 2-channels flow estimation. We build HSV images from 2-channels flow maps for better visualisation, which were then converted to grayscale images.

Figure 2.13 shows examples of dense optical flow map using Farneback method and VCN. When the vehicle is stationary (example in first column), both methods output similar results, highlighting the moving objects. We can nevertheless see that VCN output more precise optical flow map, which then should improve object detection. When im-

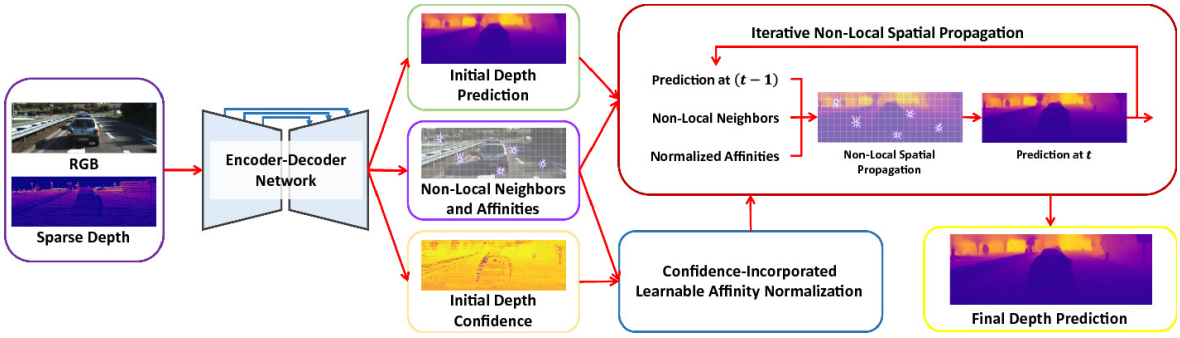


Figure 2.12 – NLSPN Architecture. Figure extracted from [74]

ages are recorded from a moving vehicle (2nd column), VCN outperforms Farneback algorithm, which has much more difficulty to match pixels because of illumination changes and large displacements of objects. Conversely, VCN, which was pretrained on scenes from KITTI Flow Estimation Benchmark, gives high-accurate results, even when the vehicle is in motion. Nevertheless, like GA-Net for depth estimation, VCN requires much more power processing than Farneback method and therefore slows down even more the whole object detection framework.

Depth from LiDAR point cloud We used linear barycentric interpolation for depth completion of LiDAR sparse map for our experiments with Multimodal-RetinaNet. Visually, the depth map obtained were quite good and the results obtained with a RetinaNet taking these maps as input were interesting. However, we pointed out the fact that these maps could be smoother and that this method works especially because KITTI's LiDAR have 64 beams.

In order to remedy these issues, Depth Completion task has been explored, using LiDAR projection and RGB images for depth map reconstruction. Non-Local Spatial Propagation Network (NLSPN) [74] is part of state-of-the-art algorithms. This DNN estimates the neighbors of each pixel beyond the local region, based on color and depth information within a wide area. NLSPN architecture is shown in figure 2.12. Starting from a RGB image and a sparse depth map from LiDAR point cloud projected image plane, an encoder-decoder network predicts an initial dense depth map with its confidence depth map as well as non-local neighbors and corresponding affinities. Following this network, a non-local spatial propagation layer conducted in an iterative manner, with a learnable affinity normalization, aggregates relevant information using the spatially-varying affinities from encoder-decoder outputs.

Figure 2.13 shows examples of completed dense map with 2D bilinear interpolation and NLSPN. Even if 2D bilinear interpolation outputs proper dense disparity maps, NLSPN upgrade maps quality with more accurate and smoother predictions. However, like GA-Net and VCN, NLSPN requires more processing power. Moreover, NLSPN method needs a RGB image for depth completion, which makes framework object detection less robust in case of sensor failure.

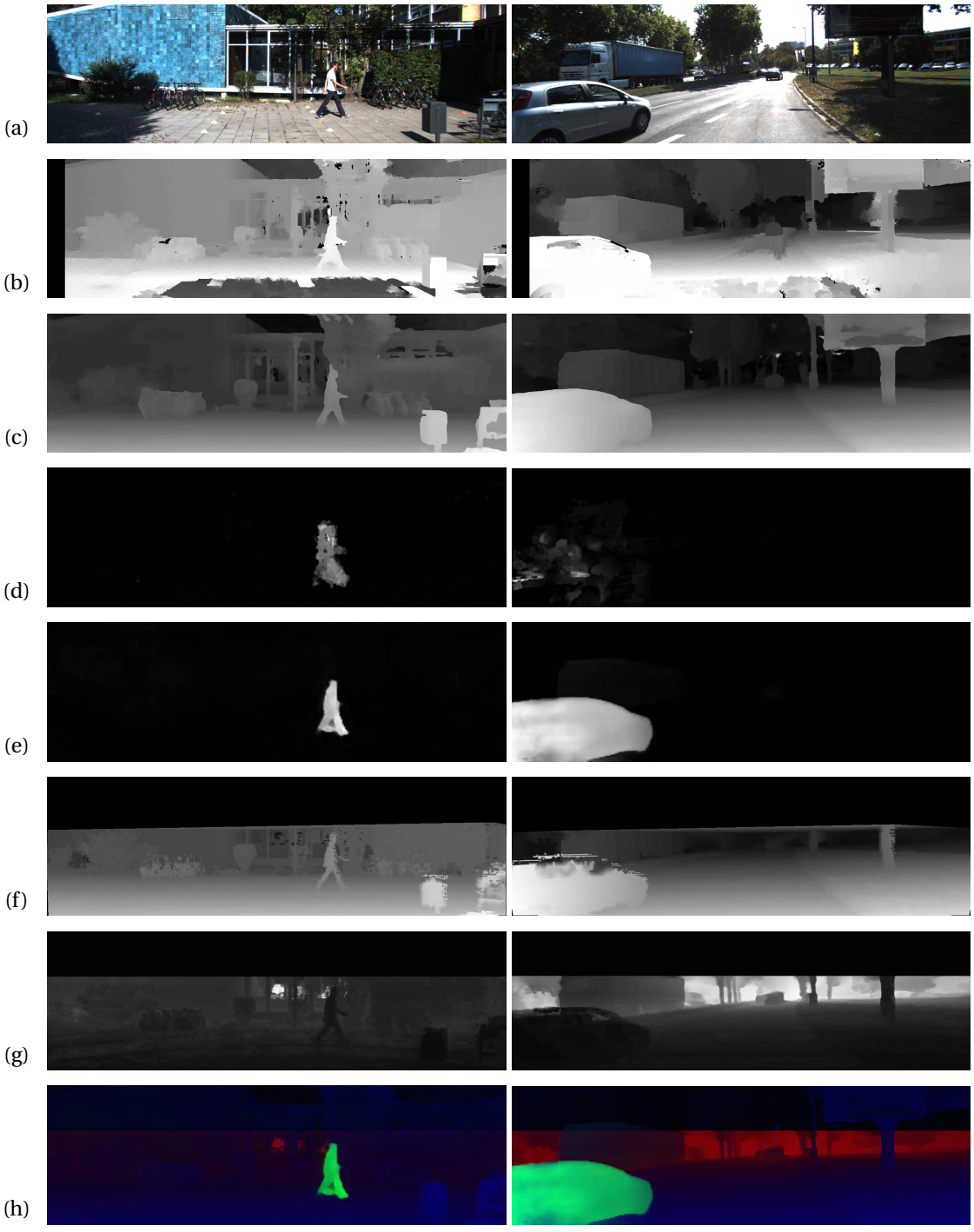


Figure 2.13 – Examples of our extracted modalities with different algorithms : a) RGB ; b) Depth map from stereovision with SGBM ; c) Depth map from stereovision with GA-Net ; d) Optical flow with Farneback ; e) Optical flow with VCN ; f) Disparity map from LiDAR point cloud with linear barycentric interpolation ; g) Depth map from LiDAR point cloud with NLSPN ; h) Stack fused DOL signal

Input Modality / Signal	Algorithms used for modality extraction	mAP	mAP $_{50}$	mAP $_{75}$	Car $_{75}$	Ped $_{50}$	Cyc $_{50}$
DP	SGBM	54.64 ± 2.57	80.75 ± 2.34	62.05 ± 3.54	81.01 ± 1.33	74.10 ± 0.74	77.43 ± 2.25
	GA-Net	56.03 ± 2.67	83.10 ± 1.92	64.07 ± 3.74	81.28 ± 0.95	78.27 ± 1.48	80.62 ± 3.04
OF	Farneback	33.04 ± 2.14	61.32 ± 2.28	32.20 ± 3.41	49.33 ± 1.80	53.90 ± 0.76	49.38 ± 3.25
	VCN	27.10 ± 1.69	50.79 ± 2.66	25.37 ± 1.66	48.27 ± 2.44	51.28 ± 2.08	48.43 ± 2.28
LD	Linear Interpolation	56.20 ± 0.92	84.29 ± 1.02	64.81 ± 1.62	80.65 ± 0.51	83.52 ± 1.36	84.13 ± 2.85
	NLSPN	57.40 ± 2.08	83.23 ± 2.18	65.80 ± 3.45	85.04 ± 0.78	81.46 ± 1.47	84.51 ± 3.08
DOL	GA-Net & VCN & NLSPN	**61.34 ± 0.79**	**87.39 ± 0.91**	**70.31 ± 1.20**	**86.49 ± 0.44**	**84.63 ± 0.69**	**85.93 ± 2.19**

Table 2.5 – Comparison of the detection score of RetinaNet on their validation sets, depending on input modalities. The best detection score for each metric is in bold.

DOL signal We could see that each algorithm we used for modality extraction outputs a grayscale image. With MM-Retina, we used these modalities as 3-channels images even if their information could be contained in only one channel. However, duplicating this channel in order to create a 3-channel image that can be processed by a pretrained backbone involves an increase of memory and power processing for our networks. Thus, instead of using these modalities separately in input data, we opted for stacking these 3 modalities in order to create a unique signal we called DOL (D for Depth, O for Optical flow and L for LiDAR). Figure 2.13 shows examples of DOL signals extracted with GA-Net, VCN and NLSPN algorithms. In these examples, DP, OF and LD modalities are respectively contained in Blue, Green and Red signal channels.

Comparison of the methods used for extraction

We saw, on figure 2.13, examples of results of the algorithms used for modalities extraction. Visually, we noticed that GA-Net for Depth extraction, VCN for Optical flow extraction and NLSPN for LiDAR completion produced smoother and more accurate modalities that respectively SGBM, Farneback algorithm and linear barycentric interpolation. However, these modalities are used for 2D object detection, and having modalities with better quality does not mean that they will help for a more efficient 2D object detection. Therefore, we evaluated these algorithms results for 2D object detection.

For these experiments, we compare depth modalities extracted with SGBM and GA-Net, optical flow modalities extracted with Farneback or VCN, dense disparity/depth map from LiDAR point cloud extracted with linear barycentric interpolation and NLSPN, and DOL signal, which is the stack fusion of the 3 extracted modalities. All these modalities are as 3-channel images. For each modality, we trained 5 RetinaNet, for 5-fold cross validation, with a ResNet50 backbone, during 100 epochs. We used Adam Optimizer with a learning rate of 10^{-4}.

Table 2.5 shows the performances of RetinaNet depending on its input modality or signal and the algorithms used to extract them. We notice that GA-Net and NLSPN algorithms produce input modalities that get better performances in 2D object detection than those extracted respectively by SGBM and linear barycentric interpolation. However, in view of their standard deviation for each metric, their improvements are not significant and therefore, we can not affirm that GA-Net and NLSPN must be definitely chosen instead of SGBM and linear barycentric interpolation. Concerning optical flow modality, RetinaNet obtains significantly better performances using Farneback algorithm rather

Data augmentation configuration	Model	Inputs	Inference time	mAP	mAP $_{50}$	mAP $_{75}$	Car $_{75}$	Ped $_{50}$	Cyc $_{50}$
DA1	RetinaNet	RGB	53 ms	66.85 ± 1.36	89.80 ± 0.68	75.67 ± 1.89	89.70 ± 0.53	85.88 ± 0.78	88.30 ± 1.95
	RetinaNet	DOL	53 ms	55.70 ± 1.70	80.78 ± 1.70	62.58 ± 2.30	84.22 ± 0.58	78.64 ± 1.38	78.88 ± 1.87
	SFD-Retina	RGB+DOL	70 ms	66.96 ± 1.42	89.92 ± 0.87	75.93 ± 1.86	89.96 ± 0.70	85.90 ± 0.75	89.48 ± 1.99
	GFD-Retina	RGB+DOL	135 ms	66.55 ± 1.00	89.37 ± 1.00	74.87 ± 1.53	89.49 ± 0.38	85.88 ± 0.50	88.11 ± 2.65
DA2	RetinaNet	RGB	53 ms	68.21 ± 0.91	**92.93 ± 0.71**	79.48 ± 0.77	90.74 ± 0.53	88.16 ± 0.78	91.55 ± 1.58
	RetinaNet	DOL	53 ms	62.16 ± 1.39	87.08 ± 0.97	71.54 ± 1.85	87.79 ± 0.48	84.11 ± 1.20	86.53 ± 2.90
	SFD-Retina	RGB+DOL	70 ms	69.54 ± 1.23	92.73 ± 0.96	**81.54 ± 1.04**	91.60 ± 0.35	88.80 ± 0.91	**91.83 ± 1.78**
	GFD-Retina	RGB+DOL	135 ms	**69.92 ± 1.05**	92.58 ± 0.57	80.76 ± 1.04	**91.91 ± 0.36**	**89.01 ± 0.46**	91.70 ± 1.42

Table 2.6 – Comparison of the detection score of RetinaNet, SFD-Retina and GFD-Retina on their validation sets, depending on input modalities. The best detection score for each metric is in bold.

than VCN for its input data extraction. These observations go completely against our visual analysis, where we found that VCN outputted flow maps of a much higher quality than Farneback algorithm. Finally, when stack fusing these modalities into a unique DOL signal, RetinaNet gets the best performances against each modality used separately.

For the future experiments, we will use GA-Net, VCN and NLSPN for extracting respectively DP, OF and LD modalities. Moreover, we opted for DOL signal instead of using each modality separately as input data, in view of RetinaNet performances during this experiment. Nonetheless, our previous hypothesis stated in section 2.2.7, suggesting that the bad quality of our input modalities could explain the inefficient fusion in MM-Retina, is rejected.

2.3.3 Experimental setup

For the next experiments, we compare Stacked Fusion Double RetinaNet and Gated Fusion Double RetinaNet to single RetinaNet taking RGB or DOL images as input. We used the same samples of KITTI 2D object detection as those used for experiments in section 2.2.3 for fair comparison with MM-Retina. This dataset is divided into 5 disjoint folders for training, with their images rescaled for having an average size of $600 \times 2000 \times 3$. In view of the large number of parameters of both SFD-Retina and GFD-Retina, we trained all the CNNs in two stages. First, we fine-tune their ResNet50 backbones separately, using a RetinaNet pretrained on COCO dataset [61] for each signal RGB and DOL. These backbones are trained independently during 100 epochs with ADAM optimizer and a learning rate of 10^{-4}. Their optimal weights are selected according to the lowest value of the validation loss. Then, we transfer and freeze these weights fine-tuned to four different CNNs : two single RetinaNet taking RGB and DOL signals as input, and one SFD-Retina and one GFD-Retina taking both signals as input. Finally, we train the CNN top layers, composed of fusion units, FPN, classification and bounding box regression subnetworks, on 100 epochs with a learning rate of 10^{-5} with ADAM optimizer. We select the optimal weights according to the loss value on validation set. This methodology allows our CNNs to converge quickly in a few iterations. Each training process is repeated five times to provide reliable results.

2.3.4 Evaluation of SFD-Retina and GFD-Retina

We used the same metrics for comparing our networks as those used in section 2.2.5: *mAP*, mAP_{50} , mAP_{75} , Car_{75} , Ped_{50} and Cyc_{50} . The performances obtained by RetinaNet, SFD-Retina and GFD-Retina are shown in table 2.6. Inference time represents the prediction time of one input data (RGB and/or DOL images, with an average size of $2000 \times 600 \times 3$) with one Nvidia Tesla V100 graphic card. We trained our CNNs with 2 data augmentation configurations : *DA1* configuration with only image horizontal flip method, and *DA2* configuration with several methods including image translation, image rotation, image shear, image zoom and image horizontal flip.

As seen on this table, we notice that with *DA1* configuration, our multimodal proposals obtain similar performances as RetinaNet taking RGB signal as input. Behind these 3 CNNs, RetinaNet using DOL signal obtained lower performances, demonstrating once again that RGB is crucial for object detection. However, the combination of DP, OF and LD modalities inside this DOL signal allows RetinaNet to reach interesting performances.

Then, with *DA2* configuration, our proposed multimodal neural networks are slightly more efficient than RetinaNet taking RGB signal as input, but these improvements are most of the time not significant regarding metrics standard deviation. We note that our multimodal CNNs are still better than RetinaNet in terms of bounding box accuracy (on the *mAP* and mAP_{75} metrics). We therefore believe that our multimodal networks here are partly limited by the size of KITTI dataset, and that by using a larger one, the performance gap between them and RetinaNet using RGB signal would be more visible.

When comparing SFD-Retina and GFD-Retina, it is difficult to decide between these two which of them is more accurate. Indeed, depending on the metrics, one slightly surpasses the other, but it is never significant. Thus, we can question the contribution of Gated Fusion Units over Stacked Fusion Units. Nonetheless, we observed significant lower loss values on validation set for GFD-Retina comparing to SFD-Retina, even if the metrics do not show the difference of these results. Therefore, one can consider that these metrics could not be fully suitable to decide between these two multimodal CNNs. Moreover, in the same way as for the comparison with RetinaNet, it would be interesting to evaluate these networks on a bigger dataset in order to better see their differences.

Finally, in terms of speed, SFD-Retina and GFD-Retina are obviously slower than a single RetinaNet, since their architecture contain two backbones. Gated Fusion Units, composed of millions of parameters, also increase GFD-Retina inference time. These inference time are reasonable for real time object detection. However, these values were recorded with our input modalities already extracted and with an efficient graphic card, and therefore could be used only for comparison of our networks.

2.3.5 Impact of each modality on CNN performances

The same way as we experimented in section 2.2.6 for Multimodal RetinaNet, we analyze the impact of each modality on our multimodal CNNs. We therefore perform an ablation study of our CNNs on their validation sets when one modality is missing, replaced by a black image of the same size as the modality. Table 2.7 shows the performances of the multimodal CNNs trained with *DA1* configuration, depending on its input data configuration and the missing modality during evaluation.

Model	Input signals	mAP $_{50}$	w/o RGB	w/o DP	w/o OF	w/o LD
RetinaNet	DOL	87.08	-	1.23	77.60	12.02
SFD-Retina	RGB+DOL	92.73	20.12	66.23	91.04	71.16
GFD-Retina	RGB+DOL	92.58	24.57	70.26	91.62	73.54

Table 2.7 – mAP_{50} of our multimodal CNNs depending on missing modalities during evaluation. Higher losses comparing to normal conditions (third column) are highlighted with a darker blue.

For RetinaNet taking DOL signal as input, its performances drop dramatically without DP or LD modality as input, with an important loss of accuracy (respectively 85.86 % and 75.07 %), showing that these modalities are crucial in this configuration. Knowing that these modalities are early fused, it is normal to see important losses when an input modality is missing as RetinaNet backbone process these 3 modalities at the same time for feature extraction. However, without OF modality, RetinaNet loses 9.48 % of accuracy, which is little comparing to the other modalities. Thus, one can consider that OF modality is used as additional information by RetinaNet.

For SFD-Retina and GFD-Retina, when a modality is missing, these networks show similar losses of accuracy. We can observe that RGB modality absence impact the most both networks. However, comparing to Multimodal RetinaNet, those networks can predict several objects, even if their accuracy are rather low. Conversely, OF modality does not impact these networks when missing.

2.4 Evaluation of our contributions on KITTI Benchmark

All our previous experiments were evaluated on validation sets of KITTI benchmark train set, since we did not have the possibility to test all our trained networks on KITTI benchmark test set. Therefore, we select one version of Multimodal RetinaNet taking the 4 modalities as input, SFD-Retina and GFD-Retina, with the lowest loss value on their validation set, and evaluate their object detection performance on KITTI Benchmark test set in order to compare them to state-of-the-art algorithms.

For evaluation in KITTI 2D Object Detection Benchmark, PASCAL criteria is used with detection Average Precision on the 3 main classes, namely *Car*, *Pedestrian* and *Cyclist*. An overlap of 70 % is required for cars, while an overlap of 50 % is required for pedestrians and cyclists. For each class, 3 scores are computed, according to the three levels of difficulty introduced in 2.2.2 : Easy, Moderate and Hard. It is to be noticed that during evaluation in KITTI 2D Object Detection Benchmark, only the objects larger than 25 pixels in height (in the original image) are evaluated. Moreover, it do not count *Van* as false positives for *Car* or *Person_Sitting* as false positive for *Pedestrian* due to their similarity in appearance.

Table 2.8 shows performances of our proposed Multimodal RetinaNet, SFD-Retina and GFD-Retina with several object detectors for *Car*, *Pedestrian* and *Cyclist* detection. We first notice that SFD-Retina and GFD-Retina surpasses Multimodal RetinaNet, which is in agreement with what we observed in section 2.3.4. Moreover, as we have seen on the same section, SFD-Retina and GFD-Retina obtain similar performances on the test set, with a slight advantage for the first one for *Car* and *Cyclist* detection.

Method	Inputs	Car			Pedestrian			Cyclist		
		E	M	H	E	M	H	E	M	H
CLOCs [73]	RGB+LD	96.77	**96.07**	**91.11**	-	-	-	-	-	-
BANet [81]	LD	**98.75**	95.61	90.64	-	-	-	-	-	-
PC-CNN-V2 [30]	RGB+LD	96.06	95.20	89.37	-	-	-	-	-	-
MV3D [18]	RGB+LD	96.47	90.83	78.63	-	-	-	-	-	-
MM_MRFC [25]	RGB+OF+LD	95.54	88.46	78.14	83.79	70.76	64.81	-	-	-
WSSN [43]	RGB+LD	-	-	-	84.91	76.42	71.86	-	-	-
F-PointNet [80]	RGB+LD	95.85	95.17	85.42	**89.83**	**80.13**	**75.05**	86.86	73.16	65.21
RangeIoUDet [60]	LD	95.74	94.61	91.98	-	-	-	**90.43**	**81.67**	**74.90**
PV-RCNN [94]	LD	98.17	94.70	92.04	68.88	58.37	55.38	86.62	80.42	73.64
RRC [87]	RGB	95.68	93.40	87.37	85.98	76.61	71.47	86.81	76.81	66.59
SDP-CRC [116]	RGB	92.06	85.00	71.71	79.22	64.36	59.16	75.63	60.72	53.00
3DOP [17]	RGB+DP	92.96	89.55	79.38	83.17	69.57	63.48	80.52	68.71	61.07
MMRetina **(Ours)**	RGB+DP+OF+LD	93.74	88.63	78.56	79.88	64.91	59.60	69.95	54.90	48.49
SFD-Retina **(Ours)**	RGB+DP+OF+LD	94.85	91.64	82.13	82.48	67.17	62.02	76.35	57.71	50.56
GFD-Retina **(Ours)**	RGB+DP+OF+LD	94.55	91.59	81.80	81.79	67.39	62.24	74.59	56.93	49.91

Table 2.8 – Detection AP (%) of several object detectors using RGB image, Depth from stereo vision (DP), Optical Flow (OF) or LiDAR point cloud (LD) on KITTI object detection benchmark [37] according to three level of difficulty : Easy (E), Moderate (M) and Hard (H).

We obtain satisfying results for both SFD-Retina and GFD-Retina, with performances that approach those from state-of-the-art algorithms. However, we obtain lower accuracy for almost all the categories, especially for the Moderate and Hard levels of difficulty for our 3 classes. Nonetheless, it is to be noticed that contrary to our proposals, most of the algorithms listed in table 2.8 perform also 3D object detection, orientation estimation and/or bird's eye view object detection on the same test, which takes 2D object detection task into the background.

To explain these results, we have several hypotheses. First, for *Pedestrian* and *Cyclist* detection, these objects are smaller and less present than *Car* objects in KITTI Benchmark. Therefore, in the context of multi-class object detection, without an explicit strategy to remedy it, CNNs will rather converge in order to minimize the maximum of errors, without focusing on the class of objects, and thus favoring objects from the most present classes.

Next, we decided to use 4 modalities, which is the most compared to all the other several objects on KITTI detection benchmark. This choice was made for our next work on robustness improvement, but it affects our multimodal CNNs in terms of power processing needed. Thus, we do not have the possibility to opt for the most accurate backbones available, that could have improved our CNNs performances.

Finally, state-of-the-art algorithms which use more than 2 modalities fuse them with an explicit strategy, using them in their best format. In our case, we opt for giving to our networks a better scalability, by using our modalities as grayscale or color images, and by designing our networks to easily add, remove or replace a modality in our input data, especially for Multimodal RetinaNet. It is therefore quite logical to obtain lower performances in our case with this strategy.

2.5 Conclusion

In this chapter, we address 2D object detection task with multimodal CNNs. We first proposed a multimodal neural network, Multimodal RetinaNet, designed to mid-fuse as many modalities as desired without exploding his number of parameters. We then introduced SFD-Retina and GFD-Retina, two multimodal 2D object detectors taking two 3-channel images as input and using different middle fusion strategy to fuse them. We experiment them taking as input 4 modalities (RGB, DP, OF and LD), in a grayscale or color image format, and with early fusion for SFD-Retina and GFD-Retina. These networks obtained satisfying performances on KITTI Benchmark test set, although less successful than state-of-the-art algorithms.

During this chapter, we also analyze the role of each modality of our input data for 2D object detection task inside our network. For our first multimodal CNN, we notice a strong imbalance between RGB modality against the other ones, due to an undapated architecture and extraction methods that visually deteriorate a lot their information. Then, with our two Double RetinaNet, we found that RGB modality was always crucial for object detection, but that our networks could potentially get away with our other modalities. However, our experiments showed that OF modality has no impact in CNN detection process, as its absence does not impact any of our networks performances. Moreover, as presented in 2.3.2, its format does not seem to be suitable to object detection, regarding low performances of RetinaNet using it instead of the other modalities. Thus, we can question its usefulness in our input data configuration. Extracting it with VCN increase processing time and therefore make our proposal CNNs difficult to work in real time. Moreover, this modality is processed from left camera images, including RGB modality. Thus, OF and RGB modalities are linked : if one of them is absent due to left camera breakdown, this second one will disappear to. This modality is therefore not useful for improving neural network robustness. In view of these findings, OF modality is useless in our input data configuration. Nonetheless, we believe that this modality could help for other tasks, like orientation estimation or object tracking. It would be interesting thus to opt for multi-task learning in order to see what this modality could bring to network performances.

With our ablation studies, we noticed that our multimodal CNNs performances drop considerably when a modality is missing (except for OF one). However, with our preliminary experiments in section 2.3.2, we noticed that each modality can ensure a correct detection alone with a CNN. Therefore, when one modality in input data is missing, we know that the information contained in the other modalities available could lead to better performances, but our multimodal CNNs do not manage to detect objects correctly. We can thus conclude that these networks are not robust to sensor breakdown.

Chapter 3

Robustness improvement of multimodal neural networks with missing modalities

Contents

3.1 Introduction

Unexpected events are to be considered on the road. This can be of any nature: bad weather, an uncommon behavior of a road traffic actor, a dazzling sun, an object on the track that can damage the vehicle, among others. Most of these situations can be understood by drivers, who will adapt their driving to avoid an accident. In the ADAS context, a vehicle must also apprehend these complex situations, but the task is not simple. For this, the use of sensors is more than essential in order to analyze the road and identify these risky events. Nonetheless, on top of that, we can add the possible failure of its sensors.

Without an explicit strategy, the analysis of road scenes is strongly affected by sensor failure, as seen in sections 2.2.6 and 2.3.5. Most of the time, when a modality is missing, the performance of our multimodal CNNs drops drastically. This observation is all the more striking as this missing modality is replaced by a black image, without any information, easy to ignore, and that the other modalities are correct. We can see that this rather easy case is not even manageable if a CNN has not been trained for.

To remedy this problem, we can deal with another strategy upstream of the network by detecting potential sensor failures. In knowledge of a failure, the perception system would react differently when the vehicle faces this situation. This process is necessary, but remains insufficient alone. Indeed, in the event that a sensor failure is detected, which decision should the vehicle take? If it has to stop, it cannot do so in the middle of the road because the passengers and other road users will be in danger. In order to direct a vehicle to a safe parking area, sensors and a detection system must be used. If the primary driver has to take over the wheel, there is a period of time, hopefully as short as possible, when the vehicle is out of control. In addition, a fully functional sensor may have its sensing disrupted by external events. From these observations, even when detecting sensor failure, the detection systems should be robust when a modality is missing.

Regarding the robustness of multimodal CNNs, we have seen in section 1.4 that there is unfortunately few works on the subject. Among them, we identify 3 main axes concerning this task: the design of specific architectures of the CNNs to support these degraded conditions, the use of datasets developed for the problem and the techniques of data augmentation to learn to manage these risky situations. The creation of robust CNN architectures is an avenue to explore, but it is necessary in all cases to learn on damaged data. For that, the creation of a dataset is interesting, but complicated in our problem, namely the failure or breakdown of sensors. Getting real data means to damage several sensors, in different ways. We can opt for synthetic data, which is similar to data augmentation techniques. Therefore, we believe that data augmentation is the best option for improving the robustness of multimodal CNNs. These techniques are easily implementable, and adaptable to any type of multimodal neural network, whatever the input data.

In this chapter, we study the case where one or more modalities are missing. We introduce several data augmentation techniques for improving multimodal CNN robustness in case of missing modalities. Our main objective is to obtain better performances in degraded conditions with missing modalities, while maintaining the performances in ideal conditions, when all modalities are correct. Through these techniques, we also seek to remove the dependencies that a multimodal network may have with one or more modalities which, when absent, drastically decrease neural network performances.

This chapter is organized as follows. Sections 3.2, 3.3 and 3.4 respectively introduce our data augmentations methods, namely Random Signal Cut (RSC), Random Channel Cut (RCC) and Random Modality Cut (RMC), for a multimodal network to learn with partially missing input data. Our methods are first described and then experimented with our multimodal CNNs, in ideal conditions as well as in degraded conditions with missing modalities. Finally, we analyze in section 3.5 one of our most robust model with additional metrics to better understand the impact of one or several modalities on its performances.

3.2 Random Signal Cut

In this chapter, we consider that a failed sensor is a sensor that does not produce any signal, although in reality, it is likely to have sensors that are not calibrated or that return noisy signals. These missing signals are replaced by black images of the same size. We showed in sections 2.2.6 and 2.3.5 that most of the time, when one modality is missing in input data, multimodal CNNs performances drops considerably, while the remaining modalities are satisfactory to allow these networks to ensure a safe detection. This implies that in case of sensor breakdown, road object detection is highly less efficient.

The principle of Random Signal Cut is to generate, during CNN training, partial input data from multimodal dataset. Therefore, during sensor fusion, Our method influence the multimodal CNNs to not base their decision entirely on one or several modalities, so that in a critical situation where these modalities are not present, the latter can manage to be satisfied with the remaining modalities for its predictions. RSC is independent of multimodal network architecture and can be applied for any task. We experiment our method on Multimodal RetinaNet (MM-Retina), our proposed multimodal CNN using middle fusion.

3.2.1 Proposed method

The goal of Random Signal Cut is to feed the network during its training with various scenarios of input data configuration with missing signals so as it can handle a correct detection in case of sensor failure. With RSC, input data are considered as several signals, that is to say several images processed independently in the network before fusion of their feature map. The principle of RSC is therefore to transform the input data by replacing randomly one or several signals by empty signals, which are black images of the same size as the originals. However, injecting null signals in training can have consequences on the performance of the CNNs, and it is thus necessary to take care not to have a too important share of them on the training set. On top of that, if we only generate data corresponding to real failure configurations, we can have a strong imbalance between the signals, with some more present in the training set than others. This can lead the CNN to focus only on the ones that are mostly present and ignore the others ones. To address these two issues, Random Signal Cut assign a Cutoff Rate to each signal, between 0 and 100 % corresponding to the percentage of input data signal absence. During CNN training, each input data signal will randomly be either normal or totally null, creating input data with different signals configurations where some of them are physically impossible

(for instance, a configuration where a depth from stereovision signal is available whereas the RGB signal from one of the stereo cameras is cut).

This method can be compared to a Dropout [98] applied not on neurons but on input signals. However, unlike Dropout technique, Random Signal Cut do not rescale input data after signals cutting. Our main idea here is to create potential failure in our sensor system, considering that the unavailability of a signal should not influence the rest of input data. That's why each signal cut is made independently of the other signals. Moreover, with Random Signal Cut, we are able to assign different adapted rates for each signal depending on its characteristics. One can consider the cutoff rate values of one multimodal CNN as hyperparameters having to be optimized.

In order to avoid null data in our network training, we lock the possibility to have all our signals unavailable. This influences the defined cutoff rate of each signal (theoretical rate) from the true proportion of signals set to zero during training (real rate). With N input signals, each with a non-zero cutoff rate, $2^N - 1$ cutoff states can be generated by Random Signal Cut, ranging from the state where all signals are present, to the presence of a single one among the input data. These different states have different probabilities depending on the cutoff rates. We define our N signals $S = \{S_1, S_2, ..., S_N\}$, each of them with a cutoff rate CR_i for $i \in [1, N]$. We pose X, a random variable to describe the cutting of our signals with Random Signal Cut, such as :

$$P(X_i = 0) = CR_i \text{ and } P(X_i = 1) = 1 - CR_i \tag{3.1}$$

Knowing that we have locked the possibility to obtain the configuration where all signals are cut, the probability that X is equal to the cutoff state $Y = \{Y_1, Y_2, ..., Y_N\}$ is :

$$P(X = Y) = \frac{\prod_{i=1}^{N} y_i}{1 - \prod_{i=1}^{N} CR_i} \tag{3.2}$$

with y_i the probability of Y_i defined as :

$$y_i = \begin{cases} CR_i & \text{if } Y_i = 0 \\ 1 - CR_i & \text{if } Y_i = 1 \end{cases} \tag{3.3}$$

Since the configuration where no signal is present is impossible, the cutoff rate CR_i applied to the signal S_i is a theoretical rate, which differs from the real cutoff rate CR'_i, defined as follows:

$$CR'_i = \frac{CR_i - \prod_{k=1}^{N} CR_k}{1 - \prod_{k=1}^{N} CR_k} \tag{3.4}$$

In the case where the same cutoff rate p is assigned for each signal, the random variable X follows a positive binomial distribution and the real cutoff rate p' is defined as :

$$p' = \frac{p - p^N}{1 - p^N} \tag{3.5}$$

This difference between the theoretical and real cutoff rates is even greater when the number of input signals is low. For instance, for 2 input signals, a theoretical rate p of 50 % will give a real rate p' of 33.33 %.

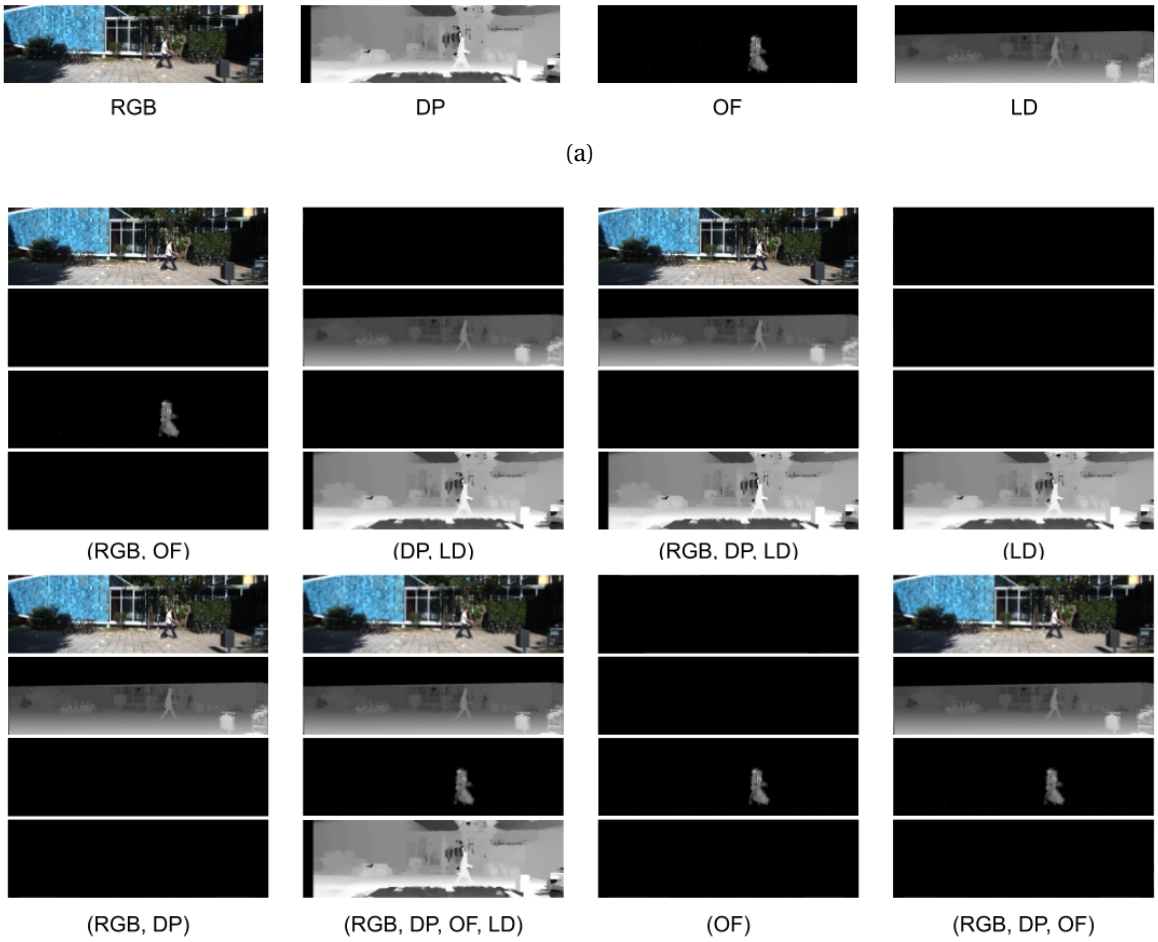


Figure 3.1 – a) Multimodal RetinaNet input data and b) examples of generated samples by Random Signal Cut.

3.2.2 Experimental setup

For this experiment, we evaluate MM-Retina with a cutoff rate ranging from 0% (no data augmentation), to 50%, in 5% steps. Random Signal Cut is applied on MM-Retina input data, consisting of RGB, DP, OF and LD modalities, all considered as signals since their feature extraction are processed independently. Figure 3.1 shows our CNN input signals, and examples of generated samples by Random Signal Cut from this input. Our CNNs are trained for multi-class object detection on KITTI 2D Object Detection Dataset, with 8 different classes and taking into account the proposed areas to be ignored. We divide the dataset into 5 disjoint folders for a 5-fold cross validation and rescale the input data such that their shorter size is fixed to 600 pixels. All our CNNs are first initialized with a ResNet18 backbone, pretrained on ImageNet, and we fine-tune them on 100 epochs with a batchsize of 16, an Adam optimizer, a learning rate of 10^{-4}, and automatic mixed precision. Finally, we select CNN optimal weights according to the lowest value of its validation loss. We used image horizontal flip as only data augmentation technique.

Evaluation configuration	Available modalities			
	RGB	DP	OF	LD
N	✓	✓	✓	✓
A_1	✗	✓	✓	✓
A_2 / C_1	✓	✗	✓	✓
A_3	✓	✓	✗	✓
A_4 / C_2	✓	✓	✓	✗
B_1	✓	✗	✗	✗
B_2	✗	✓	✗	✗
B_3	✗	✗	✓	✗
B_4 / C_4	✗	✗	✗	✓
C_3	✓	✗	✓	✗

Table 3.1 – Available modalities according to evaluation configurations for MM-Retina ablation study (✓: present modality, ✗: missing modality)

3.2.3 Evaluation configurations

To analyze the impact of our data augmentation method on the robustness of Multimodal RetinaNet, it is necessary to evaluate it in degraded configurations where one or more of them are missing. For this, we will perform a study ablation under different degraded configurations with one or more input missing modalities. From the 4 modalities of Multimodal-Retina, we have 15 possible evaluation configurations where at least one modality is present. Among them, we have chosen 10 in order to have a good representation of the robustness performances obtained. We obviously have the configuration where all the modalities are present (noted N). To this one, we added the following degraded configurations, divided into 3 groups:

— Group A : The configurations where 1 modality is missing, noted A_1, A_2, A_3 and A_4, with as missing modality respectively RGB, DP, OF and LD;

— Group B : The configurations where all the modalities except one are missing, noted B_1, B_2, B_3 and B_4, with as remaining modality respectively RGB, DP, OF and LD;

— Group C : The real failure configurations that the sensor system (consisting of 2 stereo cameras and 1 LiDAR) may encounter:

  — C_1 (equivalent to A_2 configuration) : Failure of one of the 2 stereo cameras, and where the DP modality is missing;

  — C_2 (equivalent to A_4 configuration) : Failure of the LiDAR, and where the LD modality is missing;

  — C_3 : Failure of one of the 2 stereo cameras and the LiDAR, and where DP and LD modalities are missing;

  — C_4 (equivalent to B_4 configuration) : Failure of the 2 stereo cameras, and where RGB,DP and OF modalities are missing.

A summary of all the evaluation configurations we use along with their missing modalities is shown in table 3.1.

Cutoff rate	mAP	mAP $_{50}$	mAP $_{75}$	Car $_{75}$	Ped $_{50}$	Cyc $_{50}$
0 %	**63.06 ± 1.94**	88.30 ± 1.33	71.01 ± 3.12	**87.07 ± 1.13**	84.33 ± 1.54	88.28 ± 2.41
5 %	62.84 ± 1.19	**88.39 ± 1.04**	**71.31 ± 2.26**	86.68 ± 0.38	84.66 ± 1.08	88.66 ± 1.77
10 %	62.70 ± 1.18	88.17 ± 0.76	70.10 ± 1.91	86.71 ± 0.80	83.72 ± 1.32	88.38 ± 1.94
15 %	62.62 ± 1.35	88.19 ± 0.84	70.68 ± 2.55	86.41 ± 0.78	84.21 ± 1.32	88.33 ± 1.81
20 %	61.97 ± 1.40	88.17 ± 1.00	69.15 ± 1.79	86.82 ± 0.63	84.14 ± 1.87	88.07 ± 2.15
25 %	61.46 ± 1.56	87.69 ± 1.41	69.03 ± 2.23	86.46 ± 0.69	83.70 ± 1.33	87.85 ± 1.01
30 %	61.21 ± 0.01	87.04 ± 0.34	69.20 ± 0.85	86.28 ± 0.39	84.10 ± 0.11	**89.34 ± 1.07**
35 %	60.96 ± 0.67	86.37 ± 0.80	69.27 ± 0.52	86.40 ± 0.29	**85.28 ± 1.47**	88.45 ± 1.86
40 %	60.93 ± 0.66	86.43 ± 0.45	69.31 ± 0.54	86.28 ± 0.01	84.43 ± 0.88	88.89 ± 0.51
45 %	60.23 ± 0.81	86.08 ± 0.11	67.59 ± 0.75	86.06 ± 0.43	83.67 ± 0.60	88.90 ± 1.55
50 %	59.95 ± 1.57	85.70 ± 1.45	67.45 ± 2.80	85.74 ± 0.36	83.05 ± 1.85	87.05 ± 1.63

Table 3.2 – Performances of MM-Retina depending on cutoff rate applied with Random Signal Cut in normal conditions.

3.2.4 Performances in normal conditions

First, we evaluate our multimodal CNNs with Random Signal Cut under normal conditions on their validation sets, varying the cutoff rate from 0 % to 50 %. We compare networks performances with the mean average precision *mAP*, mAP_{50} and mAP_{75} as well as the average precision Car_{75} , Ped_{50} and Cyc_{50} . Table 3.2 shows CNNs performances obtained depending on their cutoff rate.

When looking at the mean average precision, these metrics significantly decrease as CNNs cutoff rate increases. From this observation, it is therefore important to choose a cutoff rate that is not too high in order to keep these performances. It can be seen that these decreases are slightly larger on the *mAP* and mAP_{75} metrics, compared to mAP_{50} , showing that the cutoff rate partly affects the bounding box accuracy of CNN predictions. Looking in detail at CNNs performances on *Car*, *Pedestrian* and *Cyclist* classes, they are less affected, as they are part of the majority classes of KITTI, i.e. those with the largest number of object instances in the dataset. We therefore think that RSC on normal conditions affects in particular classes with few instances.

3.2.5 Performances in case of missing modalities

We analyze the impact of Random Signal Cut on Multimodal RetinaNet under degraded conditions. We perform an ablation study of our networks on their validation sets in the 10 evaluation configurations cited in section 3.2.3. Table 3.3 shows the performances of the multimodal networks in each configuration, according to the cutoff rate applied via Random Signal Cut.

Cutoff Rate	Failure Configurations (with remaining input modalities)								
	N (RGB,DP,OF,LD)	A_1 (DP,OF,LD)	A_2 / C_1 (RGB,OF,LD)	A_3 (RGB,DP,LD)	A_4 / C_2 (RGB,DP,OF)	C_3 (RGB,OF)	B_1 (RGB)	B_2 (DP)	B_3 (OF)	B_4 / C_4 (LD)
0 %	88.30	4.55	87.85	88.04	84.73	82.74	80.77	0.03	0.00	2.24
5 %	88.39	55.04	87.78	87.81	86.24	84.91	84.27	26.01	2.26	26.62
10 %	88.17	66.81	87.71	87.73	86.36	84.99	84.21	39.64	4.44	38.23
15 %	88.19	69.05	87.78	87.83	87.23	85.99	85.45	46.68	6.63	41.63
20 %	88.17	72.85	87.58	87.90	86.55	85.06	84.18	52.04	9.26	46.35
25 %	87.69	74.97	87.18	87.37	85.92	84.67	84.06	55.07	13.46	51.13
30 %	87.04	72.84	86.63	86.74	85.45	84.12	83.28	54.01	14.44	52.33
35 %	86.37	76.02	86.54	86.27	84.86	84.15	83.30	58.24	17.09	55.38
40 %	86.43	75.83	86.46	86.42	85.59	84.06	82.73	57.74	19.81	56.12
45 %	86.08	76.95	86.50	85.75	84.58	83.56	82.60	60.20	23.84	60.31
50 %	85.70	77.42	85.95	85.83	84.23	83.95	82.72	62.96	28.95	62.50

Table 3.3 – Performances (mAP_{50}) of MM-Retina in degraded conditions, depending on cutoff rate applied with Random Signal Cut. Higher losses comparing to normal conditions (second column) are highlighted with a darker blue.

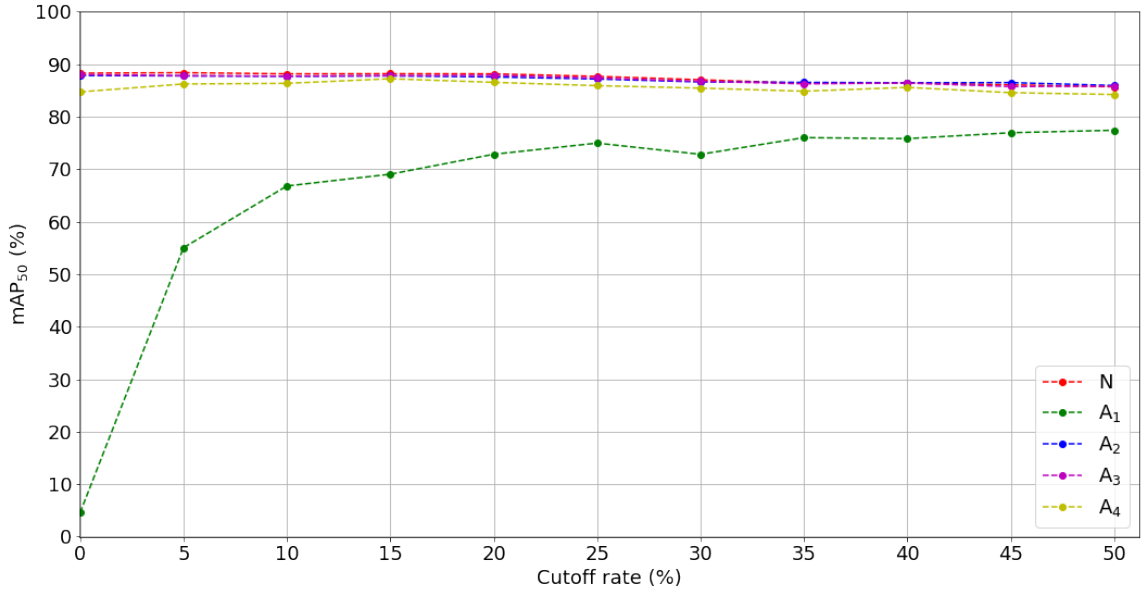


Figure 3.2 – mAP_{50} of MM-Retina depending on cutoff rate applied with RSC on group A configurations.

Impact of one missing modality

Figure 3.2 illustrates the performance of the CNNs (mAP_{50}) under normal conditions (configuration N) and when one input modality is missing (configurations A_1,A_2,A_3 and A_4). First, we see that, except for the A_1 configuration (missing RGB modality), the results vary slightly with Random Signal Cut. Since the RGB modality is very dominant in Multimodal RetinaNet, when one of the other modalities is missing, the performance losses are minimized, whatever the cutoff rate. We note a slight improvement in A_4 configuration (missing LD modality) when RSC is applied. Moreover, as the rate increases, the performance on these configurations decreases, showing that it is preferable to not apply a cutoff rate too high. On the other hand, for the A_1 configuration, RSC improves considerably the robustness of our networks, going from 4.55 % without RSC to 77.42 % with RSC and a 50 % cutoff rate. Even so, there is still a fairly significant loss of performance compared to normal conditions, which is obvious. These results show us that the modality RGB is highly essential for MM-Retina and that the other modalities do not bring much to these performances.

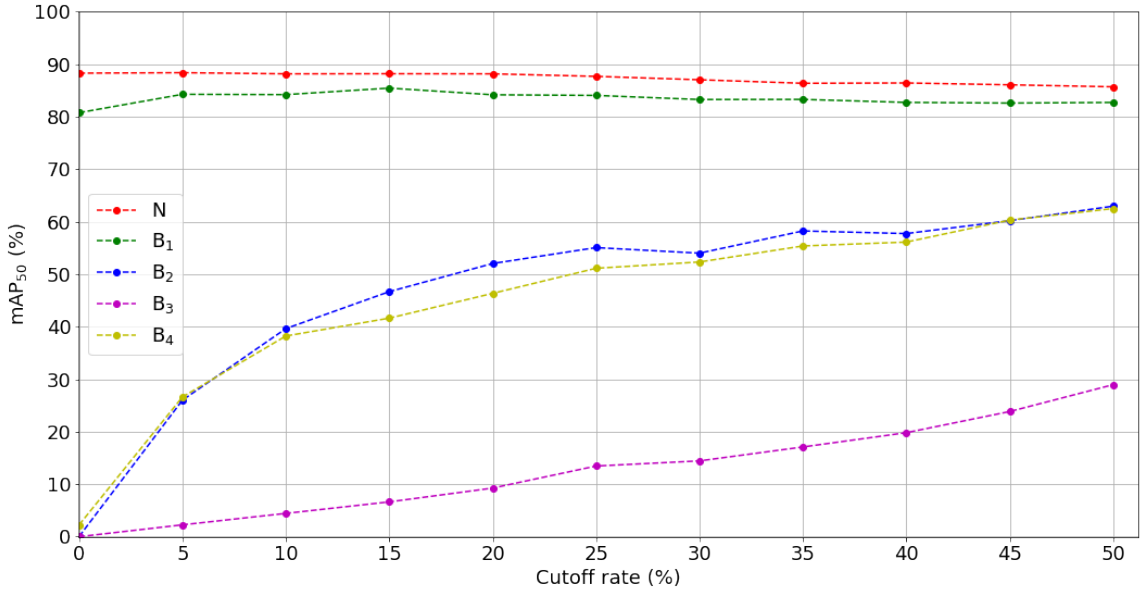


Figure 3.3 – mAP_{50} of MM-Retina depending on cutoff rate applied with RSC on group B configurations.

Impact of one remaining modality

In extreme conditions, when only one modality remains (configurations B_1,B_2,B_3 and B_4), the impact of RSC is more visible, as shown in Figure 3.3. In configuration B_1 (remaining RGB modality), our data augmentation method slightly increases CNNs performances with a low cutoff rate (as low as 5%), but this improvement decreases as the rate grows. Since the RGB modality is dominant, the higher the cutoff rate, the less the CNN will have this modality correct during its training, which justify this decrease. Moreover, the performances obtained on this configuration are quite close to the results in normal conditions, and are even finally better than when all the other modalities except RGB are present (configuration A_1). Concerning the configurations B_2, B_3 and B_4, RSC clearly improves the robustness of CNNs as the rate grows. While the performances are almost null without RSC, the latter have scores above 60 % for the configurations B_2 (remaining DP modality) and B_4 (remaining LD modality), as well as a mean average precision of 28.95 % in the configuration B3 (remaining OF modality) when a cutoff rate of 50 % is applied. One can think that Random Signal Cut allows CNNs to better exploit these modalities by giving them more importance, especially by cutting the dominant modality RGB during training. However, this does not result in better modality fusion, since the CNNs performances under normal conditions decrease at the same time.

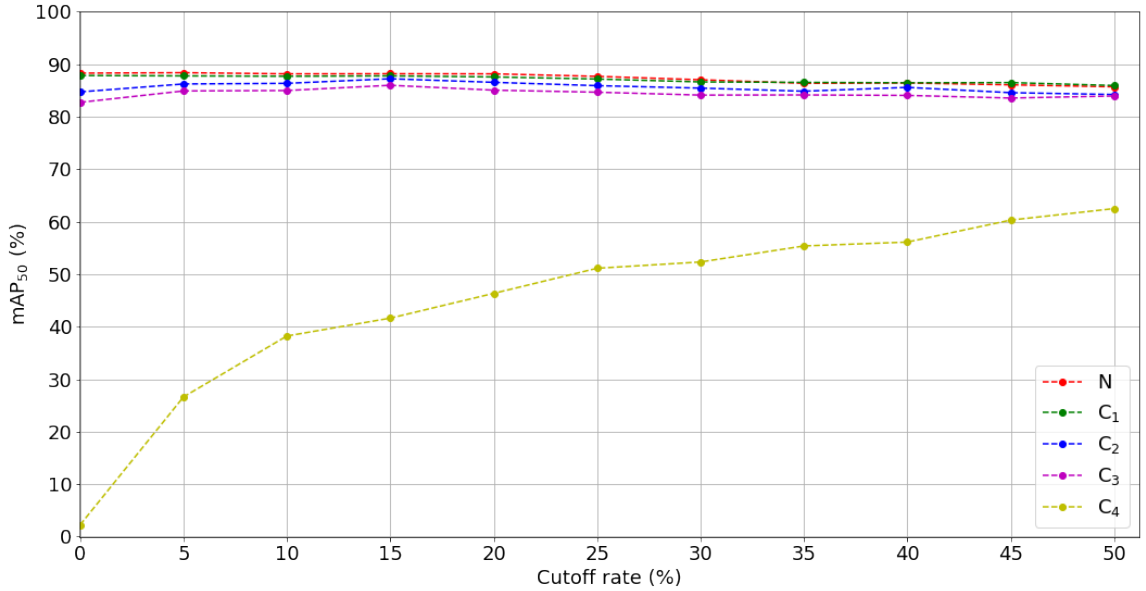


Figure 3.4 – mAP_{50} of MM-Retina depending on cutoff rate applied with RSC on group C configurations.

Impact in real failure conditions

Under real-world degraded conditions (configurations C_1, C_2, C_3, and C_4), RSC has less impact on the CNN robustness, as illustrated in Figure 3.4. For the first 3 configurations (C_1, C_2 and C_3), this is explained by the presence of the modality RGB which allows to maintain the performances of CNNs almost at the same level as in normal conditions. For the C_4 configuration (remaining modality LD), equivalent to the B_4 configuration, the improvement of the robustness by RSC is drastic. However, one should not forget that this case is theoretically quite rare since it requires the malfunctioning of both cameras at the same time. This still gives us an indication of the potential of RSC, especially if we place the sensor system in an environment where the cameras performances are low (for example in heavy fog or at night).

3.2.6 Overall discussion

To summarize the obtained results, Random Signal Cut positively impacts the robustness of Multimodal RetinaNet. In all degraded configurations, RSC significantly increased the mean average precision of our CNN. These scores improve as the signal cutoff rate increases. However, the method slightly reduces MM-Retina performances in normal conditions, so a trade-off is needed. From a cutoff rate of 35 %, the decrease in performance on *mAP*is significant compared to normal conditions, so we think it is preferable to opt for a rate around 25 %.

Nevertheless, it is difficult to affirm that MM-Retina is robust to failure, even with Random Signal Cut: when the RGB modality is absent, the performance is greatly diminished. This shows that to make a multimodal CNN robust, it is essential that its detection is not based only on one modality, as we have seen before.

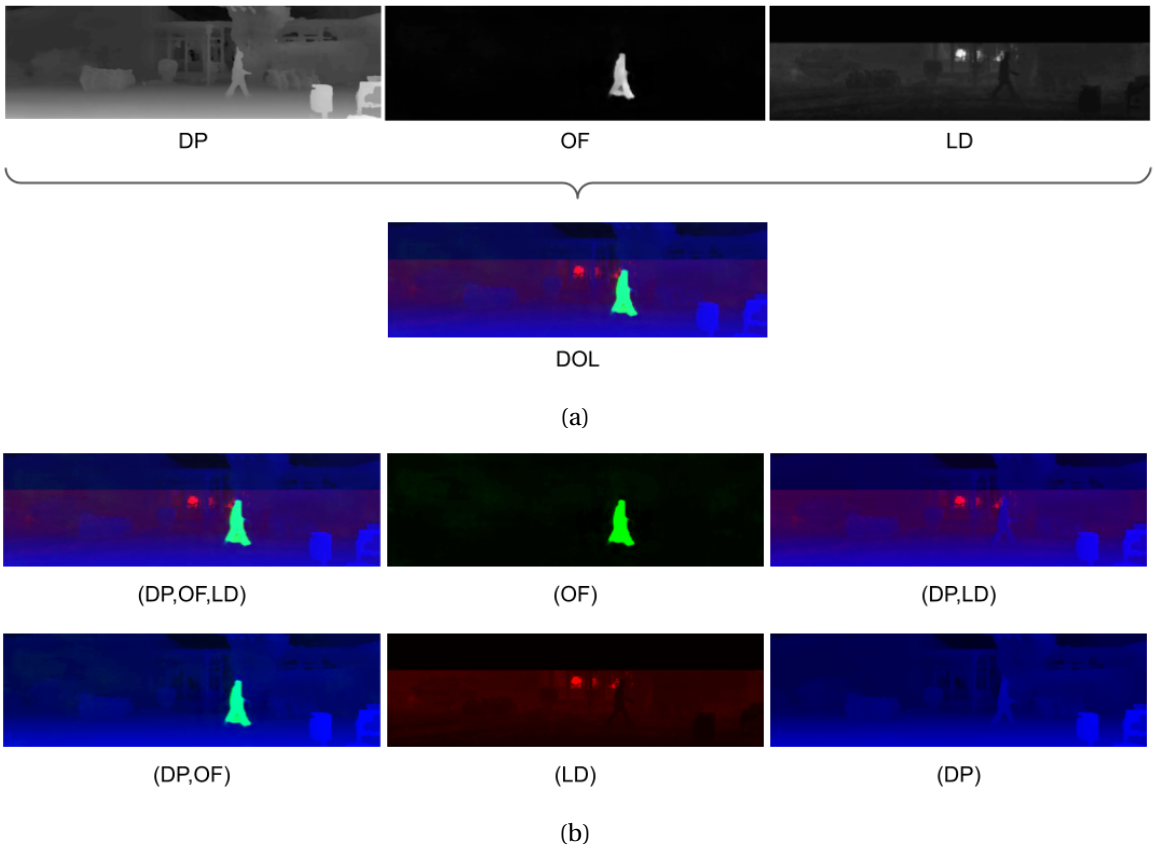


Figure 3.5 – a) RetinaNet input data and b) examples of generated samples by Random Channel Cut.

3.3 Random Channel Cut

We noticed with our first experiment that Random Signal Cut applied on Multimodal RetinaNet significantly improved its robustness when one or several modalities are missing. In this context, the absence of a signal was easily manageable for the CNN, since it uses the middle fusion. During modalities feature extraction, MM-Retina backbone had either correct or missing signals as input, meaning black images. In the second case, our CNN does not get any information from this missing signal and MM-Retina stacked fusion unit allows to filter the feature map extracted from this missing signal, having little impact on the performances.

However, when a multimodal network uses early fusion only, Random Signal Cut is unnecessary. Indeed, the input data already fused form a single signal and our method is unworkable in this case. A second problem is that the fusion is done before the feature extraction. Therefore, a missing modality impacts the feature extraction of the other ones, which can lead to a drastic decrease in performance. From these observations, we propose a new data augmentation method, Random Channel Cut (RCC), which supports early-fusion CNN input data and improve network robustness in case of missing modalities. We experiment RCC with a simple RetinaNet, taking DOL images as input.

3.3.1 Proposed method

Random Channel Cut (RCC) is a method derived from Random Signal Cut. Its objective is the same: to make a CNN learn on partial input data to improve its robustness in case of missing modalities. With RCC, input data are interpreted as signals containing one or more channels. For example, a RGB image alone forms a single signal including 3 channels: Red, Green and Blue. Contrary to RSC, where the cut was operated on a whole signal, RCC can cut each channel of a single signal, i.e. replace the channel values by 0, thus producing partial signals. To do this, Random Channel Cut assigns to each input data channel a cutoff rate, corresponding to the percentage of absence of this channel during CNN training. Each cutoff rate can be different from the others, and each channel cutoff is done independently of the other channels, even if they are linked by the same signal. Nevertheless, the configuration where all input channels are absent is locked, in order to have at least one channel present during training.

The main advantage of Random Channel Cut over Random Signal Cut is its support for early-fused input data. In the case where several modalities are included in the same signal, RSC will not generate data where only one of its modalities is missing: either the signal is complete or it is cut. On the other hand, RCC can generate partial signals, allowing the CNN to learn with signals where one of its modalities is missing. Then, given a sample of input data containing N channels, our data augmentation method can generate $2^N - 1$ different cuts. Knowing that the number of channels in the input data is greater than or equal to its number of signals, Random Channel Cut creates more diverse data than Random Signal Cut. However, this data is partly unreal, i.e., the partially cut signals generated may never actually be obtained by the sensor system. For example, our data augmentation method applied to RGB signal may generate an image where the Red and Blue channels are present, and the Green channel is absent, but a conventional camera will very rarely return such a signal. These generated signals can thus modify the reasoning of a CNN during its training, and the latter could consider the input signal as 3 distinct channels independent of the others.

Evaluation configuration	Available modalities		
	DP	OF	LD
N	✓	✓	✓
D_1	✗	✓	✓
D_2	✓	✗	✓
D_3	✓	✓	✗
D_4	✓	✗	✗
D_5	✗	✓	✗
D_6	✗	✗	✓

Table 3.4 – Available modalities according to evaluation configurations for RetinaNet ablation study (✓: present modality, ✗: missing modality).

3.3.2 Experimental setup and evaluation configurations

For this experiment, we evaluate RetinaNet with a channel cutoff rate varying from 0 % to 25 % with a step of 5 %, as well as with a channel cutoff rate of 50 %. Random Channel Cut is applied on RetinaNet input data, consisting of DOL signal including DP, OF and LD channels. Figure 3.5 shows our CNN input channels, and examples of generated samples by Random Channel Cut from this input. The CNNs have been trained for multi-class object detection on KITTI 2D Object Detection Dataset. The images of the dataset have been rescaled to have a size of around $600 \times 2000 \times 3$. The CNNs were initialized with a ResNet50 backbone, pre-trained on COCO dataset, and we fine-tuned them on 100 epochs with an Adam optimizer, a learning rate of 10^{-4}, a batchsize of 16 and automatic mixed precision. We then selected the optimal weights of the networks based on their lowest value on their validation loss. We use a 5-fold cross-validation to validate our results. We used data augmentation during CNN training with a set of methods including image rotation, horizontal flip, image translation, image zoom and image shearing.

For our next ablation study, we defined 7 evaluation configurations to analyze the impact of RCC on the robustness of RetinaNet to missing modalities. The configurations used are as follows:

— N : All input modalities are correct;

— D_1,D_2 and D_3 : One of the 3 input modalities is missing (respectively DP, OF and LD);

— D_4,D_5 and D_6 : Only one of the 3 input modalities is correct (respectively DP, OF and LD), whereas the two others are missing.

Table 3.4 sums up our evaluation configurations used for this experiment, as well as their missing modalities.

Cutoff rate	mAP	mAP $_{50}$	mAP $_{75}$	Car $_{75}$	Ped $_{50}$	Cyc $_{50}$
0 %	61.68 ± 0.58	88.18 ± 0.95	70.87 ± 0.30	86.50 ± 0.67	83.75 ± 1.93	86.93 ± 1.84
5 %	63.00 ± 1.20	**89.53 ± 0.95**	72.73 ± 2.07	**87.00 ± 0.48**	84.92 ± 1.20	87.73 ± 3.00
10 %	**63.31 ± 1.16**	89.35 ± 0.79	72.47 ± 2.01	86.89 ± 0.78	84.97 ± 2.05	**88.38 ± 2.31**
15 %	62.06 ± 1.17	88.75 ± 0.71	71.43 ± 1.22	86.14 ± 0.72	84.96 ± 1.92	88.28 ± 2.43
20 %	62.87 ± 1.56	89.33 ± 0.86	**73.26 ± 2.66**	86.60 ± 0.63	**85.42 ± 1.15**	88.20 ± 1.87
25 %	62.69 ± 1.58	89.15 ± 1.07	72.33 ± 2.25	86.42 ± 0.95	84.83 ± 1.22	88.12 ± 2.41
50 %	60.31 ± 2.33	88.08 ± 1.88	68.99 ± 3.02	84.77 ± 1.01	84.52 ± 1.85	86.54 ± 1.75

Table 3.5 – Performances of RetinaNet with DOL images as input depending on cutoff rate applied with Random Channel Cut in normal conditions.

3.3.3 Performances in normal conditions

We evaluate the impact of Random Channel Cut on the performance of RetinaNet on its validation set under normal conditions, when all modalities are present. We compare networks performances with the mean average precision *mAP*, mAP_{50} and mAP_{75} as well as the average precision Car_{75} , Ped_{50} and Cyc_{50} . Table 3.5 shows the average precision obtained by our CNNs depending on their channels cutoff rate.

First of all, we notice that except for the 50 % cutoff rate, Random Channel Cut slightly improve the mean average precision (*mAP*, mAP_{50} and mAP_{75}) of RetinaNet compared to the same network without this data augmentation technique (channel cutoff rate of 0 %). However, these improvements are not all significant. We also observe these increases on the accuracy of the *Pedestrian* and *Cyclist* classes (respectively Ped_{50} and Cyc_{50} metrics). These results are due to the nature of Random Channel Cut which is primarily a data augmentation method. Usually, data augmentation techniques are used to reduce the overfitting of a deep neural network, allowing to obtain better global performances. In our experimentation, RCC slowed down the overfitting, allowing to obtain better performances in normal conditions even if it is not its primary purpose.

Next, we can see that with a too high channel cut rate (50 %), Random Channel Cut causes a drop in overall performances on many metrics (*mAP*, mAP_{75} and Car_{75}), even if this is not significant. In the context of the autonomous vehicle, we consider that the sensors work the vast majority of the time, making this drop in performance significant. It is therefore important to properly calibrate this cutoff rate to ensure that mean average precision under normal conditions are the same as when Random Channel Cut is not used.

Cutoff Rate	Failure Configurations (with remaining input modalities)						
	N (DP,OF,LD)	D_1 (OF,LD)	D_2 (DP,LD)	D_3 (DP,OF)	D_4 (DP)	D_5 (OF)	D_6 (LD)
0 %	88.18	1.10	80.56	18.75	13.92	0.03	0.93
5 %	89.53	84.80	87.32	76.64	71.34	15.31	77.27
10 %	89.35	85.56	88.13	80.50	76.38	20.70	81.27
15 %	88.75	86.03	88.18	81.63	78.96	27.45	82.72
20 %	89.33	87.31	88.87	82.92	80.83	29.59	84.28
25 %	89.15	86.77	88.60	83.30	80.97	33.52	84.52
50 %	88.08	86.83	87.76	83.74	82.69	47.59	85.22

Table 3.6 – Performances (mAP_{50}) of RetinaNet in degraded conditions, depending on cutoff rate applied with Random Channel Cut. Higher losses comparing to normal conditions (second column) are highlighted with a darker blue.

3.3.4 Performances in case of missing modalities

We now perform an ablation study of our CNNs under degraded conditions on the evaluation configurations defined in section 3.3.2. Table 3.6 displays the mean average precision (mAP_{50}) obtained on their validation sets in different degraded configurations. These results are also plotted on figure 3.6, according to the channel cutoff rate applied on our networks.

When Random Channel Cut is not used during training (0 % cutoff rate), RetinaNet performances drops drastically in almost all evaluation configurations. Unlike Multimodal RetinaNet, our RetinaNet here does not have the modality RGB among its inputs. However, during our previous experiment in section 3.2.5, we noticed that this modality took the upper hand over the others: when it is present, MM-retina performances hardly decrease, whereas when it is missing, MM-Retina obtained poor results in mean average precision. For our new case, the slightest missing modality disturbs the network enormously because it does not rely on one of them in particular. The modality OF is however an exception, since when it is missing (configuration D_2), the mean average precision decreases slightly, even if is significant.

Next, Random Channel Cut significantly improves CNNs performances under almost all degraded conditions. Starting at a channel cutoff rate of 5 %, the mean average precision in each evaluation condition exceed 70 %, and for a rate of 20 %, these scores reach more than 80 %. Only the D_5 (DP and LD missing modalities) configuration is problematic for our networks, with notable improvements but well below the results obtained in all other configurations. This can be explained by the presence only of OF modality in the input data, and we have already seen in sections 2.2.5 and 2.3.2 that this modality alone is insufficient to ensure a correct object detection.

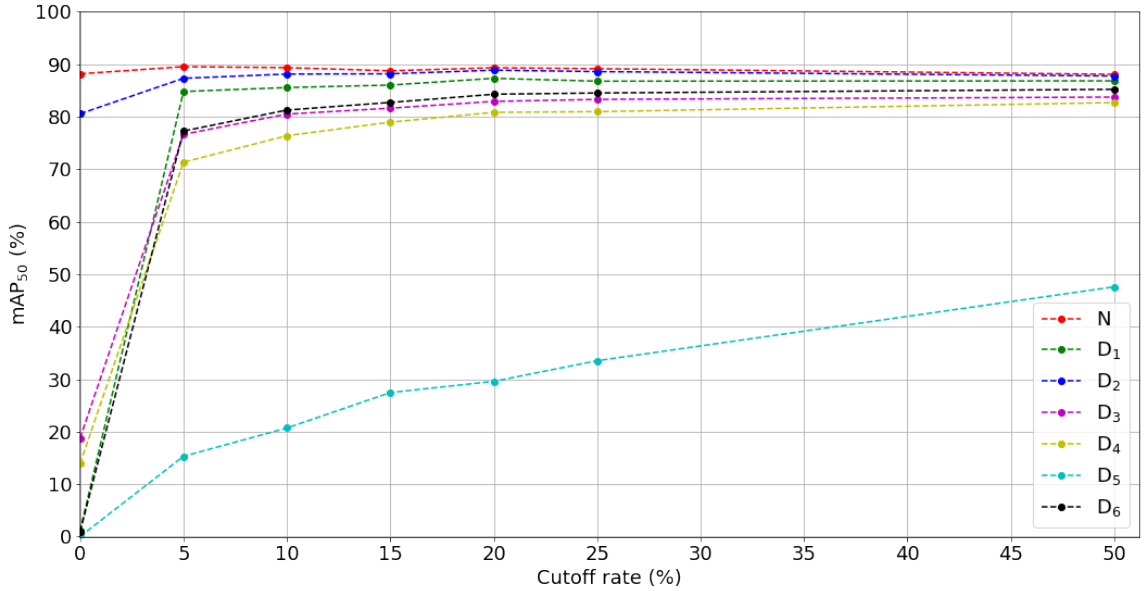


Figure 3.6 – mAP_{50} of RetinaNet depending on cutoff rate applied with RCC on degraded conditions.

3.3.5 Overall discussion

Previous experiments show that the impact of Random Channel Cut on the robustness of RetinaNet is highly positive. CNN performances in degraded conditions are all significantly improved with our data augmentation method. Moreover, not only the scores obtained in normal conditions did not drop, contrary to what we observed with RSC on MMRetina, but they are also slightly better. However, we believe that these improvements are due to the overdimensioning of the network with respect to the size of our dataset, which makes RetinaNet overfit during its training. Random Channel Cut, like most of data augmentation methods, slows down this overfitting, allowing the CNN to get better performances. Our method is therefore an efficient solution to improve multimodal neural networks using early fusion.

3.4 Random Modality Cut

We have seen with our last experiment that a CNN can learn normally with randomly missing channels in the input data via the use of Random Channel Cut. The feature extraction of the DOL signal was not disturbed by these missing modalities and the results obtained in normal conditions are at the same level, and even slightly better, than for a classical learning, with better performances in robustness.

In this section, we analyze a third case of robustness improvement with our multimodal CNN GFD-Retina, using both early fusion (construction of the DOL signal) and middle fusion (gated fusion of RGB and DOL signals). With this network, our 2 proposed data augmentation methods do not act in the same way. Random Signal Cut randomly drop the RGB signal or the DOL signal during the CNN training. In this setup, our method is not really adapted since the DOL signal is created via early-fusion. When the modalities DP, OF or LD will be missing, these configurations will not be learned during GFD-Retina training and thus risk to disturb it. With Random Channel Cut, the channels of RGB (Red, Green and Blue) and DOL (DP, OF and LD) signals are randomly removed from the input data during GFD-Retina training. RCC therefore generate, during neural network training, input data where the modality RGB will be partially complete (i.e. with one or more missing channels). However, these configurations are impossible in reality, since RGB channels are linked together. The particularity of GFD-Retina input data is that its modalities do not have the same format: a signal for the modality RGB and channels for the modalities DP, OF, LD. RSC and RCC are thus not completely suitable for our modalities.

In order to better adapt to any type of multimodal input data, we introduce in this section Random Modality Cut, a data augmentation method to improve CNN robustness in case of missing modalities. With RMC, the objective is to get closer to the possible failure configurations by abstaining from the input data format. This method is compared in this section with Random Signal Cut and Random Channel Cut, all 3 applied on GFD-Retina. This experimentation thus allows us to determine which of them best improves the robustness of our CNN while least impacting its performance under ideal conditions.

3.4.1 Proposed method

Like our previous data augmentation methods, Random Modality Cut aims at generating partial data during neural network training. However, with RMC, the format of the input modalities is ignored. The input data are thus no longer considered as several signals or several channels, but as several modalities, which can be of different types (image with one or several channels, 3D data volume, a scalar, etc.). During neural network training with Random Modality Cut, each modality of the input data can be cut, i.e. replaced by an object of the same type but null. Like as RSC and RCC, a cutoff rate is assigned to each modality, independent to the other modalities, signifying its probability of absence during the training of the network. Finally, in order not to disturb the neural network, we also prevent our method from generating null data where no modality is present.

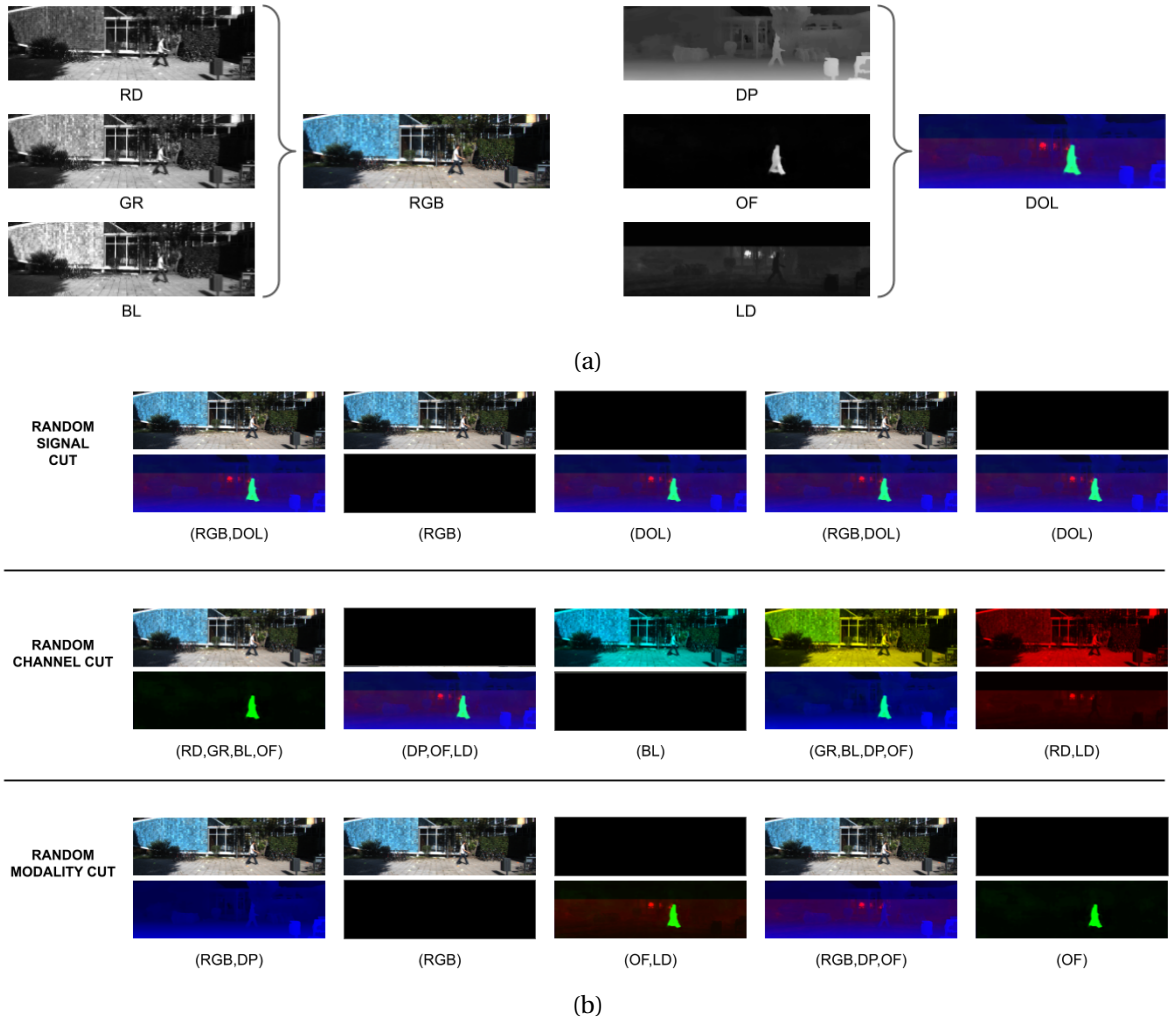


Figure 3.7 – a) GFD-Retina input data and b) examples of generated samples by Random Signal Cut, Random Channel Cut and Random Modality Cut.

The main advantage of Random Modality Cut, compared to our two previous data augmentation methods, is its great scalability, since this method can be applied on any multimodal neural network, where RSC and RCC are limited to particular multimodal CNNs. Moreover, the partial data generated by RMC are closer to reality, better reflecting the failures that a sensor system can obtain. However, with RMC, we do not distinguish the important modalities containing necessary information, from the additional modalities completing useful information but not sufficient on their own. Our method therefore has limitations since it can generate data where only the additional modalities are present during neural network training, which can strongly disturb the latter.

3.4.2 Experimental setup and evaluation configurations

In this experiment, we compare GFD-Retina using RSC, RCC and RMC with a cutoff rate varying from 0 % to 25 % with a step of 5 %, and a cutoff rate of 50 %. GFD-Retina takes as input RGB, DP, OF and LD modalities, forming 2 signals (RGB and DOL), and which can be decomposed into 6 channels : Red (RD), Green (GR) and Blue (BL) channels from RGB signal, as well as DP, OF and LD channels from DOL signal. Figure 3.7 illustrates GFD-Retina input data, and several samples generated by our 3 data augmentation methods.

Our CNNs backbones (for RGB and DOL signals) were first initialized with a ResNet50 pretrained on COCO dataset. Inside RetinaNet architecture, these backbones are trained on 100 epochs with samples of KITTI 2D Object Detection Dataset resized for having a $600 \times 2000 \times 3$ size, with an Adam optimizer, a learning rate of 10^{-4} and automatic mixed precision. We used data augmentation during backbones training with a set of methods including image rotation, horizontal flip, image translation, image zoom and image shearing. Depending on GFD-Retina data augmentation method (RSC, RCC or RMC), we additionally use the following data augmentation methods for robustness when learning backbones:

— If GFD-Retina use Random Signal Cut, our two backbones are trained without additional data augmentation method;

— If GFD-Retina use Random Channel Cut, our backbones are also trained with RCC with the same channel cutoff rate;

— If GFD-Retina use Random Modality Cut, only its backbone taking DOL signal as input is trained with Random Channel Cut with the same cutoff rate.

These differences are set so that the GFD-Retina backbones learn with the modalities in the same configurations as when training the full model. We then select the backbones optimal weights according to their lowest validation loss value, and transfer and freeze them on GFD-Retina architecture. Finally, we fine-tune full model top layers on 100 epochs with an Adam optimizer, a learning rate of 10^{-5} and automatic mixed precision. CNN final weights are also selected based on its validation loss. For each data augmentation method and for each cutoff rate, we train 5 instances of GFD-Retina for 5-fold cross validation.

For the ablation study of our developed networks, we use the same evaluation configurations as those listed in section 3.2.3, namely configuration N (normal conditions), group A configurations (when one modality is missing), group B configurations (when only one modality is present), and group C configurations (real failure configurations of our sensor system).

Note that in this experiment we compare our 3 data augmentation methods by applying the same theoretical cut-off rates. However, as they do not have the same number of modifiable entities (respectively 2 signals, 4 modalities and 6 channels for RSC, RMC and RCC), our 3 methods with the same theoretical cutoff rate will have different real cutoff rates between them. For example, for a theoretical cutoff rate of 50 %, the real cutoff rates of RSC, RCC and RMC will be respectively 33.33 %, 49.21 % and 46.67 %. Thus, these 3 methods are hardly comparable for the same theoretical cutoff rate.

Cutoff rate	Method	mAP	mAP $_{50}$	mAP $_{75}$	Car $_{75}$	Ped $_{50}$	Cyc $_{50}$
X	X	68.91 ± 1.39	93.01 ± 0.67	79.35 ± 2.60	90.59 ± 0.45	88.36 ± 1.22	92.76 ± 1.52
5 %	RSC	68.48 ± 0.89	92.93 ± 0.65	79.55 ± 1.44	90.54 ± 0.67	88.57 ± 1.16	92.65 ± 2.27
	RCC	70.11 ± 0.81	92.94 ± 0.93	79.57 ± 1.05	91.16 ± 0.44	88.32 ± 0.78	92.59 ± 2.00
	RMC	69.45 ± 1.61	93.11 ± 0.87	79.67 ± 1.69	90.93 ± 0.51	88.12 ± 1.03	**93.26 ± 1.98**
10 %	RSC	68.95 ± 1.34	93.02 ± 0.29	79.10 ± 2.10	90.79 ± 0.48	88.59 ± 0.89	92.83 ± 1.80
	RCC	70.64 ± 1.22	93.20 ± 0.70	80.99 ± 1.10	91.35 ± 0.41	88.79 ± 0.94	93.08 ± 2.00
	RMC	69.65 ± 0.93	**93.40 ± 0.51**	80.47 ± 1.48	90.96 ± 0.50	88.49 ± 1.25	92.78 ± 2.32
15 %	RSC	68.63 ± 0.92	92.92 ± 0.66	79.31 ± 0.96	90.74 ± 0.49	88.39 ± 0.69	92.60 ± 1.89
	RCC	**71.14 ± 0.92**	93.21 ± 0.34	**81.30 ± 1.17**	**91.53 ± 0.35**	**89.23 ± 1.09**	92.45 ± 1.66
	RMC	69.56 ± 1.06	92.96 ± 0.58	80.27 ± 2.12	90.78 ± 0.57	88.60 ± 0.79	93.03 ± 1.91
20 %	RSC	68.64 ± 1.75	92.70 ± 1.11	79.61 ± 2.12	90.96 ± 0.35	88.56 ± 0.66	92.48 ± 2.48
	RCC	70.94 ± 0.80	93.34 ± 0.64	79.74 ± 0.97	91.19 ± 0.47	88.67 ± 1.12	92.35 ± 1.92
	RMC	69.32 ± 1.23	93.14 ± 0.69	79.95 ± 1.45	90.92 ± 0.51	89.12 ± 0.92	93.06 ± 1.84
25 %	RSC	68.55 ± 1.43	92.78 ± 0.67	79.27 ± 1.35	90.61 ± 0.46	88.34 ± 0.87	92.35 ± 2.13
	RCC	70.39 ± 0.83	93.23 ± 0.72	80.13 ± 0.97	90.93 ± 0.42	88.62 ± 1.07	92.55 ± 2.26
	RMC	68.69 ± 0.86	93.21 ± 0.45	79.33 ± 1.86	90.40 ± 0.62	88.62 ± 0.85	92.57 ± 2.59
50 %	RSC	67.77 ± 1.06	92.41 ± 0.70	78.46 ± 1.42	89.88 ± 0.44	87.96 ± 0.72	92.32 ± 2.47
	RCC	69.87 ± 0.82	93.26 ± 0.39	80.00 ± 1.13	90.78 ± 0.38	88.79 ± 1.26	92.48 ± 0.85
	RMC	67.52 ± 1.72	92.62 ± 0.70	78.27 ± 1.95	89.65 ± 0.67	88.40 ± 0.73	92.88 ± 1.51

Table 3.7 – Performances of GFD-Retina depending on cutoff rate applied with either Random Signal Cut (RSC), Random Channel Cut (RCC) or Random Modality Cut (RMC) in normal conditions.

3.4.3 Performances in normal conditions

Our CNNs are first evaluated on their validation sets under normal conditions, depending on their data augmentation method (RSC, RCC or RMC) and their theoretical cutoff rate (from 0 % to 25 % with a step of 5 %, and 50 %). Our evaluation metrics are the mean average precision *mAP*, mAP_{50} , mAP_{75} as well as the precision Car_{75} , Ped_{50} and Cyc_{50} . Table 3.7 shows the performances reached by our multimodal CNNs.

First of all, whatever the data augmentation method applied and whatever the rate, the mean average precision mAP_{50} does not vary a lot, showing once again that our methods do not affect the performance in normal conditions. These conclusions are the same for the precision on *Car*, *Pedestrian* and *Cyclist* classes (with the metrics Car_{75} , Ped_{50} and Cyc_{50}).

However, given the more restrictive *mAP* metric, our 3 data augmentation methods impact differently GFD-Retina performances. For RSC, the performances are at the same level as without data augmentation (around 68 %). With a theoretical cutoff rate of 50 %, the mean average precision starts to decrease, but slightly and not significantly. The networks using RMC have slightly better performances than those using RSC or without

data augmentation, with scores exceeding 69 %. When the theoretical rate reaches 50 %, we also see the same drop in performance as for RSC. Finally, RCC improves even more the precision of GFD-Retina compared to the other methods previously mentioned, with a score reaching even 71.14 % for a theoretical cutoff rate of 15 %. In the same way, as the cutoff rate increases, the results decrease.

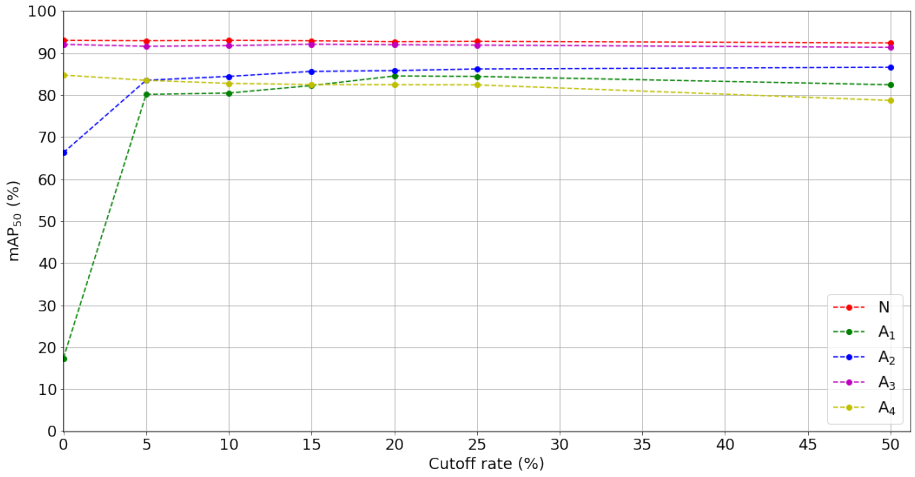
Through these results, one could assume that Random Channel Cut improves GFD-Retina performances, ahead of Random Modality Cut and then Random Signal Cut. However, we believe that these increases are mainly due to the architecture of GFD-Retina, which requires a lot of images to obtain better results. From there, our data augmentation methods produce new samples from the same starting sample, slowing down the CNN overfitting. According to the number of modifiable entities K of our input modalities (respectively 2 signals, 4 modalities and 6 channels for RSC, RMC and RCC), the number of cutoff states by our data augmentation methods is $2^K - 1$, resulting in 3 cutoff states for Random Signal Cut, 15 for Random Modality Cut and 63 for Random Channel Cut. The greater the number of cutoff states, the more diverse the data generated by our methods and the slower the overfitting of GFD-Retina will be, improving its performances. This is why networks using Random Channel Cut have the best mean average precision here. However, by using a larger multimodal dataset or by opting for a smaller multimodal CNN requiring less input data, these disparities between methods should be more reduced under normal conditions.

3.4.4 Performances in case of missing modalities

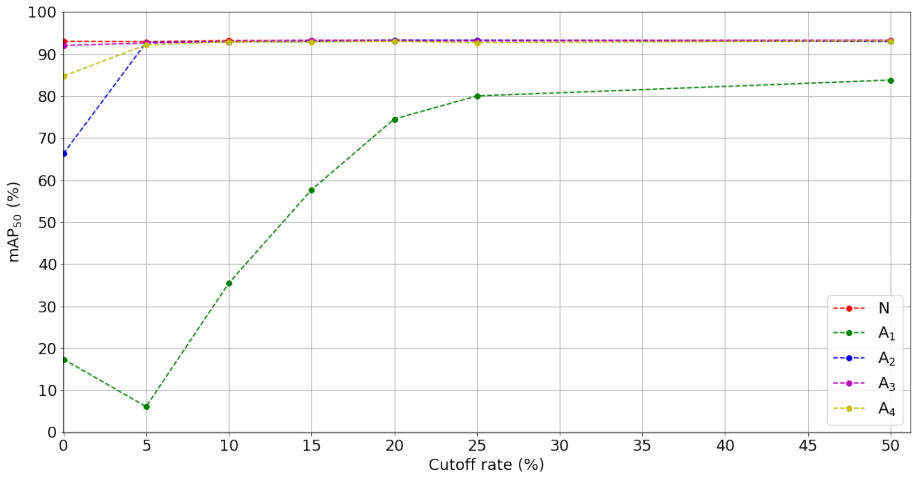
We analyze the impact of our 3 data augmentation methods under degraded conditions on GFD-Retina. Ablation study of our networks is done on their validation sets in the 10 evaluation configurations cited in section 3.2.3. Table 3.8 shows the performances of our multimodal CNNs in each configuration, according to their data augmentation method and theoretical cutoff rate applied.

Theoretical cutoff rate	Method	Real cutoff rate	Failure Configurations (with remaining input modalities)									
			N (RGB,DP,OF,LD)	A_1 (DP,OF,LD)	A_2 / C_1 (RGB,OF,LD)	A_3 (RGB,DP,LD)	A_4 / C_2 (RGB,DP,OF)	C_3 (RGB,OF)	B_1 (RGB)	B_2 (DP)	B_3 (OF)	B_4 / C_4 (LD)
X	X	X	93.01	17.36	66.29	92.06	84.75	56.06	59.97	2.72	0.00	0.02
5 %	RSC	4.76 %	92.93	80.15	83.52	91.61	83.50	77.37	85.90	15.74	0.12	5.85
	RCC	5.00 %	92.94	6.07	92.73	92.63	92.18	90.52	87.28	1.72	0.24	2.97
	RMC	5.00 %	93.11	80.28	92.89	93.14	91.89	90.23	88.62	57.00	10.78	66.60
10 %	RSC	9.09 %	93.02	80.47	84.43	91.75	82.78	74.27	89.18	14.72	0.74	12.49
	RCC	9.99 %	93.20	35.47	92.94	92.91	92.90	91.66	91.13	19.13	3.74	27.01
	RMC	10.00 %	93.40	84.15	93.08	93.15	92.26	90.64	89.93	66.66	16.74	74.33
15 %	RSC	13.04 %	92.92	82.25	85.63	92.11	82.50	79.82	90.28	14.34	0.61	15.62
	RCC	14.96 %	93.21	57.68	92.95	93.29	92.89	91.91	91.77	42.54	9.87	50.61
	RMC	15.00 %	92.96	86.40	92.98	93.03	92.39	91.40	90.76	73.51	22.68	79.03
20 %	RSC	16.67 %	92.70	84.53	85.81	91.97	82.45	76.65	91.17	15.88	0.88	20.24
	RCC	19.87 %	93.34	74.52	93.29	93.10	93.00	92.18	91.90	58.34	15.84	66.62
	RMC	19.99 %	93.14	85.26	92.94	93.19	92.81	92.04	91.28	72.86	24.36	78.03
25 %	RSC	20.00 %	92.78	84.43	86.20	91.89	82.42	77.34	91.47	16.00	1.11	23.13
	RCC	24.71 %	93.23	80.02	93.28	93.11	92.72	92.01	91.94	65.93	20.21	73.35
	RMC	24.98 %	93.21	87.58	92.68	93.04	92.44	91.24	90.76	79.38	31.49	81.26
50 %	RSC	33.33 %	92.41	82.45	86.61	91.35	78.72	74.78	91.24	13.91	0.87	30.64
	RCC	46.67 %	93.26	83.79	92.96	93.25	93.11	92.58	92.15	75.68	35.46	79.29
	RMC	49.21 %	92.62	86.95	92.50	92.67	91.73	90.45	89.68	80.76	43.22	83.37

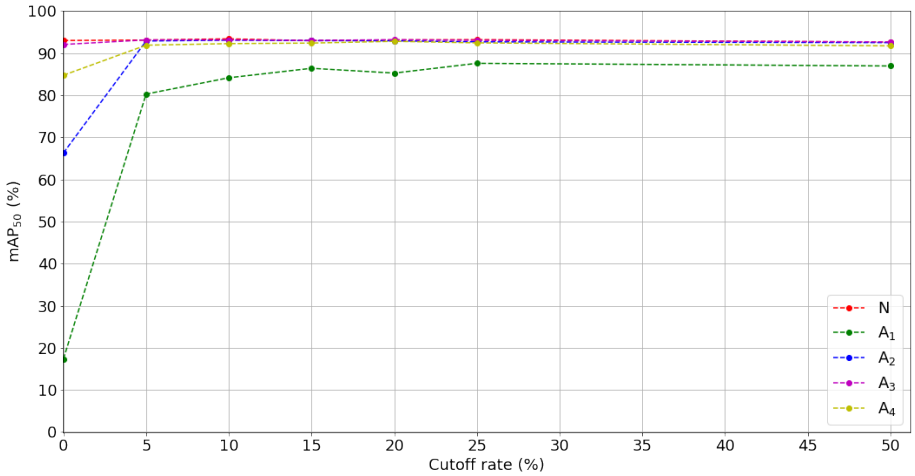
Table 3.8 – Performances (mAP_{50}) of GFD-Retina in degraded conditions, depending on cutoff rate applied with either Random Signal Cut (RSC), Random Channel Cut (RCC) or Random Modality Cut (RMC). Higher losses comparing to normal conditions (fourth column) are highlighted with a darker blue.



(a) RSC



(b) RCC



(c) RMC

Figure 3.8 – mAP_{50} of GFD-Retina depending on cutoff rate applied with RSC (a), RCC (b) or RMC (c) on group A configurations.

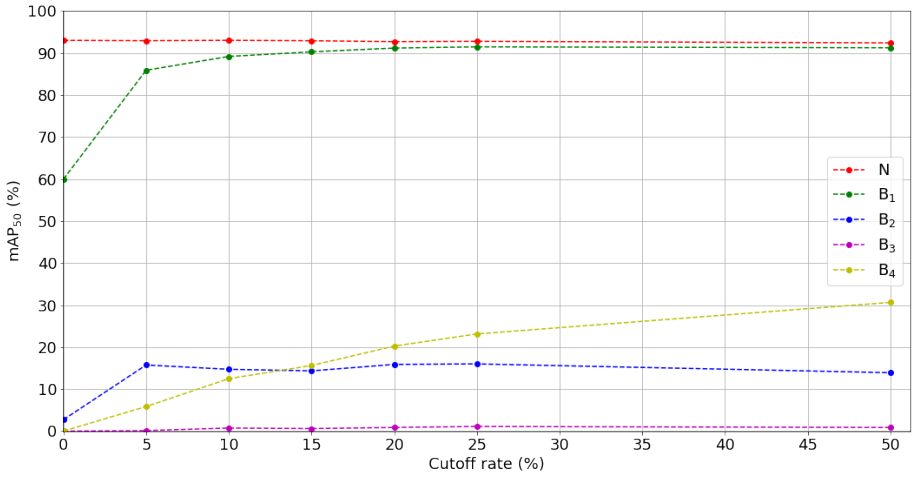
Impact of one missing modality

Figure 3.8 illustrates GFD-Retina performances (mAP_{50}) under normal conditions (configuration N) and when one input modality is missing (configurations A_1, A_2 ,A_3 and A_4) when applying RSC, RCC and RMC.

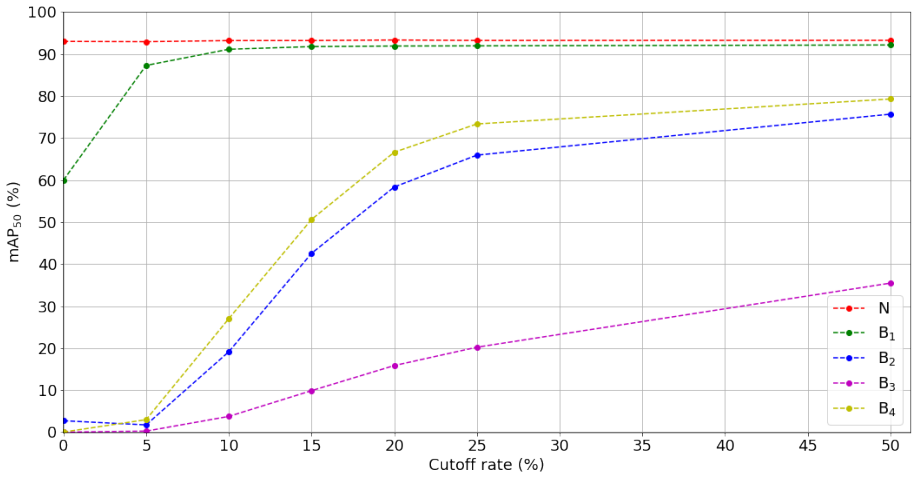
For degraded conditions with the RGB modality present (configurations A_2, A_3 and A_4), RCC and RMC perform at the level of the normal configuration N from a theoretical cutoff rate of 5 %. The performances in these same configurations with Random Signal Cut are on the other hand weaker. This is due to the absence of one of the DOL signal modalities (respectively DP, OF and LD for A_2, A_3 and A_4), which was never learned during our networks training, which disrupts both the feature extraction of the DOL signal and GFD-Retina middle fusion via the gated fusion units. The RGB modality present in input data ensures a correct detection level, resulting in slight decreases around 10 % maximum for the A_2 and A_4 configurations. For the A_3 configuration, OF modality absence does not impact networks accuracy, with a maximum decrease of 1.32 % for GFD-Retina with a theoretical cutoff rate of 5 %. We believe that this modality again does not provide useful information to GFD-Retina.

When RGB modality is missing from the input data (configuration A_1), the performance without data augmentation is disastrous (17.36 %), indicating that GFD-Retina middle fusion is not robust to failure without adapted learning. With our data augmentation methods, the scores obtained by the CNNs are much better, exceeding 80 % whatever the cutoff rate for RSC and RMC. These increases are also noticeable with RCC but are more slight, requiring a higher cutoff rate to be at the level of the other methods.

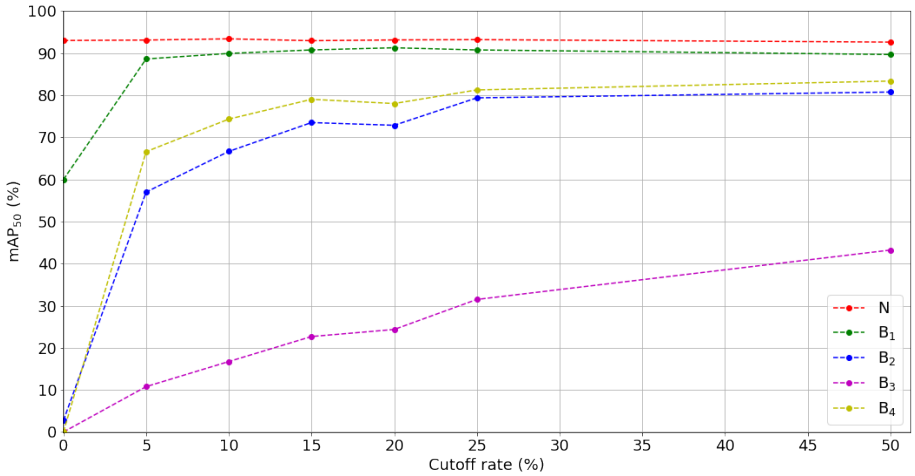
Globally, GFD-Retina is more robust in Group A configurations with Random Modality Cut.



(a) RSC



(b) RCC



(c) RMC

Figure 3.9 – mAP_{50} of GFD-Retina depending on cutoff rate applied with RSC (a), RCC (b) or RMC (c) on group B configurations.

Impact of one remaining modality

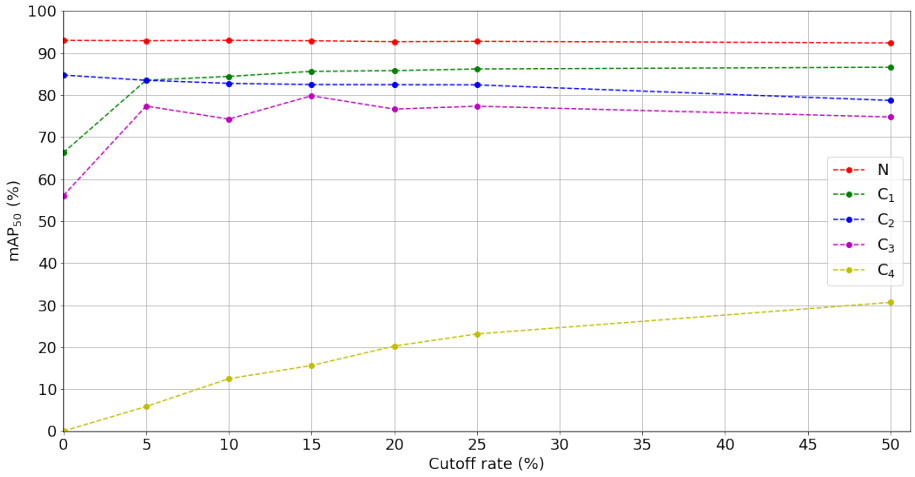
Figure 3.9 shows GFD-Retina mean average precision (mAP_{50}), depending on its data augmentation method (RSC, RCC and RMC) and its cutoff rate in extreme conditions when only one modality is present in input data.

With only RGB modality as input (B_1 configuration), GFD-Retina obtains a mean average precision of 59.97 %, representing a loss of 33.04 % compared to normal conditions. This indicates that CNN Gated Fusion Units are also not robust to the absence of the DOL signal without a strategy for robustness enhancement. Using our data augmentation methods, the performance on this configuration increases, reaching scores above 90 % with a cutoff rate of 15 % or more. This shows that our methods make the middle fusion of GFD-Retina robust, learning to make do with RGB signal when DOL signal is not present. GFD-Retina with RSC, RCC or RMC obtain fairly close results on this configuration, even almost similar when the theoretical cutoff is above 15 %.

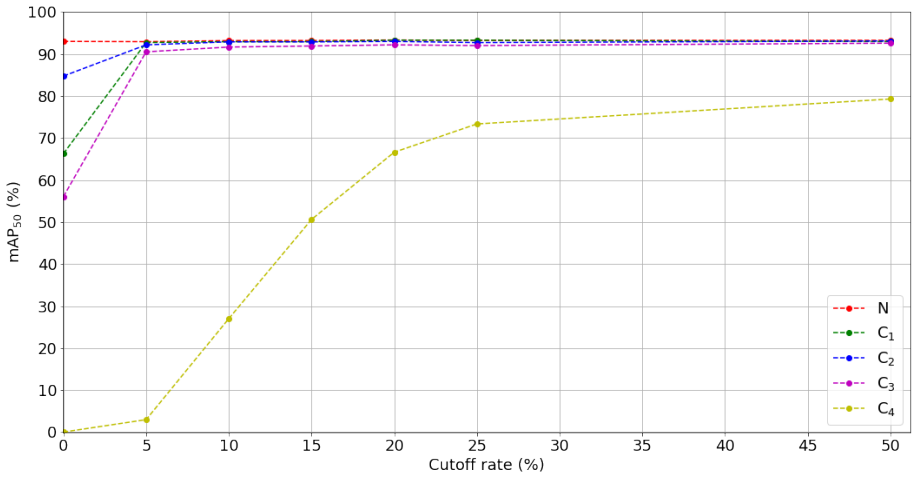
For B_2 (remaining DP modality) and B_4 (remaining LD modality) configurations, GFD-Retina has almost zero scores without data augmentation. Each of our 3 methods significantly improves the results in these configurations, even allowing to reach performances above 80 % with GFD-Retina and a cutoff rate of 50 %. GFD-Retina with RCC tends to obtain similar scores, but requires a higher cutoff rate than with RMC. Random Signal Cut on the other hand slightly improves the results, but much lower than the 2 other methods. This remains logical since these configurations are not learned during the training of our CNNs with RSC.

Finally, when only the OF modality is present (B_3 configuration), the mean average precision are null for the networks without data augmentation but also for those with Random Signal Cut. RMC obtains better performances than RCC on this configuration with equivalent theoretical cutoff rate, but all these results are under 50 %, which remains rather weak. In the context of the autonomous vehicle, this configuration is ignorable since it is impossible to obtain, because the OF modality is created from the RGB one: if this last one is absent, OF will be as well.

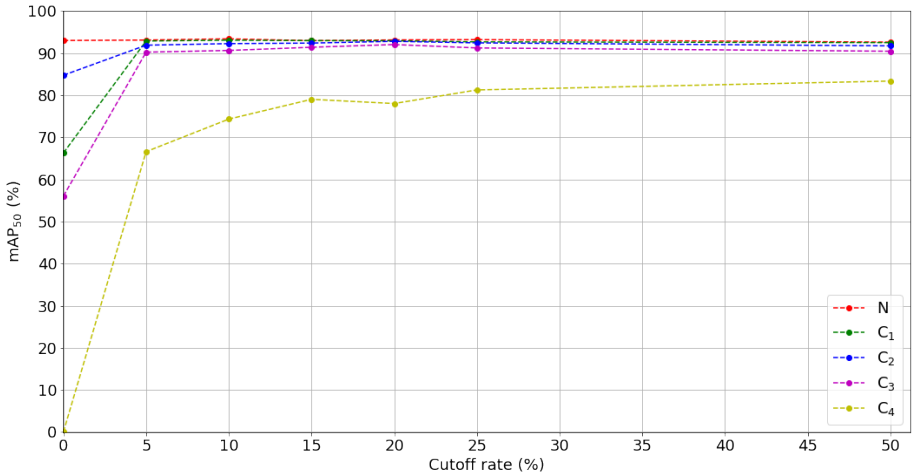
Overall, GFD-Retina with RMC gets the best performance on group B configurations.



(a) RSC



(b) RCC



(c) RMC

Figure 3.10 – mAP_{50} of GFD-Retina depending on cutoff rate applied with RSC (a), RCC (b) or RMC (c) on group C configurations.

Impact in real failure conditions

The scores of the CNNs under real degraded conditions (group C configurations), according to their applied data augmentation method and their cutoff rate, are displayed in figure 3.10. For the configurations where RGB modality is present (C_1, C_2, C_3), GFD-Retina scores without data augmentation under normal conditions are lower, without being catastrophic. Networks using RMC or RCC obtain similar performances on these configurations whatever the cutoff rate applied, while those using RSC get a smaller improvement. C_3 configuration (remaining LD modality), equivalent to B_4 configuration, determines Random Modality Cut as more robust than Random Channel Cut. However, it should be noted that this configuration is an extreme case, requiring the failure of both RGB cameras, which is theoretically very rare. Even so, in an unfavorable environment where these cameras are disturbed, one can imagine that a CNN using Random Modality Cut will better rely on LD modality to compensate the drawback.

3.4.5 Overall discussion

Our 3 data augmentation methods have a positive impact on the robustness of GFD-Retina in case of missing modalities. Under normal conditions, Random Channel Cut slightly improves the results, as previously with RetinaNet, ahead of Random Modality Cut and Random Signal Cut. However, for each of the methods, a too high cutoff rate tends to decrease these performances. For degraded configurations, RSC is not adapted for GFD-Retina input data, and increases the performance of the latter in some of these configurations, mainly when at least one of the RGB or DOL signals is present. This method is therefore not recommended when the network uses early fusion. For RCC, the robustness of GFD-Retina is improved in all degraded configurations. Random Modality Cut stands out by obtaining better results than Random Channel Cut with a lower cutoff rate. Our 2 methods are therefore effective in enhancing the robustness of multimodal CNNs using early fusion and middle fusion.

If we had to make a choice, we would opt for Random Modality Cut, which is better suited for any type of multimodal neural network, whatever its input data. Indeed, when the modalities are in the form of signals (i.e. with Multimodal RetinaNet), our method works in the same way as Random Signal Cut. For a network taking modalities in the form of channels as input (i.e. RetinaNet with DOL input), our method will act as Random Channel Cut. Finally, if our modalities are in another form (combination of 2 channels, 3D volume, 1D signal, etc.), Random Modality Cut will be able to work in any configuration, while corresponding better to the possible failure configurations. Nevertheless, the results obtained with Random Channel Cut are interesting, especially for normal conditions, and deserve to be experimented further to better determine its potential.

3.5 In-depth analysis of failure impact on a robust CNN

We have shown that our 3 data augmentation methods, applied on GFD-Retina, improve its robustness in degraded conditions when one or more modalities are missing. Random Modality Cut stands out from the other two, offering a better robustness in all degraded conditions during our experiments, and being able to be applied on any type of multimodal neural network, whatever the input data. With a cut-off rate of 25 %, we obtain a good compromise between performances in normal conditions at the same level as without the method, and limited performance losses in degraded conditions (except in extreme conditions where only one modality is present).

It is interesting to note that the performances losses of GFD-Retina combined with Random Modality Cut at 25% in degraded conditions are very slight in many degraded conditions (A_1, A_2, A_3, A_4, B_1 and C_3). Moreover, in some of them, the losses are negligible (below 1 %). We can therefore question the usefulness of some modalities, since they do not seem to impact the network when they are missing. In this section, we evaluate GFD-Retina with RMC and a cutoff rate of 25% with more specific metrics, in order to better understand the impact of missing modalities on CNN performances.

3.5.1 Evaluation on each object class

We evaluate our robust GFD-Retina under degraded conditions on each object class of the dataset. KITTI 2D Object Detection Dataset includes 51865 objects on its training set, labeled in 8 classes: *Car*, *Pedestrian*, *Cyclist*, *Van*, *Truck*, *Person sitting*, *Tram* and *Misc*. This last class includes different types of objects in small quantities in the dataset (motorcycle, caravan, trailer among others). The number of instances of each class in the training dataset varies greatly, ranging from 28742 objects labeled *Car*, to 222 objects labeled *Person sitting*. We use as reference metric the average precision with an IoU threshold of 50 % and a score threshold of 5 %. Table 3.9 shows the performances of the 5 GFD-Retina networks on their validation sets on each class of objects in normal and degraded conditions.

Object class	**Number of instances**	**Failure Configurations (with remaining input modalities)**						
		N	A_1	A_2	A_3	A_4	C_3	B_1
		(RGB,DP,OF,LD)	**(DP,OF,LD)**	**(RGB,OF,LD)**	**(RGB,DP,LD)**	**(RGB,DP,OF)**	**(RGB,OF)**	**(RGB)**
Car	28742	97.01	95.82	96.92	96.98	96.76	96.56	96.60
Van	2914	96.70	91.48	96.50	96.91	95.62	94.40	94.44
Truck	1094	97.87	94.36	97.50	97.84	97.84	96.73	96.52
Tram	511	97.39	93.49	96.73	97.11	97.29	97.03	97.17
Pedestrian	4487	88.62	84.02	88.28	88.41	87.75	86.28	85.85
Person sitting	222	80.08	67.45	77.30	79.53	78.74	74.90	72.38
Cyclist	1627	92.57	87.43	93.02	92.15	91.31	90.09	90.17
Misc	973	95.46	86.60	95.14	95.38	94.25	93.92	92.91
Total	51865	93.21	87.58	92.68	93.04	92.44	91.24	90.76

Table 3.9 – Average precision of GFD-Retina with Random Modality Cut and a 25 % cutoff rate on each object class. Higher losses comparing to normal conditions (third column) are highlighted with a darker blue.

Size category	Number of instances	Failure Configurations (with remaining input modalities)						
		N (RGB,DP,OF,LD)	A_1 (DP,OF,LD)	A_2 (RGB,OF,LD)	A_3 (RGB,DP,LD)	A_4 (RGB,DP,OF)	C_3 (RGB,OF)	B_1 (RGB)
mAP_S	16633	83.84	75.58	84.04	83.78	78.36	74.16	73.82
mAP_M	18005	89.48	83.98	89.12	89.52	88.90	85.60	87.32
mAP_L	17227	90.73	81.55	90.18	90.18	89.79	88.48	86.58

Table 3.10 – Mean Average Precision of GFD-Retina with Random Modality Cut and a 25 % cutoff rate depending on object size : Small (mAP_S), Medium (mAP_M) or Large (mAP_L). Higher losses comparing to normal conditions (third column) are highlighted with a darker blue.

First of all, under normal conditions, the average precision on the *Pedestrian* and *Person sitting* classes are lower than for all other classes, with scores below 90 %. These results are due in part to the size of these objects, which are smaller than the others, and therefore more difficult to detect. There is no particular link between the number of instances of each class and the average precision obtained in normal conditions, showing that GFD-Retina generalizes well during its learning despite the large difference of instances that there can be between some classes. In degraded conditions, the losses on the majority of the classes are approximately of the same order as those on all the classes in general. Only the class *Person sitting* has more important decreases of precision on the degraded configurations A_1, A_2, B_1 and C_3 compared to the other object classes. Considering the very small number of instances of this class (222 out of the 51,865 objects in the training dataset, i.e. 0.4 % of all objects), it is quite normal that these decreases are more amplified. We consider that when one or more modalities are missing, none of our object classes is more impacted than the others in the performance loss.

3.5.2 Evaluation according to objects size

We have previously noted that *Pedestrian* and *Person sitting* objects are smaller on average than other objects. Within the same class, the size of the objects is very variable. We now evaluate our robust multimodal networks in degraded conditions according to the size of the objects. During training of the latter, we enlarged the input images from a size of about $375 \times 1225 \times 3$ to about $600 \times 2000 \times 3$. The objects in the training dataset were thus divided into 3 categories:

— Small : the objects have a surface less than 3600 pixels;

— Medium : the objects have a surface between 3600 pixels and 14400 pixels;

— Large : the objects have a surface greater than 14400 pixels.

For each size category, we use the mean average precision with an IoU threshold of 50 % and a score threshold of 5 % (noted mAP_S, mAP_M and mAP_L for respectively the categories Small, Medium and Large). During CNN evaluation on a category, the objects of the other categories were considered as regions to be ignored, so that the detection of these objects by the network do not penalize the performances. Table 3.10 shows GFD-Retina performances with RMC and a cutoff rate of 25 % in normal and degraded conditions, for each size category.

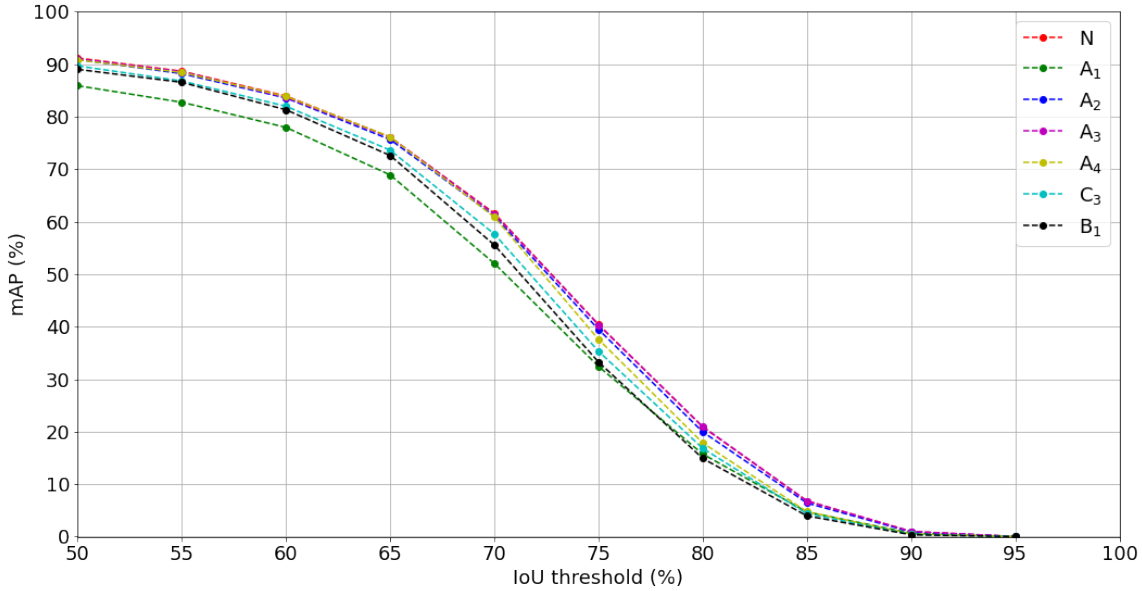


Figure 3.11 – GFD-Retina mean average precision on normal and degraded conditions depending on IoU threshold applied.

Under normal conditions, we see a clear difference in performance between the objects in the Medium and Large categories (respectively 89.48 % and 90.73%) and those in Small category (83.84 %). This discrepancy is justified by the IoU threshold of 50 %, requiring high accuracy for small objects, where errors on larger objects are less impactful. In degraded conditions, we note that the absence of DP or OF modalities (configuration A_2 and A_3) does not affect the performance of GFD-Retina on each size category. In our other degraded configurations, the decreases are more intensified, especially for configurations where one of the two input signals RGB or DOL is missing (configurations A_1 and B_1). However, for the sake of visibility, we do not display the standard deviations on table 3.10, but they are quite large (around 3 to 5 %). The behavior of each of our tested networks differs in degraded conditions. Therefore, we cannot conclude that one category of object size is more impacted positively or negatively compared to the others in degraded conditions.

3.5.3 CNN bounding boxes accuracy in degraded conditions

To evaluate GFD-Retina under degraded conditions, we previously used the mean average precision mAP_{50} as metric. With this metric, we consider that an object of the ground truth is well detected if one of the predictions of the network has the same class as the object and that the intersection of their bounding boxes divided by their union is greater than or equal to 50 %. In degraded conditions, if our network loses accuracy in these bounding boxes, the prediction can remain correct as long as the IoU is above this threshold. In order to better understand the accuracy of the bounding box coordinates of the GFD-Retina predictions when one or several modalities are missing, we now evaluate our networks using the mean average precision with a score threshold of 5 %, and varying

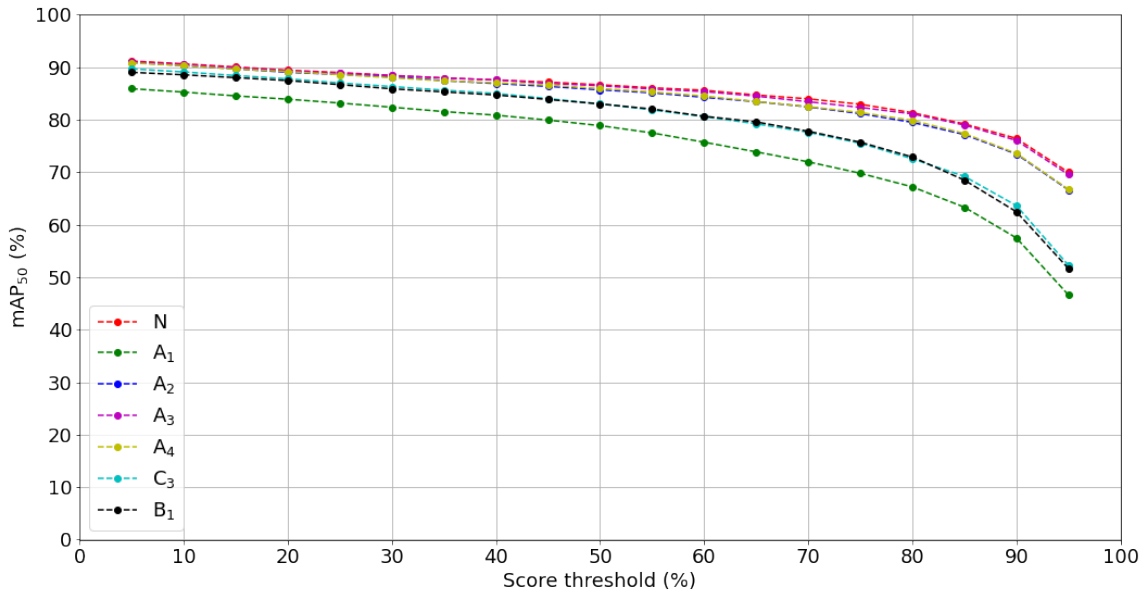


Figure 3.12 – GFD-Retina mean average precision on normal and degraded conditions depending on confidence threshold applied.

the IoU threshold from 50 % to 95 % in steps of 5 %. The higher the threshold, the closer the bounding box coordinates of the network predictions must be to the ground truth objects. Thus the performances will inevitably decrease as this threshold increases. We illustrate the mean average precision of GFD-Retina in normal and degraded conditions depending on IoU threshold applied in figure 3.11.

We can see on this figure that for the configurations A_2 (missing DP modality) and A_3 (missing OF modality), the performances are approximately the same as in normal conditions whatever the IoU threshold. For the A_4 configuration (missing LD modality), we note a deviation from a threshold of 75 %, which remains however rather light, but illustrating the importance of LD modality in the bounding boxes accuracy of GFD-Retina predictions. When the modalities DP and LD are missing (configuration C_3), the deviation from the normal conditions grows, with more important decreases as the IoU threshold is increased (up to a threshold of 80 %). Without either of these two modalities, the network has no longer a depth map on which to base its predictions, which may explain this discrepancy. Moreover, by removing the OF modality (B_1 configuration), the accuracy decreases and the difference in performance between this configuration and the one in normal conditions becomes quite significant (about 7 % when the IoU threshold is 75 %). This shows one of the roles of our modalities DP OF and LD, allowing to refine the predictions of GFD-Retina. Finally, we see that the most important loss in prediction bounding boxes accuracy occurs when the modality RGB is missing (configuration A_1), showing again the main importance of this modality in input data.

3.5.4 CNN confidence in its predictions

When GFD-Retina makes a prediction on input data, it returns a confidence score, between 0% and 100%, signifying the confidence it has in its prediction. With the mAP_{50} metric, we take into account all predictions with a confidence score above 5%. During the calculation of the metric, we sort all these predictions in decreasing order according to their confidence score. The drawback with this metric is that it does not take into account the confidence score of a prediction. Apart from sorting its predictions by descending order, the confidence score has no impact on the metric. For example, let's suppose that 2 networks have exactly the same predictions on input data, all with a confidence score greater than 5 %, but the confidence scores of the predictions of the first network are twice as important as those of the second. Considering that these predictions are correct, we will prefer the first network to the second one because it is more confident of these predictions. However, the mean average precision of these 2 networks will be exactly the same, not reflecting this important difference in confidence.

In order to better analyze these confidence scores, we used the same mean average precision mAP_{50} , with an IoU threshold of 50 %, but varying the threshold score for the selection of predictions from 5 % to 95 %, by 5 % steps. As the threshold score increases, the number of predictions taken into account for the calculation of the metric decreases, causing a loss of performance. Figure 3.12 illustrates the mean average accuracy of GFD-Retina in normal and degraded conditions according to the applied threshold score.

We first notice that the A_3 configuration (OF modality missing) obtains similar performances than in normal conditions. The modality OF has no impact on GFD-Retina when the other modalities are present, and its absence does not affect the performance of the network. The performances of the network in configuration A_2 (DP modality missing) and A_4 (LD modality missing) are roughly the same and are more decreased than those in normal conditions. This gap widens as the score threshold increases. This shows, therefore, that the absence of DP or LD modalities causes a slight loss of confidence in GFD-Retina predictions. However, this loss of confidence remains quite slight in both cases. On the other hand, when these 2 modalities are missing (C_3 configuration), the performance gap is much larger compared to normal conditions. Moreover, this gap increases strongly as the threshold score increases, until it reaches a difference of about 18 % when the threshold score is at 95 %. This loss of confidence is far from negligible, showing the importance for GFD-Retina to have at least one of these 2 modalities. When only the modality RGB remains (configuration B_1), the performance curve is similar to that of configuration C_3. Here, the modality OF does not impact the confidence of GFD-Retina in its predictions. Finally, as seen previously with our other metrics, the absence of the modality RGB (A_1 configuration) affects even more the network performances, whose decreases intensify with a high threshold score.

3.5.5 Overall discussion

With these diverse metrics, we better understand the impact of our modalities within our robust multimodal network, the role they have within GFD-Retina, and the consequences of their absence. RGB modality has a major role: when it is missing, we obtain the worst performances among the degraded configurations studied. The modalities DP,

OF and LD have a more additional role. When one of them is missing, the consequences are minimal, even negligible. On the other hand, if several of them are missing (in particular DP and LD), there are significant decreases in overall performance. One of the advantages is that DP, OF and LD are modalities computed from several sources of information (stereo images, LiDAR, sensor calibration, etc.), which can be noisy and thus disturb their extraction. With the robustness of GFD-Retina, thanks to Random Modality Cut, the absence of one of these modalities will have very slight consequences, and the presence of RGB modality alone allows for a less efficient but correct operation.

Regarding the nature of the losses in case of missing modalities, we do not have objects that are more impacted than others, whatever its class or its size. The main losses observed are in the confidence of GFD-Retina in its predictions. To a lesser extent, its bounding box precision also decrease in degraded conditions. According to the different missing modalities in input, the predictions of the network will be mostly the same, but with a variable precision of the coordinates of their bounding boxes, and a lower confidence score on them.

3.6 Conclusion

In this chapter, we tackled the problem of multimodal neural networks robustness in case of missing modalities in their input data. We found that without an explicit strategy, these networks cannot guarantee a reliable operation in degraded conditions. With our proposed data augmentation methods, our networks have seen their robustness greatly improved, whatever the degraded configuration, without affecting their performance in normal conditions. Among our 3 proposed methods, Random Modality Cut is the most efficient one, offering the best performances in robustness, and being able to be applied to any type of multimodal neural network. Further experiments with this method, on networks for various tasks and with other input modalities, are however necessary to confirm the results obtained with our CNNs for 2D object detection. With in-depth analysis of a robust network, we also found that the loss of performance outside of extreme conditions resulted in predictions of the same nature as in normal conditions, but with lower bounding box coordinate accuracy and prediction confidence.

However, we have studied here a rather binary case: each modality is either present or missing in input data. Our neural networks easily identify the absence of modalities, replaced by null data. This has allowed them to keep a fairly consistent level of performance, via our robustness enhancement strategies, in most degraded configurations. Moreover, since the missing data did not contain any information, they did not disturb our neural networks at any time. It is therefore obvious to obtain high performances, such as those obtained except in extreme conditions.

In reality, sensor failures very rarely cause such modalities. Instead, we can obtain modalities that are noisy to a greater or lesser extent, containing useful but also false information, which can strongly disrupt the network. The related consequences are to be avoided, since the non-detection of objects, as well as the detection of false positives can cause severe accidents. Our previously presented data augmentation methods are therefore a first step to CNN robustness improvement, but they cannot be sufficient in the ADAS context.

Chapter 4

Robustness improvement of multimodal neural networks with noisy modalities

Contents

4.1 Introduction

In our previous chapter, we studied the case where one or more input modalities were missing, replaced by null data of the same type. Our proposed methods obtained satisfactory results on the improvement of the robustness of multimodal neural networks. However, this case study was quite simple: a multimodal CNN can easily ignore a null modality, containing no information, and therefore learns to base its detection on the correct ones. It was then enough to inject partial multimodal data during CNN learning so that it does not discover this phenomenon only in the test phase.

However, this case study is quite far from reality. A failed sensor is not automatically dysfunctional and can therefore return damaged data. These noisy data are of different types and qualities, depending on the nature of the sensor, its failure and the recording conditions. It is possible to predict certain types of common noise in a sensor, but the need for maximum robustness combined with the diversity of noise that a modality can obtain makes the task more complicated.

Noisy modalities are a major problem for neural networks. These damaged data may contain false information generated by its noise. Therefore, the extraction of the true information can be distorted. Moreover, these noisy modalities can disturb the network, resulting in missed detection, due to the lack of information, but also in false detection. Consequently, these noisy modalities lead to a decrease of neural network performances compared to normal conditions without noise. It is therefore important to identify these noisy modalities, in order to process them as well as possible to avoid these disturbances.

One solution is to correct the noisy modality. In the field of image processing, many solutions have been proposed to attenuate the modality noise, ranging from a simple filter to apply to more sophisticated algorithms for more complex cases. These methods are effective when the modalities are slightly noisy, but they can hardly correct the noisiest ones. Moreover, it is necessary to identify the noise in order to use the most suitable method. Finally, it sometimes happens that the input signal is too noisy, and that it does not contain any usable information. The best thing to do then is to ignore this signal, but a neural network can hardly do this if it has not been designed for it.

In this chapter, we address the robustness problem of multimodal neural networks with input data containing unusable modalities. We consider that input data modalities of a CNN will each be either intact or too noisy, containing no information. Faced with such noisy modalities, our objective is not to process them to get any information, but to make the neural network ignore them.

This chapter is organized as follows. In section 4.2, we present Noise Augmentation, a data augmentation method aiming at producing strongly noisy modalities during multimodal CNN training. This approach aims at making our neural networks more robust in degraded conditions where one or more modalities are strongly noisy via different noises. Then, in section 4.3, we introduce Modality Activator Model, a preprocessing CNN for the identification and deactivation of unusable modalities. This strategy aims at eliminating the false information contained in these noisy modalities in order not to disrupt the multimodal network, while learning to deal with the absence of one or more modalities during training. In these two sections, the contributions are presented and experimented on three multimodal networks, in normal and degraded conditions.

4.2 Noise Augmentation

In order to improve the robustness of a neural network when several input modalities are unusable, it is necessary to inject them during its training. Indeed, if a neural network learns only on correct modalities, it will not be able to take in charge damaged data and will be strongly disturbed. The main problem here is therefore to obtain damaged data, noised in different ways and for each of our input modalities.

However, almost all datasets in the context of ADAS offer only correct data, i.e. data recorded in good conditions without sensor failures. Some datasets, mentioned in section 1.4, focus on the robustness problem, but contain noisy images depending on the weather conditions mainly. To our knowledge, there is no dataset including damaged data caused by sensor failures in the literature. This is due to the feasibility of creating such a dataset, requiring several sensors damaged in various ways, which is inevitably expensive.

To overcome this problem, damaged data can be artificially generated from real data. For this, there are many methods, from simple image processing algorithms to Generative Adversarial Networks [42], allowing us to generate many noisy data in different ways and of different qualities from a single sample. The main advantage of these methods is the zero cost to acquire such diverse data. However, these generated data can be far from what we get with real sensor failures. Nevertheless, these generated data remain interesting because of their diversity and their usefulness, since we want the network to ignore them instead of processing them.

In this section, we introduce Noise Augmentation, a data augmentation method aiming at injecting artificially generated unusable modalities during the training of a multimodal CNN. Our method uses different image processing noise algorithms for the generation of unusable data, containing no useful information for the neural network. We analyze the impact of the presence of unusable modalities in multimodal input data on CNNs performances, as well as the contribution of Noise Augmentation on the latter. We evaluate our method on different multimodal CNNs for object detection and analyze its performances under normal and degraded conditions.

4.2.1 Proposed method

Our proposed Noise Augmentation strategy aims to generate unusable modalities during network training by adding a strong noise on a part of input data, simulating sensors failures or malfunctions. Our objectives with this technique are to allow the neural network to learn to cope with unusable modalities that contain either no information or false information, and to make it ignore them, instead of trying to extract information that will be largely erroneous. With Noise Augmentation, each input modality will either be intact (i.e. without any noise), or strongly noisy and therefore unusable. When noise is applied to a modality, this technique randomly selects a type of noise from a set of previously defined noise generation methods, and with random parameters within the noise generation method. Thus, the same unusable modality will be represented in different ways.

However, when learning with a lot of noisy signals, a neural network succeeds in ignoring them, but at the expense of diminished performances in normal conditions. It is therefore necessary to inject a certain number of noisy signals during training to improve

CNN performances in degraded conditions, while minimizing the loss of performance in normal conditions. For that, we define for each modality a Noise Augmentation Rate (NAR), between 0 and 100 %, where 0 means that the modality will always be intact and 100 means that the modality will always be strongly noisy. Each of these rates is independent of the others, considering that the unavailability of a modality should not influence the rest of input data. We can therefore also define a different noise rate for each modality according to these characteristics, with noises of different natures, which brings us more scalability. Moreover, several modalities of the same sample can be noised via different methods. Nonetheless, we lock the possibility to have all our modalities unusable in order to have at least one intact modality in input data.

Our method here is similar in its operation to Random Modality Cut. We can consider RMC moreover as a particular case of Noise Augmentation, where the only defined noise generation method would be a zeroing of the modality. However, Noise Augmentation, via its set of noise generation methods and their randomly chosen parameters, generates very different unusable modalities instead of a single missing modality.

4.2.2 Experimental setup

We experiment Noise Augmentation on 3 multimodal CNNs for object detection, respectively RetinaNet performing early-fusion with DOL images as input, SFD-Retina and GFD-Retina performing early and middle-fusion with RGB and DOL images as input. Our CNNs are trained for multi-class object detection on KITTI 2D Object Detection Dataset, with 8 different classes and taking into account the proposed areas to be ignored. The images of the dataset have been rescaled to have a size of $608 \times 2016 \times 3$.

Concerning RetinaNet, Noise Augmentation is applied on each channel of DOL signal, i.e. DP, OF and LD channels. Our RetinaNet are initialized with a ResNet50 backbone, pre-trained on COCO dataset, and we fine-tuned them on 100 epochs with an Adam optimizer, a learning rate of 10^{-4}, a batchsize of 16 and automatic mixed precision.

For SFD-Retina and GFD-Retina, Noise Augmentation is applied on their input data modalities, including RGB signal and DP, OF and LD channels from DOL signal. Our CNNs backbones (for RGB and DOL signals) are first initialized with a ResNet50 pre-trained on COCO dataset. Inside RetinaNet architecture, these backbones are trained on 100 epochs with an Adam optimizer, a learning rate of 10^{-4} and automatic mixed precision. We apply Noise Augmentation only during DOL backbone training, with the same NAR as SFD/GFD-Retina one, since the whole RGB signal could not be unusable during RGB backbone training. We then select the backbones optimal weights according to their lowest validation loss value, and transfer and freeze them on SFD/GFD-Retina architecture. Finally, we fine-tune full model top layers on 100 epochs with an Adam optimizer, a learning rate of 10^{-5} and automatic mixed precision.

For all our CNNs, we selected their optimal weights according to their lowest value on their validation loss. On top of Noise Augmentation, we used other data augmentation methods such as image rotation, horizontal flip, image translation, image zoom and image shearing. We use 4 Noise Augmentation Rates : 0 % (classical training), 5 %, 25 % and 50 %. We applied the same NAR on each modality input. We train 5 instances of each CNN with each NAR for 5-fold cross validation.

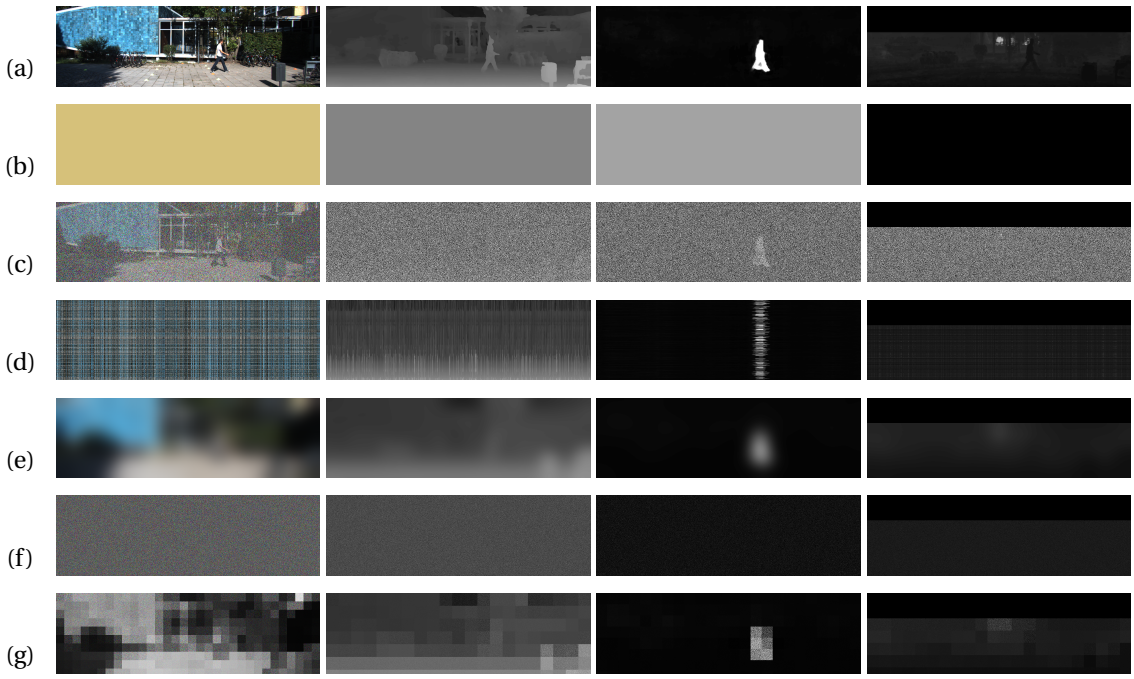


Figure 4.1 – Our CNNs input data (a) (respectively RGB, DP, OF and LD modalities) and example of generated samples by Noise Augmentation with the following noise algorithms : CST (b), RGPN (c), SHUF (d), BLUR (e), RGD (f) and LRGD (g).

4.2.3 Noise generation methods

We define a set of 6 noise generation methods for our Noise Augmentation technique. All the techniques are applied on 1-channel images. The methods used are the following:

— CST: This method returns an image of the same size filled with a single value, between 0 and 255;

— RGPN: Random Gaussian Pixel Noise is applied on pixels, with a zero-mean and a strong standard deviation;

— SHUF: Image rows and/or column are shuffled;

— BLUR : Gaussian Blur is applied on modality with a large kernel;

— RGD : Input modality is replaced by a Random Gaussian Distribution of the same mean and variance;

— LRGD : RGD is applied locally on the modality via a regular square grid.

Figure 4.1 shows an example of all our noise generation methods applied on one sample modality. It is to be noticed that when a modality has several channels, we apply the same noise with the same parameters on all its channels. Moreover, as the LiDAR modality has a black band at the top of the image due to its detection range, we have kept this band during its noise processing.

Network	Input Signal	Rate	mAP	mAP $_{50}$	mAP $_{75}$	Car $_{75}$	Ped $_{50}$	Cyc $_{50}$
RetinaNet	DOL	0 %	**57.49 ± 0.79**	84.25 ± 1.18	**65.18 ± 1.18**	**82.59 ± 0.69**	**83.17 ± 1.60**	84.21 ± 2.49
		5 %	57.48 ± 1.43	**84.96 ± 1.52**	65.09 ± 2.24	82.16 ± 0.39	82.96 ± 1.42	**85.70 ± 2.33**
		25 %	55.89 ± 1.12	84.30 ± 1.14	62.75 ± 1.52	80.60 ± 0.70	82.83 ± 1.47	84.21 ± 3.15
		50 %	51.68 ± 0.58	80.77 ± 1.31	57.30 ± 1.29	77.86 ± 0.53	80.19 ± 1.69	81.16 ± 1.66
SFD-Retina	RGB + DOL	0 %	**68.71 ± 0.87**	92.01 ± 0.55	**79.33 ± 1.29**	**88.68 ± 0.48**	88.97 ± 1.34	92.64 ± 1.00
		5 %	68.36 ± 1.45	92.11 ± 1.04	78.11 ± 2.23	88.56 ± 0.59	**89.07 ± 0.98**	**92.68 ± 1.05**
		25 %	67.56 ± 0.90	**92.40 ± 0.57**	77.85 ± 1.73	87.94 ± 0.67	89.05 ± 1.42	92.41 ± 1.46
		50 %	66.13 ± 1.09	91.65 ± 0.76	75.10 ± 1.33	86.93 ± 0.40	88.84 ± 1.03	91.75 ± 1.68
GFD-Retina	RGB + DOL	0 %	70.25 ± 1.29	92.42 ± 1.09	80.90 ± 1.49	**90.05 ± 0.33**	88.75 ± 1.05	93.10 ± 1.50
		5 %	**70.44 ± 0.92**	92.31 ± 0.86	**81.09 ± 1.11**	89.63 ± 0.33	89.27 ± 0.81	93.47 ± 1.33
		25 %	69.49 ± 1.31	**92.63 ± 0.96**	79.72 ± 1.27	89.13 ± 0.30	89.00 ± 0.92	**93.86 ± 1.51**
		50 %	68.65 ± 1.16	92.29 ± 0.75	78.85 ± 1.30	88.38 ± 0.46	**89.58 ± 0.67**	92.27 ± 1.56

Table 4.1 – Performances of RetinaNet, Stacked and Gated Fusion Double RetinaNet, depending on Noise Augmentation Rate (NAR) in normal conditions.

4.2.4 Performances in normal conditions

We evaluate our CNNs on their validation sets on normal conditions (i.e. without any noise applied), using mAP_{50} , mAP_{75} , mAP, Car_{75} , Ped_{50} and Cyc_{50} metrics. The performances obtained are reported on table 4.1.

First, we see that on the metrics with the least restrictive IoU thresh (mAP_{50} , Ped_{50} and Cyc_{50}), Noise Augmentation has little effect on network performance, regardless of the NAR applied. However, we note a significant decrease in performance for RetinaNet with a NAR of 50 % on these 3 metrics compared to a training without our method, indicating that a too high rate, like our previous data augmentation methods RSC, RCC and RMC, disturbs the network during its training.

On our other 3 metrics (mAP_{75} , mAP, Car_{75}), requiring more accurate bounding boxes, Noise Augmentation negatively impacts our CNNs. With a NAR of 5 %, the performances remain similar to those obtained without our method. On the other hand, with a NAR of 50 %, the decreases observed are significant for each of our CNN. For a NAR of 25 %, we note a tendency to the decrease of the performances also, sometimes significant, but more moderate than for a NAR of 50 %. Noise Augmentation thus decreases bounding boxes accuracy of our networks predictions as the NAR increases. These losses are all the more important for RetinaNet, having no RGB modality in input, compared to our 2 other networks.

4.2.5 Performances in degraded conditions

We are now evaluating our multimodal CNNs on their validation sets in degraded conditions. Given the multitude of degraded configurations, we separate the impact of each noise generation method from the consequences according to the affected modalities. We also evaluate our networks against a new noise generation method not learned in training.

Network		Failure Configurations (with remaining input modalities)								
Input modalities	NAR	N	A_1	A_2	A_3	A_4	B_1	B_2	B_3	B_4
		(RGB,DP,OF,LD)	(DP,OF,LD)	(RGB,OF,LD)	(RGB,DP,LD)	(RGB,DP,OF)	(RGB)	(DP)	(OF)	(LD)
RetinaNet (DOL)	0 %	84.25	-	17.30	29.93	9.04	-	0.55	1.27	1.03
	5 %	84.96	-	67.52	75.16	53.96	-	39.41	10.00	50.51
	25 %	84.30	-	76.68	81.05	71.04	-	63.02	21.02	70.04
	50 %	80.77	-	75.62	78.29	71.40	-	65.13	28.61	70.81
SFD-Retina (RGB + DOL)	0 %	92.01	17.06	73.74	79.19	69.45	59.99	0.01	0.09	0.01
	5 %	92.11	78.50	91.02	91.38	90.85	87.72	31.78	8.11	43.08
	25 %	92.40	80.36	91.72	92.06	91.66	89.15	57.39	18.99	64.28
	50 %	91.65	79.18	91.13	91.39	91.17	89.54	63.09	26.78	67.97
GFD-Retina (RGB + DOL)	0 %	92.42	10.76	71.13	77.32	63.65	53.55	0.00	0.06	0.01
	5 %	92.31	80.76	91.63	91.76	91.41	89.61	33.59	8.80	44.83
	25 %	92.63	84.03	92.33	92.52	92.04	91.04	61.78	20.81	67.97
	50 %	92.29	82.73	91.94	92.02	91.88	90.85	68.28	29.82	72.81

Table 4.2 – Performances (mAP_{50}) of RetinaNet, Stacked and Gated Fusion Double RetinaNet in degraded conditions, depending on Noise Augmentation Rate (NAR). Higher losses comparing to normal conditions (third column) are highlighted with a darker blue.

Performances depending on affected modalities

We analyze our multimodal networks depending on affected modalities as a function of the NAR applied via Noise Augmentation. We define 9 degraded configurations:

— The normal configuration (N), where all the input modalities are correct, i.e. not noisy ;

— 4 configurations where one input modality is noisy (A_1, A_2, A_3 and A_4 for noisy RGB, DP, OF and LD modalities respectively) ;

— 4 configurations where all the input modalities except one are noisy (B_1, B_2, B_3 and B_4 for respectively RGB, DP, OF and LD modalities not noisy).

We use the same methods in our Noise Augmentation set, with equal distribution, for modality noising. For our RetinaNet networks, having no RGB modality as input, the performances in degraded configurations A_1 and B_1 are not calculated. We use mAP_{50} metric for comparison for each CNN. Table 4.2 relates the performances obtained by our CNNs on their validation sets in normal and degraded conditions according to the applied NAR.

At first, without Noise Augmentation, we see that our CNNs have significant performance losses on all degraded configurations. Contrary to our previous experiments with RSC and RMC, where the absence of the modality OF (A_3 configuration) did not impact the network performance, here the applied noise disturbs the network, resulting in reduced performance. In extreme conditions where only one modality except RGB is correct (configurations B_2, B_3 and B_4), the performances are almost null. However, we notice that when the RGB modality is correct (configurations A_2, A_3, A_4 and B_1) for SFD-Retina and GFD-Retina, the losses are more moderate, showing that the RGB modality ensures the basis of the network detection.

With Noise Augmentation, our CNNs become significantly more robust to noise, in all degraded configurations. Concerning our RetinaNet, they obtain the lowest losses in degraded configurations with a NAR of 50 %. However, the rate is too high and leads to a decrease in performance in normal conditions. The best compromise among our tests is a NAR of 25 %. Nevertheless, the decreases observed even with this rate remain significant in almost all the degraded configurations. For SFD-Retina and GFD-Retina, we note two different trends depending on the degraded configurations. When one of the two signals RGB or DOL is correct in the input data (configurations A_1, A_2, A_3, A_4 and B_1), the robustness of the CNNs is strongly improved with a NAR of 5 %. By increasing this rate, we keep quite similar performances in these configurations. On the other hand, in our extreme configurations where only one of the modalities DP, OF or LD is noiseless (configurations B_2, B_3 and B_4), the robustness of the networks grows as the NAR increases. However, as mentioned earlier, an NAR of 50% negatively impacts performance under normal conditions. It is therefore preferable to opt, among our tests, for a NAR of 25%. It is interesting to note that without Noise Augmentation, SFD-Retina is more robust than GFD-Retina in all degraded configurations, despite its rather low performance. On the other hand, when we apply Noise Augmentation on our 2 networks, GFD-Retina becomes more robust, whatever the NAR, on all our degraded configurations.

Network		Noise method							
Input modalities	NAR	Ø	CST	RGPN	SHUF	BLUR	RGD	LRGD	MEAN
RetinaNet (DOL)	0 %	84.25	9.73	0.19	1.16	44.44	3.68	16.40	12.60
	5 %	84.96	44.33	35.65	48.01	62.55	53.95	57.18	50.28
	25 %	84.30	61.88	56.63	61.74	72.83	64.98	67.99	64.34
	50 %	80.77	62.18	58.34	62.79	72.64	65.66	67.69	64.88
SFD-Retina (RGB + DOL)	0 %	92.01	37.35	24.98	34.73	45.94	37.80	42.01	37.13
	5 %	92.11	68.17	65.25	67.98	76.51	73.21	74.12	70.87
	25 %	92.40	75.49	74.37	75.58	82.48	77.98	78.72	77.44
	50 %	91.65	77.04	76.05	76.95	82.80	78.96	79.75	78.59
GFD-Retina (RGB + DOL)	0 %	92.42	33.04	20.23	33.49	43.08	34.53	39.36	33.95
	5 %	92.31	69.16	65.95	70.18	77.71	73.56	75.41	72.00
	25 %	92.63	77.88	76.55	78.47	84.00	79.98	81.46	79.72
	50 %	92.29	79.21	78.34	80.15	84.91	81.16	82.74	81.08

Table 4.3 – Performances (mAP_{50}) of RetinaNet, Stacked and Gated Fusion Double RetinaNet, depending on their Noise Augmentation Rate, according to the type of generated noise of the unusable modalities. Higher losses comparing to normal conditions (third column) are highlighted with a darker blue.

Performances depending on applied noise

We evaluate our multimodal CNNs based on the noise applied to the modalities. To do this, we define 50 % of our test modalities as unusable (ensuring that each test sample had at least one noisy and one correct modality). We then modify these modalities with a single noise generation method present in the Noise Augmentation set (CONST, RGPN, SHUF, BLUR, RGD and LRGD). Afterwards, the average of these results is calculated (MEAN). Table 4.3 shows the performances of RetinaNet, SFD-Retina and GFD-Retina depending on their NAR applied for each noise generation method.

We note that our CNNs have performances that vary with the noise applied on the modalities. RGPN noise gives the worst results since the variance of the noisy modality is much higher than the original one. On the contrary, BLUR noise affects CNNs performances the least, because the noisy modality is closer to the original one. Without Noise Augmentation, the scores of our networks on the different noises vary greatly.

Second, Noise Augmentation drastically improves CNNs robustness, regardless of the noise applied. As the NAR increases, the results in degraded configurations become closer to those in normal conditions. All these results remain logical, since our multimodal networks must learn on unusable modalities to be able to ignore them later. In addition, the performance gap between our different applied noises narrows as the NAR increases. This is positive since our primary goal is to make our networks more robust to all possible types of noise.

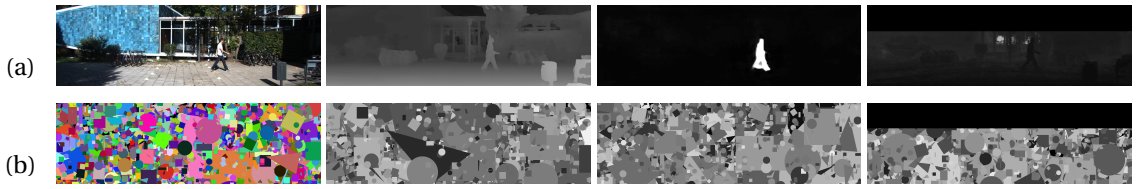


Figure 4.2 – Our CNNs input data (a) (respectively RGB, DP, OF and LD modalities) and example of generated samples by Dead Leaves Pattern method (b).

Network **(Input modalities)**	**NAR**	**NORMAL**	**MEAN**	**DLP**
RetinaNet (DOL)	0 %	84.25	12.60	0.36
	5 %	84.96	50.28	7.06
	25 %	84.30	64.34	13.40
	50 %	80.77	64.88	15.22
SFD-Retina (RGB + DOL)	0 %	92.01	37.13	28.43
	5 %	92.11	70.87	44.70
	25 %	92.40	77.44	48.93
	50 %	91.65	78.59	50.80
GFD-Retina (RGB + DOL)	0 %	92.42	33.95	24.72
	5 %	92.31	72.00	49.37
	25 %	92.63	79.72	53.60
	50 %	92.29	81.08	55.06

Table 4.4 – Performances (mAP_{50}) of RetinaNet, Stacked and Gated Fusion Double RetinaNet, depending on their Noise Augmentation Rate, in normal conditions (NORMAL) and under degraded conditions with our 6 noise generation methods (MEAN) and with DLP method applied on 50 % of dataset modalities. Higher losses comparing to normal conditions (third column) are highlighted with a darker blue.

Performances with unknown noise

The previous results obtained were expected. It is obvious to improve CNNs robustness on specific noises by injecting modalities with these noises during training. It is now interesting to see the impact of our method on a new unknown noise. For this purpose, we followed the same experimental protocol as in the previous section, namely the creation of an evaluation dataset with 50 % of correct modalities, and 50 % of noisy modalities, making sure that each sample has at least one correct and one noisy modality. We noised the unusable modalities with the Dead Leaves Pattern (DLP) method. This method consists in creating an image made of multiple geometrical shapes (polygons, circles, rectangles, etc.), of different colors, sizes. The images produced with this noise generation method do not look at all like those produced by our other methods. Figure 4.2 shows an example of the modalities produced by Dead Leaves Pattern method.

We evaluate our previously trained CNNs on this dataset, and compare the performances obtained with those under normal conditions and the average of the results on the different noises (MEAN). Table 4.4 summarizes the scores obtained according to the CNNs, input modalities and their NAR used. The performances obtained with our noisy modalities via DLP are inevitably less good, which is ultimately quite obvious. Compared to the average on the learned noises (MEAN), we note a rather significant drop in performance for each network, whatever the NAR applied. However, we can see that as the NAR increases, the performance on this dataset also increases. This shows that our method makes our network less perturbed by this noise, even if it did not encounter it during training. However, this trend remains to be confirmed with other noise generation methods. We also note that RetinaNet is more negatively impacted by this new noise compared to SFD-Retina and GFD-Retina. We believe that this is due to the absence of RGB modality in input data, which provides better detection robustness in many degraded configurations.

Noise Augmentation	Normal Conditions		Degraded Conditions							
Set Methods	**mAP**	**mAP $_{50}$**	**CST**	**RGPN**	**SHUF**	**BLUR**	**RGD**	**LRGD**	**MEAN**	**DLP**
CST	52.85 ± 0.98	81.76 ± 0.92	69.88	0.61	3.97	54.49	6.60	20.11	25.94	2.98
RGPN	51.77 ± 1.23	80.69 ± 0.97	16.70	58.52	9.37	47.57	44.68	34.60	35.24	2.83
SHUF	52.36 ± 0.94	81.43 ± 1.00	12.69	14.08	67.82	44.88	65.67	63.78	44.82	3.25
BLUR	**56.24 ± 1.13**	**84.12 ± 1.54**	9.69	0.72	2.52	79.63	6.37	22.19	20.19	0.56
RGD	53.46 ± 1.02	82.33 ± 1.16	13.78	14.20	13.62	45.78	68.85	41.54	32.96	1.08
LRGD	54.36 ± 1.55	82.43 ± 1.36	7.86	1.91	15.27	46.71	30.71	69.79	28.71	1.44
CST, RGPN, SHUF, BLUR, RGD, LRGD	51.68 ± 0.58	80.77 ± 1.31	62.18	58.34	62.79	72.64	65.66	67.69	65.09	15.22

Table 4.5 – Performances (mAP_{50}) of RetinaNet with a NAR of 50 % depending on their Noise Augmentation set of methods, under degraded conditions with a specific noise applied on 50 % of dataset modalities. Higher losses comparing to normal conditions (third column) are highlighted with a darker blue.

4.2.6 Impact of Noise Augmentation set of methods

We have seen in the previous section that when a new noise not learned by the CNNs is applied, the performances of our networks are better than without our method, but remain quite low. A solution could then be to use a maximum of noise generation methods during the training of our neural networks. The diversity of the noise generation methods would then allow the neural networks to be robust to any kind of noise. However, we found that with a too large NAR, our networks performed less well under normal conditions. This is even more visible with RetinaNet, taking as input DP, OF and LD modalities. The mean average precision mAP_{50} of the network in normal conditions goes from 84.25 % without Noise Augmentation, to 80.77 % with our method and a NAR of 50 %, that is a significant loss of 3.48 %.

We compare our RetinaNet network using DOL images as input according to the set of Noise Augmentation methods. For this, we trained 6 RetinaNet with a 50 % Noise Augmentation Rate, following the same experimental protocol as detailed in section 4.2.2. These 6 networks were each trained with a different Noise Augmentation set of methods, containing a single noise generation method (respectively CST, RGPN, SHUF, BLUR, RGD, LRGD). For each of these networks, we performed a 5-fold cross-validation. We compare these networks trained with RetinaNet with a NAR of 50 %, and a Noise Augmentation set of methods including our 6 noise generation methods. These 7 networks are evaluated under normal and degraded conditions by applying one of these 6 noises on 50 % of the modalities. The average mAP_{50} of our networks on these 6 noises (MEAN), as well as their performances with DLP noise applied are computed. Table 4.5 shows the results obtained by our networks in normal conditions (mAP and mAP_{50}), and in degraded conditions (mAP_{50}).

We can see that in normal conditions, the network trained with the 6 noise generation methods obtains almost the worst performances, in mAP and mAP_{50}, on a par with the one using RGPN in noise generation. The accumulation of noise generation methods used in data augmentation impacts negatively the performances in normal conditions. The size of the set of noise generation methods must therefore be controlled in order to limit this drop in performance.

In degraded conditions, we note that for each noise generation method, the neural network having only this same method in data augmentation obviously obtains the best performances. On the other hand, this same network performs worse on all the other noise generation methods in degraded conditions. These decreases are however variable depending on the noise in training and the one in test. For example, our RetinaNet trained on unusable noisy modalities via SHUF performs quite well on RGD and LRGD noises in evaluation. It is therefore also interesting to select well the noise generation methods that generalize the best, but this requires a lot of experimentation in order to determine them.

The network having learned with all the methods in data augmentation obtains the best performance on average (MEAN), well above the other networks. In detail, it obtains satisfactory performance in degraded conditions on each noise generation method, but most of the time below the networks having learned with only this method in data augmentation. This observation is easily explained: the larger the set of methods, the fewer samples will be seen by the network in training for each type of noise.

With modalities noised by DLP, we see a clear difference between our network having learned with the 6 methods of noise generation in data augmentation (15.22 %), against those having learned on only one of these 6 methods (at most 3.23 %). This gap of at least 11 % is significant, despite very low scores for each of our networks. This shows that the size of the set of noise generation methods improves the robustness of the networks to unknown noise. Indeed, by generating fairly diverse noisy modalities, Noise Augmentation ensures that our CNNs do not ignore a specific noise. Therefore, we believe that using even more noise generation methods with our method could further improve this robustness. However, this remains to be confirmed with additional experiments.

With this experiment, we see the limit of our contribution. With the objective of supporting any noise, the consequences on the learning of the networks are not negligible, and the performance decreases, especially in normal conditions, are significant.

4.2.7 Overall discussion

Through our experiments, we have seen the positive impact of Noise Augmentation on the robustness of our neural networks. Our method allows us to improve the performances of our 3 multimodal neural networks, whatever the degraded configuration and whatever the noise applied. The overall robustness becomes even better as the NAR increases. However, we note significant performance decreases in normal conditions when this rate is too high. The generated noises disturb our networks during their training, and we must therefore be careful not to inject too many unusable modalities in order to preserve the performances in normal conditions. Compared to our previous problem in chapter 3, where the modalities were either present or missing, but never noisy, the improvements brought by our data augmentation method are a little bit more backward, showing the negative impact that noisy modalities have, both in learning and in testing phase. Moreover, this impact strongly depends on the type of noise.

Using Noise Augmentation to improve the robustness of the CNNs in degraded conditions remains conclusive, but limited to it alone. Indeed, there is an infinity of possible noises, potentially reproducible with Noise Augmentation. However, the larger the set of noise generation methods of our data augmentation technique, the more the performances in normal conditions are reduced, as well as the robustness of the network on each type of noise. This requires even larger neural networks that can handle all the diversity of noise making the modalities unusable, which complicates the real time application. Therefore, a compromise has to be found between performance in normal conditions, speed of execution and robustness to noise.

4.3 Modality Activator Model

In order to make a CNN robust to unusable modalities, injecting noisy modalities as input is not enough, as shown with Noise Augmentation. The main problem of this strategy is the too large diversity of possible noises, generating disturbances in neural network training and leading to reduced overall performances in normal and degraded conditions. To remedy this, a preprocessing step is necessary.

A modality correction to attenuate the noise for our problem is unnecessary. The original information of the latter is much too noisy, and it is better to ignore it. However, if the correction is to be avoided in our case, an identification of the unusable modalities is interesting, since a filtering of the latter can then be performed. By deactivating these disturbing modalities, i.e. by setting them to zero, they no longer contain false information, and the neural network will be better able to handle them.

Whatever its task, a multimodal CNN often goes through a feature extraction step and a modality fusion step. In order to identify a unusable modality due to a strong noise, a feature extraction step is an essential basis for being able to then make a decision on its treatment. It is therefore interesting to use this feature extraction to identify unusable modalities, in order to deactivate them. However, it may happen that a neural network fuses the input modalities first before extracting their features, namely during an early fusion. In this case, a noisy modality is more difficult to identify in the extracted features, and the other fused modalities are impacted by the noise as well. Therefore, we believe that it is preferable to dissociate the identification and processing of a modality noise from the multimodal neural network, i.e. via a preprocessing step.

In this section, we introduce Modality Activator Model (ModAM), a CNN that preprocess input modalities, identify and deactivate those unusable. This strategy aims at eliminating the false information contained in these noisy modalities in order not to disrupt the multimodal network, while learning to deal with the absence of one or more modalities during training. We experiment ModAM with Noise Augmentation on several CNNs in normal and degraded conditions.

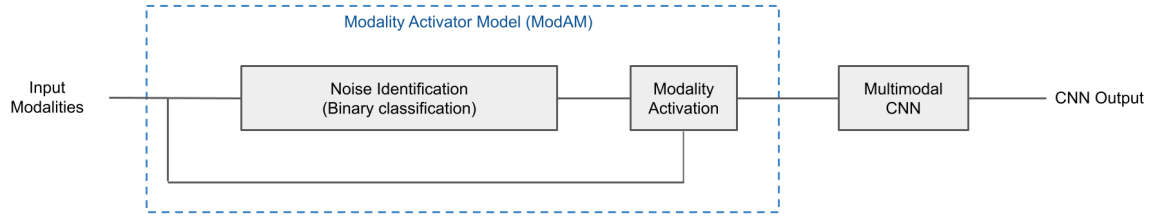


Figure 4.3 – Modality Activator Model architecture.

4.3.1 Proposed method

Our proposed Modality Activator Model (ModAM) intervenes upstream of a multimodal CNN to analyze its modalities and eliminate those that are noisy. Its architecture is illustrated in figure 4.3. ModAM noise identification is made via a binary classifier, taking as input each modalities and outputting modality activation factors. For one modality input, its modality activation factor will be either 1 (meaning the modality is correct and usable) or 0 (meaning the modality is unusable, due to strong noise). Then, ModAM activates correct input modalities and deactivates unusable ones by multiplying each channel of input data with its modality activation factor. Assuming that the model works perfectly, all channels of noisy modalities will be replaced by a null image of the same size. When a multi-channel modality obtains different modality activation factors for its channels, ModAM sets all these factors to 0 by default, preferring a false negative (non-noisy modality but set to zero) to a false positive (unfiltered noisy modality). The idea behind ModAM is to simplify all the noisy modalities by the same image containing no information, easier to handle for the multimodal CNN.

Figure 4.4 shows an example of ModAM process. We can see that among the input modalities, from top to down, the first and third ones are correct whereas the second and fourth ones are unusable. After noise identification made by ModAM classifier, each input modality have a corresponding modality activator factor: 1 fot the correct ones, and 0 for the noisy ones. We consider that for the first modality, having multiple channels, each of them got the same modality activator factor. Then, with modality activation, we obtain the corrected input modalities, where the first and third one, initially correct, remain the same, and the second and fourth modalities, initially noisy, are deactivated, resulting in a black image of the same dimension for each of them.

The main advantage of ModAM is that it is applicable to all types of multimodal CNN using image-based modalities as input, whatever the type of fusion applied. Moreover, a very noisy modality is easily distinguishable from a non-noisy modality, which means that ModAM classifier can be very light, slightly slowing down the execution time.

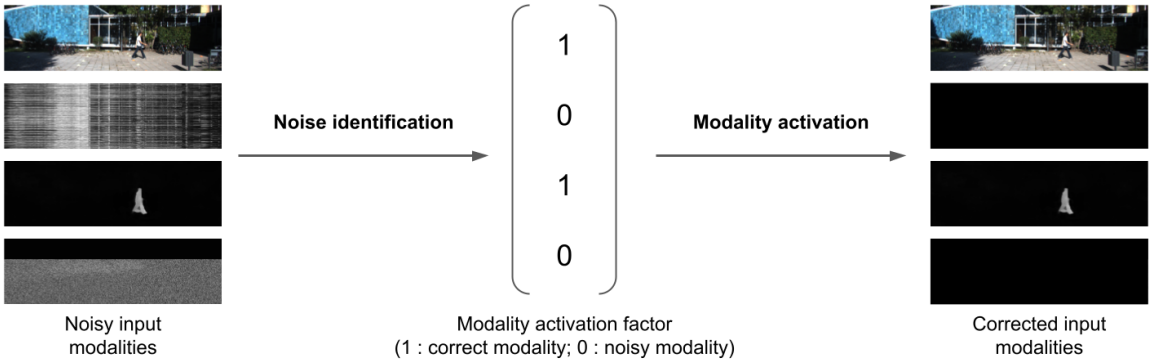


Figure 4.4 – Detailed operation of Modality Activator Model.

4.3.2 Experimental setup

In order to analyze the contribution of ModAM on the robustness of our networks, we experiment our contribution on the 3 multimodal CNNs: RetinaNet with DOL images, SFD-Retina and GFD-Retina taking RGB and DOL images as input. We used Noise Augmentation with the same noise generation method as those presented in 4.2.3 and 4 NAR: 0 % (classical learning), 5 %, 25 % and 50 %. We followed the same experimental protocol as the one detailed in section 4.2.2.

We first trained 5 instances of ModAM classifier for 5-fold cross validation. It takes as input 1-channel modalities and outputs its probability of being usuable or not. Each CNN were initialized with AlexNet [55] pre-trained on ImageNet [27]. Then we modify their first convolutional layer for taking 1-channel images as input, and their last layer for outputting only one value per modality. We next fine-tuned them on 10 epochs with Adam optimizer and a learning rate of 10^{-6}. Finally, for our multimodal CNNs using ModAM for modalities preprocessing, we transfer and freeze ModAM classifier weights before object detectors training.

Network	Input Signal	Rate	ModAM	mAP	mAP $_{50}$	mAP $_{75}$	Car $_{75}$	Ped $_{50}$	Cyc $_{50}$
RetinaNet	DOL	0 %	N	57.49 ± 0.79	84.25 ± 1.18	65.18 ± 1.18	**82.59 ± 0.69**	83.17 ± 1.60	84.21 ± 2.49
			Y	**57.79 ± 1.19**	84.35 ± 1.12	**66.11 ± 1.88**	82.49 ± 0.89	83.33 ± 0.89	83.95 ± 3.02
		5 %	N	57.48 ± 1.43	**84.96 ± 1.52**	65.09 ± 2.24	82.16 ± 0.39	82.96 ± 1.42	**85.70 ± 2.33**
			Y	57.27 ± 0.94	84.34 ± 1.15	64.72 ± 1.97	82.37 ± 0.82	**83.84 ± 0.78**	84.94 ± 2.58
		25 %	N	55.89 ± 1.12	84.30 ± 1.14	62.75 ± 1.52	80.60 ± 0.70	82.83 ± 1.47	84.21 ± 3.15
			Y	56.90 ± 0.79	84.83 ± 0.83	64.38 ± 1.15	81.09 ± 0.61	83.25 ± 1.12	84.49 ± 3.87
		50 %	N	51.68 ± 0.58	80.77 ± 1.31	57.30 ± 1.29	77.86 ± 0.53	80.19 ± 1.69	81.16 ± 1.66
			Y	54.08 ± 1.00	82.40 ± 1.01	60.66 ± 1.57	78.75 ± 0.61	82.34 ± 0.90	82.02 ± 2.63
SFD-Retina	RGB + DOL	0 %	N	**68.71 ± 0.87**	92.01 ± 0.55	**79.33 ± 1.29**	88.68 ± 0.48	88.97 ± 1.34	92.64 ± 1.00
			Y	68.58 ± 1.24	92.06 ± 0.84	78.68 ± 1.49	**88.96 ± 0.40**	89.08 ± 1.15	92.28 ± 1.35
		5 %	N	68.36 ± 1.45	92.11 ± 1.04	78.11 ± 2.23	88.56 ± 0.59	89.07 ± 0.98	92.68 ± 1.05
			Y	68.58 ± 0.91	92.25 ± 0.74	79.13 ± 0.80	88.78 ± 0.34	**89.12 ± 0.91**	**92.85 ± 0.91**
		25 %	N	67.56 ± 0.90	**92.40 ± 0.57**	77.85 ± 1.73	87.94 ± 0.67	89.05 ± 1.42	92.41 ± 1.46
			Y	68.06 ± 0.85	92.07 ± 0.77	78.60 ± 1.51	87.98 ± 0.63	88.76 ± 1.36	92.63 ± 1.26
		50 %	N	66.13 ± 1.09	91.65 ± 0.76	75.10 ± 1.33	86.93 ± 0.40	88.84 ± 1.03	91.75 ± 1.68
			Y	66.93 ± 0.87	91.46 ± 0.71	76.64 ± 0.94	87.19 ± 0.57	88.61 ± 1.36	92.07 ± 1.29
GFD-Retina	RGB + DOL	0 %	N	70.25 ± 1.29	92.42 ± 1.09	80.90 ± 1.49	**90.05 ± 0.33**	88.75 ± 1.05	93.10 ± 1.50
			Y	69.71 ± 1.28	92.12 ± 0.56	79.93 ± 1.66	89.83 ± 0.20	88.83 ± 1.13	92.70 ± 1.15
		5 %	N	**70.44 ± 0.92**	92.31 ± 0.86	**81.09 ± 1.11**	89.63 ± 0.33	89.27 ± 0.81	93.47 ± 1.33
			Y	70.24 ± 1.66	92.69 ± 0.96	80.60 ± 2.33	89.76 ± 0.68	89.37 ± 1.14	92.87 ± 1.75
		25 %	N	69.49 ± 1.31	92.63 ± 0.96	79.72 ± 1.27	89.13 ± 0.30	89.00 ± 0.92	**93.86 ± 1.51**
			Y	70.01 ± 1.15	**92.76 ± 0.75**	80.57 ± 1.78	89.15 ± 0.62	**89.62 ± 1.20**	92.49 ± 1.22
		50 %	N	68.65 ± 1.16	92.29 ± 0.75	78.85 ± 1.30	88.38 ± 0.46	89.58 ± 0.67	92.27 ± 1.56
			Y	69.01 ± 1.02	92.43 ± 0.50	79.52 ± 1.53	88.19 ± 0.73	89.28 ± 0.87	92.54 ± 1.26

Table 4.6 – Performances of RetinaNet, Stacked and Gated Fusion Double RetinaNet, depending on Noise Augmentation Rate (NAR) and the presence of ModAM (Y for Yes, N for No) in normal conditions.

4.3.3 Performances in normal conditions

We now evaluate our CNNs on their validation sets on normal conditions without unusable modalities. We use the mean average precision mAP_{50} , mAP_{75} and mAP, and the average precision Car_{75} , Ped_{50} and Cyc_{50} as our metrics. Table 4.6 shows the performances of RetinaNet, SFD-Retina and GFD-Retina depending on their NAR (0%, 5%, 25% or 50 %) and the presence of ModAM for modalities preprocessing.

We first notice that without Noise Augmentation (i.e. with a NAR of 0%), the addition of ModAM does not significantly modify the performances of our CNNs. We found that ModAM did not disable any modalities. Since the input modalities are all correct, this behavior was intended and the addition of ModAM does not influence the results.

When injecting unusable modalities with Noise Augmentation with a NAR of 5 %, we find no significant difference also between networks using ModAM and the others, on all metrics. On the other hand, with a higher NAR (25 % or 50 %), while we observed that the CNNs using only Noise Augmentation lost in performances, on the mean average precision mAP and mAP_{75} in particular, those using ModAM in modalities preprocessing have less important decreases in performance. These findings are even more visible for the RetinaNet network, where the modality RGB is not used as input data. While the false information contained by the unusable modalities disturbs the network during training, their deactivation by ModAM facilitates neural network learning, since the CNN only have to manage their absence.

Regarding our CNNs speed, the addition of ModAM slows them down slightly. For comparison purposes only, the inference time of RetinaNet goes from 53 ms per input sample without ModAM to 57 ms with it, an increase of 7.5 %. For SFD-Retina, the inference time increases from 70 ms without ModAM to 77 ms with it, an increase of 10 %. Finally, the inference time of GFD-Retina increases from 135 ms without ModAM to 144 ms with it, an increase of 6.67 %. These increases in computing time, at most 10 %, are very reasonable for real time object detection.

4.3.4 Performances in degraded conditions

In the same way as in section 4.2.5, we evaluate our multimodal CNNs according to the affected modalities, the applied noise and in front of a new noise not learned in training. We also analyze the identification of noisy modalities by ModAM classifier.

Identification of unusable modalities with ModAM

First, we evaluate ModAM classifier for its task of identification of unusable modalities. On its validation set, each classifier trained achieved a 100 % accuracy. This proves that the identification of unusable modalities, at least in our problem with 6 types of noise, is quite easy, and that it is not necessary to use a state-of-the-art CNN to perform this task. In a more complex case, there is potentially no need to use a larger classifier, since they converged here with a very low learning rate (10^-6) in only 10 epochs. To verify this, further experiments are needed.

Network			Failure Configurations (with remaining input modalities)							
Input modalities	NAR	ModAM	N (RGB,DP,OF,LD)	A_1 (DP,OF,LD)	A_2 (RGB,OF,LD)	A_3 (RGB,DP,LD)	A_4 (RGB,DP,OF)	B_1 (RGB)	B_2 (DP)	B_3 (OF)	B_4 (LD)
RetinaNet DOL	0 %	N	84.25	-	17.30	29.93	9.04	-	0.55	1.27	1.03
		Y	84.35	-	42.89	70.28	19.47	-	2.15	3.20	28.17
	5 %	N	84.96	-	67.52	75.16	53.96	-	39.41	10.00	50.51
		Y	84.34	-	78.88	81.86	67.72	-	58.80	15.23	72.45
	25 %	N	84.30	-	76.68	81.05	71.04	-	63.02	21.02	70.04
		Y	84.83	-	81.87	83.58	76.17	-	73.14	26.74	78.52
	50 %	N	80.77	-	75.62	78.29	71.40	-	65.13	28.61	70.81
		Y	82.40	-	80.75	82.57	76.63	-	75.32	36.47	78.90
SFD-Retina RGB + DOL	0 %	N	92.01	17.06	73.74	79.19	69.45	59.99	0.01	0.09	0.01
		Y	92.06	22.94	85.54	90.36	68.21	71.86	1.57	0.39	4.43
	5 %	N	92.11	78.50	91.02	91.38	90.85	87.72	31.78	8.11	43.08
		Y	92.25	80.61	91.43	92.02	91.29	88.35	51.29	12.92	66.41
	25 %	N	92.40	80.36	91.72	92.06	91.66	89.15	57.39	18.99	64.28
		Y	92.07	82.82	91.84	91.84	91.50	89.91	70.14	24.49	75.53
	50 %	N	91.65	79.18	91.13	91.39	91.17	89.54	63.09	26.78	67.97
		Y	91.46	82.17	91.50	91.33	91.14	90.07	73.99	34.76	77.84
GFD-Retina RGB + DOL	0 %	N	92.42	10.76	71.13	77.32	63.65	53.55	0.00	0.06	0.01
		Y	92.12	15.84	85.56	90.64	65.60	60.54	0.72	0.24	1.70
	5 %	N	92.31	80.76	91.63	91.76	91.41	89.61	33.59	8.80	44.83
		Y	92.69	82.16	91.80	92.33	92.06	89.51	51.44	12.78	67.62
	25 %	N	92.63	84.03	92.33	92.52	92.04	91.04	61.78	20.81	67.97
		Y	92.76	84.70	92.19	92.56	92.69	90.81	72.62	25.69	77.75
	50 %	N	92.29	82.73	91.94	92.02	91.88	90.85	68.28	29.82	72.81
		Y	92.43	83.82	92.37	92.60	92.07	91.02	76.42	35.95	79.64

Table 4.7 – Performances (mAP_{50}) of RetinaNet, Stacked and Gated Fusion Double RetinaNet in degraded conditions, depending on Noise Augmentation Rate (NAR) and the presence of ModAM (Y for Yes, N for No). Higher losses comparing to normal conditions (fourth column) are highlighted with a darker blue.

Performances depending on affected modalities

We analyze our multimodal networks under degraded conditions where one or several modalities are noisy and therefore unusable. We use the 9 degraded conditions listed previously in section 4.2.5. The unusable modalities are noised with the same methods as those used in our Noise Augmentation set (namely CST, RGPN, SHUF, BLUR, RGD and LRGD), with equal distribution. The mAP_{50} of our CNNs on their validation sets in degraded conditions, depending on their NAR and the presence of ModAM for modalities preprocessing, are shown in table 4.7.

With classical learning without injected unusable modalities, we already see clear improvements in degraded configurations for our networks using ModAM compared to the others. These performance increases are very variable depending on the networks and the degraded configurations, but it shows that a noisy modality disturbs our networks more than a deactivated modality. However, ModAM alone is not enough to make our CNNs robust, especially in extreme conditions when only one of the modalities DP, OF or LD is noiseless (configurations B_2, B_3 and B_4).

On the other hand, by applying Noise Augmentation during the training of our CNNs, we find that ModAM significantly improves their robustness to noisy modalities in the majority of cases. For RetinaNet, whatever the degraded configuration and the NAR used, the performance increases significantly. For SFD-Retina and GFD-Retina, the improvements are weaker, in particular when RGB modality is correct (A_2, A_3, A_4 and B_1 configurations).

The particular advantage with ModAM is that we can lower the NAR and keep good robustness performances. Without our preprocessing model, in order to handle a maximum of noise types, it is necessary to inject a large part of unusable modalities, and this inevitably impacts the performances in normal conditions. With an efficient preprocessing model, the noisy modalities become zero, and the data augmentation method Noise Augmentation finally acts on our CNNs in the same way as Random Modality Cut.

Network			Noise method							
Input modalities	NAR	ModAM	Ø	CST	RGPN	SHUF	BLUR	RGD	LRGD	MEAN
RetinaNet (DOL)	0 %	N	84.25	9.73	0.19	1.16	44.44	3.68	16.40	12.60
		Y	84.35	29.54	29.54	29.54	29.54	29.54	29.54	29.54
	5 %	N	84.96	44.33	35.65	48.01	62.55	53.95	57.18	50.28
		Y	84.34	62.68	62.68	62.68	62.68	62.68	62.68	62.68
	25 %	N	84.30	61.88	56.63	61.74	72.83	64.98	67.99	64.34
		Y	84.83	69.72	69.72	69.72	69.72	69.72	69.72	69.72
	50 %	N	80.77	62.18	58.34	62.79	72.64	65.66	67.69	64.88
		Y	82.40	71.39	71.39	71.39	71.39	71.39	71.39	71.39
SFD-Retina (RGB + DOL)	0 %	N	92.01	37.35	24.98	34.73	45.94	37.80	42.01	37.13
		Y	92.06	41.45	41.45	41.45	41.45	41.45	41.45	41.45
	5 %	N	92.11	68.17	65.25	67.98	76.51	73.21	74.12	70.87
		Y	92.25	76.63	76.63	76.63	76.63	76.63	76.63	76.63
	25 %	N	92.40	75.49	74.37	75.58	82.48	77.98	78.72	77.44
		Y	92.07	80.84	80.84	80.84	80.84	80.84	80.84	80.84
	50 %	N	91.65	77.04	76.05	76.95	82.80	78.96	79.75	78.59
		Y	91.46	82.46	82.46	82.46	82.46	82.46	82.46	82.46
GFD-Retina (RGB + DOL)	0 %	N	92.42	33.04	20.23	33.49	43.08	34.53	39.36	33.95
		Y	92.12	38.05	38.05	38.05	38.05	38.05	38.05	38.05
	5 %	N	92.31	69.16	65.95	70.18	77.71	73.56	75.41	72.00
		Y	92.69	77.28	77.28	77.28	77.28	77.28	77.28	77.28
	25 %	N	92.63	77.88	76.55	78.47	84.00	79.98	81.46	79.72
		Y	92.76	82.52	82.52	82.52	82.52	82.52	82.52	82.52
	50 %	N	92.29	79.21	78.34	80.15	84.91	81.16	82.74	81.08
		Y	92.43	83.42	83.42	83.42	83.42	83.42	83.42	83.42

Table 4.8 – Performances (mAP_{50}) of RetinaNet, Stacked and Gated Fusion Double RetinaNet, depending on their Noise Augmentation Rate and the presence of ModAM (Y for Yes, N for No), according to the type of generated noise of the unusable modalities. Higher losses comparing to normal conditions (fourth column) are highlighted with a darker blue.

Performances depending on applied noise

Our CNNs are evaluated depending on the noise applied. We define 50 % of our test modalities as unusable making sure that each test sample had at least one noisy and one correct modality. These unusable modalities are noised with one of our Noise Augmentation set of methods, namely CST, RGPN, SHUF, BLUR, RGD and LRGD. The average of these results according to the noise applied is calculated (MEAN). Table 4.8 shows the performances of our CNNs depending on their NAR and the presence of ModAM for each noise generation method.

Knowing our ModAM classifiers have reached an accuracy of 100 % on their validation set, each unusable modality, whatever the noise applied, is disabled during our evaluations. As a result, the networks using ModAM have identical results for each noise generation method. Our contribution here brings a considerable advantage to CNNs since with this strategy of identifying and deactivating unusable modalities, we obtain a uniformity in CNNs robustness, whatever the type of noise. As a result, the complexity of the problem becomes simpler, in particular when dealing with very disturbing noises.

In comparison with networks without ModAM, those using it obtain on average better robustness performances, whatever the network and the applied NAR. The most important improvements are on the noise RGPN, which is the most disturbing noise among our noise generation methods. Except for the noise BLUR, the addition of ModAM in preprocessing improves the performance of CNNs for all types of noise of our set of Noise Augmentation methods. For the noise type BLUR, we obtain less good performances with ModAM. These decreases are less important for SFD-Retina and GFD-Retina. Moreover, as the NAR increases, the performance gap also decreases. This observation shows that improvements are possible for ModAM on its deactivation module, by replacing the noisy modalities with a fixed matrix more convenient for each modality rather than a zero image.

Network		Without ModAM			With ModAM		
Input modalities	NAR	NORMAL	MEAN	DLP	NORMAL	MEAN	DLP
RetinaNet (DOL)	0 %	84.25	12.60	0.36	84.35	29.54	0.35
	5 %	84.96	50.28	7.06	84.34	62.68	0.65
	25 %	84.30	64.34	13.40	84.83	69.72	0.94
	50 %	80.77	64.88	15.22	82.40	71.39	0.92
SFD-Retina (RGB + DOL)	0 %	92.01	37.13	28.43	92.06	41.45	28.09
	5 %	92.11	70.87	44.70	92.25	76.63	33.33
	25 %	92.40	77.44	48.93	92.07	80.84	37.48
	50 %	91.65	78.59	50.80	91.46	82.46	38.42
GFD-Retina (RGB + DOL)	0 %	92.42	33.95	24.72	92.12	38.05	23.82
	5 %	92.31	72.00	49.37	92.69	77.28	32.64
	25 %	92.63	79.72	53.60	92.76	82.52	39.34
	50 %	92.29	81.08	55.06	92.43	83.42	44.15

Table 4.9 – Performances (mAP_{50}) of RetinaNet, Stacked and Gated Fusion Double RetinaNet, depending on their Noise Augmentation Rate and the presence of ModAM (Y for Yes, N for No), in normal conditions (NORMAL) and under degraded conditions with our 6 noise generation methods (MEAN) and with DLP method applied on 50 % of dataset modalities. Higher losses comparing to normal conditions (third column for networks without ModAM, sixth column for those with ModAM) are highlighted with a darker blue.

Performances with unknown noise

Finally, we analyze the impact of ModAM on the robustness of multimodal CNNs when an unknown noise is applied on the unusable modalities. First, we evaluated each instance of ModAM classifier on noisy modalities with DLP. It is important to note that our classifiers did not learn with noisy modalities generated with this method. Despite its maximum accuracy on their validation sets, our networks here classify all noisy modalities via DLP as correct modalities, thus applying no deactivation on them. Indeed, DLP produces unusable modalities bearing little resemblance to those generated via our other noise generation methods, and our classifiers here have learned to identify these 6 noises specifically. Improvements are thus needed on the development of ModAM classifiers, so that they learn to identify the correct modalities rather than the noisy ones.

Next, in the same way as our previous experiment, we evaluate our CNNs on their validation set with DLP noise applied. 50 % of the modalities of the dataset are noised with DLP. We compare CNNs performances depending on their NAR and the presence of ModAM. The obtained results are shown in table 4.9.

Since ModAM does not perform any filtering for this type of noise, we inevitably obtain low performances for each of our CNNs. Without Noise Augmentation, the networks using ModAM obtain similar performances as those without preprocessing. This observation is obvious since these networks have all learned on correct modalities, and have been evaluated with the same noisy data, without any deactivation by ModAM. On the other hand, we see that our networks without ModAM fare better when we apply Noise Augmentation. Indeed, our networks using ModAM learned only with either correct or deactivated modalities and thus null. The noisy unusable modalities with DLP therefore disturb the networks strongly, explaining these performances. For our CNNs without ModAM, the diversity of noisy and non deactivated modalities in training allowed them to learn to ignore these disturbing modalities, and are therefore less impacted by this new noise.

We also notice that our SFD-Retina and GFD-Retina networks are slightly more robust than RetinaNet to this unlearned noise. This is explained by the presence of the modality RGB in input, which when it is correct, ensures a minimal detection.

4.3.5 Overall discussion

The strategy of identifying and deactivating unusable modalities via Modality Activator Model, combined with Noise Augmentation, improves the robustness of our CNNs. By simplifying these disturbing modalities by null modalities without any information, our networks obtain on average better performances in degraded conditions. Their performance in normal conditions is also less impacted, thanks to the removal of the false information contained in the noisy modalities. Moreover, their Noise Augmentation Rates applied on the different modalities can be lower, while ensuring a satisfactory robustness in degraded conditions. Finally, the addition of ModAM slightly increases the prediction time of our networks but does not prevent the possibility of real-time execution.

Nevertheless, we also found an important flaw in this strategy. When an unknown noise, not learned by ModAM, is applied on our modalities, our CNNs are not robust anymore, and their performances drop drastically. Therefore, the use of ModAM requires a

lot of work to make it as reliable as possible. In our experiments, we only used 6 noise generation methods, and reached an accuracy of 100 % in only 10 iterations with a light classification network (AlexNet). By using a more advanced classifier, and by generating many more different noises, it is possible to reduce the impact of this flaw. To achieve these improvements, it will be necessary to have a larger set of data for training, and in particular more noisy images, real or artificial. It will also be interesting to study in depth the functioning of the trained classifier, in order to determine the operations it performs to make its decision. Nevertheless, it will be difficult to guarantee a total reliability in front of the diversity of possible noises which can mislead the classifier and consequently disturb the CNN.

4.4 Conclusion

In this chapter, we focus on the problem of robustness of multimodal neural networks when one or more modalities are strongly noisy and therefore unusable. In contrast to a missing modality, we found that a noisy modality disturb our neural networks because of the false information it contains. This leads to drastic performance decreases of multimodal neural networks in degraded conditions. The use of Noise Augmentation allows us to partially solve this problem by injecting unusable modalities during neural network training. However, the robustness improvements observed are often accompanied by performance losses in normal conditions. Moreover, this contribution alone is not sufficient, given the diversity of possible noises.

The combination of Noise Augmentation with Modality Activator Model has allowed significant performance improvements of our models, in normal and degraded conditions. By identifying and deactivating the unusable modalities in preprocessing, ModAM makes the robustness issue of CNNs simpler, thus mitigating some undesirable effects of Noise Augmentation. On the other hand, we have observed the limits of this strategy, in particular when faced with unknown noise not learned by ModAM or the CNN. This unknown noise strongly disturbs the CNN, leading to drastic performance drops.

From these observations, many improvements are possible. First, additional work is needed on the generation of noise, which is very limited for our experiments. On one hand, opting for specific realistic noises for each modality would be a viable option, allowing us to get closer to the real driving case. On the other hand, the generation of artificial noises offers a wider diversity of noises, and would consequently lead our multimodal networks to be more robust. However, we should not limit ourselves to certain noise generation methods as proposed with Noise Augmentation. For this, the use of Generative Adversarial Networks, for noise generation as proposed by [2] is an interesting approach, not requiring real data.

However, we have tackled a particular case of failure, where a modality was either correct or unusable because of too much noise. In reality, the probability of obtaining a totally unusable modality is quite rare. With this approach of identifying and deactivating noisy modalities, it remains to define a limit between what we consider a modality as perfect or slightly noisy and a modality too noisy to be ignored. This depends a lot on the type of noise, on the objectives to reach for road scenes understanding, as well as on the modalities concerned by this noise.

Finally, ModAM can be greatly improved. In order to make it able to handle any kind of noise, an increase in the size of the network is to be expected with the use of a more complex classifier than the one used for our experiments (AlexNet). This will lead to a slowing down of the execution speed of the complete network. To remedy this, an integration of the classifier within the CNN is an option to consider. By using the same feature extraction for the detection and for the identification of unusable modalities, it is possible to save computation time, while potentially improving the robustness of the neural networks. However, this implies abandoning any early fusion strategy, or at least guaranteeing a feature extraction of each modality independent of the others, without which this strategy is not viable.

Conclusion and perspectives

In this chapter, we sum up the contributions proposed in this thesis and discuss interesting directions for future work.

Conclusion

In this thesis, we have studied the robustness of perception systems based on multimodal deep neural networks in case of sensor malfunctions. Our research focused on two main axes: the proposal of multimodal neural networks for 2D detection of road traffic actors, and the improvement of their robustness in degraded conditions.

Multimodal neural networks for 2D object detection

In Chapter 2, we proposed several multimodal CNNs with the objective of taking as many modalities as possible as input and extracting them to improve their overall robustness. First, we introduced Multimodal RetinaNet, an architecture that can take as many image-based modalities as desired as input without exploding the size of the network. The scalability of this architecture is possible thanks to the use of a single backbone for all modalities, while opting for a middle fusion of the feature maps generated via stack fusion. However, we could see with our experiments the limits of this strategy, privileging in our case the modality RGB above the others, and thus not improving the general performances compared to a classic RetinaNet.

We then proposed two new multimodal CNN architectures, Stacked and Gated Fusion Double RetinaNet. These two networks use both early and middle fusion of 4 input modalities. Our experiments showed that multimodality slightly improved the results without being significant. With the use of data augmentation, we found that these 2 networks needed much more data than a simple RetinaNet to potentially perform better. Our contributions were then compared with state-of-the-art perception systems on the KITTI 2D Object detection benchmark. Despite results approaching the best perception systems for SFD-Retina and GFD-Retina for car detection, clear differences in performance can be seen between our contributions and the best perception systems.

During this chapter, we also highlighted the vulnerabilities of our networks when a modality was missing in input. We were able to analyze the impact of each input modality on the detection process and to conclude that RGB modality plays a crucial role above the other additional modalities.

Robustness improvement of multimodal neural networks

In Chapter 3, we addressed the problem of robustness of multimodal neural networks in case of missing input modalities. We presented 3 data augmentation methods, namely Random Signal Cut (RSC), Random Channel Cut (RCC) and Random Modality Cut (RMC). These 3 methods have a similar operation, but differ in the entities to deactivate in input. Our experiments have shown clear improvements in the robustness of multimodal neural networks, whatever the degraded conditions. RMC stood out from the other two methods, offering a better overall robustness. However, RCC is a method to be considered thanks to its more diversified data generation, moving away from the possible real failure cases, but allowing to delay the overfitting of our networks. We then analyzed in depth a robust multimodal neural network with complementary metrics in order to identify the consequences of the absence of one or more modalities. We found that the performance losses are mainly due to a reduced confidence of the network on its predictions.

Finally, in our last chapter, we studied a more complex case of robustness, when one or more modalities are strongly noisy. The use of data augmentation via our contribution, Noise Augmentation, showed significant improvements in any degraded conditions, whatever the noise applied. However, we have seen the limitations of using this method alone, namely a decrease in performance under normal conditions, as well as the inability to handle all possible types of noise without severely degrading the overall performance of the neural network. To address both of these issues, combining Noise Augmentation with our preprocessing model, Modality Activator Model (ModAM), shows improvements on almost all degraded conditions, as well as more moderate losses in normal conditions. The strategy of identifying and deactivating noisy modalities thus eliminates the false information they contain, which strongly disrupts our neural networks. However, this combination has a major vulnerability in case of unknown noise, not learned by the network.

Perspectives

We now discuss some perspectives from our contributions that could be addressed in future work.

On Multimodal Convolutional Neural Networks

Better choice of input modalities

Our 3 proposed architectures all have 4 input modalities: RGB, DP, OF and LD. We could notice the predominance of the first of them on the others. We also observed the weak contribution of the modality OF. There are several possible reasons for these differences in role. The first one is the task of our CNNs, the 2D object detection. In this configuration, our 3 additional modalities can be useful but we do not exploit here their full potential, more adapted to other challenges like depth estimation or flow estimation. Then, we represented them in image format, easier to process, but many works have shown the interest of LiDAR point cloud representation for 3D object detection for example. In our opinion, through our experiments, DP, OF and LD modalities were underexploited. Moreover, these modalities need to be extracted each with a neural network,

which increases considerably the computation time. For all these reasons, we think that the biggest margin of progress is in the choice of the input modalities and their format, to be adapted according to the task to be performed.

Integrating modality extraction within object detector architecture

DP, OF, LD are calculated before being used for 2D object detection. This reduces the execution speed of the general pipeline. In addition, if there is noise on the data needed to create these modalities, extraction is compromised. If one of the two stereo cameras is down, the extraction of the modality is useless. For these reasons, we believe that it is preferable to opt for raw data, each depending on only one sensor, with a light preprocessing, allowing to limit the possible robustness flaws. The ideal way to exploit the stereo depth would therefore be to extract the characteristics of the 2 camera images directly in the object detector architecture, and thus avoid a costly reconstruction of the image. This requires a lot of work and a more consequent architecture for 2D object detection, but nevertheless represents a considerable time saving on the general pipeline.

Opting for a specific modalities fusion

Through our experiments with Multimodal RetinaNet, we have seen the difficulty of combining scalability and overall performances. Even if many improvements are possible for this network while keeping the general idea of being able to support a non-fixed number of modalities, it will remain difficult to get close to the state of the art. Therefore, we believe that it is best to opt for a fixed architecture based on the input data. Among the different fusion schemes, middle fusion remains the best option ahead of early and late. On the fusion method, a specific fusion adapted to the input modalities is according to us a track to explore to improve 2D object detector performances. Concerning neural networks robustness, we have observed an improvement of the performances in degraded conditions on several multimodal CNNs using different fusion schemes. Even if further experiments are needed to confirm our results, we believe that our contributions on this topic can improve CNNs robustness regardless of the fusion method used, and that this choice can therefore be made independently of this issue.

On Robustness Improvement of Multimodal CNNs

Generating more realistic and modality-specific noises

The noise generation methods used during our experiments are basic and the resulted modalities are for the most part very far from reality. This does not necessarily mean that they should be ignored, but that the probability of obtaining such noises is very low. One way of improvement is therefore to opt for a more realistic noise generation in order to get closer to real cases. Using real noisy data would be even better, but it is difficult to achieve. Noisy data generation, as used with Noise Augmentation, remains the best choice in our opinion. Moreover, it would also be more relevant to generate specific noises according to the input sensors. Where Gaussian blur is likely on a camera image, this method is less appropriate for LiDAR 3D point clouds.

Using partially noisy modalities

In Chapters 3 and 4, the problems studied assumed that each input modality was either correct or unusable (missing or strongly noisy). An additional step in the continuation of our experiments is to study the case where modalities are partially noisy. This can be done by applying strong noise to only part of the modality. In this case, a noise identification strategy would require an evolution of ModAM, opting for a binary semantic segmentation rather than a classification. The use of less disturbing noises would also be interesting, in order to define a limit between correct noises that can be handled by the object detector, and noises that are too disturbing and need to be preprocessed.

Instilling explicability in noise identification strategy

In section 4.3.4, we found a major vulnerability of ModAM, when a new noise not learned by the neural network is applied. We then found that the ModAM filtering had disabled only the modalities altered by one of the Noise Augmentation methods. While adding this unknown noise to the Noise Augmentation set could solve this problem here, this solution is by no means effective. We think that in the case where we want to use a strategy of identification and deactivation of noisy modalities, it is necessary to understand the precise functioning of the identification process and the reasons of the choices made by ModAM for its deactivation. Thus, modifications could be implemented to lead ModAM classifier to identify the correct modalities, rather than the noises that the modalities contain.

Integrating noise identification into object detector architecture

The previous areas of improvement imply big changes to ModAM, and its size could increase drastically to meet these challenges. In this case, real-time operation would be compromised. To remedy this, an integration of the noise identification strategy in the architecture of the object detector itself is an avenue to explore. Indeed, noise identification requires a feature extraction, also performed by the object detector. This repeated step could thus be performed only once. Noise identification would then benefit from a more advanced and detailed feature extraction without increasing the computing time. The robustness results could then be improved, while reducing the overall computation time of the neural network. However, such a strategy is not compatible with early fusion, as explained in section 4.3. Despite this drawback, we believe that it is preferable to opt for a multimodal CNN using only middle-fusion, and thus integrate noise identification in its architecture through a subnetwork at the backbone exit to deactivate the feature maps from noisy modalities before their fusion.

Conclusion et perspectives

Dans ce chapitre, nous résumons les contributions proposées dans cette thèse et discutons des directions intéressantes pour les travaux futurs.

Conclusion

Dans cette thèse, nous avons étudié la robustesse des systèmes de perception basés sur des réseaux de neurones profonds multimodaux en cas de dysfonctionnement d'un ou plusieurs capteurs. Nos recherches ont porté sur deux axes principaux : la proposition de réseaux de neurones profonds convolutionnels multimodaux pour la détection 2D des usagers de la route, et l'amélioration de leur robustesse en conditions dégradées.

Réseaux de neurones multimodaux pour la détection 2D d'objets

Dans le chapitre 2, nous avons proposé plusieurs CNNs multimodaux dont l'objectif est de prendre en entrée autant de modalités que possible pour ensuite améliorer leur robustesse globale. Tout d'abord, nous avons présenté Multimodal RetinaNet, une architecture pouvant prendre en entrée autant de modalités basées sur l'image que souhaité sans faire exploser la taille du réseau. La modularité de cette architecture est possible grâce à l'utilisation d'un seul backbone pour toutes les modalités, tout en optant pour une fusion intermédiaire des cartes de caractéristiques générées par la *"stack"* fusion. Cependant, nous avons pu constater avec nos expériences les limites de cette stratégie, privilégiant dans notre cas la modalité RGB par rapport aux autres, et n'améliorant donc pas les performances générales par rapport à un RetinaNet classique.

Nous avons ensuite proposé deux nouvelles architectures de réseaux de neurones convolutionnels multimodaux, Stacked et Gated Fusion Double RetinaNet. Ces CNNs utilisent la fusion précoce et intermédiaire de 4 modalités d'entrée. Nos expériences ont montré que la multimodalité améliorait légèrement les résultats sans être significative. Grâce à de l'augmentation de données, nous avons constaté que ces deux réseaux requièrent beaucoup plus de données qu'un simple RetinaNet pour être potentiellement plus performants. Nos contributions ont ensuite été comparées aux systèmes de perception de l'état de l'art sur le benchmark de détection 2D d'objets KITTI. Malgré des résultats proches des meilleurs systèmes de perception pour SFD-Retina et GFD-Retina pour la détection de voitures, de nettes différences de performances peuvent être constatées entre nos contributions et les meilleurs systèmes de perception.

Au cours de ce chapitre, nous avons également mis en évidence les vulnérabilités de nos réseaux lorsqu'une modalité était manquante en entrée. Nous avons pu analyser l'impact de chaque modalité d'entrée sur le processus de détection et conclure que la modalité RGB joue un rôle crucial au-dessus des autres modalités supplémentaires.

Amélioration de la robustesse des réseaux de neurones multimodaux

Dans le chapitre 3, nous avons abordé le problème de la robustesse des réseaux de neurones multimodaux en cas d'absence de modalités d'entrée. Nous avons présenté 3 méthodes d'augmentation des données, à savoir Random Signal Cut (RSC), Random Channel Cut (RCC) et Random Modality Cut (RMC). Ces 3 méthodes ont un fonctionnement similaire, mais diffèrent par les entités à désactiver en entrée. Nos expériences ont montré de nettes améliorations de robustesse des réseaux de neurones multimodaux, quelles que soient les conditions dégradées. La méthode RMC s'est démarquée des deux autres méthodes, offrant une meilleure robustesse globale. Cependant, RCC est à considérer grâce à sa génération de données plus diversifiée, s'éloignant des éventuels cas de défaillance réels, mais permettant de retarder le sur-apprentissage de nos réseaux. Nous avons ensuite analysé en profondeur un réseau de neurones multimodal robuste avec des métriques complémentaires afin d'identifier les conséquences de l'absence d'une ou plusieurs modalités. Nous avons constaté que les pertes de performance sont principalement dues à une moindre confiance du réseau sur ses prédictions.

Enfin, dans notre dernier chapitre, nous avons étudié un cas plus complexe de robustesse, lorsqu'une ou plusieurs modalités sont fortement bruitées et par conséquent inutilisables. L'utilisation de l'augmentation des données via notre contribution, Noise Augmentation, a montré des améliorations significatives dans toutes les conditions dégradées, quel que soit le bruit appliqué. Cependant, nous avons constaté les limites de l'utilisation de cette méthode seule, à savoir une diminution des performances dans des conditions normales, ainsi que l'incapacité à gérer tous les types de bruit possibles sans dégrader fortement les performances globales du réseau de neurones. Pour résoudre ces deux problèmes, la combinaison de Noise Augmentation avec notre modèle de prétraitement, Modality Activator Model (ModAM), montre des améliorations dans presque toutes les conditions dégradées, ainsi que des pertes plus modérées dans des conditions normales. La stratégie d'identification et de désactivation des modalités bruitées élimine ainsi les fausses informations qu'elles contiennent et qui perturbent fortement nos réseaux de neurones. Cependant, cette combinaison présente une vulnérabilité majeure en cas de bruit inconnu, non appris par le réseau.

Perspectives

Nous discutons maintenant de certaines perspectives émanant de nos contributions qui pourraient être abordées dans des travaux futurs.

Sur les réseaux de neurones convolutionnels multimodaux

Un meilleur choix des modalités d'entrée

Les 3 architectures que nous proposons ont toutes 4 modalités d'entrée : RGB, DP, OF et LD. Nous avons pu constater la prédominance de la première d'entre elles sur les autres. Nous avons également observé la faible contribution de la modalité OF. Plusieurs raisons peuvent expliquer ces différences de rôle. La première est la tâche de nos CNNs, la détection 2D d'objets. Dans cette configuration, nos 3 modalités supplémentaires peuvent être utiles mais nous n'exploitons pas ici tout leur potentiel, plus adapté à d'autres tâches comme l'estimation de la profondeur ou du mouvement. Ensuite, nous les avons représentées au format image, plus facile à traiter, mais de nombreux travaux ont montré l'intérêt de la représentation du nuage de points LiDAR pour la détection d'objets 3D par exemple. A notre avis, au travers de nos expériences, les modalités DP, OF et LD ont été sous-exploitées. De plus, ces modalités doivent être extraites chacune avec un réseau de neurones, ce qui augmente considérablement le temps de calcul. Pour toutes ces raisons, nous pensons que la plus grande marge de progression réside dans le choix des modalités d'entrée et de leur format, à adapter en fonction de la tâche à réaliser.

Intégrer l'extraction de modalité dans l'architecture du détecteur d'objet

DP, OF, LD sont calculés avant d'être utilisés pour la détection 2D d'objets. Cela réduit la vitesse d'exécution de la pipeline général. De plus, s'il y a du bruit sur les données nécessaires à la création de ces modalités, l'extraction est compromise. Si l'une des deux caméras stéréo est en panne, l'extraction de la modalité est inutile. Pour ces raisons, nous pensons qu'il est préférable d'opter pour des données brutes, dépendant chacune d'un seul capteur, avec un prétraitement léger, permettant de limiter les éventuels défauts de robustesse. L'idéal pour exploiter la profondeur stéréo serait donc d'extraire les caractéristiques des images des 2 caméras directement dans l'architecture du détecteur d'objets, et ainsi éviter une reconstruction coûteuse de l'image. Cela demande beaucoup de travail et une architecture plus conséquente pour la détection 2D d'objets, mais représente néanmoins un gain de temps considérable sur la pipeline générale.

Opter pour une fusion de modalités spécifiques

Au travers de nos expériences avec Multimodal RetinaNet, nous avons pu constater la difficulté de combiner modularité et performances globales. Même si de nombreuses améliorations sont possibles pour ce réseau tout en gardant l'idée générale de pouvoir supporter un nombre non fixe de modalités, il restera difficile de se rapprocher de l'état de l'art. Par conséquent, nous pensons qu'il est préférable d'opter pour une architecture fixe en fonction des données d'entrée. Parmi les différents schémas de fusion, la fusion intermédiaire reste la meilleure option devant la fusion précoce et la fusion tardive. Concernant la méthode de fusion, une fusion spécifique adaptée aux modalités d'entrée pourrait être une piste à explorer pour améliorer les performances des détecteurs 2D d'objets. Concernant la robustesse des réseaux de neurones, nous avons observé une amélioration des performances en conditions dégradées sur plusieurs CNNs multimodaux utilisant différents schémas de fusion. Même si d'autres expériences sont nécessaires pour confirmer

nos résultats, nous pensons que nos contributions sur ce sujet peuvent améliorer la robustesse des CNNs indépendamment de la méthode de fusion utilisée, et que ce choix peut donc être fait indépendamment de cette problématique.

Sur l'amélioration de la robustesse des CNNs multimodaux

Générer des bruits plus réalistes et spécifiques à la modalité

Les méthodes de bruit utilisées lors de nos expériences sont basiques et les modalités générées sont pour la plupart très éloignées de la réalité. Cela ne signifie pas nécessairement qu'il faille les ignorer, mais que la probabilité d'obtenir de tels bruits est très faible. Une piste d'amélioration consiste donc à opter pour une génération de bruit plus réaliste afin de se rapprocher des cas réels. Utiliser des données bruitées réelles serait encore mieux, mais cela reste difficile à réaliser. La génération de données bruitées, telle qu'elle est utilisée avec Noise Augmentation, reste le meilleur choix à notre avis. Il serait d'ailleurs plus pertinent de générer des bruits spécifiques en fonction des capteurs d'entrée. Alors que le flou gaussien est probable sur une image de caméra, cette méthode est moins appropriée pour les nuages de points 3D LiDAR.

Utiliser des modalités partiellement bruitées

Dans les chapitres 3 et 4, les problèmes étudiés supposaient que chaque modalité d'entrée était soit correcte, soit inutilisable (manquante ou très bruitée). Une étape supplémentaire dans la suite de nos expériences est d'étudier le cas où les modalités sont partiellement bruitées. Cela peut être fait en appliquant un bruit fort à une partie seulement de la modalité. Dans ce cas, une stratégie d'identification du bruit nécessiterait une évolution de ModAM, optant pour une segmentation sémantique binaire plutôt qu'une classification. L'utilisation de bruits moins perturbants serait également intéressante, afin de définir une limite entre les bruits corrects pouvant être traités par le détecteur d'objets, et les bruits trop perturbants devant être prétraités.

Instiller l'explicabilité dans la stratégie d'identification des bruits

Dans la section 4.3.4, nous avons découvert une vulnérabilité majeure de ModAM, quand un nouveau bruit non appris par le réseau de neurones est appliqué. Nous avons ensuite constaté que le filtrage de ModAM n'avait désactivé que les modalités modifiées par l'une des méthodes de Noise Augmentation. Bien que l'ajout de ce bruit inconnu à l'ensemble de méthodes de Noise Augmentation puisse résoudre ce problème ici, cette solution n'est en aucun cas efficace. Nous pensons que dans le cas où l'on souhaite utiliser une stratégie d'identification et de désactivation des modalités bruitées, il est nécessaire de comprendre le fonctionnement précis du processus d'identification et les raisons des choix effectués par ModAM pour ses désactivations. Ainsi, des modifications pourraient être mises en œuvre pour amener le classificateur de ModAM à identifier les modalités correctes, plutôt que les bruits que ces modalités contiennent.

Intégrer l'identification du bruit dans l'architecture du détecteur d'objets

Les domaines d'amélioration précédents impliquent des changements conséquents de ModAM, et sa taille pourrait augmenter considérablement pour relever ces défis. Dans ce cas, le fonctionnement en temps réel serait compromis. Pour remédier à cela, une intégration de la stratégie d'identification du bruit dans l'architecture même du détecteur d'objets est une piste à explorer. En effet, l'identification du bruit nécessite une extraction de caractéristiques, également réalisée par le détecteur d'objets. Cette étape répétée pourrait donc être réalisée une seule fois. L'identification du bruit bénéficierait alors d'une extraction de caractéristiques plus poussée et plus détaillée sans augmenter le temps de calcul. Les résultats de robustesse pourraient alors être améliorés, tout en réduisant le temps de calcul global du réseau de neurones. Cependant, une telle stratégie n'est pas compatible avec la fusion précoce, comme expliqué dans la section 4.3. Malgré cet inconvénient, nous pensons qu'il est préférable d'opter pour un CNN multimodal n'utilisant que la fusion intermédiaire, et donc d'intégrer l'identification du bruit dans son architecture par le biais d'un sous-réseau à la sortie du backbone pour désactiver les cartes de caractéristiques de modalités bruitées avant leur fusion.

Bibliography

[1] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3782–3795, 2019.

[2] M. Baradad Jurjo, J. Wulff, T. Wang, P. Isola, and A. Torralba. Learning to see by looking at noise. *Advances in Neural Information Processing Systems*, 34:2556–2569, 2021.

[3] M. Bednarek, P. Kicki, and K. Walas. On robustness of multi-modal fusion—robotics perspective. *Electronics*, 9(7):1152, 2020.

[4] M. Bijelic, C. Muench, W. Ritter, Y. Kalnishkan, and K. Dietmayer. Robustness against unknown noise for raw data fusing neural networks. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2177–2184. IEEE, 2018.

[5] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, and F. Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11682–11692, 2020.

[6] R. Blin, S. Ainouz, S. Canu, and F. Meriaudeau. A new multimodal rgb and polarimetric image dataset for road scenes analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 216–217, 2020.

[7] R. Blin, S. Ainouz, S. Canu, and F. Meriaudeau. Multimodal polarimetric and color fusion for road scene analysis in adverse weather conditions. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3338–3342. IEEE, 2021.

[8] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[9] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[10] Z. Cai and N. Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1483–1498, 2019.

[11] L. Caltagirone, M. Bellone, L. Svensson, and M. Wahde. Lidar–camera fusion for road detection using fully convolutional neural networks. *Robotics and Autonomous Systems*, 111:125–131, 2019.

[12] C. Caraffi, T. Vojíř, J. Trefnỳ, J. Šochman, and J. Matas. A system for real-time detection and tracking of vehicles from a single car-mounted camera. In *2012 15th international IEEE conference on intelligent transportation systems*, pages 975–982. IEEE, 2012.

[13] S. Chadwick, W. Maddern, and P. Newman. Distant vehicle detection using radar and vision. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8311–8317. IEEE, 2019.

[14] S. Chadwick, W. Maddern, and P. Newman. Distant vehicle detection using radar and vision. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8311–8317. IEEE, 2019.

[15] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[16] R. O. Chavez-Garcia and O. Aycard. Multiple sensor fusion and classification for moving object detection and tracking. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):525–534, 2015.

[17] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1259–1272, 2017.

[18] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017.

[19] X. Chen, K.-Y. Lin, J. Wang, W. Wu, C. Qian, H. Li, and G. Zeng. Bi-directional cross-modality feature propagation with separation-and-aggregation gate for rgb-d semantic segmentation. In *European Conference on Computer Vision*, pages 561–577. Springer, 2020.

[20] Z. Chen and X. Huang. Pedestrian detection for autonomous vehicle using multi-spectral cameras. *IEEE Transactions on Intelligent Vehicles*, 4(2):211–219, 2019.

[21] R. Condat, A. Rogozan, and A. Bensrhair. Gfd-retina: Gated fusion double retinanet for multimodal 2d road object detection. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.

[22] R. Condat, A. Rogozan, and A. Bensrhair. Random signal cut for improving multimodal cnn robustness of 2d road object detection. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2020.

[23] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[24] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005.

[25] A. Daniel Costea, R. Varga, and S. Nedevschi. Fast boosting based detection using scale invariant multimodal multiresolution filtered features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6674–6683, 2017.

[26] S. de Blois, M. Garon, C. Gagné, and J.-F. Lalonde. Input dropout for spatially aligned modalities. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 733–737. IEEE, 2020.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[28] Z. Deng and L. Jan Latecki. Amodal detection of 3d objects: Inferring 3d bounding boxes from 2d ones in rgb-depth images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5762–5770, 2017.

[29] J. Dou, J. Xue, and J. Fang. Seg-voxelnet for 3d vehicle detection from rgb and lidar data. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 4362–4368. IEEE, 2019.

[30] X. Du, M. H. Ang, S. Karaman, and D. Rus. A general pipeline for 3d detection of vehicles. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3194–3200. IEEE, 2018.

[31] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard. Multimodal deep learning for robust rgb-d object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 681–687. IEEE, 2015.

[32] M. Enzweiler and D. M. Gavrila. A multilevel mixture-of-experts framework for pedestrian classification. *IEEE Transactions on Image Processing*, 20(10):2967–2979, 2011.

[33] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2154, 2014.

[34] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88 (2):303–338, 2010.

[35] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pages 363–370. Springer, 2003.

[36] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020.

[37] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[38] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, et al. A2d2: Audi autonomous driving dataset. *arXiv preprint arXiv:2004.06320*, 2020.

[39] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[40] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[41] A. González, Z. Fang, Y. Socarras, J. Serrat, D. Vázquez, J. Xu, and A. M. López. Pedestrian detection at day/night time with visible and fir cameras: A comparison. *Sensors*, 16(6):820, 2016.

[42] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[43] Z. Guo, W. Liao, Y. Xiao, P. Veelaert, and W. Philips. Weak segmentation supervised deep neural networks for pedestrian detection. *Pattern Recognition*, 119:108063, 2021.

[44] Q. Ha, K. Watanabe, T. Karasawa, Y. Ushiku, and T. Harada. Mfnet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5108–5115. IEEE, 2017.

[45] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian conference on computer vision*, pages 213–228. Springer, 2016.

[46] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[47] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on pattern analysis and machine intelligence*, 30(2):328–341, 2007.

[48] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[49] 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*, pages 4700–4708, 2017.

[50] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang. The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 954–960, 2018.

[51] 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*, pages 1037–1045, 2015.

[52] J. Kim, J. Choi, Y. Kim, J. Koh, C. C. Chung, and J. W. Choi. Robust camera lidar sensor fusion via deep gated information fusion network. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1620–1625. IEEE, 2018.

[53] J. Kim, J. Koh, Y. Kim, J. Choi, Y. Hwang, and J. W. Choi. Robust deep multi-modal learning based on gated information fusion network. In *Asian Conference on Computer Vision*, pages 90–106. Springer, 2018.

[54] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[55] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[56] S. J. Krotosky and M. M. Trivedi. On color-, infrared-, and multimodal-stereo approaches to pedestrian detection. *IEEE Transactions on Intelligent Transportation Systems*, 8(4):619–629, 2007.

[57] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[58] M. Liang, B. Yang, S. Wang, and R. Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 641–656, 2018.

[59] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7345–7353, 2019.

[60] Z. Liang, Z. Zhang, M. Zhang, X. Zhao, and S. Pu. Rangeioudet: Range image based real-time 3d object detector optimized by intersection over union. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7140–7149, 2021.

[61] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[62] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[63] G.-H. Liu, A. Siravuru, S. Prabhakar, M. Veloso, and G. Kantor. Learning end-to-end multimodal sensor policies for autonomous navigation. In *Conference on Robot Learning*, pages 249–261. PMLR, 2017.

[64] J. Liu, S. Zhang, S. Wang, and D. N. Metaxas. Multispectral deep neural networks for pedestrian detection. In *27th British Machine Vision Conference*, 2016.

[65] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen. Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2):261–318, 2020.

[66] 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*, pages 21–37. Springer, 2016.

[67] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. In *Proceedings of DARPA Image Understanding Workshop*, pages 121 – 130, April 1981.

[68] O. Mees, A. Eitel, and W. Burgard. Choosing smartly: Adaptive multimodal fusion for object detection in changing environments. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 151–156. IEEE, 2016.

[69] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and W. Wu. Mixed precision training. In *International Conference on Learning Representations*, 2018.

[70] N. Neverova, C. Wolf, G. Taylor, and F. Nebout. Moddrop: adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1692–1706, 2015.

[71] F. Nobis, M. Geisslinger, M. Weber, J. Betz, and M. Lienkamp. A deep learning-based radar and camera sensor fusion architecture for object detection. In *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–7. IEEE, 2019.

[72] S.-I. Oh and H.-B. Kang. Object detection and classification by decision-level fusion for intelligent vehicle systems. *Sensors*, 17(1):207, 2017.

[73] S. Pang, D. Morris, and H. Radha. Clocs: Camera-lidar object candidates fusion for 3d object detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10386–10393. IEEE, 2020.

[74] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. So Kweon. Non-local spatial propagation network for depth completion. In *European Conference on Computer Vision*, pages 120–136. Springer, 2020.

[75] A. Pfeuffer and K. Dietmayer. Optimal sensor data fusion architecture for object detection in adverse weather conditions. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 1–8. IEEE, 2018.

[76] A. Pfeuffer, M. Schön, C. Ditzel, and K. Dietmayer. The aduulm-dataset-a semantic segmentation dataset for sensor fusion. In *31th British Machine Vision Conference*, 2020.

[77] 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*, 40(4-5):681–690, 2021.

[78] D. O. Pop, A. Rogozan, F. Nashashibi, and A. Bensrhair. Improving pedestrian recognition using incremental cross modality deep learning. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2019.

[79] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[80] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018.

[81] R. Qian, X. Lai, and X. Li. Boundary-aware 3d object detection from point clouds. *arXiv preprint arXiv:2104.10330*, 2021.

[82] H. Rashed, A. El Sallab, S. Yogamani, and M. ElHelw. Motion and depth augmented semantic segmentation for autonomous navigation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 364–370. IEEE Computer Society, 2019.

[83] H. Rashed, M. Ramzy, V. Vaquero, A. El Sallab, G. Sistu, and S. Yogamani. Fusemodnet: Real-time camera and lidar based moving object detection for robust low-light autonomous driving. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2393–2402. IEEE Computer Society, 2019.

[84] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

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

[86] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[87] J. Ren, X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y.-W. Tai, and L. Xu. Accurate single stage detector using recurrent rolling convolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5420–5428, 2017.

[88] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[89] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[90] C. Sakaridis, D. Dai, and L. Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, 2018.

[91] J. Schlosser, C. K. Chow, and Z. Kira. Fusing lidar and images for pedestrian detection using convolutional neural networks. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2198–2205. IEEE, 2016.

[92] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[93] J. Shi and C. Tomasi. Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, pages 593–600. IEEE, 1994.

[94] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

[95] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[96] V. A. Sindagi, Y. Zhou, and O. Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7276–7282. IEEE, 2019.

[97] R. Spangenberg, T. Langner, and R. Rojas. Weighted semi-global matching and center-symmetric census transform for robust driver assistance. In *International Conference on Computer Analysis of Images and Patterns*, pages 34–41. Springer, 2013.

[98] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[99] D. Sun, X. Huang, and K. Yang. A multimodal vision sensor for autonomous driving. In *Counterterrorism, Crime Fighting, Forensics, and Surveillance Technologies III*, volume 11166, page 111660L. International Society for Optics and Photonics, 2019.

[100] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[101] Y. Sun, W. Zuo, and M. Liu. Rtfnet: Rgb-thermal fusion network for semantic segmentation of urban scenes. *IEEE Robotics and Automation Letters*, 4(3):2576–2583, 2019.

[102] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. *Advances in neural information processing systems*, 26, 2013.

[103] K. Takumi, K. Watanabe, Q. Ha, A. Tejero-De-Pablos, Y. Ushiku, and T. Harada. Multispectral object detection for autonomous vehicles. In *Proceedings of the on Thematic Workshops of ACM Multimedia 2017*, pages 35–43, 2017.

[104] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[105] A. Valada, G. L. Oliveira, T. Brox, and W. Burgard. Deep multispectral semantic scene understanding of forested environments using multimodal fusion. In *International symposium on experimental robotics*, pages 465–477. Springer, 2016.

[106] A. Valada, J. Vertens, A. Dhall, and W. Burgard. Adapnet: Adaptive semantic segmentation in adverse environmental conditions. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4644–4651. IEEE, 2017.

[107] A. Valada, R. Mohan, and W. Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *International Journal of Computer Vision*, 128(5): 1239–1285, 2020.

[108] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee, 2001.

[109] J. Wagner, V. Fischer, M. Herman, and S. Behnke. Multispectral pedestrian detection using deep fusion convolutional neural networks. In *ESANN*, volume 587, pages 509–514, 2016.

[110] Z. Wang and K. Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1742–1749. IEEE, 2019.

[111] C. Wojek, S. Walk, and B. Schiele. Multi-cue onboard pedestrian detection. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 794–801. IEEE, 2009.

[112] F. Wulff, B. Schäufele, O. Sawade, D. Becker, B. Henke, and I. Radusch. Early fusion of camera and lidar for robust road detection based on u-net fcn. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1426–1431. IEEE, 2018.

[113] P. Xiao, Z. Shao, S. Hao, Z. Zhang, X. Chai, J. Jiao, Z. Li, J. Wu, K. Sun, K. Jiang, et al. Pandaset: Advanced sensor suite dataset for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3095–3101. IEEE, 2021.

[114] D. Xu, D. Anguelov, and A. Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 244–253, 2018.

[115] B. Yang, M. Liang, and R. Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *Conference on Robot Learning*, pages 146–155. PMLR, 2018.

[116] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2129–2137, 2016.

[117] G. Yang and D. Ramanan. Volumetric correspondence networks for optical flow. *Advances in neural information processing systems*, 32, 2019.

[118] J. J. Yebes, L. M. Bergasa, and M. García-Garrido. Visual object recognition with 3d-aware features in kitti urban scenes. *Sensors*, 15(4):9228–9250, 2015.

[119] J. H. Yoo, Y. Kim, J. Kim, and J. W. Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In *European Conference on Computer Vision*, pages 720–736. Springer, 2020.

[120] Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. In *International Conference on Learning Representations (ICLR)*, 2020.

[121] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr. Ga-net: Guided aggregation net for end-to-end stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 185–194, 2019.

[122] S. Zhang, R. Benenson, and B. Schiele. Citypersons: A diverse dataset for pedestrian detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3221, 2017.

[123] Y. Zheng, I. H. Izzat, and S. Ziaee. Gfd-ssd: gated fusion double ssd for multispectral pedestrian detection. *arXiv preprint arXiv:1903.06999*, 2019.

[124] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.