Contributions à la reconnaissance de véhicules dans des grilles d'occupation pour un système de perception embarqué

Nils Defauw

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : EEATS - Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)
Spécialité : Automatique - Productique
Unité de recherche : CEA Grenoble (hors LETI et LITEN)

Contributions à la reconnaissance de véhicules dans des grilles d'occupation pour un système de perception embarqué

Contributions to vehicle recognition in occupancy grids for an embedded perception system

Présentée par :

Nils DEFAUW

Direction de thèse :

Suzanne LESECQ DIRECTRICE DE RECHERCHE, CEA Grenoble / CEA-Leti	Directrice de thèse
Tiana RAKOTOVAO DOCTEUR EN SCIENCES, CEA Grenoble / CEA-List	Co-encadrant de thèse
Olivier ANTONI INGENIEUR DE RECHERCHE, CEA Grenoble / CEA-List	Co-encadrant de thèse
Marielle MALFANTE DOCTEURE EN SCIENCES, CEA Grenoble / CEA-List	Co-encadrante de thèse

Rapporteurs :

Olivier BERDER
PROFESSEUR DES UNIVERSITES, Université de Rennes
Luc JAULIN
PROFESSEUR DES UNIVERSITES, Université de Bretagne Occidentale

Thèse soutenue publiquement le **8 mars 2024**, devant le jury composé de :

Didier GEORGES, PROFESSEUR DES UNIVERSITES, Grenoble INP - UGA	Président
Suzanne LESECQ, DIRECTRICE DE RECHERCHE, CEA Grenoble / CEA-Leti	Directrice de thèse
Olivier BERDER, PROFESSEUR DES UNIVERSITES, Université de Rennes	Rapporteur
Luc JAULIN, PROFESSEUR DES UNIVERSITES, Université de Bretagne Occidentale	Rapporteur
Éric NASSOR, DOCTEUR EN SCIENCES, Canon Research Centre France	Examinateur

École Doctorale Électronique, Électrotechnique, Automatique, Traitement du Signal
Laboratoire CEA-List, CEA Grenoble

Thèse de Doctorat de l'Université Grenoble Alpes

Contributions à la reconnaissance de véhicules dans des grilles d'occupation pour un système de perception embarqué

Nils Defauw

Rapporteur	Olivier Berder IRISA - ENSSAT, Université de Rennes
Rapporteur	Luc Jaulin Lab-STICC, ENSTA Bretagne
Examinateur (Président)	Didier Georges GIPSA-lab, Grenoble INP - UGA
Examinateur	Eric Nassor Canon Research Centre France
Directrice de thèse	Suzanne Lesecq
Encadrants	Tiana Rakotovao, Olivier Antoni et Marielle Malfante

8 mars 2024

Nils Defauw

Contributions à la reconnaissance de véhicules dans des grilles d'occupation pour un système de perception embarqué

Thèse de Doctorat de l'Université Grenoble Alpes, 8 mars 2024
Rapporteurs : Olivier Berder et Luc Jaulin
Examinateurs : Didier Georges (Président) et Eric Nassor
Directrice de thèse : Suzanne Lesecq
Encadrants : Tiana Rakotovao, Olivier Antoni et Marielle Malfante

Université Grenoble Alpes
Laboratoire CEA-List, CEA Grenoble
École Doctorale Électronique, Électrotechnique, Automatique, Traitement du Signal
110 Rue de la Chimie
38400 Saint-Martin-D'Hères

Résumé

Ce manuscrit de thèse de doctorat présente les travaux réalisés durant trois ans sur le sujet de la reconnaissance des véhicules présents dans la scène à partir du modèle d'environnement de la grille d'occupation bidimensionnelle. Ce sujet s'inscrit dans le contexte du véhicule autonome et constitue une étape importante du système de perception permettant à un véhicule de créer une représentation interne de son environnement qui sera ensuite utilisée afin de prendre des décisions de navigation adaptées. La contrainte inhérente aux véhicules autonomes d'exécution en temps réel des algorithmes développés est spécialement adressée par la recherche des méthodes les plus économiques en termes de coûts calculatoire et mémoire.

Trois contributions principales sont présentées dans ce manuscrit. La première consiste en la création d'un jeu de données de grilles d'occupation annotées avec les positions et dimensions des véhicules présents. Cette première contribution est un prérequis aux méthodes d'apprentissage profond de reconnaissance des véhicules proposées en deuxième et troisième contributions. La deuxième contribution est l'étude de différents modèles de réseaux de neurones convolutifs inspirés de l'état de l'art en détection d'objets sur images afin de détecter les véhicules présents dans des grilles d'occupation via la prédiction de boîtes englobantes orientées. La dernière contribution est la proposition d'une nouvelle approche de réduction de la dimensionnalité des grilles d'occupation via leur transformation vers le domaine fréquentiel ainsi que la proposition d'une architecture neuronale de segmentation des véhicules présents dans les grilles d'occupation à partir de cette représentation de faible dimension.

Les différentes méthodes de reconnaissance de véhicules proposées sont évaluées selon la qualité des résultats prédits et les coûts calculatoires de prédiction qui sont matérialisés par le temps de prédiction par grille d'occupation ainsi que le coût en mémoire de chaque prédiction. La première approche inspirée de réseaux de neurones convolutifs utilisés pour la détection d'objets sur images permet de valider l'utilisation de grilles d'occupation comme modèle d'environnement suffisamment riche pour permettre la reconnaissance des véhicules présents dans l'environnement. Cette première approche permet en effet la détection de la majorité des véhicules tout en maintenant un temps de prédiction faible compris entre 10ms et 100ms par grille d'occupation sur le GPU mobile utilisé durant ces expériences. La deuxième approche

de reconnaissance de véhicules à l'aide d'une segmentation de grille d'occupation opérée dans le domaine fréquentiel offre quant à elle de nouvelles perspectives de reconnaissance d'objets. Cette dernière offre en effet des segmentations de qualité pour des coûts calculatoires entièrement paramétrables, mais généralement réduits d'un facteur dix par rapport aux méthodes convolutives de la première approche. Par ailleurs, cette nouvelle approche pourrait être utilisée plus globalement pour la segmentation d'images et ouvre de nouvelles perspectives de recherche par sa proposition de réseaux de neurones opérant dans le domaine fréquentiel.

Abstract

This PhD thesis manuscript presents the work carried out over a three-year period on the subject of recognizing vehicles present in the scene from the two-dimensional occupancy grid environment model. This subject takes place in the autonomous vehicle context, and represents an important step in the perception system enabling a vehicle to create an internal representation of its environment, which is then used to make appropriate navigation decisions. The constraint inherent to autonomous vehicles of real-time execution of the algorithms developed is specifically addressed by the search for the most economical methods in terms of computational and memory costs.

Three main contributions are presented in this manuscript. The first is the creation of a dataset of occupancy grids annotated with the positions and dimensions of the vehicles present in the scene. This first contribution is a prerequisite for the deep learning methods for vehicle recognition proposed in the second and third contributions. The second contribution is a study of different convolutional neural network models inspired by state of the art models for image-based object detection to detect vehicles present in occupancy grids via oriented bounding boxes prediction. The final contribution is the proposal of a new approach for reducing the dimensionality of occupancy grids via their transformation to the frequency domain, and the proposal of a neural architecture for segmenting vehicles present in occupancy grids from this low-dimensional representation.

The different vehicle recognition methods proposed are evaluated according to the quality of the predicted results and the computational costs of prediction, which are materialized by the prediction time per occupancy grid and the memory cost of each prediction. The first approach, inspired by convolutional neural networks

used for object detection on images, validates the use of occupancy grids as a sufficiently rich environment model to enable recognition of vehicles present in the environment. This first approach enables the detection of the majority of vehicles while maintaining a low prediction time of 10ms to 100ms per occupancy grid on the mobile GPU used during these experiments. The second vehicle recognition approach, based on frequency-domain segmentation of occupancy grids, offers new perspectives on object recognition. This approach offers high-quality segmentations at fully parameterizable computational costs, which are generally reduced by a factor of ten compared with the convolutional methods of the first approach. In addition, this new approach could be used more generally for image segmentation, and opens up new research prospects by proposing neural networks operating in the frequency domain.

Remerciements

Je voudrai tout d'abord remercier les membres du jury Pr Olivier Berder, Pr Luc Jaulin, Pr Didier Georges et Dr Éric Nassor d'avoir accepté d'évaluer ce travail de recherche et de s'être déplacés jusqu'à Grenoble pour assister à la soutenance. Je remercie aussi Pr Didier Georges et Dr Erwan Piriou pour les passionnants échanges que nous avons eus au cours des trois réunions du comité de suivi individuel qui se sont tenues annuellement durant cette thèse.

À mes encadrants Olivier, Tiana, Marielle et Suzanne, merci pour tout le travail accompli et pour les passionnantes discussions scientifiques que nous avons eues au cours de nos réunions régulières. Ce fut un long chemin, parfois semé d'embûches, mais je suis fier du résultat que nous avons obtenu et qui se trouve désormais entre vos mains.

À mon co-thésard/co-bureau Romain, merci pour toutes ces conversations quotidiennes sur la difficile vie de thésard sous Linux au sein d'une structure conçue autour de Windows, pour ces échanges enflammés sur la supposée surcharge pondérale de mes slides et finalement pour toutes ces discussions scientifiques ou non que nous avons eues durant ces trois années.

Plus largement à tous mes collègues que j'ai eu la chance de pouvoir côtoyer durant ces trois années, merci pour tous les bons souvenirs que je ne suis pas prêt d'oublier. Je pense en particulier aux anciens thésards désabusés avec qui nous allions boire des coups à la Passoire, à l'équipe cycliste Triban-Canyon-Ekoi qui m'a emmené faire mes premières sorties en montagne et m'a donné le goût du vélo, à toutes les sorties à Michel Musique, aux soirées jeux de société qui finissaient inlassablement à des horaires improbables et à cette fameuse journée labo où nous sommes montés en vélo/tandem au lac de Paladru pour y faire du paddle.

À la Kak'tous Corporation, merci pour cette semaine incroyable à Autrans qui a fait de nous d'authentiques start-uppers prêts à changer le monde grâce au pouvoir du Tequilo. Et merci pour toutes ces sorties au Dr D., pour ces pétanques au Parc Paul-Mistral et pour tous ces moments passés ensemble par la suite.

À Éléonore, merci pour ton précieux soutien durant ces trois années qui n'ont pas toujours été faciles, je n'y serai pas arrivé sans ton aide. Mention spéciale à ma

soutenance pour laquelle tu t'es chargée de l'organisation du pot alors que je ne savais plus où donner de la tête.

À Yara, Théo et Hugo, merci d'être là depuis plus de dix ans, toujours motivés pour une soirée billard ou bière-pong.

À mes parents et à tous mes proches venus assister à ma soutenance, merci d'avoir été là malgré la distance et l'horaire plutôt matinal. J'espère que vous aurez compris quelque chose malgré le vocabulaire assez hermétique inhérent aux travaux de recherche.

Finalement, merci à toi, lecteur, de l'intérêt porté à ces travaux. J'espère que tu trouveras dans ce manuscrit un éclairage sur ce sujet de recherche et des résultats d'intérêt.

Table des matières

1 Introduction

L'automatisation des véhicules routiers est un défi scientifique et technologique majeur dans le cadre de la transformation des moyens de mobilité à l'ère du numérique. Les enjeux de cette innovation en termes de santé publique, d'accessibilité aux transports et d'écologie sont nombreux. En particulier, la mise au point d'un tel système peut avoir un effet important sur la mortalité routière, la congestion des axes de circulation, la pollution générée par le trafic routier et l'accessibilité de la route aux personnes souffrant de handicap ou n'ayant pas les moyens financiers de posséder un véhicule personnel [FK15].

Un véhicule autonome requiert l'utilisation de capteurs lui permettant de percevoir son environnement pour ensuite prendre des décisions de conduite adaptées. Généralement, de nombreux capteurs différents sont montés sur un véhicule autonome pour d'une part maximiser l'information disponible grâce à leur multiplicité et pour d'autre part mitiger les faiblesses spécifiques à chaque capteur grâce à leur hétérogénéité [Van+18].

Les différentes informations ainsi recueillies doivent par la suite être combinées en une unique représentation de l'environnement suffisamment riche pour permettre la navigation autonome au sein de celui-ci. Cette tâche de transformation, de fusion et d'interprétation des données issues des différents capteurs est appelée perception. La représentation résultant de ce processus de perception est appelée modèle d'environnement [Bad+21].

La tâche de perception est réalisée en temps réel par un ordinateur embarqué à bord du véhicule autonome. Les contraintes en termes de coût, de consommation électrique et de volume occupé par l'ordinateur de bord limitent fortement les ressources de calcul disponibles [Shi+17]. Ainsi, l'ensemble des calculs réalisés lors de la tâche de perception doivent être suffisamment légers pour permettre leur exécution en temps réel sur ce matériel fortement contraint.

La navigation autonome nécessite l'usage d'un modèle d'environnement contenant des informations complexes telles que la nature, la position ou la vitesse des différents obstacles présents dans la scène. L'obtention de ces informations issues d'une interprétation des mesures de capteurs est généralement réalisée à l'aide de

méthodes d'apprentissage profond entraînées directement sur les mesures brutes produites par les différents capteurs embarqués [Che+17]. Ces méthodes, bien qu'efficaces pour l'interprétation des mesures de capteurs, ont pour principal désavantage leur spécialisation à un ensemble de capteurs fixé.

À la différence des méthodes précitées, il existe des modèles d'environnement génériques permettant la fusion de plusieurs capteurs hétérogènes en une unique représentation de l'environnement. La grille d'occupation bidimensionnelle est l'un de ces modèles d'environnement d'intérêt couramment utilisé dans le contexte du véhicule autonome et plus généralement de la robotique mobile [ME85]. Il s'agit d'une représentation matricielle de l'environnement vu du dessus représentant de manière probabiliste l'occupation de l'espace par des obstacles. La grille d'occupation bidimensionnelle offre comme avantages la possibilité de fusionner les mesures issues de différents capteurs en une unique grille, la possibilité de représenter l'incertitude d'occupation issue de mesures bruitées ou de la fusion de données contradictoires et finalement un coût de calcul faible compatible avec les contraintes matérielles importantes du véhicule autonome [Rak17]. Dans la suite de ce manuscrit, l'appellation *grille d'occupation* sans plus de précision désigne une grille d'occupation bidimensionnelle.

Par construction, une grille d'occupation est une matrice dont chaque cellule correspondant à une zone de l'espace contient la probabilité estimée de présence d'un obstacle dans cette zone. Elle ne fournit en revanche pas les informations essentielles à la navigation autonome telles que la nature et la géométrie des différents obstacles présents dans la scène. Afin de permettre l'usage de grilles d'occupation dans un contexte de navigation autonome, il est donc nécessaire de développer des algorithmes permettant d'interpréter ces grilles d'occupation en temps réel avec un ordinateur de bord soumis aux contraintes inhérentes au véhicule autonome. Cette approche proposée permet de conjuguer la généricité des grilles d'occupation et le besoin d'informations complexes nécessaires à toute tâche de navigation autonome.

Le présent travail de recherche se concentre en particulier sur une tâche d'interprétation d'importance : la reconnaissance des véhicules présents dans la scène. Afin de mener à bien ce travail, une première contribution consiste en la création d'un jeu de données de grilles d'occupation annotées avec les positions et les dimensions des véhicules. La constitution d'un tel jeu de données annoté est un prérequis à l'entraînement supervisé des modèles d'apprentissage profond au cœur des contributions suivantes. Ensuite sont présentées les deux approches de reconnaissance de véhicules élaborées durant ce travail de recherche. La première de ces contributions

présente une méthode de détection de véhicules sur grilles d'occupation par la production de boîtes englobantes orientées à l'aide de réseaux de neurones convolutifs. La deuxième de ces contributions présente une nouvelle approche de réduction de la dimensionnalité des grilles d'occupation via un passage dans le domaine fréquentiel. En outre, l'utilité de cette transformation est démontrée par la mise au point d'un réseau de neurones utilisant cette représentation de faible dimension pour segmenter les véhicules présents dans les grilles d'occupation. Cette dernière approche permet d'obtenir des gains importants en termes de coût calculatoire, ce qui permet d'envisager son implémentation dans un véhicule autonome.

Ce chapitre présente le contexte général du véhicule autonome et le modèle d'environnement de la grille d'occupation utilisé comme point de départ de ce travail de recherche. S'ensuit une présentation du sujet d'interprétation de grilles d'occupation avec d'importantes contraintes calculatoires. Suivent ensuite une description des contributions apportées ainsi qu'un rappel de l'organisation de ce manuscrit.

1.1 Contexte : la grille d'occupation comme modèle d'environnement générique pour le véhicule autonome

Cette section présente le contexte du sujet de recherche traité durant ce travail de thèse. Une première partie présente le contexte général du véhicule autonome : sa définition, ses enjeux, un historique de son développement et son principe général de fonctionnement. Une deuxième partie aborde le choix du modèle d'environnement de la grille d'occupation : les capteurs habituellement présents sur un véhicule autonome, les différentes approches de construction d'un modèle d'environnement permettant la navigation autonome et finalement les raisons du choix de la grille d'occupation comme point de départ de ce travail de recherche.

1.1.1 Le véhicule autonome : définition, enjeux, historique et principe de fonctionnement

Un véhicule autonome est un véhicule équipé de capteurs lui permettant de percevoir son environnement, d'un ordinateur de bord prenant des décisions de conduite à partir des informations recueillies et d'actionneurs exécutant ces décisions en agissant

sur les contrôles du véhicule [Bad+21]. L'objectif d'un tel système est de permettre au véhicule de naviguer de manière autonome sur route ouverte (environnement dit *non structuré*) sans l'aide d'un conducteur humain tout en préservant la sécurité des occupants et des autres usagers de la route et en respectant par ailleurs la législation en vigueur. La figure 1.1 présente une photographie du prototype de voiture autonome Navlab 5 qui fut la première voiture autonome à traverser les États-Unis d'est en ouest en 1995.

Figure 1.1. : Le prototype Navlab 5, premier véhicule autonome à traverser les États-Unis en autonomie 98% du temps en 1995 [@Uni].

L'appellation *véhicule autonome* couramment utilisée pour désigner certains véhicules peut laisser penser que l'autonomie serait une technologie monolithique dont seraient pourvus ou non certains véhicules. En réalité, ce terme désigne un ensemble de technologies numériques d'automatisation de certaines tâches de conduites autrement dévolues au conducteur du véhicule [Bad+21]. En ce sens, les systèmes éprouvés de maintien électronique de l'adhérence du pneu durant les freinages tels que l'ABS, les systèmes plus récents faisant appel à des algorithmes de perception avancés équipant certains véhicules commercialisés tels que l'aide au maintien dans la voie, ou encore les systèmes au stade de prototypes industriels permettant une navigation complètement autonome en ville sont tous des éléments du *véhicule autonome*. Un véhicule dit *autonome* désigne généralement un véhicule capable dans certaines conditions et possiblement sous supervision humaine de gérer seul toutes les commandes du véhicule.

En 2014, la *Society of Automotive Engineers* proposa un système de classification des véhicules selon six niveaux d'autonomie [SAE21] :

— Le niveau 0 correspond à un véhicule ne disposant pas d'actionneurs capables d'agir automatiquement sur le véhicule.

— Le niveau 1 correspond à la prise en charge de la conduite latérale ou longitudinale exclusivement, par exemple par l'utilisation d'un régulateur de vitesse ne contrôlant que l'accélération du véhicule.

— Le niveau 2 correspond à la prise en charge de la conduite latérale et longitudinale simultanément, mais avec l'obligation pour le conducteur humain de garder ses yeux sur la route pour reprendre la main en cas d'erreur.

— Le niveau 3 correspond à la prise en charge de toutes les commandes du véhicule sans obligation pour le conducteur humain de regarder la route, mais devant se tenir prêt à reprendre la main si le véhicule le lui demande.

— Le niveau 4 correspond à la prise en charge de toutes les commandes du véhicule de manière complètement autonome, mais uniquement dans certaines zones géographiques.

— Le niveau 5 correspond à une autonomie complète. Dans ce niveau d'autonomie, la présence d'un volant ou de pédales n'est pas requise.

Il est usuellement considéré qu'un véhicule est autonome s'il correspond au moins au niveau 3 d'autonomie.

À l'heure de l'écriture de ce manuscrit, il n'existe pas de véhicule de niveau 5 pouvant se passer d'un conducteur humain en toute circonstance, mais il existe de nombreux véhicules capables d'assurer des tâches de conduites normalement dévolues à un conducteur humain. Par exemple certains véhicules sont désormais capables de gérer seuls l'accélération, le maintien dans la voie et les dépassements sur autoroute [ISO18].

Enjeux

Le développement du véhicule autonome est motivé par différents enjeux de santé publique, sociétaux, écologiques et économiques. Le plus évident de ces enjeux est celui de la réduction de la mortalité sur la route. Dans le monde, approximativement 1,3 million de personnes perdent la vie chaque année dans des accidents de la circulation. Plus de la moitié de ces victimes sont des usagers vulnérables tels que les piétons, les cyclistes et les motocyclistes. Par ailleurs, les accidents de la route sont la première cause de mortalité de la classe d'âge 5-29 ans dans le monde [@Wor22].

En 2020, l'assemblée générale des Nations Unies a adopté une résolution visant à diminuer de moitié le nombre de morts et de blessés suite à des accidents de la route d'ici à 2030 [Uni20]. De nombreux comportements humains problématiques ont été identifiés comme principaux facteurs de risque des accidents de la route. Parmi

ceux-ci, on peut citer le non-respect des limites de vitesse (une augmentation de la vitesse moyenne de 1% conduit à une augmentation du risque d'accident fatal de 4%), l'utilisation du téléphone portable pendant la conduite (cet usage mène à une multiplication par quatre de la probabilité d'avoir un accident), ou encore l'usage de produits psychoactifs comme l'alcool ou les stupéfiants. Il est donc supposé que le véhicule autonome permette une nette diminution des accidents de la route par la suppression de tous ces facteurs de risque.

Une autre problématique pouvant être adressée par le recours au véhicule autonome est celui de l'encombrement des villes dont la réduction entraînerait aussi mécaniquement une réduction des temps de trajet. D'une part, les embouteillages pourraient être réduits par une conduite plus fluide et la communication entre les véhicules (V2V - *Vehicle-To-Vehicle*) et des véhicules aux infrastructures (V2I - *Vehicle-To-Infrastructure*) [FK15]. Une étude de 2012 a d'ailleurs estimé que l'usage de régulateurs automatiques de distance entre véhicules dans 90% du parc automobile accroitrait la capacité des routes de 80% [SSL12]. D'autre part, l'usage de véhicules autonomes pourrait réduire considérablement les besoins de places de stationnement en ville. En effet, un véhicule autonome serait en mesure d'aller se garer de lui-même en périphérie après avoir déposé ses usagers à leur destination.

L'avènement du véhicule autonome aurait aussi un effet en termes d'accessibilité à la mobilité routière. D'une part, un véhicule autonome permettrait aux personnes inaptes à conduire (personnes à mobilité réduite, jeunes, ...) de se déplacer en voiture sans nécessiter de conducteur humain. D'autre part, un véhicule autonome aurait la capacité d'effectuer de nombreux trajets dans la journée pour différentes personnes. Il a par exemple été estimé qu'un unique véhicule autonome pourrait remplacer dix véhicules individuels durant une expérience de véhicules autonomes partagés à Austin, Texas [FK18]. Cette mise en commun de la propriété automobile permettrait aux personnes n'ayant pas les moyens financiers de posséder un véhicule individuel de pouvoir tout de même ponctuellement se déplacer en voiture.

Sur le plan économique, il est attendu une transformation du marché du véhicule d'une économie de biens à une économie de services [SDC24]. Il existe déjà quelques sociétés de taxis exploitant des véhicules autonomes pour conduire leurs clients à destination à un coût réduit en raison de l'absence de chauffeur [@Way ; @Cru]. Des changements sont aussi attendus dans le domaine du transport de marchandises, que ce soit par des livraisons autonomes à toute heure aux pieds des habitations [SAS23] ou dans le domaine du transport longue distance avec la possibilité pour un camion autonome de rouler de jour comme de nuit sans interruptions [Van+23].

Finalement, nombre de ces changements amèneraient à un moindre impact écologique du transport routier. En effet, la diminution des embouteillages due à une meilleure fluidité du trafic et la réduction du nombre de véhicules produits amèneraient à des économies de carburant et de matières premières dont l'extraction et la consommation sont très polluantes [AGS23].

Le développement du véhicule autonome amène néanmoins des défis à résoudre dans les domaines précédemment cités. Des questions se posent par exemple sur les risques de piratage informatique de leurs systèmes de navigation [PS15] et sur l'entité désignée comme juridiquement responsable en cas d'accident. Il est aussi nécessaire de réfléchir collectivement à la requalification de tous les chauffeurs professionnels dont le métier risque de disparaître [Van+23]. Finalement, sur le plan écologique, la baisse des coûts des transports en raison des économies réalisées grâce au véhicule autonome pourrait amener à un *effet rebond* et donc à une augmentation de l'utilisation des moyens de transport routiers, ce qui aurait pour conséquence d'annuler tout effet écologique positif [Ona+23].

Historique

Le plus ancien exemple de voiture semi-autonome fut présenté en 1977 par le laboratoire japonais *Tsukuba Mechanical Engineering Laboratory* [Sri+22]. Cette voiture était capable à l'aide de deux caméras et d'un ordinateur analogique de suivre un marquage spécifique sur la route à une vitesse de 30 km/h. En 1984, l'université Carnegie-Mellon débuta son programme de recherche en véhicule autonome avec des subventions de la DARPA (*Defense Advanced Research Projects Agency*). Ce programme donna lieu à la célèbre lignée de plateformes de recherche en véhicule autonome *Navlab* [@The] dont la première version fut mise au point en 1986. En parallèle à cette initiative états-unienne, le programme européen de recherche *Prometheus* fut mis en place en 1987 pour développer un véhicule complètement autonome [Xie+93]. Ces deux programmes de part et d'autre de l'océan Atlantique donnèrent lieu à d'importantes avancées de la fin des années 1980 au début des années 1990.

En 1995, le prototype Navlab 5 réalisa la première traversée d'est en ouest des États-Unis en autonomie 98% du temps et à une vitesse moyenne de 103 km/h [@Car]. Pour dynamiser les recherches, la DARPA organisa en 2004 une compétition nommée *DARPA Grand Challenge*. L'objectif de cette compétition était que les véhicules compétiteurs parcourent 240 kilomètres dans le désert des Mojaves en Californie en moins de dix heures. Il n'y eu aucun vainqueur cette année-là, mais la compétition fut

reconduite l'année suivante et quatre véhicules finirent dans le temps imparti [@Def]. Finalement, la compétition fut à nouveau organisée en 2007, mais cette fois avec un environnement routier simulé par la présence de véhicules conduits par des humains et l'obligation de respecter la législation routière californienne. Durant cette compétition, six véhicules finirent la course dans le temps imparti [@Mas07].

Les années 2010 marquèrent le passage du véhicule autonome d'un projet de recherche étudié par des universités à une perspective industrielle crédible. Elles donnèrent lieu à la création de multiples entreprises autour du véhicule autonome. En 2009, *Google* démarra son programme *Google Self-Driving Car Project* qui devint par la suite l'entreprise *Waymo* et développa la *Pribot* [@IEE14] : première voiture autorisée à rouler sur route ouverte en 2008. En 2018, Waymo fut le premier à proposer un service de robotaxis complètement autonomes dans la ville de Phoenix en Arizona [@Way18]. En 2021, *Honda* annonça la commercialisation de son premier véhicule de niveau d'autonomie 3 (dont le conducteur n'est pas obligé de superviser la conduite du véhicule en permanence) homologué pour la route au Japon [@Hon21]. En 2022, l'entreprise *Cruise* annonça la disponibilité d'un service de taxis autonomes dans la ville de San Francisco en Californie [@Cru22].

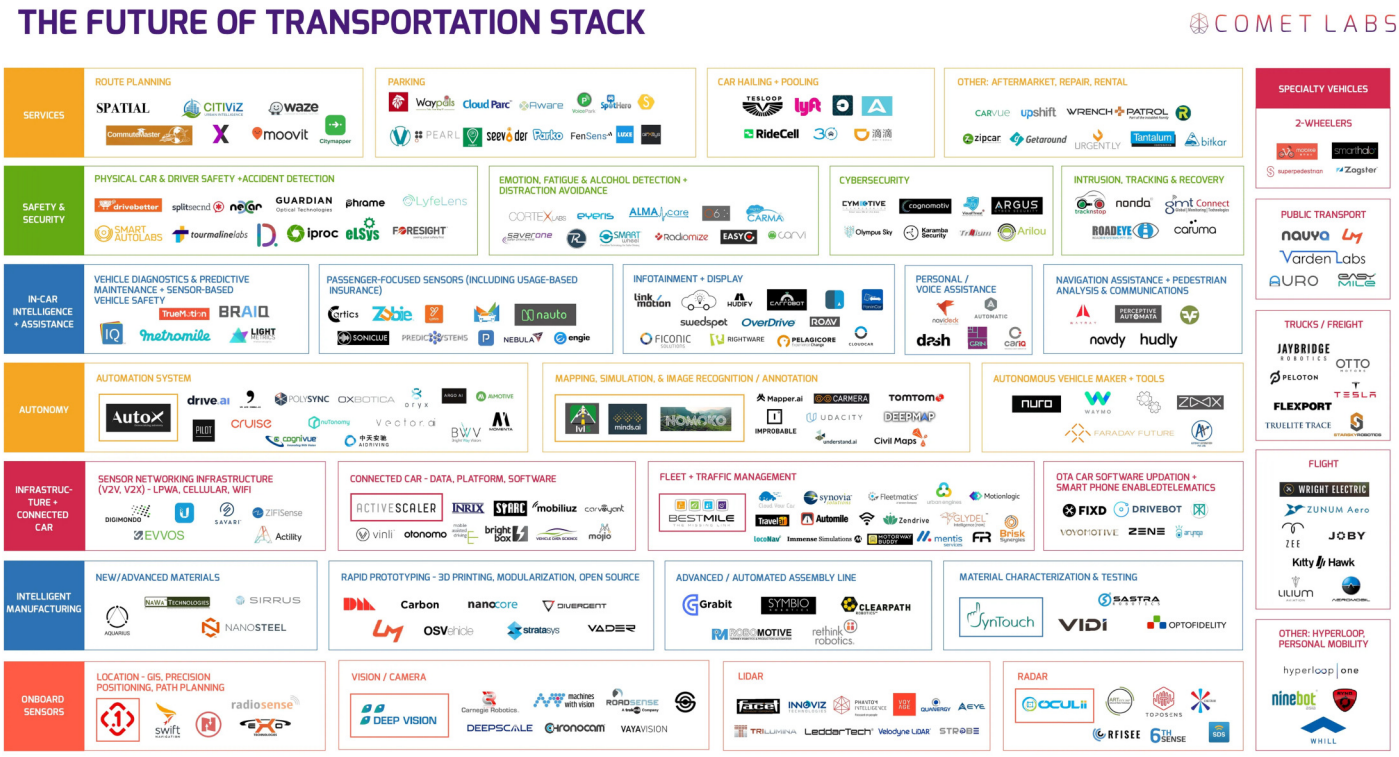

Figure 1.2. : Aperçu des principaux acteurs en activité dans le secteur du véhicule autonome [@Ste17].

À l'heure de la rédaction de ce manuscrit, le nombre d'entreprises en activité dans le secteur du véhicule autonome est très important. On y retrouve les constructeurs de véhicules historiques, les différents équipementiers leur fournissant des systèmes d'assistance à la conduite, et aussi des entreprises nouvellement créées proposant des services de livraison ou de taxis autonomes. La figure 1.2 montre un aperçu des entreprises faisant partie du secteur du véhicule autonome.

Principe de fonctionnement

Un véhicule autonome est équipé de multiples capteurs mesurant différentes propriétés physiques de leur environnement. Des capteurs tels qu'un GNSS (*Global Navigation Satellite System*) ou un IMU (*Inertial Measurement Unit*) permettent au véhicule de se positionner dans l'espace. D'autres capteurs comme les caméras, les RaDAR (*Radio Detection And Ranging*) ou les LiDAR (*Light Detection And Ranging*) permettent de percevoir l'environnement immédiat en mesurant la position des obstacles, leur forme, leur couleur, etc [Van+18].

Les mesures de ces différents capteurs sont ensuite traitées par un système de *perception* dont le rôle est d'interpréter ces différentes mesures pour fournir une représentation numérique de la scène dans laquelle se trouve le véhicule autonome. Cette représentation appelée *modèle de l'environnement* regroupe différentes informations telles qu'une cartographie des différentes zones et obstacles présents dans la scène, voire leur reconnaissance en tant qu'objets distincts (voitures, piétons, cyclistes, murs, trottoirs, chaussée, signalisation, ...) et même parfois leur dynamique. Elle peut aussi contenir des informations telles que la position et la dynamique du véhicule autonome au sein d'un système de coordonnées globales. Elle peut aussi inclure une cartographie embarquée qui aurait été chargée dans la mémoire de l'ordinateur de bord [Bad+21].

Ensuite, un système de *prise de décision* utilise ce modèle de l'environnement pour décider du comportement devant être adopté par le véhicule. Le choix du comportement à adopter dépend de la destination souhaitée, du modèle de l'environnement produit par le système de perception et de l'ensemble des règles données au système pour s'assurer qu'il agisse en respectant la réglementation. Généralement, le fonctionnement du système de prise de décision commence par le choix d'une suite de points de passage par lesquels le véhicule devra passer. Ensuite, un algorithme combine ces points de passage avec sa connaissance de la physique du véhicule et de l'environnement alentour pour décider d'un chemin à suivre physiquement réaliste et qui évite les collisions. Finalement, un dernier système transforme ce chemin en

une suite de commandes à envoyer aux actionneurs du véhicule pour le faire tourner, accélérer ou freiner [Bad+21]. La figure 1.3 illustre ces différents systèmes menant des mesures des capteurs aux commandes effectuées par les actionneurs.

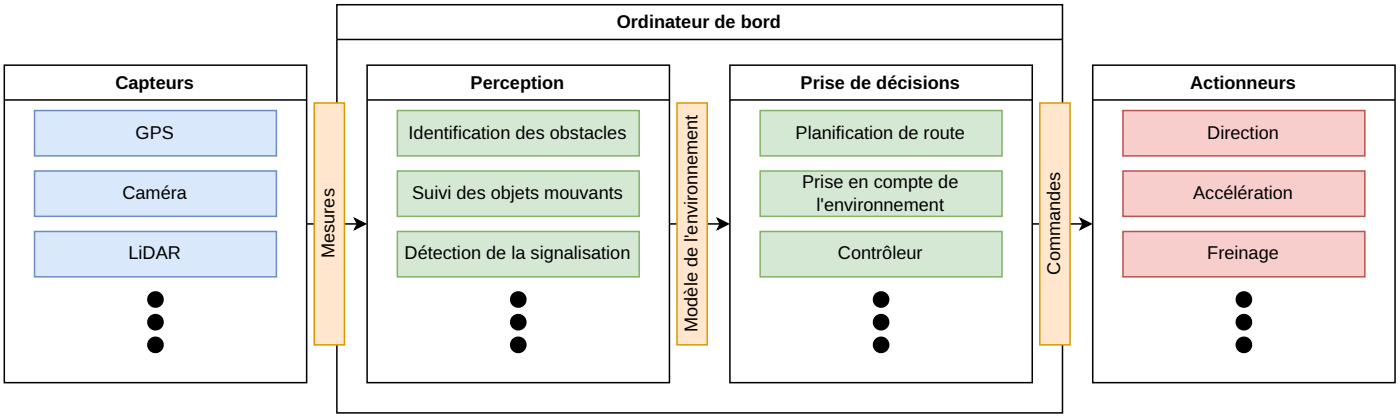

Figure 1.3. : Schéma des différents systèmes présents dans le fonctionnement d'un véhicule autonome. Les capteurs mesurent les propriétés de l'environnement. Ces mesures sont traitées au sein de l'ordinateur de bord par le système de perception qui construit le modèle d'environnement. Ensuite, le système de prise de décision décide des actions à effectuer. Finalement, les actionneurs du véhicule exécutent ces commandes.

L'ensemble des algorithmes constituant les systèmes de perception et de prise de décision doivent être exécutés en *temps réel* pour garantir la sécurité des occupants du véhicule. Cette contrainte de *temps réel* empêche les calculs d'être réalisés sur un serveur distant ce qui induirait de la latence. Cela force donc l'usage d'un ordinateur de bord. Cet ordinateur de bord est fortement contraint en termes de taille, de consommation électrique et de coût. Ces contraintes pratiques induisent des contraintes calculatoires (capacité mémoire, nombre d'opérations à la seconde, ...). Les différents algorithmes utilisés doivent donc respecter ces contraintes en étant capables de produire les résultats escomptés en temps réel dans le budget calculatoire et mémoire alloué [Shi+17]. Ainsi, le développement d'algorithmes de navigation autonome consiste en la recherche d'un compromis entre la qualité des résultats qui permet de garantir la sécurité des usagers et un coût calculatoire suffisamment faible pour être utilisé en temps réel sur le matériel *embarqué*.

1.1.2 La grille d'occupation : des capteurs de distance au modèle d'environnement permettant leur fusion

Le système de perception d'un véhicule autonome est chargé de l'interprétation des mesures des différents capteurs afin de construire une représentation numérique de

la scène appelée *modèle d'environnement*. Le choix d'un tel modèle d'environnement dépend de multiples facteurs, tels que les types de capteurs utilisés ainsi que leur potentielle hétérogénéité, les informations nécessaires à la prise de décisions de navigation, ou encore le budget calculatoire et mémoire alloué à la création du modèle d'environnement en temps réel.

Cette section présente les différents types de capteurs couramment embarqués sur un véhicule autonome et différentes approches de modélisation de l'environnement utilisées. Finalement, le modèle d'environnement de la *grille d'occupation* utilisé durant ce travail de thèse de doctorat est présenté, ainsi que ses différents avantages et inconvénients.

Capteurs intéroceptifs et extéroceptifs

Dans le présent contexte, les capteurs sont des dispositifs de mesure embarqués à bord des véhicules. Ces derniers fournissent des informations sur la situation de conduite. Un des exemples les plus anciens de capteur automobile est le compteur de vitesse présent en tant qu'équipement standard dans les véhicules à moteur depuis les années 1910 [Wik23]. Les premiers capteurs avaient pour but de fournir directement au conducteur une information lui permettant d'adapter sa conduite.

L'arrivée progressive de l'électronique numérique a permis l'utilisation de capteurs plus complexes dont les mesures doivent être interprétées par un calculateur afin d'être utilisables. Parmi ces nouveaux capteurs, certains sont destinés à fournir une information directement au conducteur comme c'était les cas pour les capteurs analogiques. Un exemple de ce type de capteur sont les capteurs de véhicule dans l'angle mort équipant de plus en plus de voitures modernes [Bal+23]. D'autres capteurs de ce type sont intégrés à des systèmes capables d'agir directement sur les commandes du véhicule dans certaines situations de danger. C'est le cas par exemple des systèmes de correction électronique de trajectoire (*Electronic Stability Program* ou *ESP*) capables d'agir sur les freins du véhicule afin d'en redonner le contrôle au conducteur quand un dérapage est détecté [Tan+23].

À la différence de ces multiples systèmes d'aide à la conduite chargés d'assister un conducteur humain, un système de véhicule autonome a pour but de remplacer complètement le conducteur humain dans sa tâche de conduite. Ce niveau d'autonomie signifie que la perception de l'environnement aussi bien proche que lointain doit être effectuée par l'ordinateur de bord et non plus par le conducteur. Pour permettre cette perception de l'environnement, un véhicule autonome doit embarquer des capteurs de plus longue portée que ceux utilisés par les systèmes d'aide à la conduite

traditionnels. Par ailleurs, ces derniers doivent être généralement plus précis, car utilisés pour des tâches de perception plus complexes.

En pratique, aucun capteur n'est en mesure de fournir des informations précises à courte, moyenne et longue portée en tous types de conditions météorologiques et de luminosité. Afin de pallier cette limitation, les véhicules autonomes embarquent généralement de multiples capteurs de technologies différentes afin d'apporter de la complémentarité et de la redondance d'information. Les mesures de ces différents capteurs sont par la suite fusionnées en un unique modèle d'environnement comme expliqué ci-après dans ce chapitre.

Les capteurs utilisés sont classiquement séparés en deux grandes familles : les capteurs *intéroceptifs* d'une part et les capteurs *extéroceptifs* d'autre part [Van+18]. Les capteurs intéroceptifs mesurent des propriétés propres au véhicule autonome sur lequel ils sont montés. Des exemples de capteurs intéroceptifs utilisés pour les véhicules autonomes sont :

— Un *Global Navigation Satellite System* ou *GNSS* utilise une méthode de triangulation en utilisant sa distance à des satellites dont la position est connue pour déterminer les coordonnées globales de latitude et de longitude du véhicule. Cette information est très utile quand elle est couplée à des cartographies enregistrées dans l'ordinateur de bord.

— Une *Inertial Measurement Unit* ou *IMU* est un ensemble de capteurs mesurant au cours du temps la force et la direction des accélérations subies par le véhicule ainsi que son orientation dans l'espace. Ce type de capteur permet à court terme de reconstruire la trajectoire suivie par le véhicule, mais son utilisation seule conduit à des accumulations d'erreurs au cours du temps.

— Un *odomètre*, plus connu dans sa version automobile sous le nom de compteur kilométrique, mesure la distance parcourue par le véhicule au cours du temps, à l'aide de capteurs placés dans la boîte de vitesse de ce dernier.

Les capteurs extéroceptifs quant à eux mesurent des propriétés de l'environnement extérieur. Ces derniers permettent d'obtenir de l'information sur ce qui existe autour du véhicule comme la position des obstacles. Des exemples de capteurs extéroceptifs courants sont :

— Une *caméra* permet d'enregistrer un flux vidéo de ce qu'il se passe autour du véhicule. Ce capteur peu coûteux permet d'obtenir une vision très détaillée de la scène, mais nécessite l'utilisation d'algorithmes complexes de reconnaissance d'image pour pouvoir être exploité. Par ailleurs, une photographie de caméra ne contient pas d'information de distance aux obstacles. Une caméra de véhicule autonome est généralement dans le domaine visible, mais il est aussi possible

dans certains cas d'usages d'utiliser d'autres bandes du spectre lumineux, par exemple avec une caméra infrarouge [Kan+11].

— Un *capteur de distance* désigne un capteur capable de mesurer en trois dimensions la position des autres obstacles. Ces capteurs fonctionnent généralement par l'émission d'une onde qui sera reflétée par les obstacles et dont le reflet sera mesuré par le capteur. La mesure du temps de trajet de l'onde permet par la suite de calculer la distance à laquelle est l'obstacle sur lequel s'est reflétée l'onde. Ces capteurs sont aussi appelés capteurs *Time-Of-Flight* ou *TOF*. Le choix du type d'onde émise caractérise les différents capteurs de distance existants :
  — Un *LiDAR* (*Light Detection And Ranging*) utilise des ondes lumineuses émises par des lasers pour mesurer la distance aux obstacles. Ces capteurs produisent en sortie un nuage de points dans l'espace en trois dimensions dont chaque point correspond à la position d'un obstacle ayant reflété le laser. Bien que coûteux, ces capteurs sont très utilisés pour le véhicule autonome car ils offrent une excellente précision à courte et moyenne portée.
  — Un *RaDAR* (*Radio Detection And Ranging*) utilise des ondes radio pour mesurer la distance aux obstacles. Ces capteurs peu coûteux sont très utilisés car ils permettent d'analyser la scène à longue portée. En revanche, en raison d'importantes erreurs angulaires, la sortie d'un RaDAR nécessite d'importants traitements numériques pour être exploitée.
  — Un *SoNAR* (*Sound Navigation And Ranging*) utilise des ondes acoustiques pour mesurer la distance aux obstacles. Ces capteurs sont principalement utilisés à très courte portée pour détecter des obstacles, par exemple dans un système d'aide au stationnement.
  — Une *caméra TOF* est simplement une caméra couplée à un capteur TOF ce qui permet d'ajouter une information de distance en plus de l'information de couleur de chaque pixel de l'image générée.

Les différents capteurs présentés fournissent des informations de différentes natures sur la situation de conduite. Afin d'en exploiter le plein potentiel, il est nécessaire d'interpréter et de fusionner ces mesures au sein d'une représentation unifiée appelée modèle d'environnement.

Approches de construction d'un modèle d'environnement sémantique

La construction d'un modèle d'environnement à partir des données des différents capteurs revient à l'interprétation conjuguée des mesures de tous les capteurs

embarqués afin d'en extraire le plus d'informations possibles sur l'environnement. Le but de cette opération est d'obtenir une représentation utile pour la prise de décisions de navigation adaptées. Une telle représentation doit contenir des informations complexes telles que la nature, la position et la dimension des différents obstacles (en particulier les obstacles dynamiques, par exemple les voitures, les camions, les vélos ou les piétons) [Bad+21]. Dans la suite de ce manuscrit, nous qualifierons de *sémantiques* ces informations complexes obtenues par interprétation des mesures des capteurs. Cette tâche essentielle de sémantisation des mesures de capteurs est une tâche complexe généralement effectuée par des *modèles d'intelligence artificielle* et plus particulièrement d'*apprentissage profond*. Deux approches de construction de ce modèle d'environnement sémantique se distinguent.

Une majorité de travaux propose d'utiliser directement les mesures de capteurs comme entrées des modèles d'intelligence artificielle d'ajout de sémantique [Che+17 ; ZT18]. L'intérêt principal de ces méthodes réside dans l'utilisation du maximum d'information brute issue des capteurs dont la compréhension est laissée au modèle d'intelligence artificielle. Les mesures de capteurs couramment utilisées par ces approches sont des nuages de points LiDAR, des signaux RaDAR ou des images de caméra. La figure 2.8 présentée plus loin dans ce manuscrit illustre ce type d'approche.

Ces méthodes d'apprentissage de modèles d'intelligence artificielle directement à partir des données brutes de capteurs produisent généralement des interprétations sémantiques de très bonne qualité, mais admettent aussi quelques limites. Premièrement, les mesures brutes des capteurs sont généralement des données de taille conséquente et parfois peu structurées comme les nuages de points qui sont stockées informatiquement sous forme de listes de points. Ces mesures à la structure complexe utilisées en entrée des modèles induisent généralement des coûts calculatoires importants des modèles d'ajout de sémantique. Deuxièmement, les modèles d'apprentissage profond développés ne permettent pas d'observer la manière dont les différentes mesures de capteurs sont utilisées pour produire les interprétations sémantiques finales. Enfin, ces modèles ne sont pertinents que pour un ensemble de capteurs fixé et tout changement ne serait-ce que d'un capteur ou de sa position sur le véhicule nécessite un réentraînement complet du modèle d'intelligence artificielle.

Afin de pallier ces problèmes, une deuxième approche de construction d'un modèle d'environnement sémantique est possible. Celle-ci consiste en l'utilisation d'un premier modèle d'environnement non sémantique dont le but est de fusionner les mesures des différents capteurs en une représentation formalisée, puis par

la suite d'entraîner un modèle d'apprentissage profond sur ce premier modèle d'environnement afin d'y ajouter des informations sémantiques telles que la nature des obstacles présents dans la scène. Cette approche permet de pallier les limites de l'approche d'entraînement directement à partir des données brutes des capteurs. En effet, l'utilisation d'un modèle d'environnement non sémantique en entrée des modèles d'apprentissage profond permet de s'assurer d'une taille d'entrée fixe et structurée et l'influence des différents capteurs peut être observée directement sur le modèle d'environnement intermédiaire. Par ailleurs, cette approche offre de la généricité, car désormais l'ajout d'un nouveau capteur nécessite seulement la fusion de ses mesures dans le modèle d'environnement intermédiaire, mais ne nécessite pas le réentraînement du modèle d'apprentissage profond d'ajout de sémantique. En revanche, un inconvénient de cette approche comparativement à la première est la perte d'information de capteur durant la fusion de ses mesures dans le modèle d'environnement intermédiaire. La figure 1.4 illustre cette différence dans les approches d'extraction de sémantique à partir de mesures de capteurs.

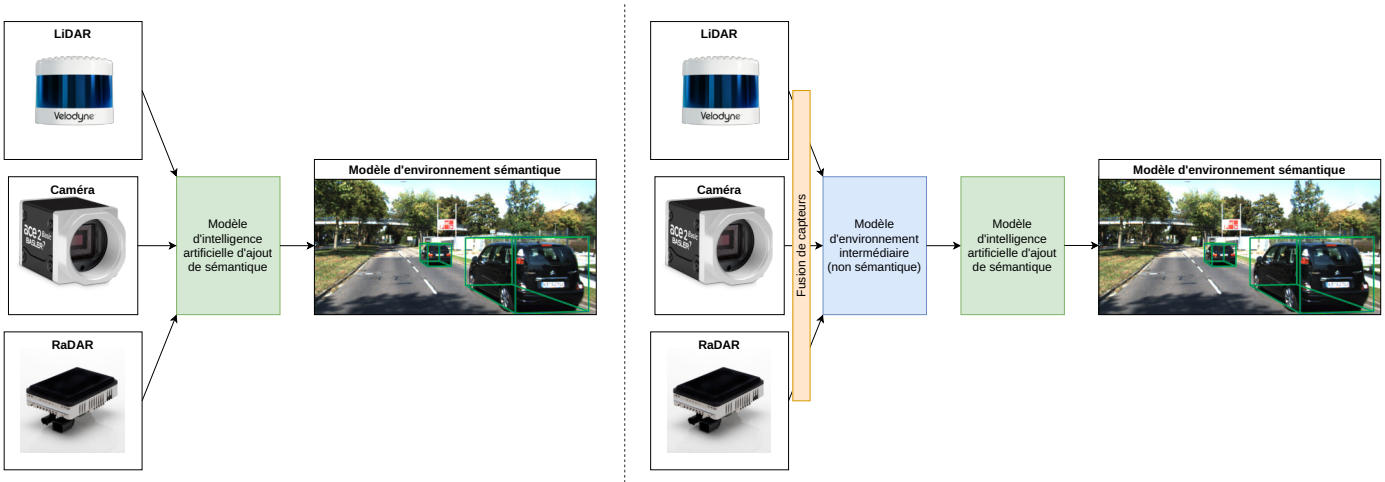

Figure 1.4. : Deux approches existent pour ajouter de la sémantique à des mesures de capteurs. Gauche : Les mesures brutes des capteurs sont directement utilisées en entrée d'un modèle d'extraction de sémantique. Droite : Les mesures des capteurs sont fusionnées en une représentation unique (ou modèle d'environnement non sémantique) qui est ensuite utilisée comme entrée d'un modèle d'extraction de sémantique.

La deuxième approche proposée est celle considérée pour ces travaux de recherche. Le choix restant à effectuer est celui d'un modèle d'environnement intermédiaire permettant de fusionner les données des différents capteurs.

La grille d'occupation comme modèle d'environnement intermédiaire

L'approche choisie durant ce travail de recherche d'utilisation de la fusion des données de multiples capteurs en une unique représentation nécessite le choix de cette représentation ou modèle d'environnement intermédiaire. Un tel modèle d'environnement doit fusionner des données issues de capteurs de différentes technologies dans un même espace de coordonnées. Par ailleurs, du fait de l'usage de multiples capteurs devant être fusionnés, il est nécessaire que le modèle d'environnement choisi puisse représenter l'incertitude liée à des mesures contradictoires. Pour ces raisons, les modèles d'environnement de type *grilles d'occupation* [ME85] ont été retenus.

Il existe deux familles de grilles d'occupation : les grilles d'occupation bidimensionnelles et les grilles d'occupation tridimensionnelles. Ces modèles consistent en une représentation de l'espace dans une grille 2D ou 3D obtenue par une subdivision régulière de l'espace en cellules. Dans le cas d'une grille 2D, la projection choisie est généralement une projection en vue de dessus motivée par le fait que tous les objets se déplacent sur le même plan correspondant à la surface du sol.

Les cellules de grilles d'occupation contiennent un indicateur quant à l'occupation de la cellule par un obstacle. Cet indicateur peut être une simple valeur binaire indiquant si la cellule contient un obstacle ou non. Il peut aussi s'agir d'une probabilité d'occupation auquel cas l'incertitude d'occupation de la cellule peut être représentée. Dans certains cas, des mesures additionnelles sont présentes telles que l'intensité lumineuse mesurée par un capteur LiDAR [Wir+18], ou encore une information de vélocité des obstacles présents dans les cellules [Hoe+18]. L'objectif de généricité du modèle d'environnement nécessite qu'aucune information spécifique à un certain type de capteur ne soit utilisée dans ce travail.

Les grilles d'occupation utilisées dans ce travail sont des grilles d'occupation bidimensionnelles. Ce choix permet d'assurer une compacité des grilles d'occupation plus importante qu'avec les grilles tridimensionnelles ce qui est cohérent avec l'objectif de légèreté des méthodes développées. Il s'agit d'une représentation matricielle probabiliste bidimensionnelle en vue de dessus de l'environnement dont les cellules contiennent des probabilités d'occupation. Dans ce contexte, la valeur numérique 0 signifie que la cellule est vide et la valeur 1 qu'elle est occupée. Les valeurs intermédiaires permettent de quantifier l'incertitude quant à l'occupation de la cellule. La valeur médiane $0,5$ représente une absence totale d'information ; elle est très utile pour représenter l'état des cellules non observées (comme celles se trouvant derrière un obstacle par exemple). Ce choix d'une variation continue de probabilité

d'occupation permet de prendre en compte les incertitudes dues à des mesures bruitées ou à des incohérences entre deux capteurs différents.

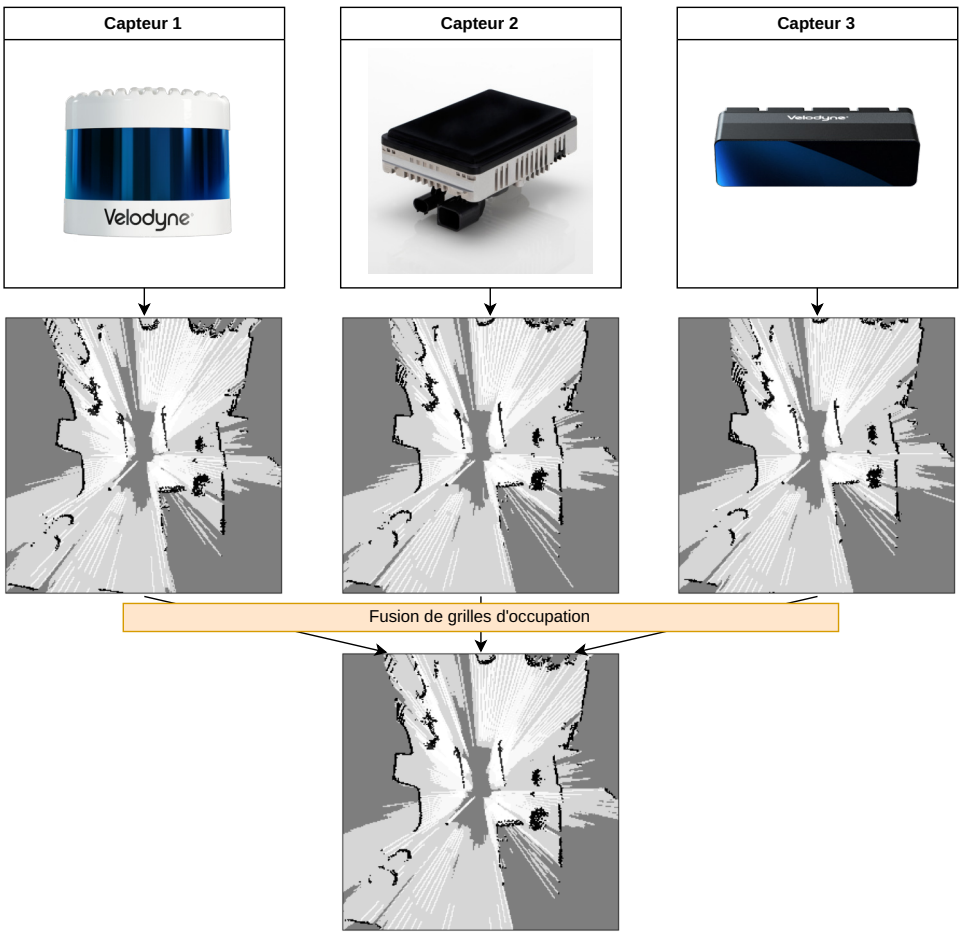

Figure 1.5. : Illustration du procédé de création d'une grille d'occupation à partir de trois capteurs différents. Dans cet exemple, les trois grilles d'occupation correspondant à trois capteurs différents sont fusionnées en une unique grille d'occupation dont les probabilités d'occupation sont encodées en niveaux de gris allant du blanc pour une probabilité d'occupation nulle au noir pour une probabilité d'occupation certaine.

La création d'une telle grille d'occupation peut être réalisée à partir de n'importe quel capteur de distance à la condition de disposer d'un modèle de bruit du capteur. Par ailleurs, deux grilles d'occupation construites à partir des mesures de deux capteurs différents peuvent être fusionnées en une unique grille dont les probabilités d'occupation sont mises à jour à l'aide d'un mécanisme de fusion bayésienne. Ainsi, à condition d'avoir un moyen de calculer une grille d'occupation pour chaque capteur embarqué, il est possible d'obtenir une unique grille d'occupation issue de la fusion des mesures de tous ces capteurs. Finalement, un dernier avantage de ces grilles d'occupation est que leur coût calculatoire est très faible ce qui permet de les calculer

en temps-réel [Rak17]. Une illustration d'une fusion de trois capteurs de distance est donnée sur la figure 1.5.

En raison de l'absence d'informations de profondeur dans les images capturées par une caméra non TOF, il n'est pas possible d'intégrer de données d'images dans une grille d'occupation. Il s'agit d'une limitation de ces dernières. À notre connaissance, il n'existe pas de modèle d'environnement capable de fusionner les données de tous les capteurs présents sur un véhicule autonome. Les grilles d'occupation, capables de fusionner les mesures de n'importe quel capteur de distance, semblent être le modèle d'environnement le plus générique.

Ainsi, le modèle d'environnement de la grille d'occupation permet une représentation de l'environnement générique, car capable de fusionner des mesures de différents capteurs de distance. Ce modèle d'environnement est par ailleurs probabiliste donc capable de représenter des incertitudes, avec un coût calculatoire faible et une résolution de cellule configurable selon les besoins de précision de la représentation. Pour ces raisons, les grilles d'occupation sont couramment utilisées dans les applications de navigation autonome. La tâche abordée par ce travail de recherche sera l'exploitation de telles grilles d'occupation bidimensionnelles par des modèles légers de reconnaissance de véhicules.

1.2 Sujet : La reconnaissance de véhicules dans des grilles d'occupation en temps réel pour un système de perception embarqué

Le modèle d'environnement de la grille d'occupation est une représentation de l'occupation de l'espace très utile grâce à ses capacités de représentation de l'incertitude, de fusion de multiples capteurs hétérogènes et de possibilité de calcul en temps réel sur du matériel contraint. Malgré ces qualités, la grille d'occupation ne permet pas à elle seule la prise de décisions de navigation en raison du manque d'information au niveau des objets présents dans la scène. En effet, les informations contenues dans une grille d'occupation ne portent que sur l'état de chaque cellule individuelle [ME85].

Une interprétation de la grille d'occupation est nécessaire pour identifier les objets composant la scène et ainsi être en mesure de prendre les décisions adéquates. Ainsi, savoir qu'un ensemble de cellules contigües appartient au même objet permet de savoir que ces cellules se déplaceront toutes dans la même direction. De plus, être en

mesure de classifier cet objet comme par exemple un véhicule motorisé, un cycliste ou un piéton permet d'anticiper le comportement de cet objet. En effet, une voiture, un cycliste ou un piéton ne se déplacent pas à la même vitesse, ne circulent pas de la même manière, ne sont pas soumis aux mêmes règles de circulation. Enfin, ils ne présentent pas la même vulnérabilité.

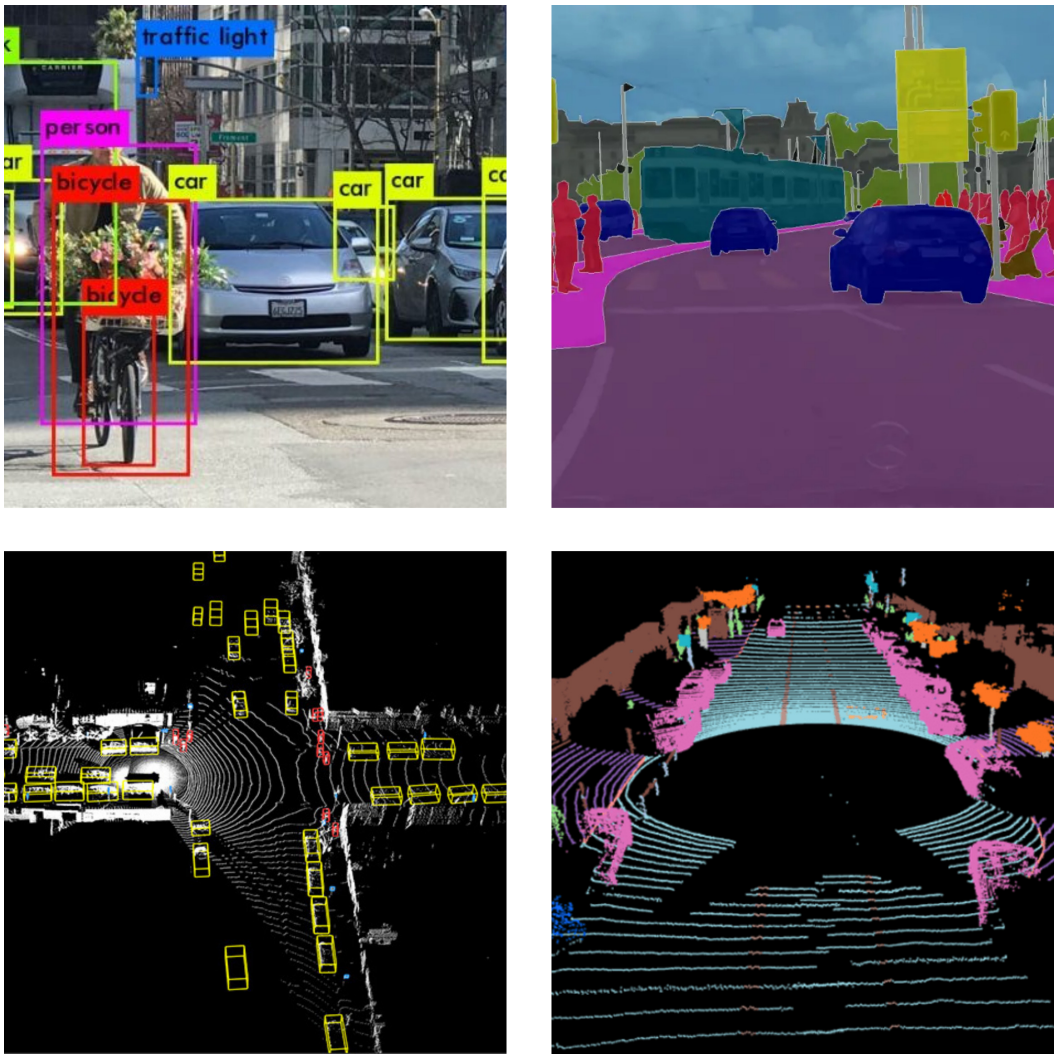

Figure 1.6. : Différentes tâches de reconnaissance d'objets utilisées pour le véhicule autonome. De gauche à droite et de haut en bas : Prédiction de boîtes englobantes 2D sur image ; Segmentation d'image ; Prédiction de boîtes englobantes 3D sur nuage de points ; Segmentation de nuage de points.

La reconnaissance des différents objets composant la scène dans le cadre du véhicule autonome peut se décliner sous différentes tâches. Une tâche de *détection* sur image vise par exemple à englober les objets visibles sur l'image par un rectangle horizontal appelé *boîte englobante* [Gir+14]. Cette tâche de détection existe aussi

en trois dimensions à partir d'une représentation 3D comme un nuage de points et vise à produire des boîtes englobantes tridimensionnelles [Eng+17]. Les boîtes englobantes produites peuvent être annotées par une classe correspondant au type d'objet présent dans la boîte englobante (voiture, piéton, etc.).

Une autre tâche permettant d'identifier les objets présents dans la scène est une tâche de *segmentation* d'images. Dans celle-ci, plutôt que de produire des boîtes englobantes, chaque pixel est annoté selon l'objet auquel il appartient [LSD15]. Il est aussi possible de réaliser la segmentation d'un nuage de points auquel cas ce sont les points le composant qui sont annotés [Mil+19]. Enfin, deux types de segmentation existent : la *segmentation sémantique* n'attribue une annotation à chaque élément que selon le type d'objet auquel il appartient alors que la *segmentation des instances* attribue une annotation différente à deux éléments correspondant au même type d'objet, mais pas à la même instance de l'objet, par exemple deux pixels correspondant à deux voitures différentes [DHS16]. La figure 1.6 illustre les tâches de détection et de segmentations appliquées à deux entrées différentes : des images et des nuages de points.

Les méthodes utilisées pour ces différents types de reconnaissance d'objets reposent sur des méthodes d'apprentissage profond. Dans le cas de traitement d'images, les méthodes utilisent des réseaux convolutifs prenant en entrée directement les images [LSD15 ; Red+16 ; DHS16]. Les méthodes à base de nuages de points sont plus complexes et fonctionnent généralement à l'aide de réseaux de neurones encodant les points dans un tenseur utilisé ensuite en entrée d'un réseau de neurones convolutif [Mil+19].

Cette thèse a pour objet l'interprétation de grilles d'occupation et plus particulièrement la reconnaissance des véhicules composant la scène.

Les contraintes spécifiques au véhicule autonome d'exécution en temps réel des algorithmes sur du matériel contraint sont particulièrement prises en compte dans le travail proposé ici par la mesure systématique des temps d'exécution des algorithmes développés sur du matériel limité en ressources de calcul. Cela permet d'obtenir une estimation des coûts calculatoires des méthodes proposées et permet leur comparaison. De plus, le terme de réseaux de neurones *légers* utilisé tout au long de ce manuscrit désigne des réseaux de neurones pouvant être exécutés en temps réel au taux de rafraichissement des capteurs sur des cibles matérielles automobiles embarquées telles que les ordinateurs de bords Nvidia DRIVE Thor [@Kan22], la série de puces Renesas R-Car [@Ren] ou encore la série de puces NXP i.MX [@NXP].

Ce manuscrit présente deux approches différentes permettant d'identifier les véhicules sur une grille d'occupation. La première approche s'inspire des réseaux de neurones convolutifs utilisés pour de la prédiction de boîtes englobantes sur des images. En effet, une grille d'occupation, en tant que matrice de probabilités d'occupation, présente de nombreuses similarités avec une image ce qui rend son exploitation par des réseaux de neurones convolutifs possible. Une deuxième approche propose d'effectuer la segmentation de grilles d'occupation à l'aide de réseaux denses fonctionnant dans le domaine fréquentiel. À notre connaissance, cette approche n'a pas été reportée dans la littérature et permet des segmentations de qualité avec un coût calculatoire restreint.

1.3 Contributions

Ce manuscrit présente les contributions à l'état de l'art en interprétation de grilles d'occupation suivantes :

— Dans un premier temps, on présente la constitution d'un jeu de données contenant des grilles d'occupation annotées avec des boîtes englobantes orientées pour les véhicules, piétons et cyclistes. Ce jeu de données est analysé qualitativement et quantitativement pour identifier les biais éventuels qui s'y trouvent.

— Dans un deuxième temps, un ensemble d'architectures convolutives de détection de véhicules sur des grilles d'occupation par la production de boîtes englobantes orientées est présenté. Les réseaux de neurones en question sont comparés selon la qualité de leurs résultats d'une part, et selon le temps nécessaire à la production de ces résultats pour une grille d'autre part. Une des architectures est particulièrement mise en avant pour sa capacité à produire des résultats très qualitatifs en un temps de l'ordre de la dizaine de millisecondes.

— La dernière contribution présentée consiste en la réduction de la dimensionnalité d'une grille d'occupation via son passage dans le domaine fréquentiel. Les fréquences les moins importantes du spectre obtenu sont éliminées ce qui permet la réduction effective de la dimension de cette représentation de grille d'occupation. Finalement, différentes architectures correspondant à différentes dimensions de spectre permettant de réaliser une segmentation des véhicules de la grille d'occupation dans le domaine fréquentiel sont présentées. L'entièreté du processus présenté de segmentation de grilles d'occupation compressées dans le domaine fréquentiel offre des résultats performants pour

un temps d'exécution et des tailles de réseaux de neurones dix fois inférieures aux méthodes de la deuxième contribution.

1.4 Plan du manuscrit

Le chapitre 2 présente les différents états de l'art nécessaires à la compréhension du travail présenté. Ce chapitre est composé de trois parties : une présentation théorique des grilles d'occupation, une présentation théorique des méthodes d'apprentissage profond, des réseaux de neurones convolutifs et de leurs applications dans le domaine de l'interprétation de grilles d'occupation, et finalement une présentation théorique de méthodes de transformations d'images dans le domaine fréquentiel ainsi que leurs applications en lien avec les travaux de la troisième contribution de ce travail de thèse.

Le chapitre 3 présente la première contribution consistant en la mise au point d'un jeu de données de grilles d'occupation annotées.

Le chapitre 4 présente la détection de véhicules sur grilles d'occupation à l'aide de réseaux de neurones convolutifs. Cette contribution a amené à la publication de deux articles pour la conférence GRETSI'22 [Def+22] et le journal MDPI Sensors [Def+23]. Un brevet a par ailleurs été déposé concernant ces travaux.

Le chapitre 5 présente la représentation de grilles d'occupation dans un espace fréquentiel de faible dimension et le développement de réseaux de segmentation de véhicules dans cet espace de faible dimension. Un article en cours de rédaction détaillant ces travaux sera soumis pour la conférence EUSIPCO'24. Par ailleurs, un brevet est en cours de dépôt concernant cette contribution.

Finalement, le chapitre 6 conclut et propose des perspectives de recherche pour de futurs travaux.

2 État de l'art

L'objectif de cette thèse est le développement d'algorithmes d'apprentissage profond légers pour la reconnaissance de véhicules sur des grilles d'occupation. Ce chapitre présente les prérequis nécessaires à la compréhension des différentes contributions qui seront présentées par la suite.

La première partie de ce chapitre aborde le modèle d'environnement de la grille d'occupation, utilisé comme point de départ des algorithmes de perception élaborés dans ce manuscrit. La création de grilles d'occupation monocapteur ainsi que la fusion de multiples grilles d'occupation monocapteur en une unique grille multicapteurs y sont présentées.

La deuxième partie présente le fonctionnement des méthodes d'apprentissage profond, et en particulier des modèles convolutifs utilisés en vision par ordinateur. S'ensuit une présentation des travaux de l'état de l'art pour l'interprétation de grilles d'occupation à l'aide de réseaux de neurones convolutifs.

Finalement, une troisième partie présente le fonctionnement des représentations spectrales d'images ainsi que leur application à des tâches de perception. Ces représentations spectrales sont au cœur de la dernière contribution apportée par ce manuscrit.

2.1 Procédé de construction d'une grille d'occupation multicapteurs et propriétés

Une grille d'occupation vise à fournir une représentation des obstacles présents dans l'environnement à partir des mesures issues des différents capteurs de distance [ME85]. Dans ce modèle d'environnement, l'espace est représenté de manière discrète sous la forme d'une grille contenant des cellules de taille donnée. Chacune de ces cellules contient une probabilité d'occupation qui modélise l'incertitude quant à la présence d'un obstacle dans cette cellule. Une probabilité de 0 signifie que la cellule ne contient pas d'obstacle avec certitude et une probabilité de 1 signifie que la cellule contient un obstacle avec certitude. Les valeurs intermédiaires

permettent de quantifier l'incertitude sur l'état de la cellule, une probabilité de 0.5 représentant une absence totale d'information sur l'état de celle-ci. Une grille d'occupation peut avoir deux [Elf89] ou trois dimensions [MS15]. Par ailleurs, le système de coordonnées utilisé peut être cartésien [Elf89 ; MS15], polaire [Bad+08] ou encore sphérique [Pat+07]. Dans la suite de ce manuscrit sont considérées les grilles d'occupation cartésiennes à deux dimensions correspondant à une projection en vue de dessus de l'environnement.

2.1.1 Construction d'une grille d'occupation monocapteur

Le processus de construction d'une grille d'occupation à partir des données des capteurs de distance comporte plusieurs étapes [TBF05]. Dans un premier temps, il est nécessaire de décider des deux paramètres que sont la portion d'espace recouverte par la grille d'occupation et la résolution de cette dernière. La portion d'espace couverte par la grille d'occupation dépend de multiples facteurs. Un de ces facteurs est la portion d'espace intéressante pour la navigation. Il est souvent plus important durant la conduite de connaître les obstacles présents devant le véhicule que ceux présents derrière, ce qui peut amener à faire le choix de ne considérer que l'espace présent en avant par rapport au véhicule. Ensuite, les différents capteurs de distance embarqués à bord du véhicule ont une portée limitée. Il est donc inutile de calculer une grille d'occupation couvrant un espace sur lequel les capteurs de distance ne peuvent pas fournir de données.

En plus de la portion d'espace couverte par la grille d'occupation, il est nécessaire de décider de sa résolution. La résolution d'une grille d'occupation désigne la densité de cellules qui la composent. Par exemple, une grille d'occupation comportant 100 cellules au mètre carré (soit des cellules de 10 centimètres de côté) aura une résolution supérieure à une grille d'occupation comportant 25 cellules au mètre carré (soit des cellules de 20 centimètres de côté). Le choix de la résolution d'une grille d'occupation dépend de l'usage qui en est fait. Une meilleure résolution permet d'augmenter la précision quant au contour des obstacles, mais a un impact important sur le coût calculatoire de la production de cette grille d'occupation et sur son coût de stockage en mémoire [Rak17]. Ainsi, un équilibre doit être trouvé entre précision et coût et cet équilibre dépend à la fois de la tâche à réaliser et aussi de la puissance des calculateurs embarqués. La figure 2.1 schématise la sélection de la portion d'espace couverte par la grille d'occupation ainsi que la grille d'occupation idéale dans cette situation.

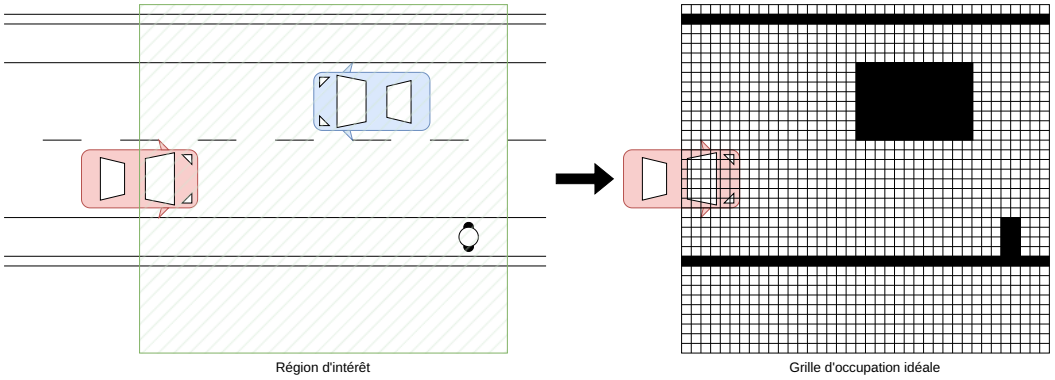

Figure 2.1. : Illustration du processus de sélection de l'espace à considérer et de la grille d'occupation idéale associée.

Une grille d'occupation contient une information sur les obstacles présents dans la scène. Le terme obstacle désigne ici tout objet matériel pouvant être présent dans un contexte de navigation. Ces différents obstacles sont de taille et de forme variables. Les différents capteurs de distance embarqués à bord du véhicule ont pour tâche d'identifier la position de ces obstacles par l'usage d'ondes lumineuses, électromagnétiques ou sonores. Cependant, aucun de ces capteurs n'étant parfait, les mesures produites ne dépendent pas seulement des obstacles, mais aussi des conditions météorologiques de l'environnement et des propriétés mécaniques et électroniques de ces capteurs. Ces éléments de variabilité introduisent une incertitude quant aux mesures retournées par les capteurs de distance [Van+18].

Ces incertitudes inhérentes aux capteurs sont prises en compte dans un modèle associé au capteur [TBF05]. Étant donné l'état de l'environnement s, la mesure z retournée par un capteur est modélisée par une distribution de probabilité $p(z \mid s)$. Cette modélisation sous forme probabiliste de la mesure d'un capteur permet de s'abstraire des différentes conditions affectant celles-ci en considérant la mesure comme un tirage aléatoire conditionné uniquement par l'état de l'environnement. En pratique, ce modèle de capteur $p(z \mid s)$ peut être estimé de manière fréquentielle par la prise de nombreuses mesures dans des conditions différentes.

La construction d'une grille d'occupation consiste en le calcul de la probabilité pour chaque cellule d'être occupée par un obstacle. Soit C_i la variable aléatoire correspondant à l'état d'occupation de la cellule i, à valeurs dans $\{\text{Occupé}, \text{Libre}\}$. Soit O_i l'évènement "$C_i = $ Occupé" et L_i l'évènement "$C_i = $ Libre". La probabilité d'occupation de la cellule i en fonction de la mesure de capteur z est $P(O_i \mid z)$.

Par application du théorème de Bayes [Joy21] l'on obtient ensuite :

$$P(O_i \mid z) = \frac{P(z \mid O_i)P(O_i)}{P(z \mid O_i)P(O_i) + P(z \mid L_i)P(L_i)}$$

On introduit ensuite l'hypothèse suivante :

Hypothèse 1 (Hypothèse de non-information préalable à la mesure)

$$P(O_i) = P(L_i) = \frac{1}{2}$$

Finalement, le calcul de $P(O_i \mid z)$ dépend des valeurs $P(z \mid O_i)$ et $P(z \mid L_i)$. Ces valeurs sont des points de la fonction $P(z \mid C_i)$ qui est appelé *modèle de capteur inverse*. Ces modèles de capteur inverses sont conçus manuellement à l'aide des caractéristiques de bruit propres à chaque capteur.

2.1.2 Fusion de multiples grilles monocapteur en une unique grille multicapteurs

Un véhicule comporte généralement plusieurs capteurs de distance permettant d'obtenir des mesures fiables quelles que soient les conditions météorologiques et lumineuses extérieures. Chacun de ces capteurs peut être utilisé pour la création d'une grille d'occupation spécifique à l'aide de la formule démontrée précédemment.

Par la suite, ces grilles d'occupation monocapteur peuvent être fusionnées en une grille d'occupation multicapteurs à l'aide de mécanismes de fusion Bayésienne. Supposons que l'on dispose de deux capteurs fournissant deux mesures z_1 et z_2. Supposons par ailleurs que les mesures des deux capteurs sont indépendantes :

Hypothèse 2 (Indépendance conditionnelle des mesures de capteurs) *Soit* $s_i \in \{O_i, E_i\}$. *Alors :*

$$p(z_2 \mid s_i \wedge z_1) = p(z_2 \mid s_i)$$

Il est alors possible par utilisation de la formule de Bayes [Joy21] d'obtenir la probabilité d'occupation d'une cellule i sachant les deux mesures z_1 et z_2 :

$$P(O_i \mid z_1 \wedge z_2) = \frac{P(O_i \mid z_1) \cdot P(O_i \mid z_2) \cdot (P(O_i) - 1)}{P(O_i) \cdot (P(O_i \mid z_1) + P(O_i \mid z_2) - 1) - P(O_i \mid z_1) \cdot P(O_i \mid z_2)}$$

Il est donc possible de calculer les probabilités d'occupation $P(O_i \mid z_1 \wedge z_2)$ de la grille d'occupation fusionnant les mesures des deux capteurs z_1 et z_2 à partir des grilles d'occupation propres à chaque capteur contenant les probabilités d'occupation $P(O_i \mid z_1)$ et $P(O_i \mid z_2)$. La démonstration complète en est donnée par [Rak17].

Finalement, les grilles d'occupation sont un modèle d'environnement permettant la fusion de multiples capteurs de distance. Le seul prérequis à une telle fusion est d'être en mesure de fournir un modèle de capteur $p(z \mid s)$ ou directement un modèle de capteur inverse $p(s \mid z)$. Cet avantage, associé au faible coût calculatoire [Rak17] de ces grilles d'occupation en font un modèle d'environnement très utilisé en robotique pour représenter l'espace environnant le robot.

2.2 Méthodes d'apprentissage profond pour la détection de véhicules sur grilles d'occupation

Le travail de recherche présenté dans ce manuscrit a pour objectif la reconnaissance des véhicules présents dans des grilles d'occupation. Comme dans la plupart des tâches de perception, des méthodes d'intelligence artificielle et plus précisément d'apprentissage profond seront utilisées pour réaliser cette tâche. Cette section vise à résumer les bases théoriques sur lesquelles s'appuie l'apprentissage profond. Une première partie présentera l'apprentissage profond de manière globale. Une seconde partie présentera les méthodes d'apprentissage profond utilisées pour le traitement d'image et qui seront au cœur de la contribution présentée au chapitre 4. Une dernière partie présentera des applications de ces méthodes pour réaliser des détections d'objets dans le contexte du véhicule autonome.

L'intelligence artificielle est un domaine d'étude en mathématiques appliquées visant à mettre au point des systèmes informatiques capables de simuler l'intelligence humaine [Tho20]. Cette thématique vient du paradoxe apparent entre d'une part la rapidité surhumaine à laquelle un système informatique exécute les calculs spécifiés par un algorithme et d'autre part l'extrême difficulté que les informaticiens ont à faire reproduire par un ordinateur des tâches intuitives pour un être humain comme la reconnaissance d'une image ou la tenue d'une discussion cohérente. La difficulté majeure dans les différentes tâches d'intelligence artificielle réside dans l'impossibilité pour un humain d'expliquer ses propres processus de cognition inconscients par une suite d'instructions simples pouvant être spécifiée sous forme d'algorithme informatique.

La plupart des techniques d'intelligence artificielle développées sont inspirées du fonctionnement de l'apprentissage humain. On enseigne à un humain à reconnaître un objet en lui présentant des exemples de cet objet. À partir de ces exemples, le cerveau est capable d'identifier les motifs importants caractéristiques de cet objet. Un algorithme de son côté nécessite une suite d'instructions détaillant quels facteurs permettent de décider si oui ou non l'objet est présent dans une image.

Les chercheurs en intelligence artificielle se sont penchés sur le fonctionnement du cerveau pour tenter de comprendre la manière dont il apprend et ainsi tenter de le reproduire informatiquement. La découverte du fonctionnement des neurones biologiques a permis la création de techniques d'apprentissage automatique sous la forme de réseaux de neurones artificiels. Ces techniques d'apprentissage automatique permettent un apprentissage par l'exemple similaire à l'apprentissage humain [LBH15].

Aujourd'hui, l'intelligence artificielle est devenue pratiquement synonyme d'apprentissage automatique à l'aide de réseaux de neurones artificiels. Ces derniers ont permis des avancées majeures dans les domaines de la compréhension et du traitement des images [KSH17], des flux audio [Hin+12] et vidéo [Kar+14], dans le traitement et la production de langage naturel [Bro+20]. Ils ont aussi apporté des solutions à des problèmes combinatoires trop complexes pour être traités par des algorithmes classiques comme certains jeux de stratégie [Sil+16] ou des problématiques médicales comme la prédiction de la structure des protéines à partir de leur séquence d'acides aminés [Jum+21]. Dans un grand nombre de ces exemples, les réseaux de neurones développés ont dépassé les capacités humaines et ces techniques sont aujourd'hui utilisées pour répondre à un grand nombre de problèmes, y compris des problèmes auxquels l'intelligence humaine ne permet pas de répondre.

2.2.1 Principes généraux de l'apprentissage profond

Cette sous-section introduit les définitions, résultats et méthodes permettant une compréhension élémentaire des méthodes d'apprentissage profond utilisées dans ce manuscrit. Le lecteur souhaitant approfondir ses connaissances à ce sujet est invité à lire [GBC16] qui englobe les notions présentées ici ainsi que d'autres plus avancées afin de fournir une compréhension fine des méthodes d'apprentissage profond.

Apprentissage automatique

L'apprentissage automatique désigne un ensemble de méthodes de résolution de problèmes utilisant des exemples de données du problème pour trouver une solution adaptée. Le terme d'apprentissage est utilisé par analogie avec le raisonnement humain qui résout les problèmes par apprentissage à partir d'exemples plutôt qu'à l'aide d'une suite fixée d'instructions préalablement connue comme c'est le cas des algorithmes informatiques classiques. L'ensemble des exemples utilisé pour l'apprentissage d'une solution est appelé un jeu de données.

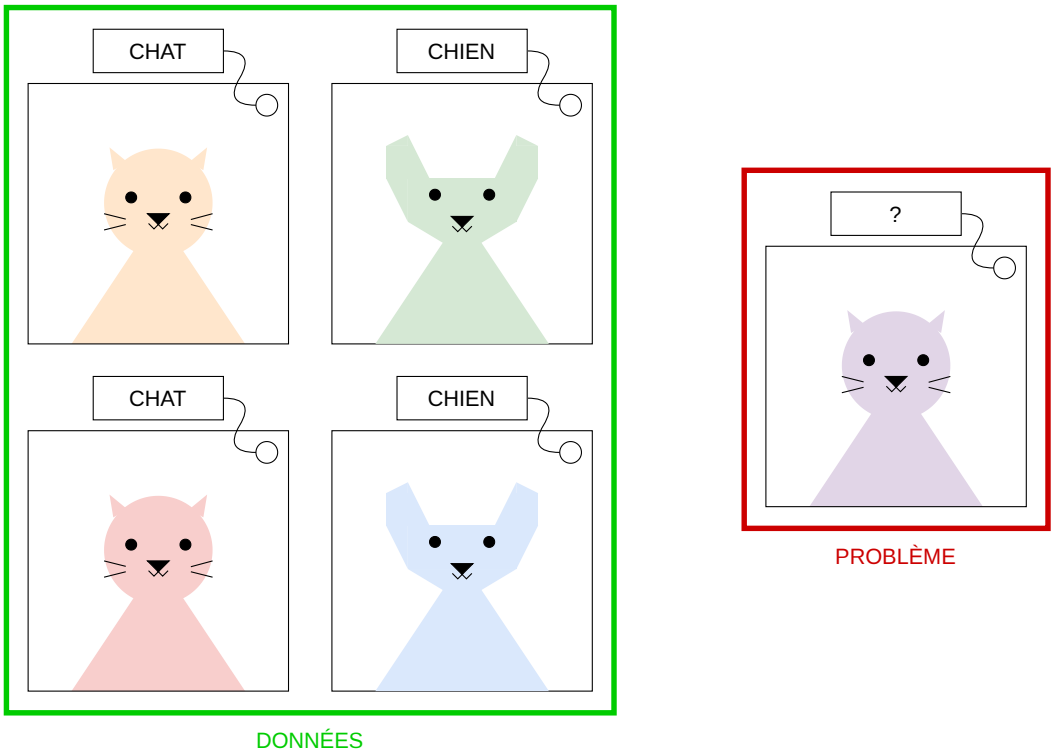

Figure 2.2. : Illustration d'une problématique d'apprentissage automatique de classification d'images en "CHIEN" ou "CHAT".

Une tâche d'apprentissage automatique peut être supervisée ou non supervisée. Une tâche *d'apprentissage supervisé* vise à modéliser une fonction $f : A \rightarrow B$ par une fonction approchée $\hat{f}$ à partir d'exemples de points de la fonction à modéliser. Dans ce cas de figure, le jeu de données $\mathcal{D}$ est un ensemble de points de la fonction f à modéliser : $\mathcal{D} = \{(x_i, f(x_i)) \in A \times B\}$. Les tâches d'apprentissage supervisé sont appelées tâches de classification quand B est un ensemble discret ou tâches de régression quand B est un ensemble continu. Prenons l'exemple d'une tâche de prédiction du type d'un animal en "CHIEN" ou "CHAT" sur une photographie à partir d'un jeu de données contenant des photographies d'animaux avec pour chacune

de ces photographies une étiquette indiquant le type d'animal représenté. Dans cet exemple, l'ensemble A est l'ensemble des photographies de chats et de chiens possibles. L'ensemble B est quant à lui l'ensemble $\{\text{CHIEN}, \text{CHAT}\}$. Finalement, la fonction f est la fonction associant pour toute photographie de A le type correct de l'animal. Étant donné que l'ensemble B est discret (car de taille finie), cette tâche de prédiction est une tâche de classification. La figure 2.2 illustre une tâche de classification de photographies d'animaux.

Une tâche d'*apprentissage non supervisé* quant à elle est une tâche d'apprentissage automatique pour laquelle il n'existe pas d'exemples de prédiction. Dans ce cas de figure, le jeu de données $\mathcal{D}$ contient uniquement des exemples de données d'entrée du problème : $\mathcal{D} = \{x_i \in A\}$. Des exemples classiques de tâches d'apprentissage automatique non supervisées sont les tâches de partitionnement de données. Dans ces exemples, on ne dispose que de points de données dans un ensemble A et l'objectif est de former des groupes de points homogènes sans disposer d'informations supplémentaires sur ces points.

L'objectif d'une tâche d'apprentissage automatique est de trouver une solution à un problème à partir d'exemples. En pratique, cette solution est une fonction paramétrisée spécifique dans un espace de recherche. La recherche de la bonne fonction est définie comme une tâche d'optimisation visant à trouver les paramètres qui minimisent une certaine métrique dépendant de la tâche à résoudre. Ce problème d'optimisation est finalement résolu de manière exacte ou approchée par une méthode variant selon la formulation de la tâche d'optimisation et l'espace de recherche des solutions.

Apprentissage profond et réseaux de neurones artificiels

L'apprentissage profond est devenu l'approche principale pour effectuer toutes sortes de tâches d'intelligence artificielle. Cette méthode d'apprentissage automatique a pour spécificité l'utilisation de réseaux de neurones comme fonctions paramétriques à optimiser à l'aide d'un algorithme efficace de descente de gradient connu sous le nom d'*algorithme de rétropropagation*. Ces réseaux de neurones sont nés d'une volonté de s'inspirer du fonctionnement du cerveau humain.

Un neurone biologique est une cellule recevant et transmettant un signal électrique appelé influx nerveux. Elle reçoit de multiples influx nerveux des neurones voisins et les combine en un nouvel influx nerveux qui est passé au neurone suivant. Ainsi, le cerveau est composé d'un large réseau de neurones combinant et communiquant l'information au moyen de signaux électriques.

Les neurones artificiels utilisés en apprentissage automatique sont une modélisation simplifiée de ces neurones biologiques inventée en 1943 par McCulloch et Pitts [MP43]. Formellement, un neurone artificiel est une fonction $\phi_{w_0,w_1,\cdots,w_n}$ à plusieurs entrées $x_1, \cdots, x_n \in \mathbb{R}$ et disposant d'une fonction croissante $\sigma : \mathbb{R} \to \mathbb{R}$ appelée fonction d'activation, paramétrisée par un ensemble de poids $(w_0, w_1, \cdots, w_n) \in \mathbb{R}^{n+1}$. À partir des entrées x_i, le neurone artificiel calcule la valeur :

$$\phi_{w_0,w_1,\cdots,w_n}((x_1, \cdots, x_n)) = \sigma(w_0 + w_1 x_1 + \cdots + w_n x_n)$$

La figure 2.3 illustre la représentation d'inspiration biologique communément utilisée pour représenter un neurone artificiel.

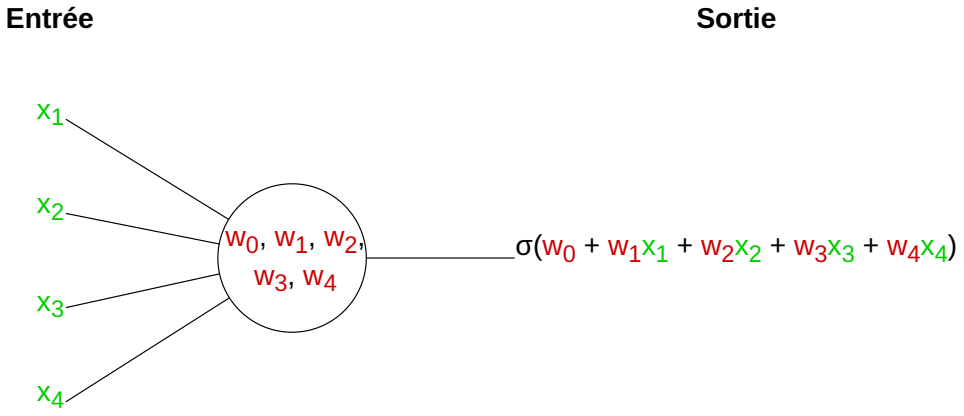

Figure 2.3. : Illustration du fonctionnement d'un neurone artificiel. Les entrées du neurone sont représentées en vert. Les poids du neurone sont représentés en rouge.

L'ensemble des fonctions $\{\phi_{w_0,w_1,\cdots,w_n}, (w_0, w_1, \cdots, w_n) \in \mathbb{R}^{n+1}\}$ correspondant à toutes les paramétrisations possibles d'un neurone sont toutes des fonctions simples ne permettant pas de modéliser des fonctions complexes. Pour cette raison les neurones sont généralement utilisés sous forme de réseaux de la même manière que le réseau de dizaines de milliards de neurones biologiques du cerveau humain.

Un réseau de neurones consiste en un graphe orienté de neurones dont les arêtes symbolisent le transfert de la sortie d'un neurone à l'entrée d'un autre neurone. Ainsi, dans l'exemple d'un réseau constitué de deux neurones à une entrée $\phi^{(1)}_{w_0^{(1)},w_1^{(1)}}$ et $\phi^{(2)}_{w_0^{(2)},w_1^{(2)}}$ dont $\phi^{(2)}$ prend en entrée la sortie de $\phi^{(1)}$, la fonction totale calculée par le réseau sera :

$$\phi^{(2)} \circ \phi^{(1)}(x) = \sigma^{(2)}(w_0^{(2)} + w_1^{(2)} \sigma^{(1)}(w_0^{(1)} + w_1^{(1)} x))$$

Un réseau de neurones utilise en pratique des millions de neurones artificiels. Ces neurones se classent en couches en fonction de leur distance aux données d'entrée du réseau. Ainsi, un neurone recevant directement en entrée les entrées du réseau appartiendra à la première couche, un neurone recevant en entrée les sorties de neurones de la première couche appartiendra à la deuxième couche, etc. Les neurones appartenant à la dernière couche sont appelés neurones de sortie car leurs sorties forment le vecteur de sortie du réseau de neurones. Le nombre de couches d'un réseau de neurones est appelé *profondeur* du réseau. Le nombre maximal de neurones contenu dans une même couche du réseau de neurones est appelé *largeur* du réseau. La figure 2.4 illustre une architecture de profondeur 3 et de largeur 5 permettant de classifier des images de chats et de chiens.

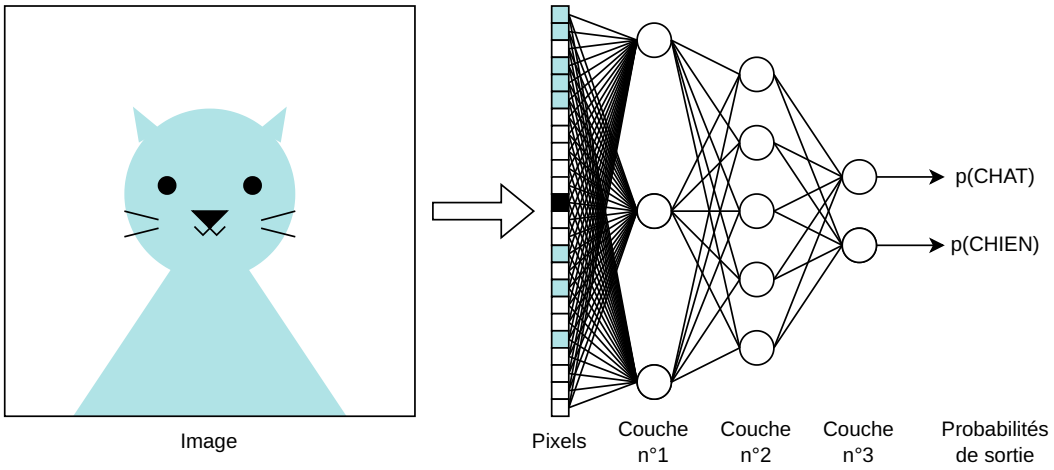

Figure 2.4. : Exemple de réseau de neurones à trois couches pour classifier une image en "CHIEN" ou "CHAT".

La structure du graphe d'un réseau de neurones est appelée *architecture* et représente l'ensemble des fonctions calculables par ce réseau. Il a été démontré qu'un réseau de neurones à deux couches est un approximateur universel à la condition que les fonctions d'activation σ utilisées dans le réseau de neurones ne soient pas linéaires [Cyb89 ; HSW89]. Mathématiquement, ce résultat affirme que pour toute fonction continue $f : \mathbb{R}^n \to \mathbb{R}^m$, tout sous-ensemble compact $K \subset \mathbb{R}^n$ et tout $\epsilon > 0$, il existe un réseau de neurones à deux couches réalisant la fonction $\hat{f}$ tel que :

$$sup_{x \in K} \|\hat{f}(x) - f(x)\| < \epsilon$$

De manière plus intuitive, ce résultat indique qu'il est toujours possible de trouver un réseau à deux couches approximant une fonction continue $f : \mathbb{R}^n \to \mathbb{R}^m$ aussi proche que souhaité sur un sous-ensemble compact de son domaine. Des théorèmes

duaux ont montré qu'un réseau de largeur fixée et de profondeur arbitraire était aussi un approximateur universel [KL20].

Méthode d'optimisation de descente de gradient

Les théorèmes d'approximation universelle affirment que quelle que soit la fonction continue à modéliser il existe un réseau de neurones s'en approchant autant que souhaité. Mais ces mêmes théorèmes n'indiquent ni l'architecture ni l'ensemble des poids à appliquer au réseau pour réaliser cette approximation.

Le choix de la bonne architecture pour répondre à un problème donné est une question scientifique ouverte. La majorité des architectures publiées sont issues d'un processus de recherche manuel. Un domaine scientifique appelé NAS (*Neural Architecture Search*) [ZL17] vise à chercher de manière automatique une architecture à partir des données d'un problème mais les résultats obtenus sont encore préliminaires et les algorithmes proposés particulièrement coûteux.

Une caractéristique centrale des réseaux de neurones est en revanche l'existence de méthodes d'optimisation très efficaces permettant de trouver des poids adaptés à la fonction à modéliser. Soit une fonction $f : Z \subset \mathbb{R}^n \to \mathbb{R}^m$ à modéliser dont ne sont connus qu'un ensemble fini de points $\mathcal{D} = \{(x_i, f(x_i)); i \in \{1, \cdots, s\}\}$ appelé jeu de données. Soit $\hat{f}_0 : \mathbb{R}^n \to \mathbb{R}^m$ un réseau de neurones dont les poids ont été initialisés de manière aléatoire. Étant donné un point x_i du jeu de données, $\hat{f}_0(x_i)$ peut être absolument n'importe quelle valeur dans $\mathbb{R}^m$. Soit $\mathcal{L} : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^+$ une fonction de coût différentiable. Cette fonction de coût $\mathcal{L}$ est utilisée pour représenter la différence entre la sortie du réseau de neurones $\hat{f}(x_i)$ et la sortie attendue $f(x_i)$. Elle vaut 0 quand $\hat{f}(x_i)$ est égal à $f(x_i)$, c'est-à-dire $\forall y \in \mathbb{R}^m : \mathcal{L}(y, y) = 0$.

L'objectif de l'entraînement du réseau de neurones est de reproduire la fonction à modéliser f, tel que $\forall x \in \mathbb{R}^n : \hat{f}(x) = f(x)$. Étant donné que la fonction f n'est connue que sur les points du jeu de données $\{x_i; i \in \{1, \cdots, s\}\}$ cet objectif est remplacé par l'objectif approché $\forall x_i \in \mathcal{D} : \hat{f}(x_i) = f(x_i)$. Pour pouvoir s'approcher de cette fonction, il est nécessaire de pouvoir quantifier l'écart entre le réseau actuel et l'objectif. Cet écart est modélisé à l'aide de la fonction de coût, l'objectif d'optimisation est donc de minimiser $\sum_{x_i \in \mathcal{D}} \mathcal{L}(\hat{f}(x_i), f(x_i))$.

La méthode d'optimisation utilisée pour entraîner un réseau de neurones est appelée *algorithme de la descente de gradient*. Cette méthode minimise une fonction différentiable $g : \mathbb{R}^p \to \mathbb{R}$ à partir d'un point donné $\theta^{(0)} \in \mathbb{R}^p$ par le calcul de son gradient. Le gradient est la généralisation à plusieurs variables de la notion de dérivée d'une

fonction à une seule variable. Il s'agit du vecteur des dérivées partielles de la fonction g par rapport à chacun de ses paramètres $(\theta_1, \cdots, \theta_p)$.

$$\nabla g = \begin{bmatrix} \frac{\partial g}{\partial \theta_1} \\ \vdots \\ \frac{\partial g}{\partial \theta_p} \end{bmatrix}$$

Évalué en un point de l'espace d'entrée $\theta^{(0)} \in \mathbb{R}^p$, $\nabla g(\theta^{(0)})$ est un vecteur représentant la direction de plus forte croissance de la fonction g au point $\theta^{(0)}$. La méthode de la descente de gradient consiste à soustraire le gradient pondéré par un coefficient α appelé *taux d'apprentissage* aux paramètres de la fonction pour faire décroitre g :

$$\theta^{(1)} = \theta^{(0)} - \alpha \nabla g(\theta^{(0)})$$

Il ne reste plus qu'à réitérer cette étape n fois jusqu'à l'obtention d'une valeur de $g(\theta^{(n)})$ satisfaisante.

Algorithme de rétropropagation du gradient pour un réseau de neurones

L'entraînement d'un réseau de neurones à l'aide de l'algorithme de la descente de gradient nécessite le calcul du gradient de l'erreur du réseau de neurone modélisé par sa fonction de coût en fonction de la valeur de ses poids. Pour décrire ce calcul, il est d'abord nécessaire de décrire le fonctionnement en couches d'un réseau de neurones.

Un réseau de neurones est composé de multiples couches qui sont elles-mêmes composées de multiples neurones. Chacun de ces neurones calcule une valeur à partir des valeurs de sortie issues des neurones de la couche précédente. Supposons que l'on dispose d'un réseau de l couches et notons $n^{(i)}$ le nombre de neurones contenus dans la couche $i \in \{1, \cdots, l\}$. Notons par ailleurs $n^{(0)}$ la dimension de l'entrée du réseau de neurones. Finalement, notons $\boldsymbol{x} = (x_1, \cdots, x_{n^{(0)}})$ le vecteur d'entrée du réseau de neurones et $\boldsymbol{z}^{(i)} = (z_1^{(i)}, \cdots, z_{n^{(i)}}^{(i)})$ le vecteur de sortie de la couche i.

Soit $\boldsymbol{w}_j^{(i)} = (w_{j,0}^{(i)}, \cdots, w_{j,n^{(i-1)}}^{(i)})$ l'ensemble des poids du neurone j de la couche i. Ce neurone va calculer :

$$z_j^{(i)} = \sigma(w_{j,0}^{(i)} + w_{j,1}^{(i)} z_1^{(i-1)} + \cdots + w_{j,n^{(i-1)}}^{(i)} z_{n^{(i-1)}}^{(i-1)})$$

Il est possible de représenter l'ensemble des calculs opérés par la couche i à l'aide de la notation matricielle :

$$\boldsymbol{z}^{(i)} = \sigma(W^{(i)}\hat{\boldsymbol{z}}^{(i-1)})$$

en définissant :

$$W^{(i)} = \begin{pmatrix} w_{1,0}^{(i)} & w_{1,1}^{(i)} & \cdots & w_{1,n^{(i-1)}}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n^{(i)},0}^{(i)} & w_{n^{(i)},1}^{(i)} & \cdots & w_{n^{(i)},n^{(i-1)}}^{(i)} \end{pmatrix}$$

et :

$$\hat{\boldsymbol{z}}^{(i-1)} = \begin{pmatrix} 1 \\ z_1^{(i-1)} \\ \vdots \\ z_{n^{(i-1)}}^{(i-1)} \end{pmatrix}$$

Cette écriture permet de représenter chaque couche du réseau de neurones comme une unique opération matricielle utilisant les résultats de la couche précédente. Par la suite, les couches du réseau de neurones seront définies de la manière suivante :

$$\begin{aligned} \hat{f}^{(i)} :& \mathbb{R}^{n^{(i-1)}} \to \mathbb{R}^{n^{(i)}} \\ & \boldsymbol{z}^{(i-1)} \mapsto \sigma(W^{(i)}\hat{\boldsymbol{z}}^{(i-1)}) \end{aligned}$$

Finalement, l'ensemble du réseau de neurone est la composition de toutes ses couches :

$$\begin{aligned} \hat{f} :& \mathbb{R}^{n^{(0)}} \to \mathbb{R}^{n^{(l)}} \\ & \boldsymbol{x} \mapsto \hat{f}^{(l)} \circ \hat{f}^{(l-1)} \circ \cdots \circ \hat{f}^{(1)}(\boldsymbol{x}) \end{aligned}$$

Le calcul du gradient d'une fonction est le calcul de toutes ses dérivées partielles. Du fait de la structure d'un réseau de neurones comme composée de couches de neurones, il est possible d'utiliser le théorème de dérivation des fonctions composées.

Théorème 1 (Dérivation des fonctions composées [Str23]) *Soit $f : I \subset \mathbb{R} \to \mathbb{R}$ et $g : J \subset \mathbb{R} \to \mathbb{R}$ avec $f(I) \subset J$. Soient $x \in \mathbb{R}$, $y = f(x)$ et $z = g(f(x))$.*

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$$

L'ensemble des variables à optimiser dans le cadre d'un réseau de neurones est l'ensemble des poids de chacune de ses couches. Cet ensemble de poids est généralement noté $\theta = (W^{(1)}, \cdots, W^{(l)})$. Dans le cadre de l'entraînement de réseaux de neurones, la fonction différentiable utilisée pour l'optimisation est la fonction de l'erreur moyenne de prédiction sur le jeu de données $g : \theta \mapsto \frac{1}{s}\sum_{i=1}^{s} \mathcal{L}(\hat{f}_\theta(\boldsymbol{x}_i), f(\boldsymbol{x}_i))$. Le calcul du gradient de la fonction g est le calcul des dérivées partielles de chacun de ses poids.

Théorème 2 (Rétropropagation des dérivées partielles [LeC+89]) *Soit $w_{j,k}^{(i)}$ le $k^{ème}$ poids du neurone j de la couche i. Soit $L = \mathcal{L}(\boldsymbol{z}^{(l)}, y)$ l'erreur entre la sortie du réseau de neurones et la valeur attendue. La dérivée partielle du poids $w_{j,k}^{(i)}$ est*

$$\frac{\partial L}{\partial w_{j,k}^{(i)}} = \frac{\partial L}{\partial \boldsymbol{z}^{(l)}} \frac{\partial \boldsymbol{z}^{(l)}}{\partial \boldsymbol{z}^{(l-1)}} \cdots \frac{\partial \boldsymbol{z}^{(i+1)}}{\partial \boldsymbol{z}^{(i)}} \frac{\partial \boldsymbol{z}^{(i)}}{\partial w_{j,k}^{(i)}}.$$

On peut remarquer dans la formule ci-avant que la dérivée partielle d'un poids de la couche i dépend de la valeur des dérivées des couches i à l. Il est ainsi possible de réutiliser les calculs de dérivée de la couche i pour calculer les dérivées partielles de la couche $i-1$. Cette observation permet de concevoir un algorithme de calcul du gradient très efficace appelé *rétropropagation du gradient*. Ainsi la propriété des réseaux de neurones comme fonction composée de nombreuses couches différentiables permet un calcul efficace du gradient et donc une application de l'algorithme d'optimisation de la descente de gradient. Par ailleurs, le calcul du gradient est hautement parallélisable ce qui permet d'utiliser des accélérateurs matériels de calcul parallèle tels que des GPU pour accélérer grandement l'entraînement [KSH17].

Limitations inhérentes à l'algorithme de descente de gradient

L'utilisation de l'algorithme de descente de gradient incarné dans le cadre des réseaux de neurones par l'algorithme de rétropropagation permet un entraînement rapide de réseaux comportant potentiellement des milliards de poids à entraîner. Néanmoins, cet algorithme d'optimisation comporte également des failles importantes qui doivent être prises en compte lors de l'entraînement.

La première problématique à adresser tient à la formulation du problème d'optimisation. Cette dernière définit l'objectif de l'entraînement comme la minimisation de l'erreur entre la prédiction du réseau de neurones et la réponse attendue sur les exemples contenus dans le jeu de données. Mais l'objectif réel de l'entraînement vise

à modéliser la fonction qui associe un exemple à la réponse attendue sur l'ensemble du domaine de la fonction, et pas seulement sur les exemples du jeu de données.

Lorsque la fonction effectuée par le réseau de neurones après l'entraînement produit de bons résultats sur les exemples contenus dans le jeu de données mais pas sur des exemples qui n'en faisaient pas partie, on parle alors d'un phénomène de surapprentissage. Ce phénomène qui touche aussi d'autres méthodes d'apprentissage automatique peut être atténué par l'usage de méthodes de régularisation [GBC16]. Ces méthodes visent à pénaliser la complexité du modèle appris par exemple par l'ajout de la norme des poids du modèle à la fonction d'erreur à minimiser car il a été observé que les valeurs de poids extrêmes sont souvent révélatrices d'un surapprentissage du modèle.

Quelles que soient les méthodes de régularisation employées, il est impossible d'éliminer totalement le surapprentissage. Il est donc devenu standard d'isoler une partie du jeu de données qui ne sera pas utilisé pour l'apprentissage et d'évaluer régulièrement au cours de l'entraînement l'erreur obtenue par le modèle non seulement sur le jeu de données d'entraînement mais aussi sur ce nouveau jeu de données appelé jeu de validation. De cette manière, un cas de surapprentissage est aisément repéré par la déconnexion entre l'erreur sur le jeu d'entraînement qui décroît au fil de l'apprentissage alors que l'erreur sur le jeu de validation augmente.

Une autre problématique liée à l'usage de l'algorithme de descente de gradient est que celui-ci ne garantit que de converger vers un minimum local de la fonction d'erreur en fonction des poids. Dans le cadre d'une fonction d'erreur convexe, tout minimum local est un minimum global, aussi le problème ne se pose pas. Il a cependant été montré que dans la majorité des tâches d'apprentissage profond la fonction d'erreur n'est pas convexe [Kaw16].

Pour pallier ce problème, les algorithmes d'optimisation utilisés en pratique sont des variantes de l'algorithme de descente de gradient. La plus importante d'entre elles est l'algorithme de descente de gradient stochastique [Bot10]. Dans celui-ci, plutôt que d'entraîner le réseau de neurones à chaque itération à minimiser l'erreur sur tout le jeu de données d'entraînement, chaque itération ne va concerner qu'une petite fraction du jeu de données choisie aléatoirement dans le jeu de données d'entraînement complet. Cette méthode permet d'une part de faciliter les calculs par une importante diminution du nombre de prédictions à faire à chaque itération et d'autre part de changer la position des minimums locaux de la fonction d'erreur à chaque itération puisque l'objectif à minimiser change en permanence. En plus du passage à une descente de gradient stochastique, la plupart des méthodes modernes

d'optimisation de réseaux de neurones sont des algorithmes à momentum intégrant de l'inertie au gradient par l'ajout de dérivées d'ordre supérieur [KB17].

Méthodologie de mise au point d'un modèle d'apprentissage profond

Dans la pratique, la résolution d'un problème par une méthode d'apprentissage profond comporte de nombreuses étapes qui seront affinées jusqu'à l'obtention d'un modèle répondant convenablement au problème posé. Ces différentes étapes seront illustrées par la reprise de l'exemple de classification d'images de chiens et de chats évoqué précédemment.

Dans un premier temps, il est nécessaire de correctement définir le problème posé. Dans le cas de la classification d'images d'animaux évoquée plus haut, il s'agit dans un premier temps d'identifier les besoins. Quelles sont les différents types d'animaux à classifier ? Les images utilisées en entrée seront-elles des photographies, des dessins, et seront-elles de définitions différentes ? Quel pourcentage d'erreur de classification est acceptable au vu de l'usage qui sera fait de ces prédictions ? Quelles ressources de calcul et quel temps d'inférence maximal sont envisageables ?

Les réponses à ces différentes questions permettront par la suite de définir le jeu de données à constituer en vue de l'entraînement ainsi que les méthodes utilisées par la suite. En particulier, il est possible que des méthodes d'apprentissage profond à base de réseaux de neurones ne soient pas les mieux adaptées en raison d'un coût calculatoire trop élevé ou de l'existence de méthodes plus simples ayant déjà fait leurs preuves.

Les réseaux de neurones n'acceptent généralement que des entrées et sorties de taille fixe. Par ailleurs, ils ne produisent en sortie que des vecteurs de nombre réels. Il est donc parfois nécessaire de légèrement modifier le problème pour respecter ces contraintes. Ainsi dans notre exemple de classification, il sera nécessaire de redimensionner les images pour qu'elles fassent toutes la même définition. De la même manière, un réseau de neurones ne pouvant pas directement produire de classe en sortie, il sera nécessaire de modifier la problématique par la production de probabilités d'appartenance à une classe, ce qui permettra par la suite de sélectionner la classe prédite qui sera celle ayant la plus grande probabilité.

Il est ensuite nécessaire de choisir une métrique permettant d'évaluer l'erreur de prédiction du modèle. Cette métrique sera utilisée comme objectif d'optimisation à minimiser. Dans les cas où cette métrique ne serait pas différentiable, il est possible de choisir une approximation différentiable appelée fonction de coût de cette métrique

qui sera utilisée comme objectif d'optimisation. La métrique restera dans ce cas utile pour évaluer la performance du modèle après l'entraînement.

Une fois toutes ces questions abordées, la constitution du jeu de données peut être effectuée. Cette étape est cruciale pour le bon fonctionnement de l'apprentissage : c'est souvent l'étape la plus complexe à réaliser. Un bon jeu de données devra être de taille suffisante, contenir des exemples de bonne qualité avec le moins d'erreurs d'étiquetage possible, et il devra être équilibré et représentatif de la diversité des exemples possibles.

Par la suite, l'architecture du premier réseau de neurones à entraîner doit être choisie. Généralement, la première architecture testée sera inspirée d'architectures de l'état de l'art sur une problématique similaire. Le choix de l'architecture est un problème difficile qui repose principalement sur l'expertise et l'expérience. La première architecture choisie pourra être par la suite modifiée pour tenter d'en améliorer les résultats ou d'en diminuer le coût calculatoire.

L'entraînement d'un réseau de neurones dépend de nombreux paramètres. L'algorithme d'optimisation utilisé en est un, et il amène avec lui d'autres paramètres. Par exemple, il va être nécessaire de choisir le nombre d'échantillons utilisés à chaque itération dans le cas d'une descente de gradient stochastique, le critère d'arrêt à utiliser permettant de stopper l'entraînement. Chaque algorithme d'optimisation comporte des paramètres qui lui sont propres comme divers taux d'apprentissage permettant de définir la taille des sauts opérés dans la direction du gradient à chaque itération. La définition de l'objectif d'optimisation peut inclure des méthodes de régularisation en plus de la minimisation de la fonction de coût, ce qui constitue alors un paramètre supplémentaire.

Le jeu de données constitué pour la tâche d'apprentissage est systématiquement divisé en deux parties au minimum. Le jeu de données d'entraînement contiendra les exemples utilisés lors de l'apprentissage. Le jeu de données de test sera lui utilisé après l'apprentissage pour évaluer la performance du modèle à l'aide de la métrique préalablement choisie. Généralement, une troisième partie du jeu de données est utilisée comme jeu de validation. Ce jeu de données sera utilisé dans tous les cas où le critère d'arrêt de l'entraînement n'est pas fixé à l'avance mais dépend des performances réalisées par le modèle à chaque itération. En particulier, ce jeu de données de validation permet d'identifier les situations de surapprentissage quand elles surviennent et de stopper l'apprentissage à ce moment-là.

Une fois l'entraînement terminé, le modèle est entraîné à prédire l'intégralité du jeu de test. Ces prédictions permettent d'analyser la qualité du modèle ainsi entraîné de

manière quantitative à l'aide de la métrique mais aussi de manière qualitative en observant les principaux critères menant à une mauvaise prédiction par exemple.

Les résultats du premier modèle obtenu permettent d'identifier les failles de celui-ci ce qui mène généralement à la modification de certains paramètres du modèle et à un réentraînement. De bons résultats sont le produit d'un nombre important de réitérations de ces étapes.

2.2.2 Réseaux de neurones convolutifs adaptés au traitement d'images

Chaque couche présentée plus haut a pour particularité que chaque neurone produit une unique sortie. Ces couches standards sont appelées couches denses et furent les premières à apparaître dans la littérature [Ros58]. Lors de tâches de traitement d'images telles que des tâches de classification, les images utilisées en entrée étaient d'abord mises à plat en un unique vecteur unidimensionnel puis traitées par le réseau [Fuk75]. Cette méthode de mise à plat a pour effet de supprimer le lien existant entre des pixels adjacents de l'image.

Les images ont certaines propriétés particulières qui peuvent être exploitées par un réseau de neurones. La localité d'une part se définit par le fait que deux pixels proches seront plus en lien que deux pixels éloignés car ils feront probablement partie d'un même objet. Par ailleurs, une image a pour propriété une invariance par translation. Ce terme signifie qu'un certain objet présent par exemple en haut à droite de l'image sera représenté de la même manière par les pixels s'il se trouve en bas à gauche de l'image.

Les couches de convolution [Fuk80 ; LeC+98] sont un type spécialisé de couches de neurones adaptées au traitement des images tirant parti des différentes propriétés de ces dernières. Dans ces couches, les neurones ne prennent pour entrée qu'un sous-ensemble local de pixels de l'image plutôt que l'entièreté des pixels. Cette technique, qui permet de ne faire traiter à un neurone qu'une fenêtre de l'image permet de tirer parti de la localité des données dans une image. Par ailleurs, pour tenir compte de l'invariance par translation des images, les mêmes poids et donc le même neurone sont utilisés sur toutes les fenêtres possibles de l'image. Un tel schéma de fenêtre glissante appliqué à l'image d'entrée est appelé un *filtre de convolution*. Une couche de convolution implique généralement de nombreux filtres permettant d'extraire des caractéristiques différentes présentes au sein de l'image. Les sorties de ces différents filtres sont ensuite empilées comme différents canaux d'une image. Par la suite,

une autre couche de convolution peut être réappliquée sur la sortie de la première couche de convolution. La figure 2.5 illustre un neurone opérant une convolution sur une image.

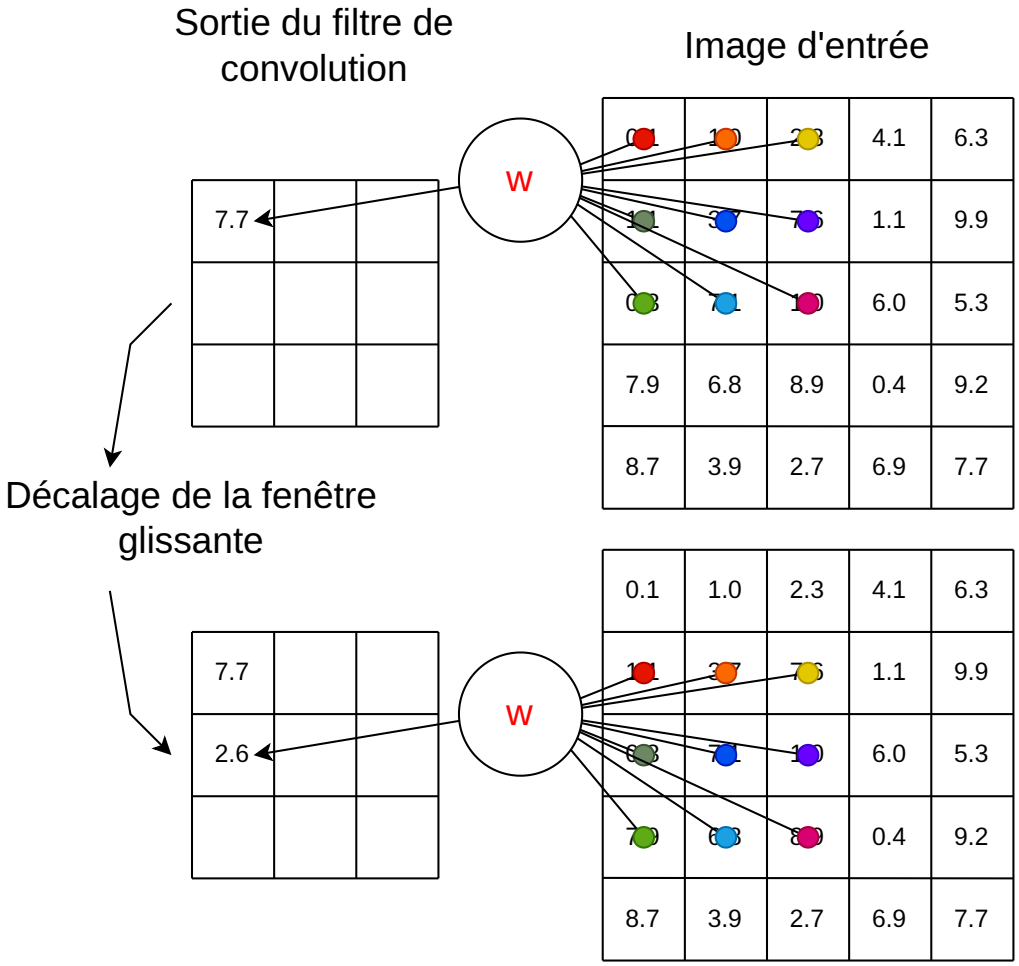

Figure 2.5. : Neurone opérant une convolution 3x3 sur une image 5x5. L'entrée du neurone est à droite et sa sortie est à gauche. La première ligne représente le calcul de la première valeur de sortie. La deuxième ligne représente l'étape suivante dans laquelle la fenêtre glissante a été décalée vers le bas et le neurone reproduit l'opération en conservant les mêmes poids, mais avec des données d'entrée différentes.

Soit $x_{i,j}$ les différents pixels composant l'image d'entrée de taille $m * n$. Soit un neurone convolutif de taille de fenêtre 2×2 comportant les poids $(w_{1,1}, w_{1,2}, w_{2,1}, w_{2,2})$ appliqués à la fenêtre plus un poids de biais w_0. Alors la sortie du neurone sera une matrice de taille $(m-1) * (n-1)$ dont les cellules vaudront :

$$z_{i,j} = \sigma(w_0 + w_{1,1}x_{i,j} + w_{1,2}x_{i,j+1} + w_{2,1}x_{i+1,j} + w_{2,2}x_{i+1,j+1})$$

Dans cet exemple, le nombre de poids à entraîner présents dans la couche de convolution est de 5. Si l'on utilisait une couche dense pour produire une sortie de même taille, le nombre de poids à entraîner serait de $(m-1)*(n-1)*(m*n+1)$. Ainsi, l'utilisation de couches de convolution permet une réduction drastique du nombre de poids à entraîner qui n'est plus dépendante des dimensions de l'image d'entrée au prix d'une expressivité plus limitée comparativement à une couche dense.

Les premiers réseaux exploitant des convolutions utilisaient généralement aussi des couches denses, par exemple en tant que dernière couche pour des tâches de classification [LeC+98]. Plus récemment, un certain nombre de réseaux de neurones permettent de se passer complètement de couches denses [LSD15 ; RF17]. Ces derniers sont appelés réseaux complètement convolutifs (*Fully Convolutional Network* ou FCN).

Un autre type de couche utilisé pour le traitement d'images sont les couches de sous-échantillonnage par valeur maximale. Ces couches qui ne contiennent pas de poids ont pour fonction de diminuer la dimension d'une représentation intermédiaire du réseau de neurones. Leur fonctionnement consiste en l'utilisation d'une fenêtre glissante de taille déterminée dont la sortie correspondra à la valeur maximale observée dans la fenêtre glissante. La figure 2.6 illustre le principe de fonctionnement d'une couche de sous-échantillonnage par valeur maximale.

Ces différentes couches de réseaux de neurones adaptées au traitement d'images sont à la base des architectures de détection d'objets sur images. Par analogie, une grille d'occupation peut être vue comme une image en niveau de gris dont la luminosité est représentée par la probabilité d'occupation de chaque cellule. Il est donc possible d'adapter ces architectures de détection d'objets sur images pour en faire des architectures de détection d'objets sur grilles d'occupation. La sous-section suivante donne des exemples de tels travaux de détection d'objets dans un contexte de navigation autonome.

2.2.3 Application des réseaux convolutifs dans un contexte de navigation autonome

Cette thèse a pour objectif la détection des véhicules présents sur une grille d'occupation. Les résultats présentés par la suite sont inspirés de manière plus générale de divers travaux de détection d'objets dans le contexte du véhicule autonome ainsi que d'autres tâches sémantiques concernant plus spécifiquement les grilles

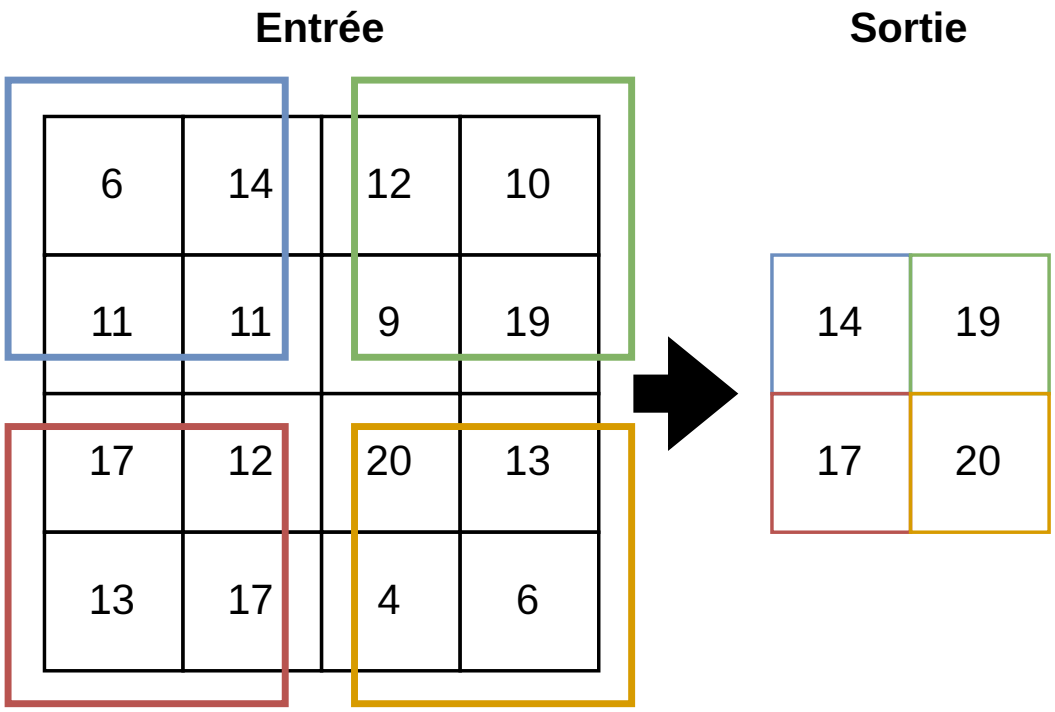

Figure 2.6. : Entrée et sortie d'une couche de sous-échantillonnage par valeur maximale avec une taille de fenêtre glissante de 2x2.

d'occupation. Cette section détaille les travaux déjà existants dans ces différents domaines techniques.

La problématique de détection d'objets est apparue en premier sur les images. Il s'agissait de pouvoir tracer un rectangle éventuellement annoté d'une classe autour des différents objets présents sur la photographie. Ces rectangles sont appelés *boîtes englobantes*. Pour ce faire, deux types de méthodes existent : les détecteurs mono-étape et les détecteurs bi-étapes.

Les *détecteurs bi-étapes* sont apparus en premier avec R-CNN [Gir+14 ; Gir+16]. Ils se composent de deux réseaux de neurones entraînés séparément. Un premier réseau, appelé le réseau de proposition de régions, identifie des régions de l'image susceptibles de contenir un objet. Le deuxième réseau va ensuite être appliqué sur chacune de ces régions pour affiner la boîte englobante et déterminer le type d'objet présent à l'intérieur. Bien que produisant des résultats de bonne qualité, le fait de devoir appliquer le deuxième réseau sur un nombre important de régions de l'image induit un coût calculatoire important. La qualité des résultats obtenus fait qu'il en existe néanmoins d'importants successeurs [He+15 ; Gir15 ; Ren+17 ; He+17].

Les *détecteurs mono-étape* sont quant à eux apparus avec YOLO [Red+16]. Ils ont pour objectif de résoudre les problématiques de performance calculatoire des détecteurs bi-étapes. Leur fonctionnement est plus simple en ce qu'ils ne sont composés que d'un seul réseau de neurones qui produit directement un certain nombre de boîtes englobantes à partir d'une image. Bien que produisant des boîtes englobantes légèrement moins bonnes que les détecteurs bi-étapes, ces détecteurs sont devenus très populaires pour leur capacité à pouvoir fonctionner en temps-réel sur du matériel contraint. Il existe de nombreux détecteurs mono-étape : [Liu+16 ; RF17 ; RF18 ; BWL20 ; Li+22 ; WBL22].

La figure 2.7 illustre la différence entre les détecteurs bi-étapes et les détecteurs mono-étape.

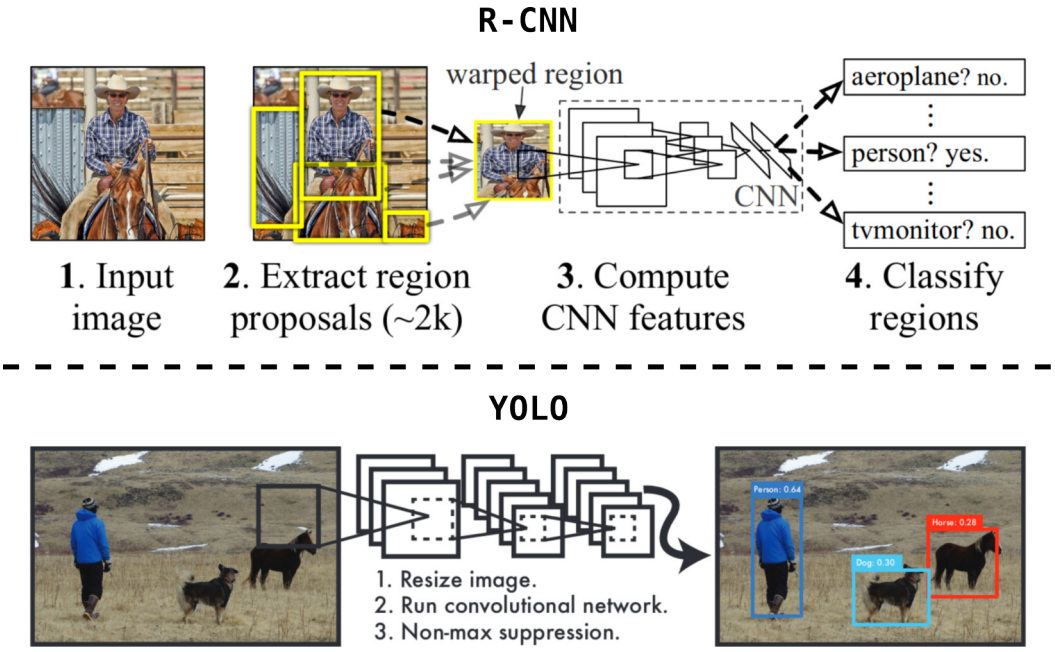

Figure 2.7. : Comparaison de détecteurs bi-étapes illustrés par R-CNN en haut avec les détecteurs mono-étape illustrés par YOLO en bas. On y observe dans le premier cas que la détection d'objets se fait en deux étapes, une première pour sélectionner des régions d'intérêt et une deuxième pour affiner la boîte englobante et la classe de chaque région. Dans le deuxième cas, ces deux opérations sont effectuées par le même réseau de neurones. Illustrations issues de R-CNN [Gir+14] et YOLO [Red+16].

Dans le cadre de la détection d'objets pour les tâches de navigation autonome, les travaux se sont fortement inspirés des détecteurs d'objets sur images existants. Les différents travaux publiés dans ce domaine varient généralement par le type de capteur utilisé et par la manière d'adapter les détecteurs d'objets sur images prévus pour des données à deux dimensions à l'espace en trois dimensions.

La plupart des méthodes de détection d'objets pour la navigation autonome s'appuient sur des capteurs LiDAR produisant des nuages de points correspondant aux obstacles dans l'espace à trois dimensions. La première approche utilisée a été de transformer l'espace en une grille 3D dont les cellules, appelées *voxels*, contiennent des informations sur les points contenus à l'intérieur. Cette transformation permet d'utiliser des couches de convolution 3D de la même manière que les couches de convolution 2D utilisées par les méthodes de détection sur images. Les premières méthodes apparues proposent d'exploiter la "parcimonie" des grilles 3D par une méthode n'évaluant les convolutions que sur les voxels contenant au moins un point [ZP15 ; Eng+17 ; ZT18]. D'autres méthodes n'exploitant pas cette technique utilisent des convolutions 3D classiques [Li17]. D'autres méthodes basées sur les voxels fonctionnent à partir d'images stéréoscopiques [Che+18], détectent dans un premier temps différentes parties des objets avant de les regrouper en boîtes englobantes [Shi+20], ou encore utilisent un nouveau type d'architecture appelé les *transformers* pour permettre au réseau de neurones de faire le lien entre des voxels éloignés [Mao+21b].

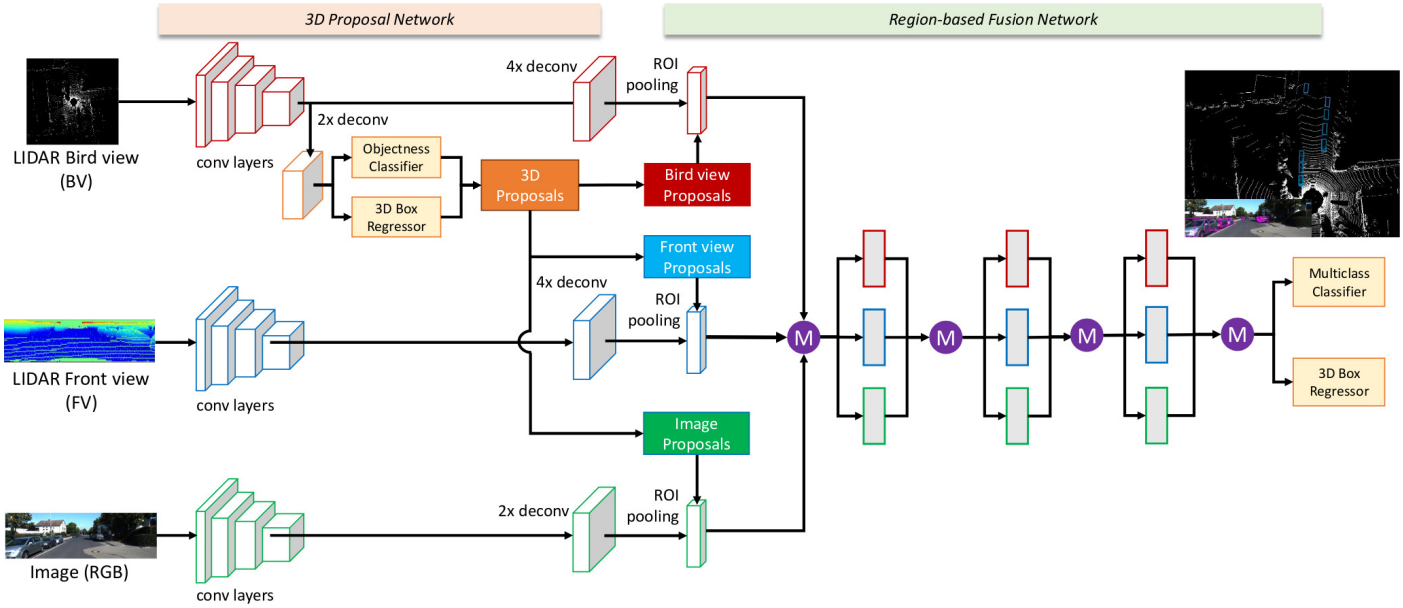

Figure 2.8. : Illustration d'une architecture permettant d'exploiter la présence d'un LiDAR et d'une caméra en conjonction pour produire des boîtes englobantes 3D des objets présents dans la scène. Cette architecture illustre l'approche discutée à la sous-section 1.1.2 d'utiliser directement les données brutes issues des capteurs pour une tâche de perception sans passer par un modèle d'environnement intermédiaire. Illustration issue de [Che+17].

D'autres méthodes de détection permettent de se passer de transformation du nuage de points en travaillant directement sur ces différents points. Ces méthodes segmentent les points selon qu'ils appartiennent à l'avant-plan ou à l'arrière-plan.

Elles produisent ensuite des boîtes englobantes selon les points situés à l'avant-plan [SWL19 ; Pan+20 ; Zha+22 ; Shi+23].

Certaines méthodes de détection d'objets proposent d'exploiter la présence de caméras sur les véhicules autonomes en conjonction avec les LiDAR afin d'améliorer les résultats via la fusion des données de ces deux types de capteurs [Che+17 ; Ku+18 ; Qi+18 ; Qia+20 ; Pie+21 ; WZZ23]. La figure 2.8 illustre l'architecture de [Che+17].

La problématique de cette thèse étant la détection d'objets sur des grilles d'occupation, qui sont une représentation de l'espace en 2D vue de dessus, les méthodes de l'état de l'art d'intérêt ici sont celles transformant les nuages de points en représentations 2D. Par exemple, une méthode propose de transformer un nuage de points en coordonnées cylindriques pour ensuite y appliquer un détecteur mono-étape produisant des boîtes englobantes qui sont ensuite passées dans l'espace 3D via la transformation inverse [LZX16]. D'autres méthodes proposent de projeter le nuage de points sur le plan du sol, donnant ainsi une vue de dessus de l'environnement similaire à celle fournie par une grille d'occupation [YML18 ; Wan+21a]. Parmi ces méthodes, certaines utilisent des détecteurs mono-étapes telles que PIXOR [YLU18] qui utilise sa propre architecture, PointPillars [Lan+19] et SE-SSD [Zhe+21] qui utilisent SSD [Liu+16] ou encore YOLO3D [Ali+19] et Complex-YOLO [Sim+19] qui utilisent YOLOv2 [RF17]. D'autres méthodes utilisent des détecteurs bi-étapes comme BirdNet [Bel+18], BirdNet+ [Bar+21], LiDAR R-CNN [LWW21]. Certaines publications proposent quant à elles de fournir une quantité d'incertitude sur les boîtes englobantes produites [FRD18 ; Fen+19] voire de produire une distribution de probabilité sur les boîtes englobantes [Wan+20 ; Fen+22]. Des méthodes permettent d'utiliser une projection en vue de dessus pour faire du suivi des objets détectés dans le temps [LYU18].

Un certain nombre de travaux utilisent directement comme données d'entrée des grilles d'occupation. Les premiers travaux apparus traitent de détection de véhicules garés sur un parking par l'usage de classificateurs à base de forêts d'arbres décisionnels [Dub+14] ou par l'usage de méthodes d'analyse de composants connectés couplées à de l'apprentissage profond pour la classification [Lom+15 ; Lom+16 ; Lom+17a ; Lom+17b]. Des travaux de suivi d'objets ont été implémentés [STW17 ; Ste+20].

Des travaux de prédictions de grilles futures à partir des grilles passées existent dans la littérature [HBD18 ; IDK19 ; SHD19 ; Wir+19a ; Lee+20 ; LIK21 ; Asg+22 ; Man+22]. D'autres travaux s'intéressent aussi à l'aspect temporel via la segmenta-

tion des cellules d'une grille d'occupation en vecteurs de mouvements [Sch+20 ; Sch+21].

Finalement, certains travaux se sont intéressés à la détection d'objets sur grilles d'occupation dynamiques [Hoe+18] ou non [Wir+18 ; Wir+19b]. La figure 2.9 illustre le fonctionnement de [Wir+18] pour détecter les objets présents dans un nuage de points via un passage à une grille d'occupation spécifique au capteur LiDAR utilisé.

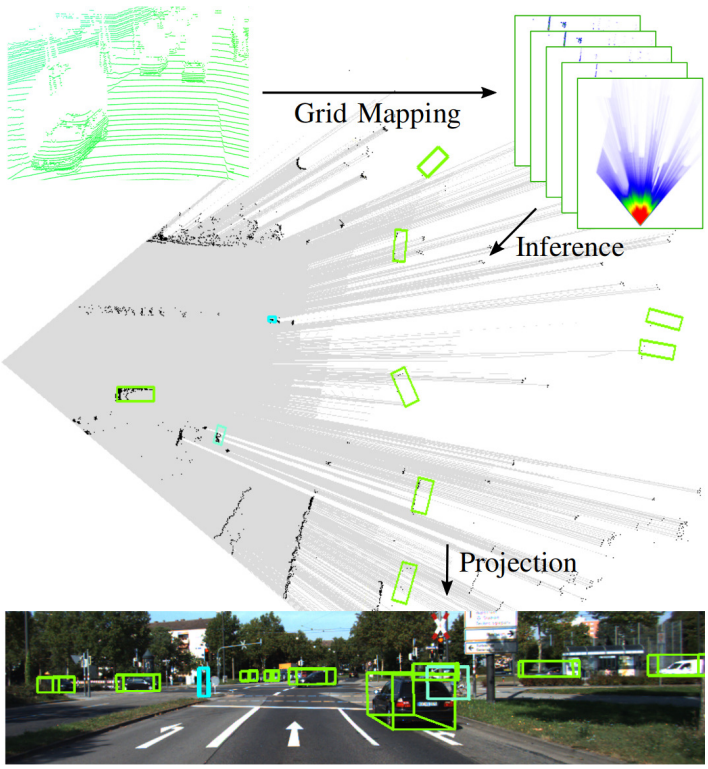

Figure 2.9. : Fonctionnement du détecteur d'objets sur nuage de points de [Wir+18]. Ici, le nuage de points est transformé en une grille d'occupation contenant des données spécifiques au LiDAR telles que la réflectivité des obstacles. Ensuite, un détecteur bi-étapes produit des boîtes englobantes 2D sur la grille d'occupation. Finalement, les boîtes englobantes 2D sont converties en boîtes englobantes 3D grâce aux informations de hauteur contenues dans le nuage de points LiDAR.

Les différents travaux présentés dans cette section, compte tenu de leur objectif de détection d'objets dans un contexte de navigation autonome, serviront de base aux travaux de détection de véhicules sur grilles d'occupation présentés au chapitre 4. En particulier, les travaux de détection d'objets avec une projection en vue de dessus des données de capteur tels que PIXOR [YLU18] qui propose sa propre architecture de réseau de neurones et les différents travaux utilisant des architectures de réseaux

de neurones telles que YOLOv2 [RF17] et YOLOv3 [RF18] seront au cœur de la deuxième contribution de ce manuscrit (chapitre 4).

2.3 Transformées des images dans le domaine fréquentiel et applications pour la reconnaissance de véhicules sur grilles d'occupation

Les méthodes de traitement d'images à partir de réseaux de neurones convolutifs apparues durant la dernière décennie [KSH17] sont aujourd'hui incontournables. En 2012, pour la première fois, un réseau de neurones convolutif appelé AlexNet [KSH17] a remporté le concours de reconnaissance d'images ImageNet avec une confortable avance de plus de 10% sur la métrique d'évaluation. Cette percée des réseaux de neurones convolutifs a mené à un important développement de ce type de techniques, parfois au détriment des méthodes de traitement des images antérieures qui fonctionnaient à l'aide de transformations et filtres heuristiques sans apprentissage [Sze22].

Pourtant, la communauté scientifique du traitement des images a développé de nombreux outils mathématiques d'analyse très utiles et complémentaires aux réseaux de neurones convolutifs. En particulier, les différentes transformées des images dans le domaine fréquentiel, dont la plus connue est la transformée de Fourier [Bra00], permettent une représentation duale des images très utile pour de nombreuses tâches et pouvant être calculées à très faible coût.

Cette section présente ces techniques de transformées en fréquence des images. Une première partie présente les principes mathématiques de ces transformations. Une seconde partie présente les différentes applications pouvant en être faites en rapport avec la tâche de reconnaissance de véhicules sur grilles d'occupation. L'utilisation pouvant être faite de ces méthodes pour aider à la reconnaissance de véhicules sur grilles d'occupation est une contribution de ce travail de thèse présentée au chapitre 5.

2.3.1 Principes de fonctionnement des transformées vers le domaine fréquentiel

La notion de transformation d'un signal de son domaine d'origine vers le domaine fréquentiel est apparue avec les séries de Fourier en 1822 [Bra00] : elles permettent de décomposer tout signal périodique en une série (somme infinie) de sinusoïdes d'amplitudes variables et de fréquences multiples d'une fréquence fondamentale. Le domaine d'origine est par convention appelé domaine temporel dans le cas où le signal $f(t)$ ne dépend que d'une variable. Il est appelé domaine spatial dans le cas où le signal $f(x, y)$ dépend de deux variables. La fonction associant pour chaque fréquence son amplitude est appelée spectre de la fonction originale.

Les séries de Fourier peuvent être généralisées aux fonctions non périodiques via la *transformée de Fourier* (*FT*).

$$\mathcal{F}(f) : \xi \mapsto \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} f(x) \mathrm{e}^{-\mathrm{i}\xi x} \mathrm{d}x$$

Cette fonction $\mathcal{F}(f)$ est exactement le spectre de la fonction f qui associe à chaque fréquence ξ son amplitude $\mathcal{F}(f)(\xi)$.

À partir du spectre $\mathcal{F}(f)$ d'une fonction f, il est possible de reconstruire exactement la fonction f :

$$f : x \mapsto \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \mathcal{F}(f)(\xi) \mathrm{e}^{+\mathrm{i}\xi x} \mathrm{d}\xi$$

Ainsi, une fonction f et son spectre $\mathcal{F}(f)$ sont deux représentations duales, la première dans le domaine spatial (dans le cas où le signal f est une image) et la deuxième dans le domaine fréquentiel.

La *transformée de Fourier discrète* (*DFT*) est une variante de la transformée de Fourier adaptée au traitement de signaux discrets tels que ceux utilisés en traitement du signal numérique. Dans ce cas de figure, l'intégrale de la transformée de Fourier est remplacée par une somme infinie. Par ailleurs, un algorithme appelé *transformée de Fourier rapide* (*FFT*) permet de calculer très efficacement le spectre d'un signal discret avec une complexité en $O(n \log n)$, n étant le nombre de points du signal [CT65].

Dans les applications de traitement des images, la transformée de Fourier rapide est très utilisée. En effet, le spectre d'une image peut être interprété : les basses fréquences correspondent à des changements graduels dans l'image alors que les hautes fréquences correspondent à des changements abrupts. Ainsi, filtrer les hautes fréquences puis reconstruire une image à partir de ce nouveau spectre permet d'adoucir les contours des objets dans une image alors que filtrer les basses fréquences

permet de ne conserver que les contours des objets dans l'image reconstruite. Par ailleurs, les très hautes fréquences correspondent généralement à du bruit et leur suppression est peu perceptible dans l'image reconstruite. Pour cette raison, de nombreuses techniques de compression d'images stockent l'image sous la forme d'un spectre dont les hautes fréquences ont été supprimées [ISO94].

La *FFT* a pour principal inconvénient que les coefficients de fréquences obtenus sont des nombres complexes ce qui peut poser quelques difficultés calculatoires. Pour pallier ce problème, la *transformée en cosinus discrète* (*DCT*) produit les mêmes types de spectre que la *FFT* mais avec des coefficients réels. Par ailleurs, un algorithme de calcul optimisé de la transformée en cosinus discrète, appelé *transformée en cosinus rapide* (*FCT*), permet un coût calculatoire de la transformée en cosinus discrète similaire à celui de la transformée de Fourier rapide [Mak80]. Pour cette raison, la *FCT* est la transformée dans le domaine fréquentiel majoritairement utilisée en traitement du signal numérique.

La transformée en cosinus discrète d'un signal discret $(f_n), n \in \{1, \cdots, N\}$ est calculée par la formule suivante :

$$\mathcal{F}((f_n)) : k \mapsto \sum_{n=1}^{N} f_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}\right)k\right]$$

La transformée en cosinus discrète inverse permettant d'obtenir le signal (f_n) à partir de son spectre $(\mathcal{F}((f_n))_k)$ est obtenue par la formule suivante :

$$\mathcal{F}^{-1}((\mathcal{F}((f_n))_k)) : n \mapsto \frac{1}{2}\mathcal{F}((f_n))_1 + \sum_{k=2}^{N} \mathcal{F}((f_n))_k \cos\left[\frac{\pi}{N} n\left(k + \frac{1}{2}\right)\right]$$

Dans le cas d'une image, le signal bidimensionnel $f_{i,j}$ produira par application de la *FCT* un spectre bidimensionnel de mêmes dimensions $\mathcal{F}(f)_{k,l}$. Chaque fréquence bidimensionnelle correspondra à une composante de l'image de même dimension que l'image originale. Finalement, l'image originale sera égale à la somme de chacune de ces composantes pondérée par la valeur du spectre de l'image pour cette fréquence. La figure 2.10 illustre la reconstruction d'une image à partir d'un spectre de quatre cellules de côté.

La transformée des images dans le domaine fréquentiel à l'aide de la transformée en cosinus rapide permet d'obtenir une représentation duale de l'image possédant des propriétés intéressantes. En effet, les fréquences du spectre sont ordonnées des basses fréquences en haut à gauche du spectre aux hautes fréquences en bas à droite. Cette propriété est exploitée entre autres par le format de compression JPEG [ISO94]

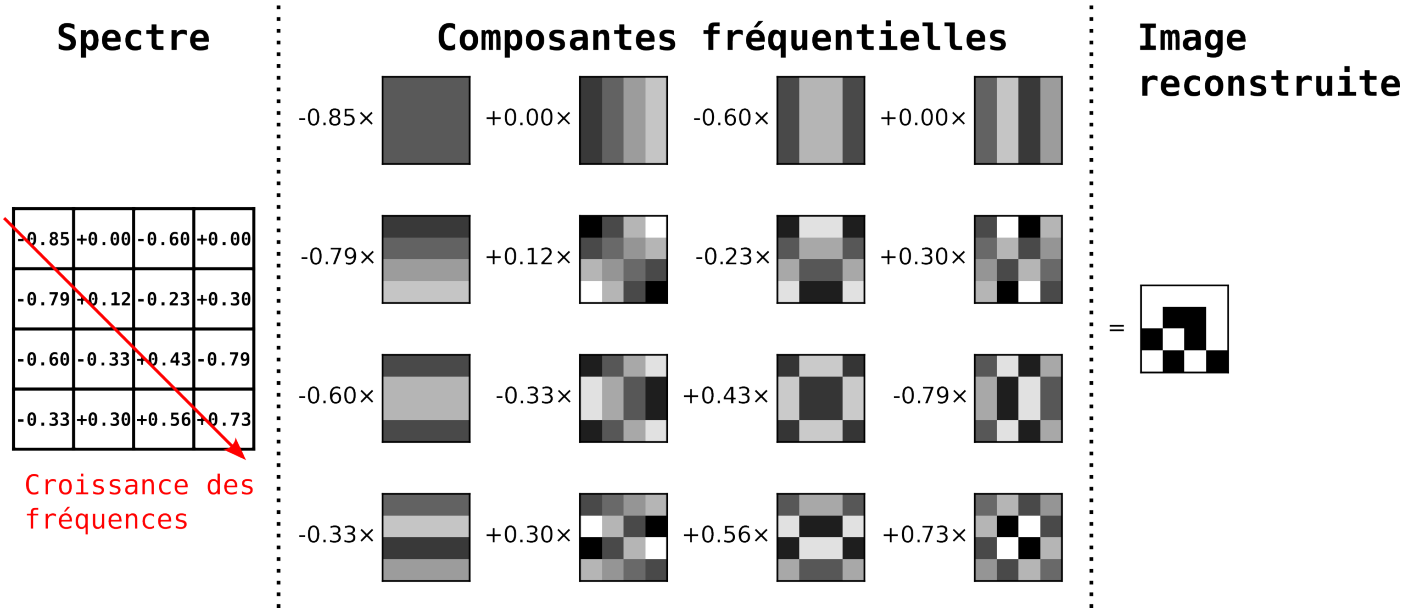

Figure 2.10. : Gauche : Spectre d'une image ; Milieu : Base de composantes fréquentielles ; Droite : Reconstruction de l'image originale par somme du produit de chaque valeur du spectre par la composante fréquentielle associée.

qui réduit la taille des images en les stockant sous forme de spectre dont les hautes fréquences ont subi une quantification des amplitudes. L'effet des hautes fréquences étant généralement faible sur l'image reconstruite, cette procédure permet de réduire considérablement la taille des images en en dégradant très peu la qualité. Cette propriété sera utilisée au chapitre 5 pour compresser des grilles d'occupation par la suppression de leurs hautes fréquences. L'effet concret en termes de dégradation de cette suppression sera étudié.

Un autre type de transformation très utilisée en compression d'images est la transformation en ondelettes. Cette transformation qui est au cœur du format de compression JPEG 2000 [ISO19] utilise une ondelette mère, qui est un signal de moyenne nulle et d'intensité nulle partout sauf sur un intervalle donné. À partir de cette ondelette mère, une base d'ondelettes est dérivée par translation et dilatation de l'ondelette mère. L'analyse d'un signal consiste en le calcul de l'intégrale du produit d'une ondelette avec le signal pour chacune des ondelettes de la base.

La transformation en ondelettes est très utilisée pour ses propriétés de conservation de la localité des motifs de l'image dans la résultante de la transformation. Le procédé de translation de l'ondelette mère peut être assimilé à une convolution de l'ondelette mère avec le signal. Le procédé de dilatation de l'ondelette mère est équivalent à l'usage de différents filtres de convolution. Pour ces raisons, la transformation en ondelettes n'est pas utilisée dans ce manuscrit car considérée

trop proche conceptuellement des réseaux de neurones convolutifs présentés au chapitre 4.

2.3.2 Applications des transformées vers le domaine fréquentiel à la perception pour la navigation autonome

L'objectif des travaux présentés dans le chapitre 5 est d'étudier le potentiel d'usage de la transformée en cosinus discrète pour la projection des grilles d'occupation dans le domaine fréquentiel afin de faciliter la reconnaissance des véhicules sur ces dernières. À cette fin, cette sous-section présente un état de l'art préalable de l'usage de méthodes fréquentielles couplées à de l'apprentissage profond pour des tâches de perception.

Cette étude bibliographique n'a pas permis d'identifier de précédents travaux utilisant des méthodes fréquentielles pour des tâches de détection d'objets. En revanche, une légère extension du domaine d'application visé à des tâches de classification d'images permet d'extraire quelques travaux similaires.

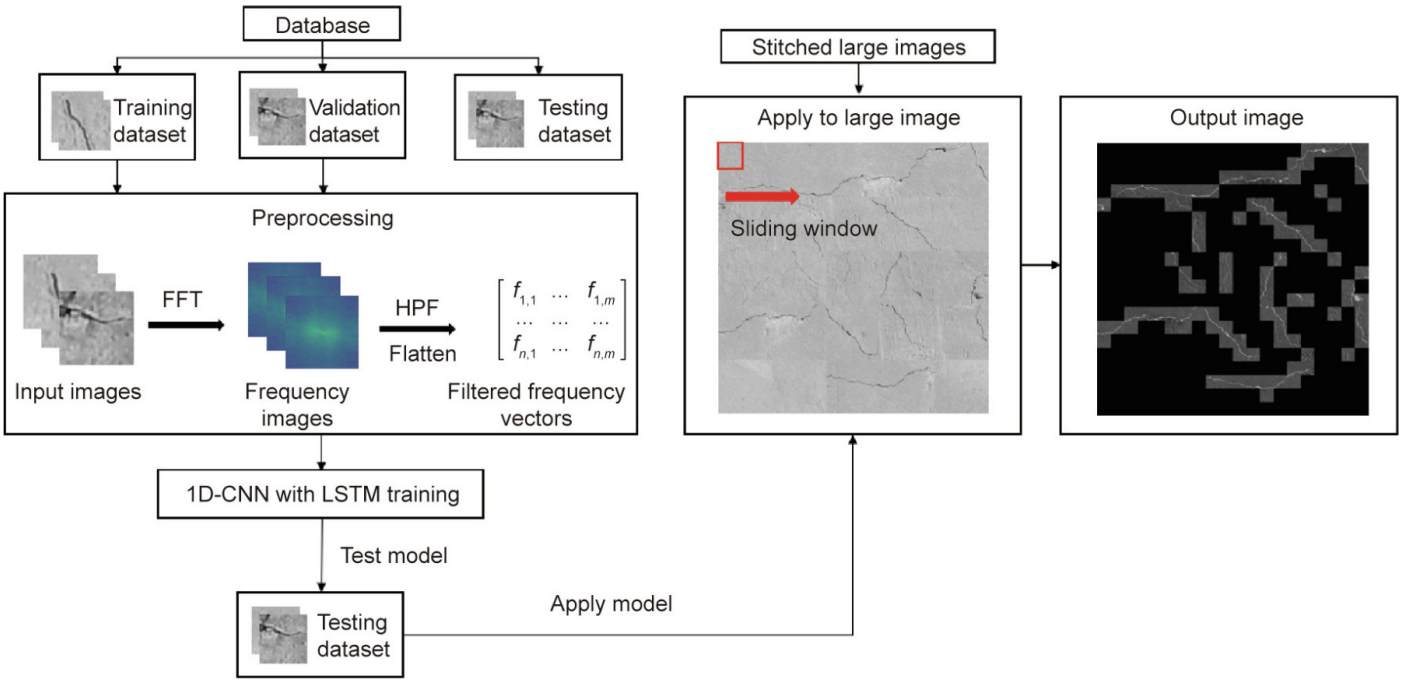

Figure 2.11. : Illustration de la méthode proposée par [Zha+21] pour la détection de fissures dans des ponts observés par imagerie satellite. La méthode utilise la transformée de Fourier discrète pour obtenir un spectre de l'image originale qui est ensuite traité par un réseau de neurones convolutif.

Ainsi, [Zha+21 ; KS22] proposent d'utiliser le spectre obtenu par application de la *FFT* à une image satellite de pont pour en détecter les fissures à l'aide d'un réseau de neurones convolutif sur ledit spectre. D'autres usages des méthodes de transformée dans le domaine fréquentiel tels que des méthodes de détection d'images retouchées numériquement [Li+23] ou de génération d'images utilisant entre autres des spectres en entrée de réseau de neurones [Cai+21] sont à noter. L'architecture de [Zha+21] est présentée sur la figure 2.11.

Cette étude bibliographique a mis en évidence le très faible nombre de travaux existants couplant l'usage de spectres d'images avec des méthodes d'apprentissage profond.

Pour cette raison, les travaux présentés au chapitre 5 utilisent des architectures conçues durant les travaux du doctorat et les résultats obtenus avec ces dernières sont considérés comme novateurs.

2.4 Conclusion

Ce chapitre a présenté les principales connaissances nécessaires à la compréhension des contributions de ce manuscrit.

La section 2.1 a présenté :

— La méthode de construction d'une grille d'occupation à partir des mesures d'un capteur et d'un modèle de capteur associé.

— La méthode permettant de fusionner différentes grilles d'occupation en une unique grille d'occupation multicapteurs.

Ces éléments permettent de comprendre les caractéristiques de ce modèle d'environnement qui est utilisé dans ce manuscrit pour la reconnaissance de véhicules présents dans l'environnement.

La section 2.2 a présenté :

— Le principe de fonctionnement d'un réseau de neurones profond, la formalisation d'un neurone artificiel, la méthode utilisée pour entraîner un réseau à l'aide de données, et finalement des considérations pratiques concernant l'élaboration d'un modèle d'apprentissage profond.

— Les méthodes d'apprentissage profond spécifiques au traitement d'images : les couches de convolution et les couches de sous-échantillonnage par valeur maximale.

— Les principaux travaux de détection d'objets dans un contexte de navigation autonome. *Ces travaux utilisant les couches de convolution seront le point de départ des méthodes de détection de véhicules sur grilles d'occupation présentées au chapitre 4.*

La section 2.3 a présenté :

— Les différentes transformées permettant de passer un signal du domaine spatial au domaine fréquentiel. Y ont été abordées la transformée de Fourier, la transformée en cosinus, et finalement la transformée en ondelettes. Grâce à ses propriétés de production de spectre contenant des nombres réels, la transformée en cosinus sera utilisée dans la contribution présentée au chapitre 5.

— Les principaux travaux conjuguant des spectres d'images et des méthodes d'apprentissage profond pour le traitement d'images. Le faible nombre de travaux dans ce domaine montre un champ de recherche peu exploré à l'heure actuelle. *Nous postulons que la contribution présentée au chapitre 5 permet d'apporter des connaissances nouvelles dans ce champ scientifique d'apprentissage profond dans le domaine fréquentiel.*

Les chapitres suivants présentent les trois contributions majeures de ce manuscrit :

1. La constitution d'un jeu de données de grilles d'occupation annoté
2. L'entraînement de réseaux de neurones convolutifs de reconnaissance de véhicules sur grilles d'occupation
3. L'entraînement d'un réseau de neurones capable de segmenter les véhicules présents sur une grille d'occupation à partir d'une représentation fréquentielle de faible dimension de la grille d'occupation originale

3 Constitution d'un jeu de données de grilles d'occupation annotées

Ce chapitre présente la première contribution de ce travail de thèse qui consiste en la production d'un jeu de données adapté aux tâches d'apprentissage profond ultérieures. Une première partie présente les jeux de données relatifs au véhicule autonome dont fait partie le jeu de données retenu ainsi que les raisons de ce choix. Une deuxième partie présente le processus de transformation de ce jeu de données vers des grilles d'occupation qui sont le point de départ des développements de ce travail de thèse. Une troisième et dernière partie présente une analyse quantitative et qualitative du jeu de données obtenu à la suite du processus de transformation.

3.1 Principaux jeux de données automobiles existants

La tâche de reconnaissance de véhicules sur grilles d'occupation étudiée dans ce travail est menée à bien par des méthodes d'apprentissage automatique. L'usage de ces méthodes repose sur l'entraînement de modèles à l'aide d'un grand nombre d'exemples de données du problème. Un tel ensemble d'exemples est appelé *jeu de données* et son choix est un élément crucial de la performance des modèles entraînés.

Dans le cas précis de la reconnaissance de véhicules sur grilles d'occupation, il est nécessaire d'avoir des exemples composés de grilles d'occupations ainsi que de boîtes englobantes correspondant aux véhicules présents dans ces grilles. Le laboratoire d'accueil de cette thèse dispose d'un outil permettant de générer des grilles d'occupation probabilistes à partir de nuages de points issus d'un ou plusieurs LiDARs. Les boîtes englobantes correspondant aux véhicules sont quant à elles présentes dans un grand nombre de jeux de données publics. La majorité de ces boîtes englobantes sont 3D, mais elles peuvent être projetées en boîtes englobantes 2D en vue de dessus simplement en éliminant leur composante verticale.

Le choix du jeu de données qui sera utilisé dans la suite de ce manuscrit a été réalisé à partir d'un certain nombre de critères. D'une part la taille du jeu de données qui se mesure en nombre de scènes ainsi que la diversité desdites scènes permet l'entraînement de modèles plus robustes. Par diversité, on entend la présence de scènes à différents moments de la journée, sous différentes conditions météorologiques, dans différents types d'environnement (centre-ville, banlieue résidentielle, autoroute, campagne) éventuellement dans différentes parties du monde. Une grande diversité ainsi qu'un nombre élevé de scènes permet un entraînement plus à même de fonctionner dans diverses situations. Elle empêche partiellement une trop grande spécialisation du modèle à résoudre uniquement le type de scènes présentes dans le jeu de données.

En plus de ces critères principaux, il est important que le jeu de données choisi contienne plusieurs capteurs LiDAR montés sur le véhicule qui a pris les mesures, car cela permet la construction de grilles d'occupation plus précises suite à la fusion des grilles d'occupation individuelles de chacun de ces capteurs.

Finalement, dans une optique de perspectives futures de recherche, il est intéressant que le jeu de données choisi fournisse plus d'informations sur la scène telles que la présence de boîtes englobantes correspondant aux piétons ou aux cyclistes ainsi qu'un historique des scènes capturées par les capteurs ce qui permettrait d'exploiter l'axe temporel des données récoltées dans les futurs modèles développés.

Pour résumer, les différents critères d'évaluation du jeu de données choisi sont les suivants classés par ordre d'importance :

1. Nombre de scènes présentes dans le jeu de données
2. Diversité des scènes proposées
3. Multiplicité des capteurs LiDAR embarqués
4. Boîtes englobantes correspondant aux objets "piéton" et "cycliste"
5. Présence de séquences temporelles constituées de scènes successives

Le tableau 3.1 présente les principaux jeux de données automobiles contenant à la fois des nuages de points issus d'un LiDAR et des boîtes englobantes correspondant aux véhicules. Les jeux de données existants à l'heure de la rédaction de ce manuscrit sont indiqués. Néanmoins, le choix du jeu de données utilisé a eu lieu à la mi-2020 et une ligne rouge sur le tableau délimite les jeux de données disponibles à ce moment-là de ceux qui ne l'ont été que plus tard et n'ont donc pas fait partie des jeux de données candidats.

Jeu de données	Année	Nombre de scènes	LiDARs	Classes			Conditions		Localisation	Séquences
				Voiture	Piéton	Cycliste	Pluie	Nuit		
KITTI [GLU12]	2012	14 999	1 x Toit	✓	✓	✓	?	✗	DE	✗
H3D [Pat+19]	2019	27 721	1 x Toit	✓	✓	✗	?	?	US	?
Argoverse [Cha+19]	2019	≈22 600 (113 séquences)	2 x Côtés	✓	✓	✓	✓	✓	US	✓
A2D2 [Gey+20]	2020	12 490	5 x Toit	✓	✓	✓	✓	?	?	?
A*3D [Pha+20]	2020	39 179	1 x Toit	✓	✓	✓	✓	✓	SG	?
nuScenes [Cae+20]	2020	≈200 000 (1 000 séquences)	1 x Toit	✓	✓	✓	?	✓	US, SG	✓
Waymo Open [Sun+20]	2020	≈200 000 (1 000 séquences)	1 x Toit + 4 x Côtés	✓	✓	✓	?	✓	US	✓
Lyft L5 [Hou+20]	2020	? (170 000 séquences)	1 x Toit + 2 x Côtés	✓	✓	✓	?	?	?	✓
AIODrive [Wen+21]	2021	? (100 séquences)	3 x Toit (simulés)	✓	✓	✓	✓	✓	Simulateur	✓
Cirrus [Wan+21b]	2021	? (12 séquences)	2 x Côtés	✓	✓	✓	?	?	?	✓
ONCE [Mao+21a]	2021	? (20 séquences)	1 x Toit	✓	✓	✓	✓	✓	CN	✓
WoodScape [Yog+21]	2021	10 000	1 x Toit	✓	✓	✓	?	?	US, EU, CN	✓
PandaSet [Xia+21]	2021	8 240	1 x Toit + 1 x Côtés	✓	✓	✓	?	✓	US	✓
Argoverse 2 [Wil+21]	2021	? (1 000 séquences)	2 x Toit	✓	✓	✓	✓	?	US	✓
KITTI-360 [LXG22]	2022	80 000	1 x Toit + 1 x Côtés	✓	✓	✓	?	?	DE	✓

Tableau 3.1. : Principaux jeux de données automobiles contenant des nuages de points LiDAR ainsi que des boîtes englobantes pour les véhicules. La ligne rouge horizontale marque la délimitation entre jeux existants et à venir au moment du choix du jeu de données à l'été 2020. Certains jeux de données indiquent un nombre de séquences plutôt qu'un nombre de scènes. Ces séquences sont des collections de situations ayant duré plusieurs secondes et durant lesquelles plusieurs mesures ont été prises à différents moments. Chaque séquence contient donc plusieurs scènes, mais il n'est pas toujours possible d'avoir accès au nombre de scènes composant chaque séquence.

Certains jeux de données indiquent comme taille le nombre de scènes différentes proposées quand d'autres indiquent le nombre de séquences temporelles de scènes. Une conversion grossière entre ces deux types de valeur peut être fait en sachant que la fréquence typique des capteurs LiDAR utilisés est de 10Hz et qu'une séquence de conduite dure généralement approximativement 20 secondes. Ces deux chiffres permettent d'estimer qu'une séquence est constituée d'environ 200 scènes. Cette conversion grossière en nombre de scènes pour les jeux de données indiquant un nombre de séquences est représentée entre parenthèses sur le tableau 3.1.

Parmi les jeux de données disponibles, deux catégories de taille se dégagent : les petits jeux de données KITTI [GLU12], H3D [Pat+19], Argoverse [Cha+19], A2D2 [Gey+20] et A*3D [Pha+20] qui contiennent moins de 40 000 scènes différentes et les gros jeux de données nuScenes [Cae+20] et Waymo Open [Sun+20] qui contiennent plus de 200 000 scènes. Ces deux derniers sont très similaires en termes de taille et de diversité, mais une différence importante pour notre cas d'usage les démarque : nuScenes ne propose qu'un nuage de points LiDAR par scène alors que Waymo Open en propose cinq à l'aide de 4 LiDARs latéraux en plus du LiDAR de toit qui est aussi présent sur nuScenes. Finalement, nuScenes et Waymo Open proposent tous deux des boîtes englobantes pour les piétons et les cyclistes et la présence de séquences temporelles de scènes.

En raison de cette taille importante et de la présence de cinq LiDARs, Waymo Open a été choisi comme jeu de données d'entraînement pour la suite de cette étude.

3.2 Transformation en jeu de données de grilles d'occupation

Waymo Open [Sun+20] contient 1 000 séquences de 20 secondes avec des LiDARs séquencés à 10Hz, ce qui donne environ 200 scènes par séquence. Le jeu de données étant destiné à être utilisé pour l'entraînement d'un réseau de neurones, un prédécoupage est prévu en 798 séquences pour le jeu d'entraînement, 101 séquences pour le jeu de validation et 101 séquences pour le jeu de test. Ce jeu de données a été enregistré dans les villes américaines de Phoenix, Mountain View et San Francisco. Chacune des scènes est composée de cinq nuages de points correspondant aux cinq LiDARs montés sur le véhicule. Chaque scène contient aussi des boîtes englobantes en 3D avec les étiquettes "voiture", "piéton" ou "cycliste" en fonction du type d'objet présent dans la boîte.

La transformation des nuages de points de chaque scène en grille d'occupation est réalisée à l'aide d'un outil développé dans le laboratoire d'accueil de cette thèse. Les grilles d'occupation produites pour chaque scène sont le résultat de la fusion bayésienne des cinq grilles d'occupation correspondant aux cinq nuages de points présents dans chaque scène.

Les grilles produites ont une résolution de $10\text{cm} \times 10\text{cm}$ pour chaque cellule. Cette résolution particulière a été choisie car elle permet de représenter finement les détails d'une scène sans être trop fine ce qui augmenterait de manière quadratique la taille des grilles d'occupation et donc le coût calculatoire des inférences ultérieures. Les grilles mesurent 256 cellules de côté, soit 25,6 mètres et sont centrées sur le véhicule de mesure car les LiDARs embarqués sont capables de faire des mesures aussi bien à l'avant qu'à l'arrière du véhicule. Pour éliminer au maximum les obstacles détectés par le LiDAR trop bas (le sol) ou trop hauts (le feuillage des arbres, ...) les points des différents nuages de points retenus sont ceux compris entre 50cm et 80cm de hauteur. Les autres points sont éliminés avant le calcul de la grille d'occupation. Ce prétraitement des nuages de point permet d'éviter l'apparition d'artefacts sur les grilles d'occupation correspondants à de faux obstacles.

Les boîtes englobantes 3D présentes dans le jeu de données sont représentées initialement par un tuple $(x, y, z, l, w, h, \theta)$.

— (x, y, z) correspond à la position du centre de la boîte englobante.

— (l, w, h) sont les dimensions de la boîte englobante.

— θ est un angle d'orientation autour de l'axe vertical. Il représente la direction dans laquelle est orientée le véhicule.

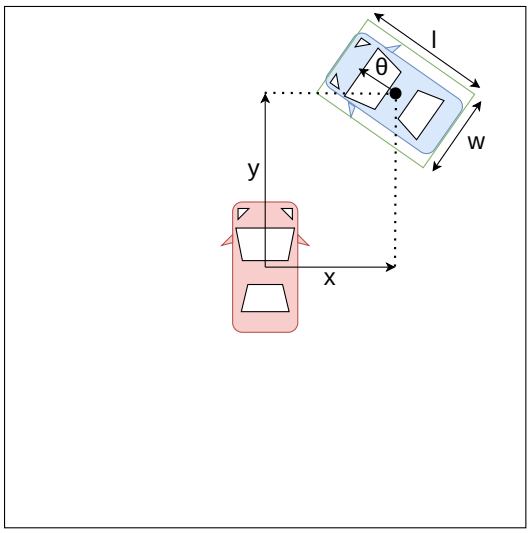

Figure 3.1. : Coordonnées de codage d'une boîte englobante 2D.

Ces boîtes englobantes sont projetées en 2D simplement par la suppression des composantes verticales z et h. La figure 3.1 illustre le codage d'une boîte englobante par ses coordonnées sur la grille d'occupation.

3.3 Analyse qualitative et quantitative

Un exemple de grille d'occupation obtenue après transformation est donné dans la figure 3.2. Les boîtes englobantes correspondant aux différents véhicules, piétons et vélos sont représentées avec une couleur différente selon la classe de l'objet. Les probabilités d'occupation de chaque cellule de la grille d'occupation sont représentées en niveaux de gris allant du blanc pour une probabilité nulle à du noir pour une probabilité d'occupation certaine. Un niveau de gris intermédiaire correspondant à une probabilité de 0,5 représente une absence d'informations sur l'occupation de la cellule.

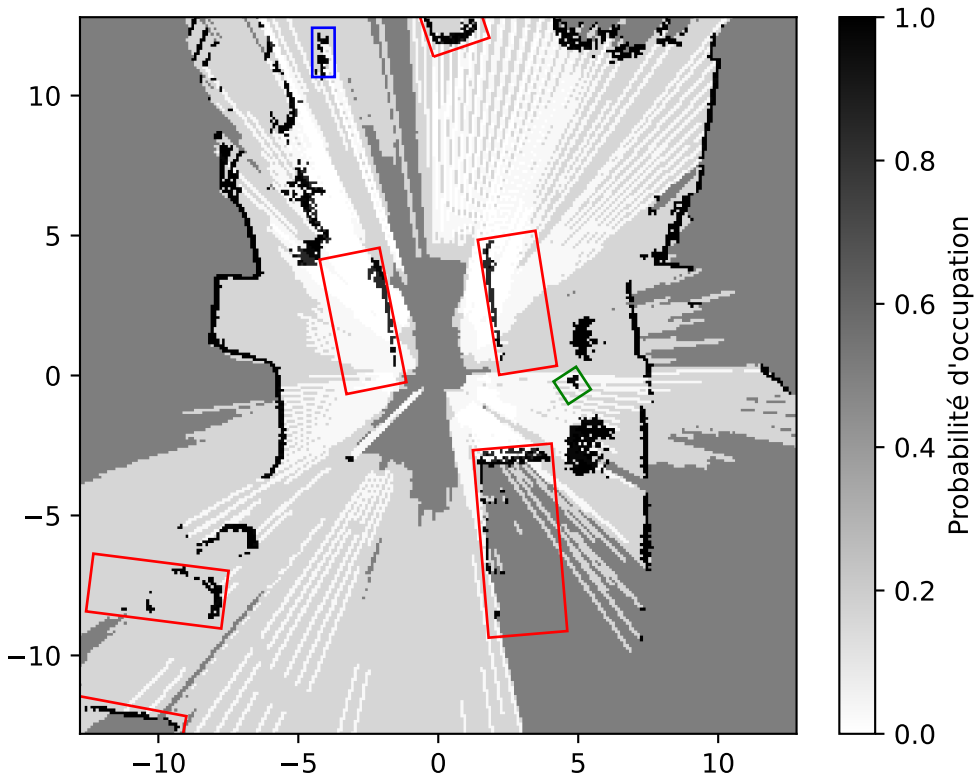

Figure 3.2. : Exemple de grille d'occupation générée par la transformation du jeu de données Waymo Open. Les probabilités d'occupation sont encodées en niveau de gris, du blanc correspondant à une probabilité de 0 au noir correspondant à une probabilité de 1. Les boîtes englobantes sont colorées de la manière suivante : rouge = voiture, vert = piéton, bleu = vélo.

Ces grilles générées permettent d'observer les obstacles en noir correspondant à des objets mouvants ou à des murs. Sur certaines grilles d'occupation courantes de l'état de l'art, on observe généralement une zone inconnue derrière chaque obstacle en raison de l'occlusion de l'arrière-plan par l'obstacle. Ce n'est pas nécessairement le cas sur ces grilles en raison de la présence de multiples capteurs qui ont permis de les générer ce qui permet dans certains cas à un des capteurs d'observer l'environnement derrière un obstacle alors qu'un autre de ces capteurs a une vision occluse par l'objet.

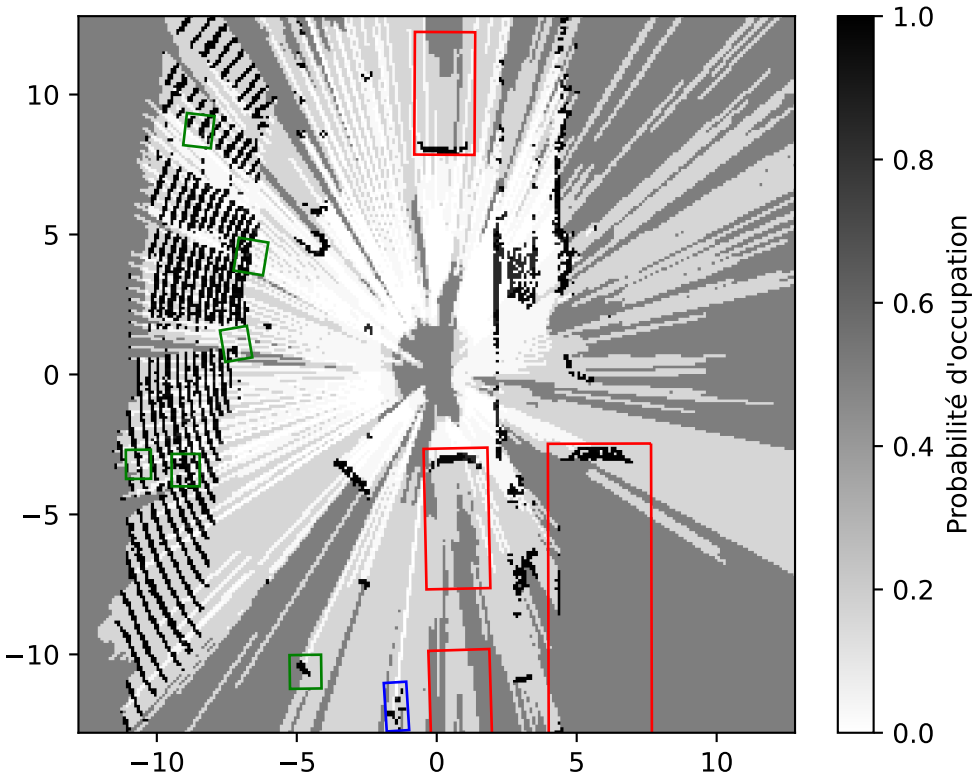

Figure 3.3. : Exemple d'artefacts du sol pouvant apparaître sur le jeu de données.

La stratégie employée de ne considérer que les points compris entre 50cm et 80cm de hauteur pour la génération de grilles d'occupation afin d'éviter les faux obstacles dus au sol est limitée par la présence de pentes sur certaines scènes du jeu de données. Cette limite peut être observée sur certaines grilles d'occupation par la présence d'artefacts sous la forme de cercles concentriques de faux obstacles. La figure 3.3 présente une de ces grilles. Ces artefacts sont problématiques car ils peuvent empêcher la représentation correcte d'objets présents dans la scène tels que les cinq piétons présents dans la partie gauche de la figure. Une stratégie d'élimination de ces artefacts serait l'utilisation d'algorithmes de détection du sol dans les nuages de points employés. Malheureusement, de tels algorithmes n'étaient

pas encore présents dans l'outil de génération de grilles d'occupation utilisé au moment de la constitution du jeu de données.

Une analyse statistique du jeu de données a été menée afin de dégager les caractéristiques des différents objets présents dans les grilles d'occupation.

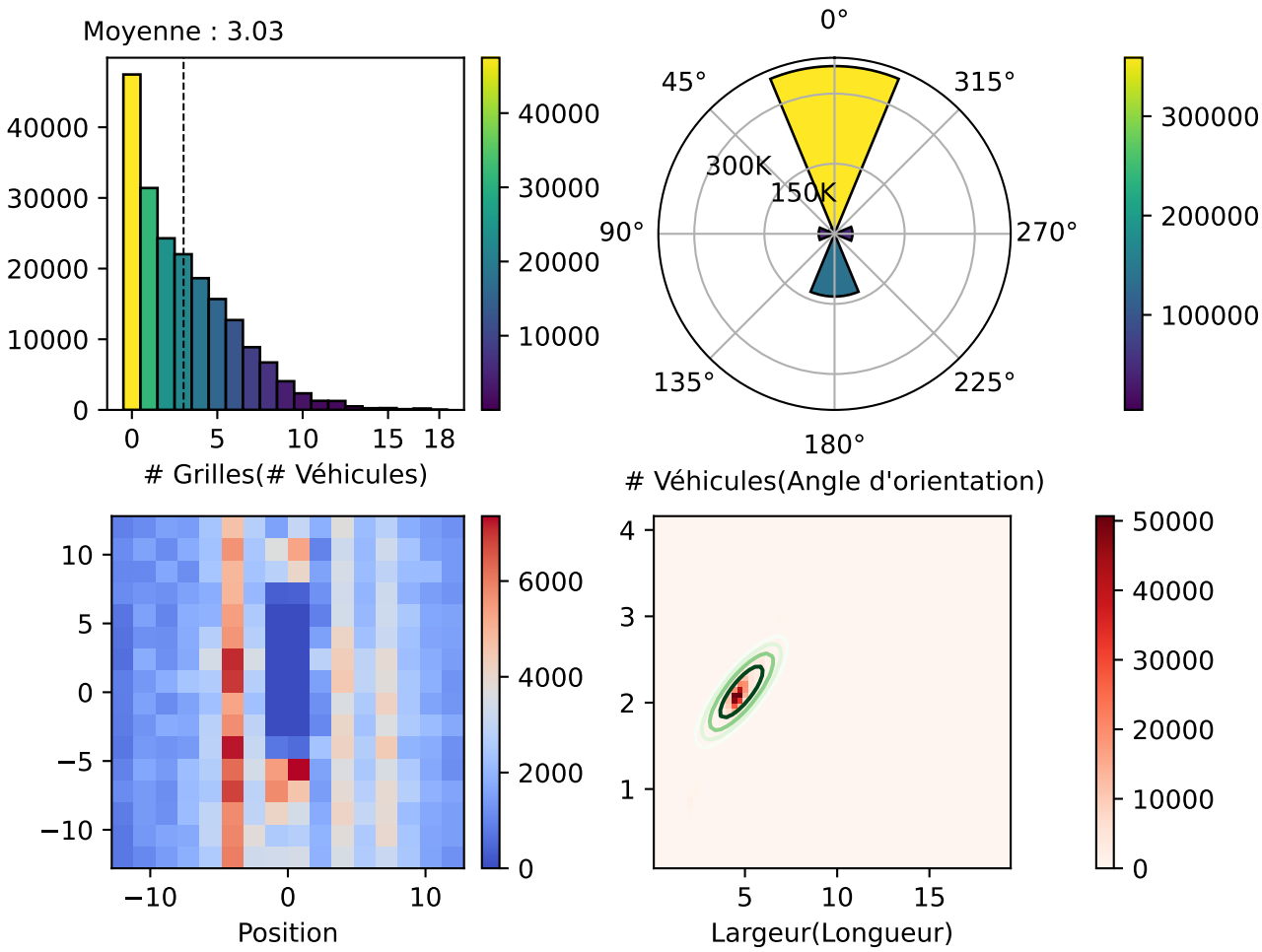

Figure 3.4. : Statistiques sur les véhicules dans le jeu de données. Sur un total de 198 068 grilles d'occupation et 599 293 véhicules. De gauche à droite et de haut en bas : Histogramme du nombre de grilles d'occupation par nombre de véhicules présents ; Histogramme polaire du nombre de véhicules par angle d'orientation ; Carte thermique de la position des véhicules dans les grilles d'occupation ; Carte thermique des dimensions des véhicules.

La figure 3.4 présente dans le sens de lecture usuel un histogramme du nombre de véhicules par grilles d'occupation, un histogramme polaire des angles d'orientation des véhicules, une carte thermique de la répartition des centres de ces véhicules sur les grilles d'occupation, ainsi que la distribution des dimensions de ces véhicules. On y observe 24% de grilles d'occupation sans véhicules, mais avec un nombre moyen de véhicules par grille de 3,03 et une bonne diversité entre 0 à 10 véhicules

par grille d'occupation. Les angles d'orientation des véhicules montrent un nombre très important de véhicules orientés à 0°, avec aussi une importante proportion de véhicules allant en sens inverse. Les orientations latérales à 90° par rapport à l'orientation du véhicule de mesure sont peu représentées, car n'ayant lieu que durant les intersections routières. La carte thermique de répartition des centres des véhicules montre une présence majoritaire des véhicules sur deux files, celle du véhicule de mesure et celle à sa gauche ce qui est cohérent avec la règle de conduite à droite en vigueur aux États-Unis, pays dans lequel a été enregistré le jeu de données. Finalement, les dimensions des véhicules sont variées mais assez concentrées autour de la moyenne de 4,8 mètres de longueur par 2,1 mètres de largeur.

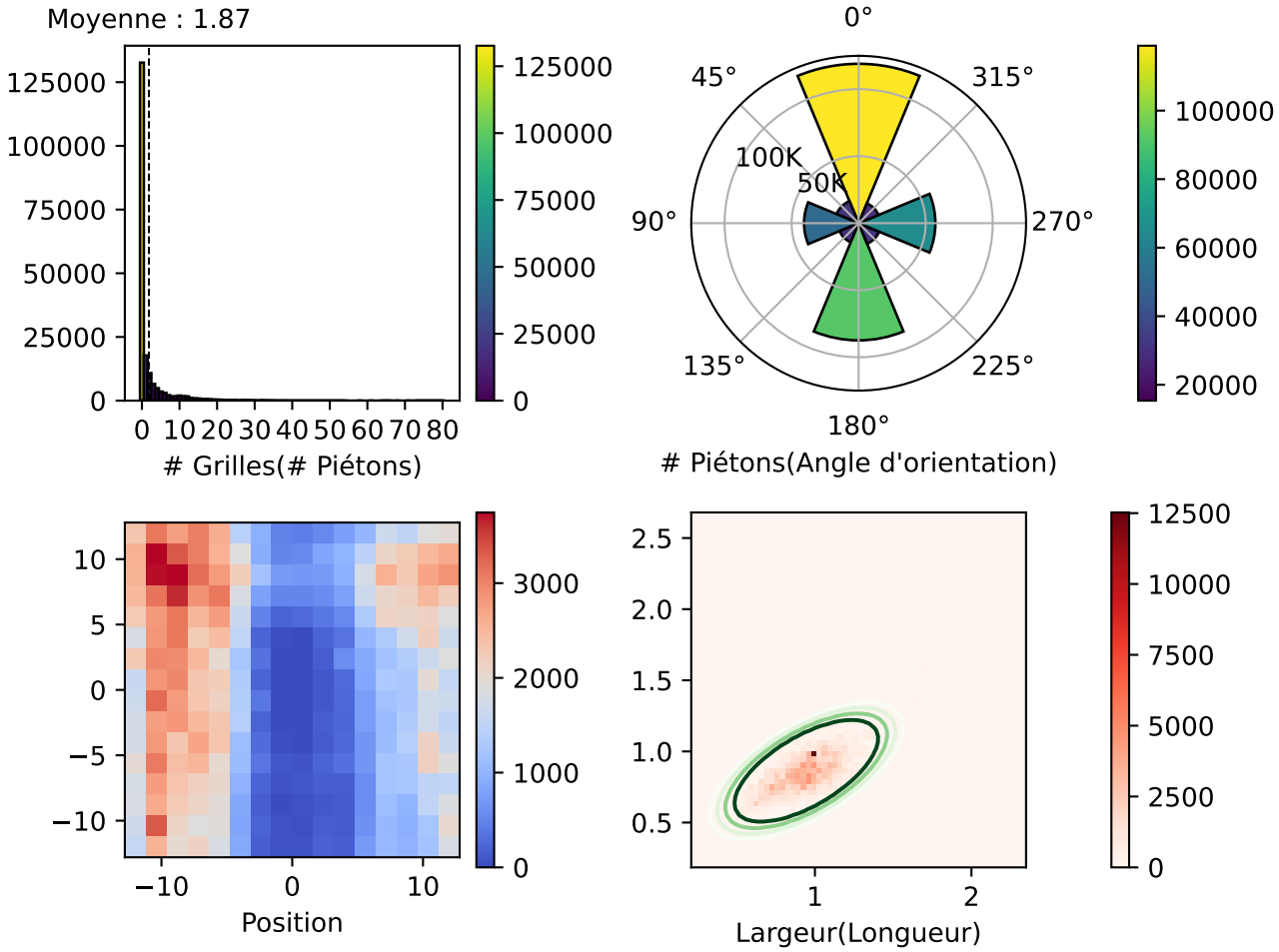

Figure 3.5. : Statistiques sur les piétons dans le jeu de données. Sur un total de 198 068 grilles d'occupation et 370 070 piétons. De gauche à droite et de haut en bas : Histogramme du nombre de grilles d'occupation par nombre de piétons présents ; Histogramme polaire du nombre de piétons par angle d'orientation ; Carte thermique de la position des piétons dans les grilles d'occupation ; Carte thermique des dimensions des piétons.

La figure 3.5 présente les mêmes informations mais cette fois-ci pour les piétons. On y observe 67% de grilles d'occupation sans piétons, mais néanmoins un nombre moyen de piétons par grille de 1,87 en raison de la présence de grilles d'occupation comportant de nombreux piétons. Les angles d'orientation des boîtes englobantes des piétons permettent d'observer le même phénomène de distribution non uniforme que pour les voitures, mais avec une meilleure diversité. La carte thermique de la position des piétons montre une distribution majoritaire sur les côtés droit et gauche des grilles d'occupation qui correspondent généralement aux trottoirs. Finalement, les dimensions sont réparties autour de la moyenne de 0,9 mètre par 0,9 mètre.

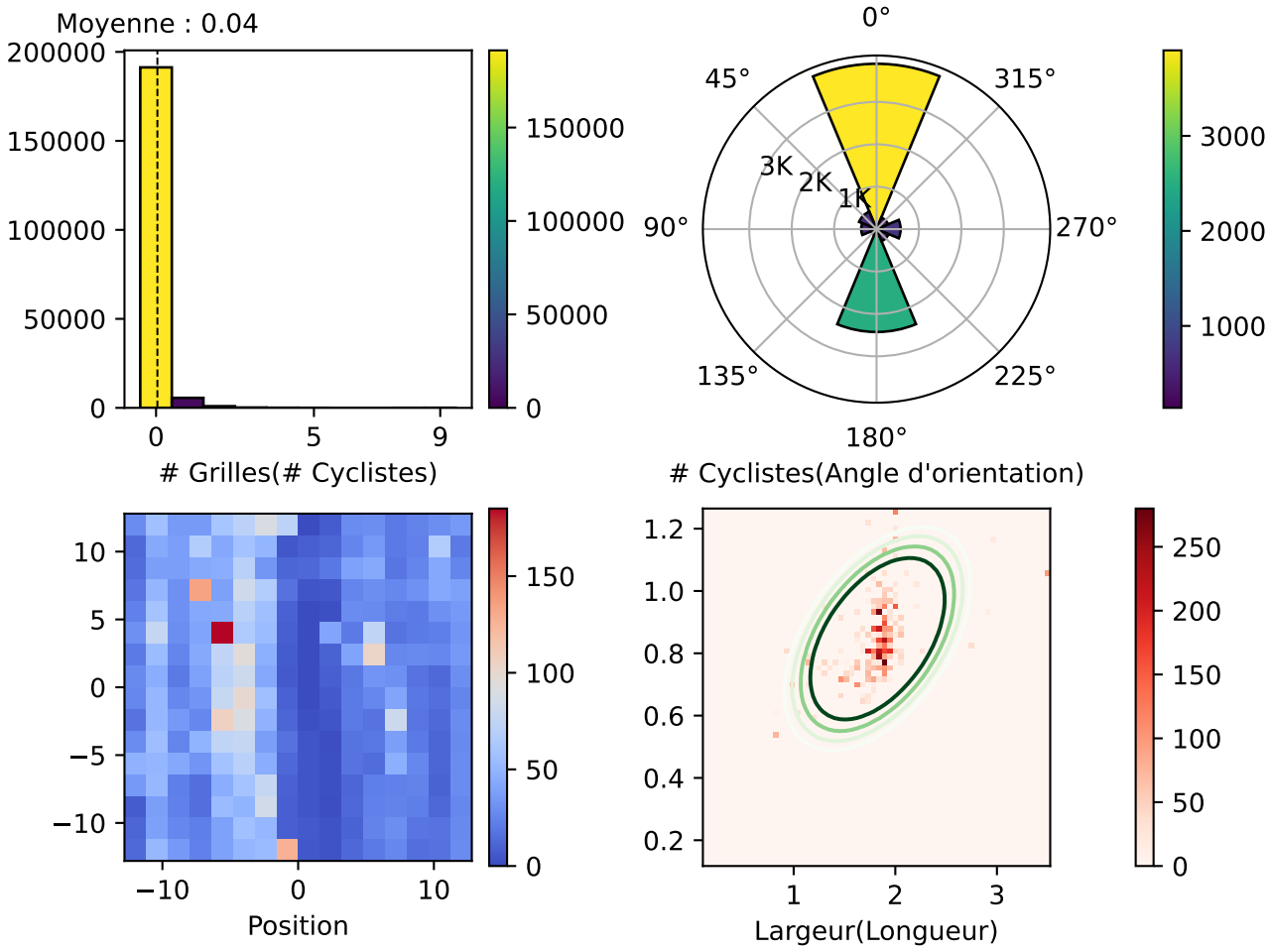

Figure 3.6. : Statistiques sur les vélos dans le jeu de données. Sur un total de 198 068 grilles d'occupation et 8 439 vélos. De gauche à droite et de haut en bas : Histogramme du nombre de grilles d'occupation par nombre de cyclistes présents ; Histogramme polaire du nombre de cyclistes par angle d'orientation ; Carte thermique de la position des cyclistes dans les grilles d'occupation ; Carte thermique des dimensions des cyclistes.

La figure 3.6 présente les informations statistiques pour les vélos. Ceux-ci ne sont présents que sur 3% des grilles du jeu de données avec une moyenne de 0,04 vélo par grille d'occupation. Ce trop faible nombre de vélos dans le jeu de données ne permet pas d'obtenir des statistiques fiables et pour cette raison les trois dernières figures ne seront pas commentées, car jugées trop incertaines.

3.4 Conclusion

Le jeu de données généré à partir du jeu de données de navigation autonome Waymo Open [Sun+20] contient toutes les données nécessaires à la conception de modèles d'apprentissage profond pour la reconnaissance de véhicules sur des grilles d'occupation. Les grilles d'occupation générées sont riches en information sur les obstacles grâce à la présence de multiples capteurs LiDAR. Certaines de ces grilles contiennent des artefacts matérialisés par de faux obstacles en raison du sol qui n'est pas plat. L'impact de ces artefacts sur la qualité des modèles conçus devra être évalué par la suite.

Les boîtes englobantes correspondant aux véhicules et aux piétons sont en nombre suffisant pour permettre l'entraînement d'un modèle de reconnaissance avec en moyenne 3 voitures et 2 piétons par grille d'occupation. En revanche, les boîtes englobantes correspondant aux vélos sont bien trop peu nombreuses pour permettre leur usage avec une moyenne quasi-nulle de vélos par grille d'occupation. La présence de nombreuses grilles d'occupation ne contenant ni véhicules ni piétons ni vélos seront d'intérêt durant la reconnaissance d'objets pour évaluer la propension des modèles à produire des faux positifs.

Bien que le sujet de cette thèse soit la reconnaissance de véhicules, la présence de piétons dans ce jeu de données est d'intérêt car cela permettra dans des travaux ultérieurs d'évaluer la capacité des modèles développés dans les deux contributions suivantes à s'adapter à de petits objets tels que des piétons. Les deux chapitres suivants présentent les deux contributions de ces travaux pour la reconnaissance des véhicules présents sur les grilles d'occupation. Ces deux contributions s'appuient sur le jeu de données présenté dans ce chapitre.

4 Détection de véhicules sur grilles d'occupation à l'aide de réseaux de neurones convolutifs

Ce chapitre présente la deuxième contribution de ce travail de thèse. L'objectif de cette contribution est d'évaluer la performance de divers modèles inspirés de l'état de l'art en détection d'objets pour la navigation autonome à détecter des véhicules dans des grilles d'occupation. Les modèles considérés sont entraînés à partir du jeu de données présenté au chapitre 3. La performance de ces modèles est évaluée selon des critères de qualité des détections produites mais aussi selon des critères de légèreté des calculs nécessaires pour une exécution en temps-réel sur matériel fortement contraint propre aux applications de navigation autonome. Ces travaux ont été publiés à la conférence GRETSI'22 [Def+22] et dans le journal MDPI Sensors [Def+23]. Un brevet a par ailleurs été déposé au sujet de ces travaux.

L'état de l'art relatif aux architectures de réseaux de neurones convolutifs implémentées dans cette section a été présenté à la partie 2.2.3. Une première partie de cette section présente ces travaux existants ayant servi d'inspiration ainsi que les raisons ayant motivé le choix de ces travaux spécifiquement. Ensuite, une seconde partie présente en détail le fonctionnement des différents détecteurs de véhicules proposés. Finalement, une dernière partie présente et analyse les résultats obtenus avec chacun de ces détecteurs.

4.1 Modèles de détection de la littérature considérés

Les grilles d'occupation sur lesquelles détecter les véhicules sont une représentation probabiliste de l'occupation de l'environnement présentée en vue de dessus au format matriciel [ME85]. Les représentations au format matriciel en vue de dessus sont très présentes dans l'état de l'art en détection d'objets pour la navigation autonome car elles ont de nombreux avantages :

— La représentation en vue de dessus de l'environnement (obtenue par exemple par projection d'un nuage de points LiDAR sur le sol) permet de représenter l'environnement avec seulement deux dimensions d'espace ce qui permet d'alléger considérablement les calculs à effectuer à la différence d'une représentation tridimensionnelle plus coûteuse à traiter.

— Une projection en vue de dessus permet de s'assurer qu'aucun obstacle n'en obstrue un autre et offre en plus l'avantage de ne pas varier la taille des objets en fonction de leur distance comme ça peut être le cas avec les projections 2D coniques obtenues par exemple sur le capteur d'un appareil photo. Pour des raisons physiques évidentes, il est néanmoins impossible d'obtenir des informations sur un objet occulté par un autre depuis le point de vue du capteur, ce qui explique la raison pour laquelle ces représentations en vue de dessus ont une manière de représenter l'absence d'informations sur une zone de l'espace par une approche spécifique (par exemple avec une probabilité d'occupation de 0,5 dans le cas des grilles d'occupation).

— Le format matriciel permet d'identifier la représentation à une image et d'utiliser l'état de l'art en détection d'objets sur images.

Pour ces différentes raisons, il existe un nombre important de travaux de détection d'objets en navigation autonome utilisant ces types de représentations matricielles en vue de dessus. Ces travaux sont ici utilisés comme source d'inspiration pour la détection d'objets sur grilles d'occupation.

La plupart des travaux de détection d'objets en vue de dessus utilisent des architectures de réseaux de neurones dérivées d'architectures de détection d'objets sur image. Les deux familles des détecteurs mono-étape et bi-étapes issues de l'état de l'art en détection d'objets sur images sont donc ici aussi présentes. Étant donné l'importance de l'exécution en temps réel des détecteurs d'objets pour les applications de navigation autonome, nous avons choisi de nous concentrer sur les réseaux de neurones dérivés de détecteurs mono-étape qui offrent généralement de bien plus faibles temps d'inférence.

Parmi ces détecteurs, certains sont basés sur YOLOv2 [RF17] comme YOLO3D [Ali+19] et Complex-YOLO [Sim+19]. D'autres sont basés sur SSD [Liu+16] comme PointPillars [Lan+19] et SE-SSD [Zhe+21]. PIXOR [YLU18] quant à lui fait exception en ce qu'il propose sa propre architecture mono-étape non dérivée de l'état de l'art en détection d'objets sur image. La figure 4.1 présente les différentes familles de détecteurs d'objets pour la navigation autonome utilisées dans ce travail.

Au vu de ces origines diverses des architectures utilisées dans les détecteurs d'objets en vue de dessus pour la navigation autonome, il semble utile de comparer l'archi-

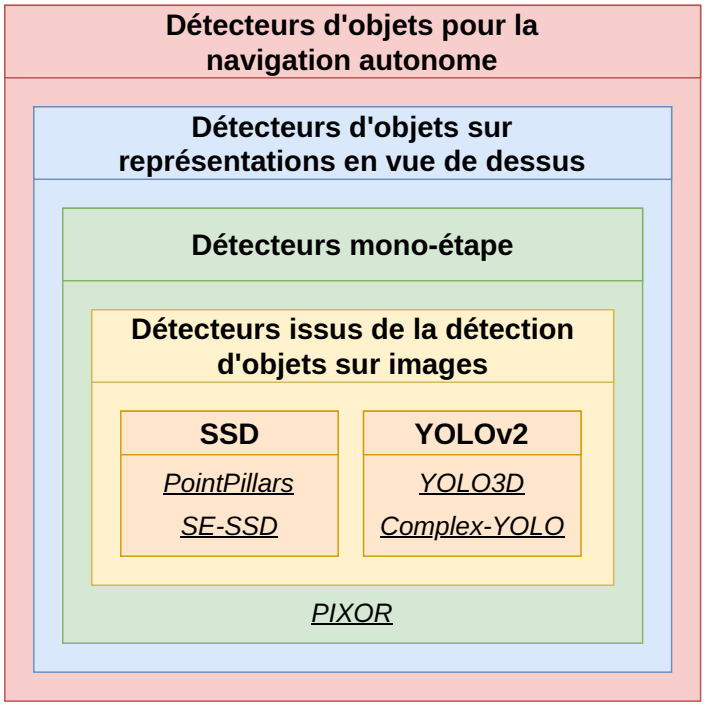

Figure 4.1. : Catégories de détecteurs d'objets pour la navigation autonome.

tecture de PIXOR [YLU18] conçue spécifiquement pour la navigation autonome à des architectures de détection issues de la détection sur image telles que la famille des détecteurs YOLO. Parmi ces derniers, les choix d'implémentation ont été de :

— Ne pas inclure YOLOv1 [Red+16] car ce réseau n'est pas un réseau complètement convolutif. En outre, il comporte une couche dense ce qui amène à un très grand nombre de poids dans le réseau peu compatible avec l'objectif de faible coût calculatoire et mémoire.

— Inclure YOLOv2 [RF17] car ce dernier est complètement convolutif et permet des temps d'inférence extrêmement faibles. De plus, il s'agit de l'architecture utilisée par YOLO3D et Complex-YOLO.

— Inclure YOLOv3 [RF18] car ce dernier, bien que moins rapide que YOLOv2, offre sur les images des résultats plus qualitatifs que YOLOv2.

— Ne pas inclure YOLOv4 [BWL20] car trop similaire à YOLOv3 qui offre de moins bons résultats que YOLOv2 que ce soit en termes de qualité des détections produites ou de temps d'inférence. Ces résultats sont détaillés à la fin de ce chapitre.

— Ne pas inclure YOLOv5, YOLOv6 [Li+22] et YOLOv7 [WBL22] car ces derniers n'existaient pas au moment de la réalisation de ces expériences.

Finalement, les architectures d'inspiration utilisées dans ce chapitre sont celles de PIXOR, de YOLOv2 et de YOLOv3. Ces deux dernières ont été modifiées de diverses manières pour obtenir une comparaison plus juste avec celle de PIXOR, par exemple en en modifiant le nombre de poids entraînables ou la taille des régions individuelles pouvant contenir une boîte englobante.

4.2 Description des différents détecteurs proposés

Cette section présente en détail les différents modèles entraînés. Y sont abordées différentes propriétés de ces modèles, telles que les formes des entrées et des sorties des réseaux de neurones, les procédures d'entraînement utilisées et l'architecture des réseaux de neurones. Finalement, un tableau récapitulatif décrit les propriétés de chacun des modèles entraînés.

4.2.1 Entrées et sorties

Dans le paradigme des détecteurs mono-étape, une grille d'occupation est utilisée en entrée d'un réseau de neurones convolutif comme s'il s'agissait d'une image à un seul canal codant des niveaux de gris. La sortie produite est une matrice de taille inférieure à la grille d'occupation originale qui représente une subdivision régulière de la grille d'occupation en régions. Chacune de ces régions de la matrice de sortie encode :

— La géométrie d'une ou de plusieurs boîtes englobantes (coordonnées du centre, dimensions, angle d'orientation).

— Un score de confiance pour chacune de ces boîtes englobantes représentant la probabilité que ces dernières correspondent bien à un objet de la grille d'occupation originale.

La figure 4.2 représente les entrées et sorties du détecteur permettant d'obtenir des boîtes englobantes orientées à partir d'une grille d'occupation.

Le facteur entre la taille du côté de la grille initiale et la taille du côté de la subdivision est appelé facteur de réduction d'échelle. Par exemple, pour une grille d'occupation originale de 256 cellules de côté et une subdivision de cette grille en 8 par 8 régions, alors le facteur de réduction d'échelle sera de 32. Par ailleurs, ce facteur de réduction d'échelle est aussi égal au côté en cellules de grilles d'occupation de chacune des régions de la subdivision.

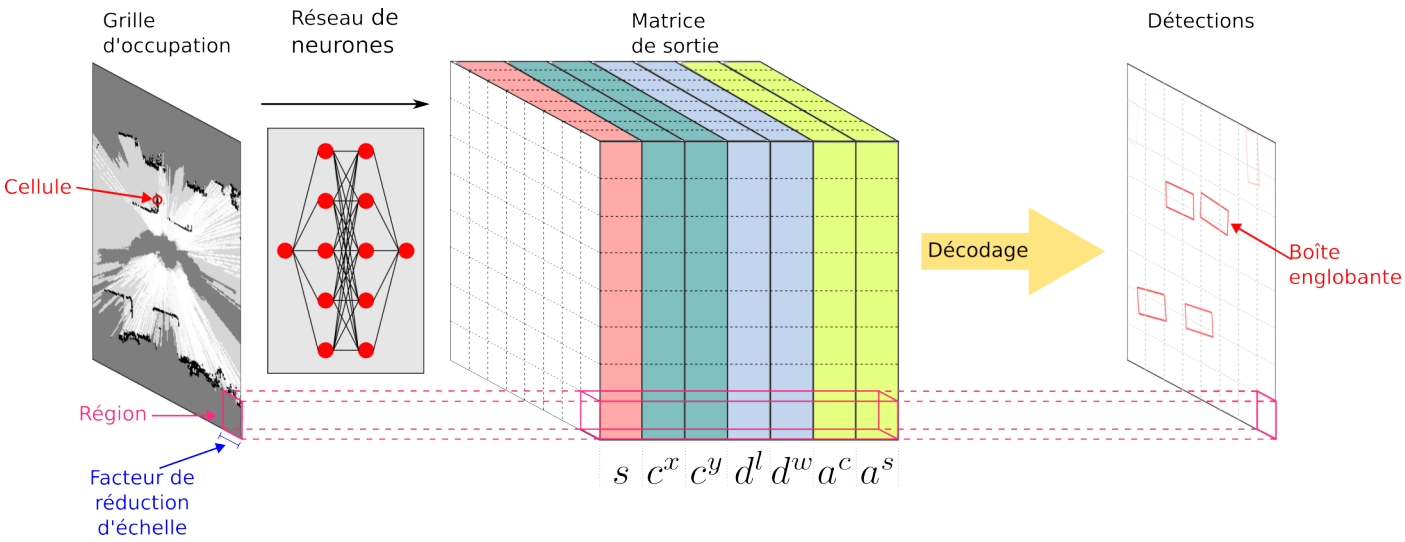

Figure 4.2. : Entrée et sortie des modèles de détection de véhicules sur grilles d'occupation. Dans cet exemple le facteur de réduction d'échelle est de 32 car la grille d'occupation est divisée en 8 par 8 régions. Chaque région mesure donc 32 par 32 cellules.

Chacune des régions de la subdivision de sortie du réseau de neurones encode une boîte englobante par un tuple :

$$r = (s, c^x, c^y, d^l, d^w, a^c, a^s)$$

où :

— $s \in [0, 1]$ est un score de confiance utilisé pour représenter la confiance du détecteur dans le fait que la boîte englobante correspond effectivement à un objet de la grille d'occupation. Une fonction d'activation sigmoïde est utilisée en sortie du réseau de neurones pour s'assurer que ce score de confiance soit bien dans l'intervalle spécifié.

— c^x et c^y représentent la position du centre de la boîte englobante par rapport au coin supérieur gauche de la région.

— d^l et d^w représentent le logarithme des dimensions de la boîte englobante.

— a^c et a^s représentent le cosinus et le sinus de l'angle d'orientation de la boîte englobante. Ce choix de découplage de l'angle d'orientation en deux valeurs indépendantes est classiquement utilisé pour les problèmes de prédiction de boîtes englobantes orientées [YLU18].

Après l'inférence, la matrice de subdivision obtenue en sortie est décodée en une liste de boîtes englobantes à l'aide d'une procédure de décodage. Un enjeu crucial de cette procédure est de transformer le nombre fixe de boîtes englobantes prédites

dans chaque région de la subdivision en une liste de taille variable. En effet, toutes les grilles d'occupation ne contiennent pas le même nombre d'objets et le nombre de régions de la subdivision est simplement le nombre maximal de boîtes englobantes pouvant être prédites par le détecteur. Pour faire cette transformation, le score de confiance est utilisé : ne sont conservées que les boîtes englobantes prédites dont le score de confiance est supérieur à un certain seuil. Le choix d'un seuil est dépendant de l'application visée : un seuil haut permettra d'éliminer au maximum les faux positifs au risque de ne pas détecter certains véhicules alors qu'un seuil bas permettra d'éliminer les faux négatifs mais risquera de produire plus de faux positifs.

Dans le cas d'une subdivision très fine de la grille d'occupation avec un facteur de réduction d'échelle faible, il peut arriver que plusieurs régions de la subdivision prédisent la même boîte englobante. Dans ce cas, un algorithme d'élimination des doublons, appelé *algorithme de suppression non maximale* [RK82], permet de ne conserver que les boîtes englobantes avec le score de confiance le plus élevé parmi les doublons.

4.2.2 Procédure d'entraînement

Pour entraîner le réseau de neurones ainsi créé, il est nécessaire de calculer l'erreur entre les prédictions du modèle et la sortie attendue. Ces sorties sous la forme de matrices de subdivision doivent donc être créées à partir des listes de véhicules présents dans le jeu de données.

Deux types de transformation sont utilisées dans ce travail : l'association *un-à-un* et l'association *un-à-plusieurs* :

— Dans l'association *un-à-un*, chaque boîte englobante présente dans le jeu de données d'entraînement va être associée à la région de la subdivision qui contient le centre de cette boîte englobante. Ce choix est utilisé dans les cas où la subdivision est grossière car dans ces cas la taille d'une région sera du même ordre de grandeur que la taille d'une boîte englobante.

— Dans l'association *un-à-plusieurs*, chaque boîte englobante présente dans le jeu de données va être associée à toutes les régions de la subdivision strictement contenues à l'intérieur de la boîte englobante. Cette méthode d'association est plus souvent utilisée dans le cas de subdivisions très fines où de multiples régions peuvent être raisonnablement associées à une même boîte englobante. Dans ce cas, le modèle étant entraîné à produire de nombreuses détections correspondant à la même boîte englobante, l'usage de l'algorithme de suppression non maximale est obligatoire pour éliminer tous les doublons produits.

Une fois l'association entre régions de la subdivision et boîtes englobantes effectuée, chaque région associée, dont on va noter $(\textit{origine}^x, \textit{origine}^y)$ les coordonnées du coin supérieur gauche, va recevoir l'encodage de la boîte englobante (x, y, l, w, θ) suivant :

$$
\begin{aligned}
s &= 1 \\
c^x &= \frac{(x - \textit{origine}^x)}{\textit{côté}} \\
c^y &= \frac{(y - \textit{origine}^y)}{\textit{côté}} \\
d^l &= \log l \\
d^w &= \log w \\
a^c &= \cos\theta \\
a^s &= \sin\theta
\end{aligned}
$$

Les régions qui n'auront pas été assignées à une quelconque boîte englobante seront encodées $(0, 0, 0, 0, 0, 0, 0)$.

La fonction de coût utilisée est une combinaison de la fonction de coût de classification $\mathcal{L}_{cls}$ qui est une entropie croisée binaire sur les scores de confiance s et la fonction de coût de régression $\mathcal{L}_{reg}$ qui est une norme L_1 lissée sur les paramètres de régression $(c^x, c^y, d^l, d^w, a^c, a^s)$ pour les régions assignées à une boîte englobante vérité terrain. Spécifiquement, pour une matrice de sortie prédite $\hat{y} = (\hat{r}_{i,j})$ et la matrice correspondante vérité terrain $y = (r_{i,j})$ issue de la transformation des listes de boîtes englobantes du jeu de données d'entraînement avec $r_{i,j} = (s_{i,j}, c^x_{i,j}, c^y_{i,j}, d^l_{i,j}, d^w_{i,j}, a^c_{i,j}, a^s_{i,j})$, alors la fonction de coût est :

$$
\begin{aligned}
\mathcal{L}(\hat{y}, y) &= \sum_{i,j} \mathcal{L}_{cls}(\hat{s}_{i,j}, s_{i,j}) + \sum_{i,j} s_{i,j} \left[\sum_{p \in \{c^x, c^y, d^l, d^w, a^c, a^s\}} \mathcal{L}_{reg}(\hat{p}_{i,j}, p_{i,j}) \right] \\
\text{où } \mathcal{L}_{cls}(a, b) &= -(a \log(b) + (1 - a) \log(1 - b)) \\
\mathcal{L}_{reg}(a, b) &= \begin{cases} 0.5 \cdot |a - b|^2, & \text{si } |a - b| \leqslant 1 \\ |a - b| - 0.5, & \text{si } |a - b| > 1 \end{cases}
\end{aligned}
$$

L'algorithme d'optimisation utilisé pour l'entraînement des modèles est l'algorithme Adam [KB17] qui est un type d'algorithme de descente de gradient stochastique couramment utilisé en raison de son efficacité calculatoire et sa capacité à entraîner de grands modèles d'apprentissage profond. Le taux d'apprentissage utilisé est de 0.001, le taux de décroissance exponentielle est de 0.9 pour l'estimation du

premier moment et de 0.999 pour l'estimation du second moment. Finalement, la fonction d'erreur sur le jeu de données de validation est calculée à chaque fin d'époque d'entraînement et l'apprentissage est arrêté dès que celle-ci commence à augmenter. Cette procédure appelée arrêt précoce permet d'éviter tout phénomène de surapprentissage du modèle [GBC16].

4.2.3 Architectures

Quatre architectures différentes de réseaux de neurones sont utilisées :

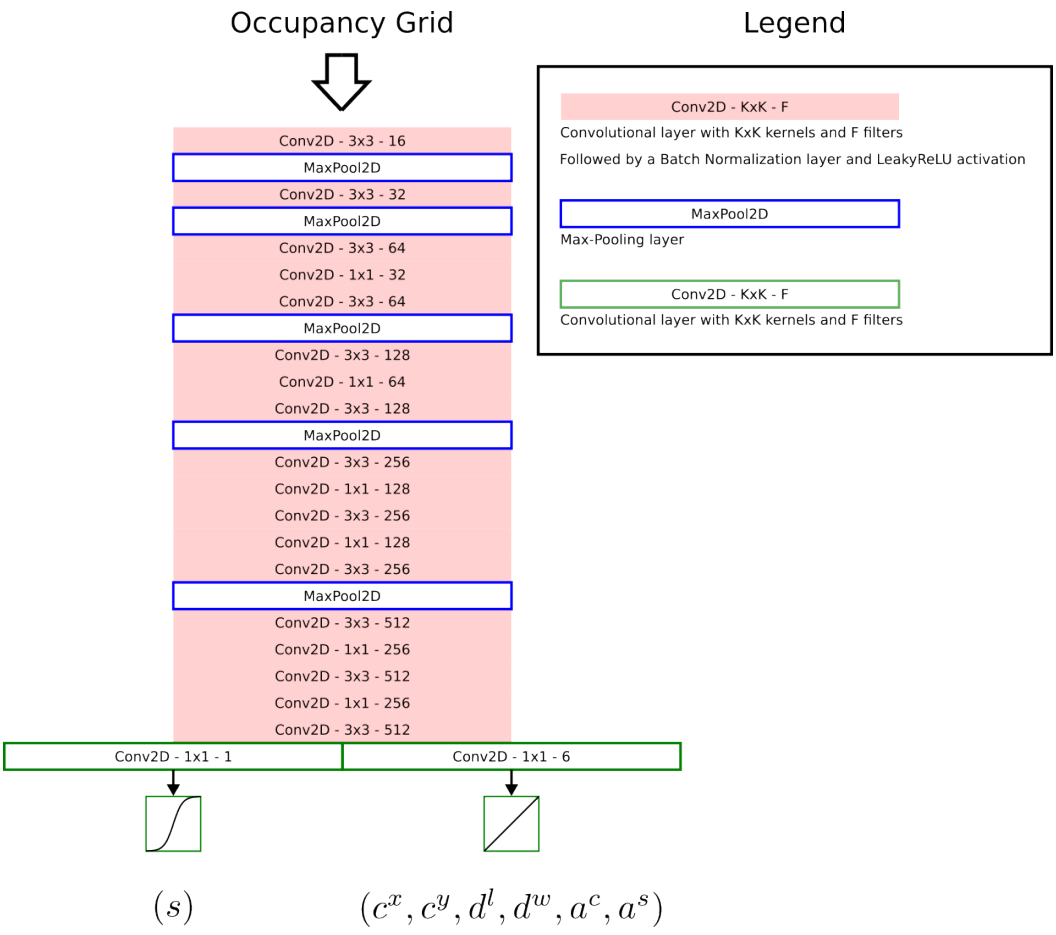

Figure 4.3. : Architecture de **GRID-YOLOv2**.

— **GRID-YOLOv2** (Figure 4.3) : Cette architecture inspirée de celle de YOLOv2 [RF17] extrait les caractéristiques des grilles d'occupation via une alternance de couches de convolution dans lesquelles se trouvent les poids à entraîner et de couches de sous-échantillonnage par valeur maximale permettant de diminuer la taille des vecteurs de caractéristiques jusqu'à atteindre le

facteur de réduction d'échelle de 32. Deux couches de convolution parallèles permettent ensuite de calculer le score de confiance de chaque région pour la première et les six coordonnées des boîtes englobantes pour la seconde. La première de ces deux couches est suivie d'une fonction d'activation sigmoïde pour confiner les scores de confiance dans l'intervalle $[0, 1]$.

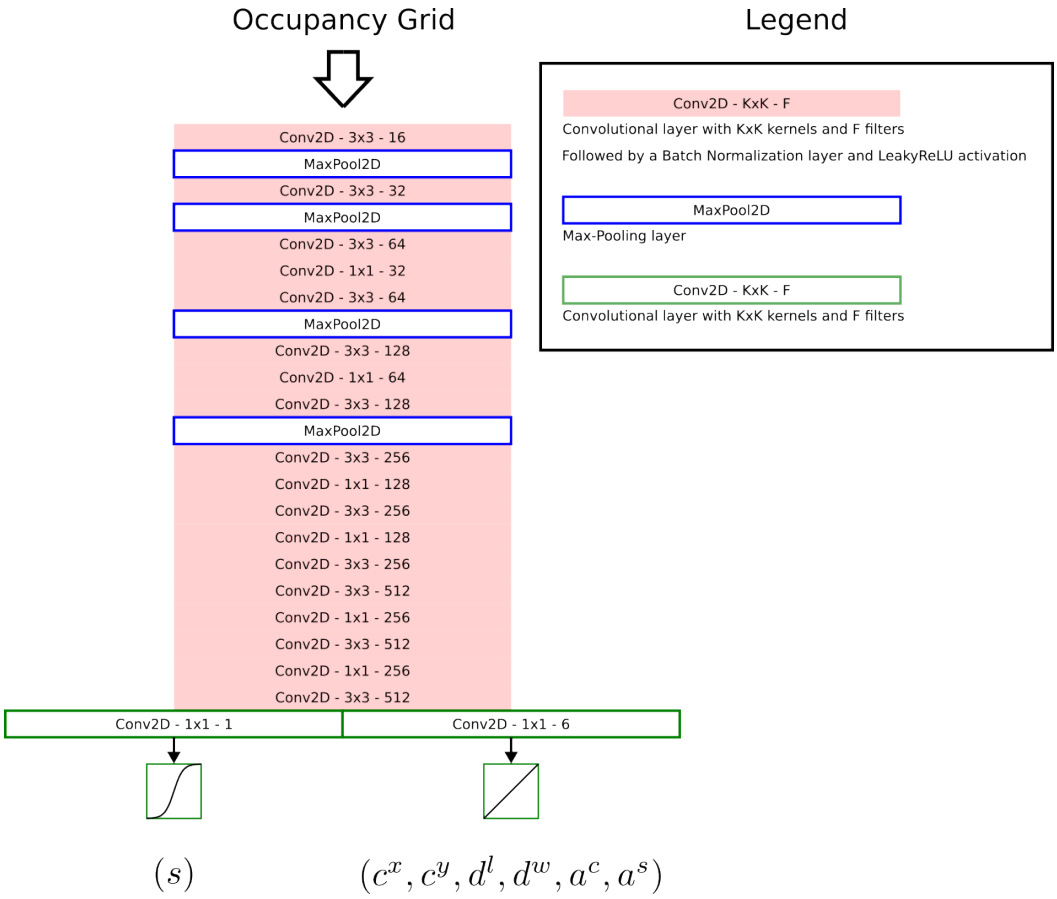

Figure 4.4. : Architecture de **GRID-YOLOv2-noMP**.

— **GRID-YOLOv2-noMP** (Figure 4.4) : Cette architecture est identique à **GRID-YOLOv2** à la différence que l'on souhaite ici obtenir un facteur de réduction d'échelle de 16 pour affiner la subdivision de sortie du détecteur et pouvoir proposer plus de boîtes englobantes par grille d'occupation. Cette modification est atteinte par la suppression de la dernière couche de sous-échantillonnage par valeur maximale dans **GRID-YOLOv2**.

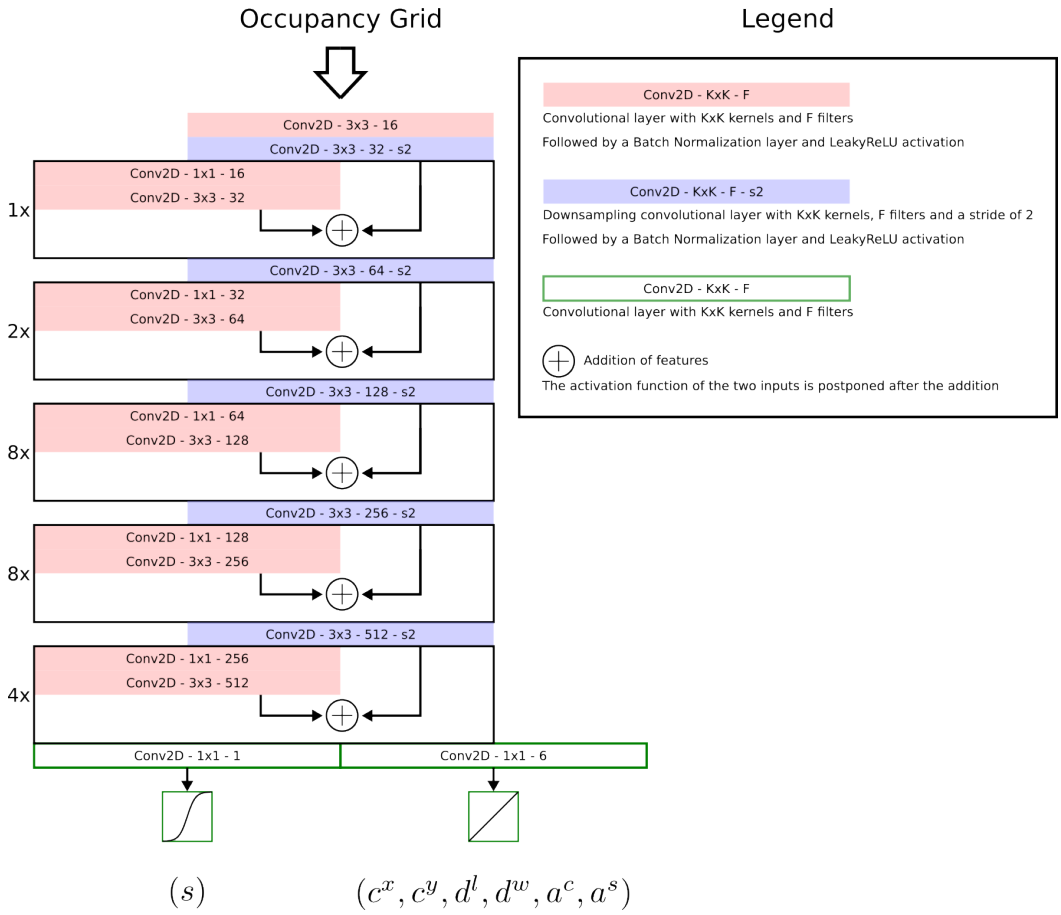

Figure 4.5. : Architecture de **GRID-YOLOv3**.

— **GRID-YOLOv3** (Figure 4.5) : Cette architecture inspirée de celle de YOLOv3 [RF18] est une alternance de blocs de couches d'extraction de caractéristiques et de couches de convolution avec un pas de 2 qui permettent de réduire la taille des vecteurs de caractéristiques à un facteur de réduction d'échelle de 32. Ces couches de convolution avec un pas de 2 sont l'équivalent dans cette architecture de la combinaison d'une couche de convolution avec un pas de 1 et d'une couche de sous-échantillonnage par valeur maximale dans les architectures basées sur YOLOv2. Les blocs d'extraction de caractéristiques contiennent des couches de convolution et des connexions sautées qui aident à l'apprentissage du modèle.

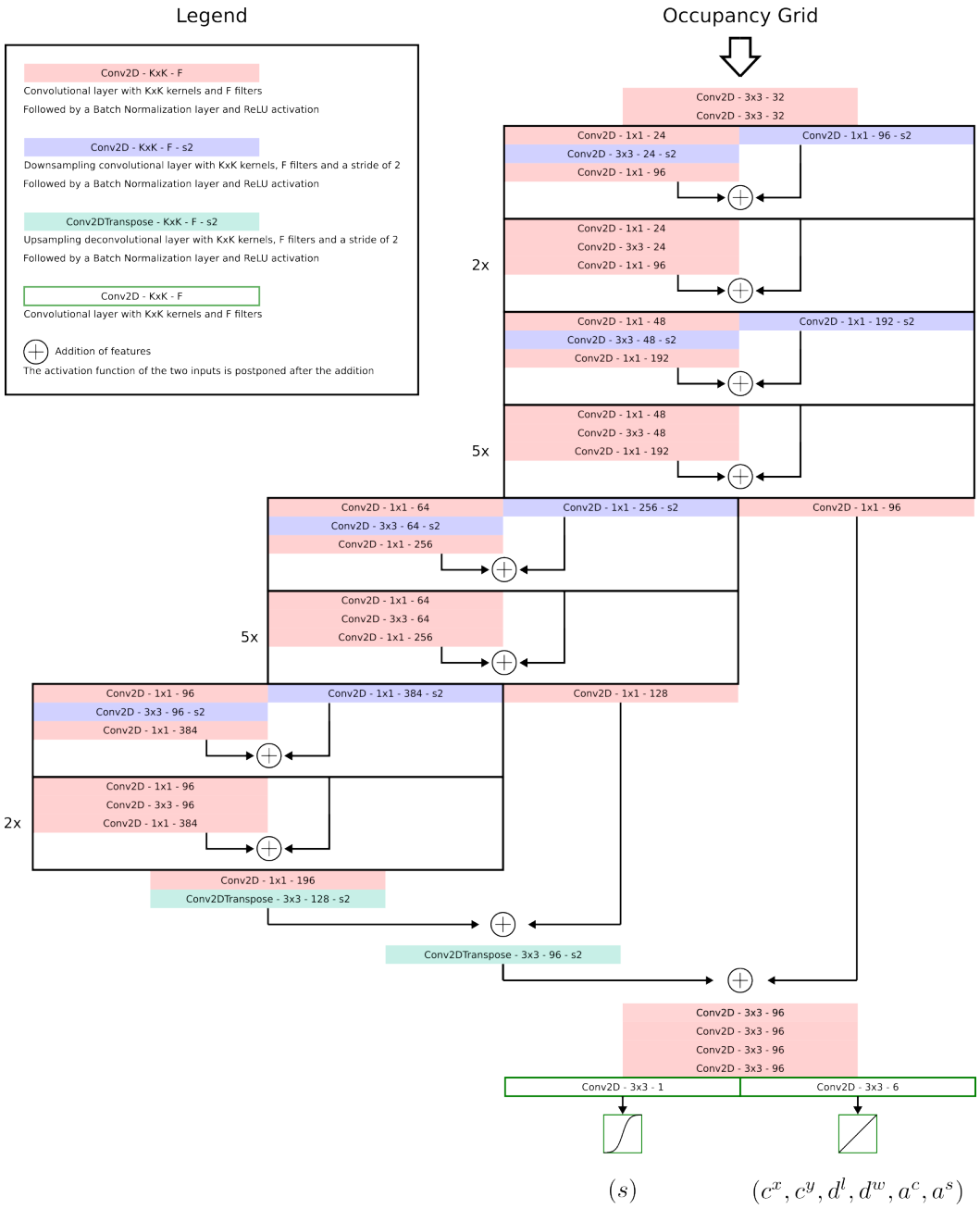

Figure 4.6. : Architecture de **GRID-PIXOR**.

— **GRID-PIXOR** (Figure 4.6) : Cette architecture tirée directement de l'article original [YLU18] calcule trois niveaux de granularités de caractéristiques de la même manière que le réseau U-Net [RFB15] avant de les recombiner puis de produire les coordonnées de chaque boîte englobante à partir de ces caractéristiques. Les caractéristiques sont extraites à l'aide de blocs résiduels qui sont composés de connexions sautées comme pour l'architecture **GRID-YOLOv3**.

4.2.4 Résumé des différents modèles proposés

La table 4.1 résume les propriétés de chacun des détecteurs de véhicules implémentés. Leur architecture ainsi que leur facteur de réduction d'échelle et leur nombre de poids entraînables y sont spécifiés. La méthode utilisée pour associer les boîtes englobantes vérités terrain aux régions lors de la création des matrices de sortie attendues est spécifiée. Finalement, si une méthode de post-traitement est utilisée pour supprimer les doublons, le nom de cette méthode est précisé.

Nom	Réduction d'échelle	Architecture	# poids	Association	Post-traitement
D1	32	**GRID-YOLOv2**	4 961 463	un-à-un	-
D2	16	**GRID-YOLOv2-noMP**	4 961 463	un-à-un	-
D3	32	**GRID-YOLOv3**	10 158 679	un-à-un	-
D4	4	**GRID-PIXOR**	2 133 379	un-à-plusieurs	NMS

Tableau 4.1. : Description des différents détecteurs entraînés.

4.3 Métriques évaluées, résultats et analyse

Cette section présente les métriques utilisées pour l'évaluation qualitative et quantitative des détecteurs entraînés. S'ensuit une présentation des résultats obtenus par chaque détecteur pour chacune de ces métriques. Finalement, une analyse qualitative des résultats obtenus est proposée aussi bien en termes de qualité des détections produites qu'en termes de temps d'inférence.

4.3.1 Métriques utilisées pour l'analyse des modèles

L'objectif des expériences réalisées dans ce chapitre est d'évaluer la capacité et la performance de plusieurs détecteurs d'objets à détecter des véhicules sur des grilles d'occupation. Cette performance est évaluée en termes de qualité des détections produites mais aussi de temps d'exécution. Il est donc nécessaire d'utiliser deux métriques différentes pour mesurer ces deux points d'évaluation.

La qualité des détections est évaluée grâce à la métrique de précision moyenne (*AP* ou *Average Precision*) classiquement utilisée pour évaluer les détecteurs d'objets [Rus+15]. Cette métrique est issue de l'évaluation des classificateurs binaires. Elle a été adaptée pour fonctionner avec des détecteurs d'objets. Dans son domaine

d'évaluation original, la précision moyenne est obtenue à partir du nombre de vrais positifs, de faux positifs et de faux négatifs du classificateur. Pour l'adapter à l'évaluation des détecteurs d'objets, il faut pouvoir classer les boîtes englobantes selon ces trois catégories.

La mesure du ratio d'intersection sur union (*IoU*) entre boîtes englobantes prédites et boîtes englobantes vérités terrain est utilisée pour classer ces dernières en vrais/faux positifs/négatifs. Concrètement, le ratio de l'aire de l'intersection sur l'aire de l'union entre deux boîtes englobantes est calculé. Si ce ratio est supérieur à un seuil prédéfini, ces deux boîtes sont considérées comme associées, ce qui signifie que l'on considère que la boîte englobante vérité terrain a été correctement détectée par la boîte englobante prédite.

Le choix de ce seuil de ratio d'intersection sur l'union représente la précision géométrique attendue du détecteur. Un seuil plus élevé demande une meilleure superposition des boîtes englobantes prédites et vérités terrain et mènera donc à une précision moyenne plus faible. Dans ce travail, nous avons décidé de considérer les métriques de précision moyenne pour deux seuils différents : un seuil de 0.5 et un seuil de 0.7. Ces deux seuils sont tous deux classiquement utilisés dans la littérature pour l'évaluation de détecteurs d'objets. L'annexe A.1 explique plus en détail la procédure de calcul de la précision moyenne.

Le coût calculatoire des détecteurs est quant à lui évalué par la mesure de la durée entre la mise à disposition de la grille d'occupation à l'entrée du détecteur et la récupération des boîtes englobantes prédites en sortie. Ce temps se décompose en un temps d'inférence qui est le temps pris par l'évaluation du réseau de neurones et le temps de post-traitement, le cas échéant, qui mesure le temps pris par l'algorithme de suppression non-maximale pour éliminer les doublons.

Les temps ont été mesurés sur un ordinateur équipé d'un processeur Intel Core i7-10850H et d'une carte graphique Nvidia Quadro RTX 3000 Mobile. Le code a été écrit en Julia [Bez+17] qui est un langage compilé juste-à-temps et par conséquent l'opération d'inférence est exécutée une première fois avant la mesure des temps d'inférence pour forcer une compilation préalable et ne pas inclure ce temps de compilation dans les temps mesurés. Le logiciel d'apprentissage profond utilisé est TensorFlow [Aba+16].

L'algorithme de suppression non maximale utilisé en tant que post-traitement effectue une boucle pour comparer géométriquement toutes les boîtes englobantes prédites. Cet algorithme a donc une complexité quadratique en fonction du nombre de boîtes englobantes prédites, nombre qui dépend lui-même du seuil de score de

confiance choisi pour considérer une prédiction comme positive. Dans les mesures présentées par la suite, ce seuil a été choisi à 0.5.

4.3.2 Résultats quantitatifs

Le tableau 4.2 présente les différentes métriques collectées sur les quatre détecteurs entraînés.

Détecteur	$AP_{0.7}$	$AP_{0.5}$	Temps d'inférence	Temps de post-traitement	Temps total
D1	0.55	0.82	**11ms**	-	**11ms**
D2	0.79	0.89	**11ms**	-	**11ms**
D3	0.50	0.82	35ms	-	35ms
D4	**0.82**	**0.90**	45ms	5ms	50ms

Tableau 4.2. : Précisions moyennes, Temps d'inférence et de post-traitement. Le GPU utilisé pour la mesure des temps est un Nvidia Quadro RTX 3000 Mobile. Le seuil de score de confiance choisi pour le calcul des temps de post-traitement est 0.5.

Au vu de ces résultats, les détecteurs peuvent se scinder en deux groupes. **D2** et **D4** obtiennent des métriques de précision moyenne très similaires dans l'intervalle $[0.79, 0.82]$ pour la précision moyenne à 0.7 et dans l'intervalle $[0.89, 0.90]$ pour la précision moyenne à 0.5. **D1** et **D3** obtiennent quant à eux des métriques de précision moyenne beaucoup plus faibles avec une précision moyenne à 0.7 comprise dans l'intervalle $[0.50, 0.55]$.

Concernant les temps d'inférence, les détecteurs **D1** et **D2** dérivés d'une architecture YOLOv2 obtiennent les meilleures performances avec des temps totaux de 11ms par grille d'occupation. Le détecteur **D3** dérivé d'une architecture YOLOv3 arrive en second avec 35ms observés par grille d'occupation. Finalement, le détecteur **D4** issu de l'architecture de PIXOR arrive en dernier avec 50ms par grille d'occupation dont 5ms de temps de post-traitement.

Ces résultats semblent suggérer que bien que **D4** propose les meilleurs résultats en termes de qualité des détections, **D2** propose un bien meilleur équilibre qualité/rapidité avec un temps total de prédiction près de cinq fois plus faibles pour une faible réduction de précision moyenne.

4.3.3 Résultats qualitatifs et analyse

Une analyse plus fine des résultats obtenus permet de mettre en évidence les caractéristiques de chaque détecteur. La figure 4.7 présente les courbes de précision en fonction du rappel des quatre détecteurs pour un seuil de ratio d'intersection-sur-l'union de 0.7. La précision est le ratio des boîtes englobantes correctement prédites parmi toutes les boîtes englobantes prédites. Le rappel est le ratio des boîtes englobantes correctement prédites parmi toutes les boîtes englobantes à prédire. Une précision élevée signifie que très peu de boîtes prédites ne devraient pas l'être mais cela ne dit rien sur les boîtes englobantes qui auraient dû être prédites. Un rappel élevé signifie que très peu de boîtes englobantes à prédire ne l'ont pas été mais cela ne dit rien sur les boîtes englobantes prédites qui n'auraient pas dû l'être. Les points qui composent ces courbes sont obtenus en faisant varier le seuil de score de confiance utilisé pour considérer une région comme contenant une boîte englobante. Ainsi, abaisser ce seuil revient à considérer plus de boîtes englobantes prédites et donc à récupérer plus de boîtes englobantes vérités terrain mais cela augmente aussi le risque de récupérer des faux positifs.

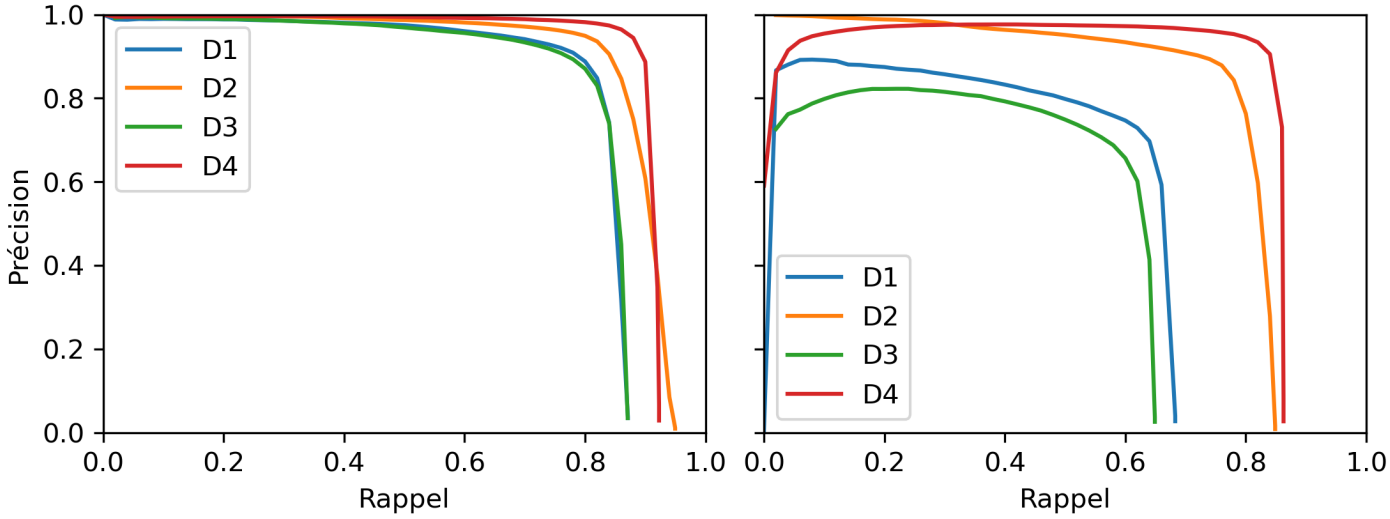

Figure 4.7. : Courbes de la précision en fonction du rappel pour les quatre détecteurs présentés. Gauche : Seuil d'IoU de 0.5 ; Droite : Seuil d'IoU de 0.7.

Ces courbes permettent par la suite de calculer la précision moyenne qui n'est autre que l'aire sous cette courbe. Elles sont donc utiles pour tenter de comprendre les résultats obtenus. Il est généralement admis que ces courbes doivent être décroissantes. Ce n'est pas le cas ici pour **D1** et **D3** et **D4** qui connaissent une légère baisse de précision alors même que le rappel baisse sur l'intervalle $[0, 0.1]$. Cette non-monotonie signifie que pour ces détecteurs, certaines boîtes incorrectement prédites (des faux

positifs) le sont avec un score de confiance très élevé alors que celui-ci devrait être bas. Ces occurrences de faux positifs de ces détecteurs peuvent être remarquées sur la figure 4.8 qui présente des détections produites par ces détecteurs sur deux grilles d'occupation différentes.

On peut aussi observer sur les courbes que **D1** et **D3** obtiennent un rappel maximal inférieur à 0.7 alors que **D2** et **D4** ont un rappel maximal supérieur à 0.8. Ce résultat peut s'interpréter comme l'impossibilité pour les deux premiers détecteurs de détecter certaines boîtes englobantes vérités terrain. Ce résultat s'observe aussi par la présence de faux négatifs sur la figure 4.8. Ce résultat peut être mis en lien avec le facteur de réduction d'échelle de 32 commun à ces deux détecteurs qui produit des régions trop grandes par rapport à la taille typique d'une boîte englobante.

Finalement, le comportement qu'a **D2** à produire quelques doublons peut être observé sur la figure 4.8 mais sans que cela ait un impact important sur les métriques de précision moyenne. Ce comportement pourrait être mitigé par l'utilisation de l'algorithme de suppression non-maximale permettant de supprimer les doublons.

Les résultats obtenus en termes de temps totaux d'exécution sont très différents selon les architectures utilisées. Les architectures de **D3** et **D4** respectivement inspirées de celles de YOLOv3 et de PIXOR contiennent bien plus de couches que celles de **D1** et **D2** qui sont inspirées de l'architecture de YOLOv2. En conséquence, leurs temps d'inférence sont, eux aussi, beaucoup plus longs. On remarque cependant en regardant le nombre de poids entraînables dans le tableau 4.1 que **D4** contient le moins de poids entraînables. Il semble donc que ce soit la séquentialité et le nombre de couches des réseaux qui expliquent leur temps d'inférence élevé plutôt que le nombre de poids entraînables contenus à l'intérieur.

Concernant le temps de post-traitement de **D4**, étant donné que l'algorithme de suppression non maximale utilisé est quadratique selon le nombre de boîtes englobantes prédites, le temps de post-traitement de **D4** est élevé car ce détecteur a un facteur de réduction d'échelle de 4 ce qui entraîne un grand nombre de régions sur la grille d'occupation et donc un nombre important de boîtes englobantes prédites.

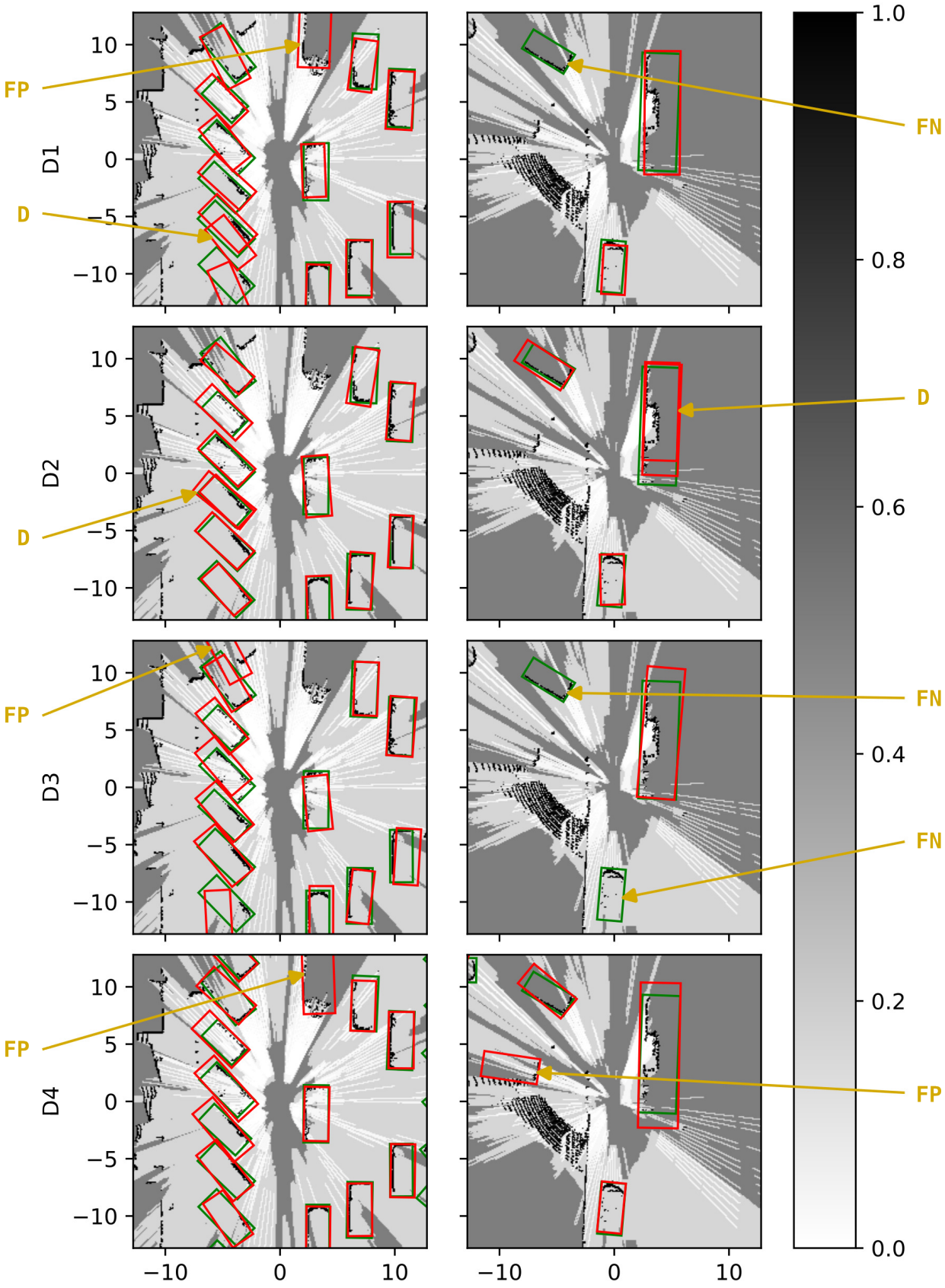

Figure 4.8. : Exemples de prédictions. Les boîtes englobantes vertes sont les vérités terrain, les boîtes englobantes rouges sont les prédictions des détecteurs. Les probabilités d'occupation sont encodées en niveaux de gris. Les faux positifs (FP), faux négatifs (FN) et doublons (D) sont indiqués avec des flèches dorées.

Les résultats obtenus sur ces détecteurs permettent d'affirmer que l'usage de réseaux simples issus de l'état de l'art en détection d'objets sur images tels que celui de **D2** inspiré de YOLOv2 [RF17] permet de détecter des véhicules sur des grilles d'occupation avec des métriques de précision moyenne élevées et des temps d'inférence faibles. Par comparaison, une architecture de détection spécifiquement conçue pour la navigation autonome telle que celle de PIXOR [YLU18] procure des résultats très similaires quantitativement qui s'expliquent par une plus grande propension à produire des faux positifs malgré un avantage en termes de rappel maximal atteignable.

Concernant les temps d'exécution totaux, **D2** est capable de produire des détections en seulement 11 millisecondes. Ces résultats sont cependant à nuancer au vu de la configuration de l'ordinateur ayant effectué les calculs. Les ordinateurs embarqués utilisés en robotique mobile sont moins puissants et risquent donc de conduire à des temps d'inférence supérieurs.

4.4 Conclusion

La contribution présentée dans ce chapitre propose une méthodologie de reconnaissance de véhicules sur grilles d'occupation par l'implémentation de réseaux de neurones mono-étape inspirés de la littérature. Parmi les modèles implémentés, deux catégories se dégagent :

— Les modèles **D1**, **D2** et **D3** sont inspirés des modèles de détection d'objet sur images YOLOv2 [RF17] et YOLOv3 [RF18]. Ces derniers ont été modifiés pour en diminuer le nombre de poids par la réduction du nombre de filtres de convolution présents dans chaque couche. La sortie de la dernière de ces couches a par ailleurs été modifiée pour prédire un angle d'orientation pour chaque boîte englobante détectée.

— Le modèle **D4** est quant à lui issu de la méthode PIXOR [YLU18] de détection d'objets dans des nuages de points LiDAR.

Les prédictions obtenues sur diverses grilles d'occupation démontrent la validité de cette approche. Les résultats suivants sont à noter :

— Le réseau **D4** issu du modèle PIXOR spécifiquement développé pour une application de perception automobile propose les meilleures boîtes englobantes avec des métriques de précision moyenne $\mathbf{AP}_{0.7}$ de 0.82 et $\mathbf{AP}_{0.5}$ de 0.90.

— Les réseaux **D1** et **D3** inspirés de YOLOv2 et YOLOv3 proposent des boîtes englobantes de véhicules de moindre qualité. Cette qualité de boîte englobante est évaluée par des métriques de précision moyenne $\mathbf{AP}_{0.7}$ n'excédant pas 0.55 et $\mathbf{AP}_{0.5}$ valant 0.82 pour les deux détecteurs.

— Le réseau **D2**, identique à **D1** à la différence que la matrice de sortie a été modifiée pour produire au maximum non pas 8×8 détections par grille d'occupation mais 16×16, produit des résultats similaires bien que légèrement inférieurs à ceux de **D4** avec des métriques de précision moyenne $\mathbf{AP}_{0.7}$ de 0.79 et $\mathbf{AP}_{0.5}$ de 0.89.

— En dépit de sa conception pour la perception automobile, **D4** a un temps d'exécution total de 50ms sur notre ordinateur portable de test équipé d'un GPU dédié. Ce temps d'exécution par grille est incompatible avec un objectif d'exécution en temps réel à 10Hz.

— En revanche, le réseau **D1** et sa version modifiée **D2**, permettant d'atteindre des métriques de précision moyenne similaires à **D4**, offrent pour principal avantage d'avoir des temps d'exécution par grille de seulement 11ms soit presque cinq fois plus faibles que le réseau **D4**.

— Finalement, la modification apportée au réseau **D1** pour en faire le réseau **D2** n'a pas été expérimentée sur le réseau **D3** inspiré de YOLOv3 car ce dernier propose des métriques de précision moyenne plus faibles que **D1** pour un temps d'exécution supérieur (35ms par grille d'occupation).

Ces résultats permettent d'établir l'architecture **D2** comme un possible candidat à la détection de véhicules sur grilles d'occupation dans un contexte embarqué temps réel. Néanmoins, les temps d'inférence sont ici obtenus à l'aide d'un ordinateur de bureau équipé d'un GPU. Dans des conditions de navigation autonome, la prédiction de boîtes englobantes en temps réel peut devoir être exécutée sur du matériel ne disposant pas de ressources de calcul suffisantes. Pour cette raison, le chapitre suivant présente une approche différente de segmentation de véhicules permettant de réduire le coût calculatoire pour mieux l'adapter à l'application visée de calcul sur processeur embarqué.

5

Segmentation de véhicules sur grilles d'occupation compressées à l'aide de réseaux de neurones denses

La présente contribution a pour objectif de proposer une nouvelle approche de traitement des grilles d'occupation, plus efficace en termes de coût de calcul. Pour ce faire, l'idée générale exploitée par les méthodes proposées dans ce chapitre sera l'utilisation d'un prétraitement pour extraire les informations essentielles d'une grille d'occupation avant le traitement de ces informations par un réseau de neurones afin d'identifier les véhicules présents dans l'environnement. Ce prétraitement d'extraction d'information repose sur la transformée en cosinus discrète [Bra00] permettant de projeter les grilles d'occupation dans le domaine spectral et par la suite la réduction de la dimension de ces spectres par l'usage de masques éliminant les hautes fréquences.

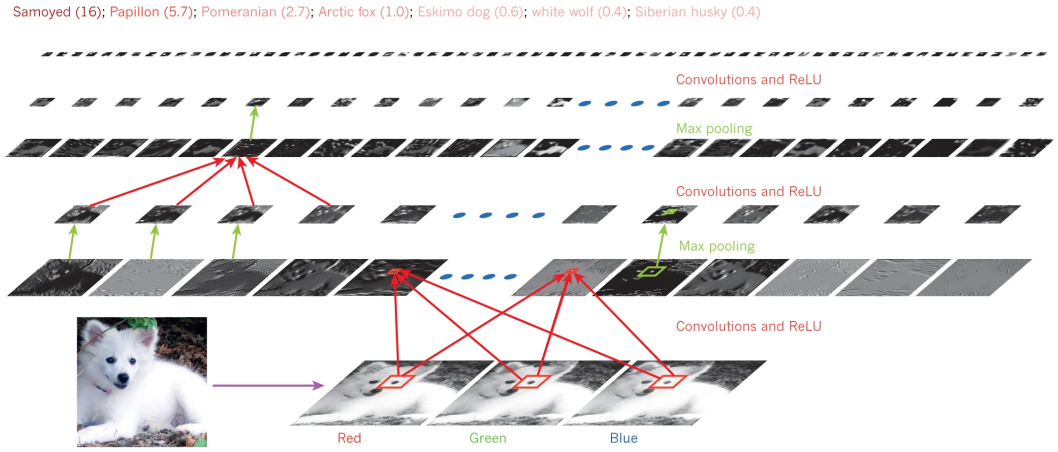

Figure 5.1. : Exemple de sorties des différentes couches de convolution composant un réseau de classification d'images sur un exemple d'image de chien. Cette illustration est issue de [LBH15].

L'entraînement d'un réseau de neurones équivaut à l'élaboration d'une fonction permettant d'identifier, d'extraire, de transformer les motifs pertinents dans les données d'entrée pour la réalisation de la tâche attendue. L'aspect modulaire des réseaux de

neurones composés d'un assemblage de multiples briques de base, appelées couches, permet de décomposer la fonction totale effectuée par le réseau de neurones en un graphe de fonctions plus spécialisées dont les entrées sont les sorties des couches précédentes dans le graphe (comme expliqué précédemment à la section 2.2). Par exemple, les couches de convolution utilisées en traitement d'images sont composées d'un empilement de convolutions appelées "filtres" et dont la visualisation des poids appris permet d'identifier le type de motif détecté [LBH15]. La figure 5.1 donne un exemple de sorties d'un réseau de neurones convolutif de classification d'images.

Il a été observé [LBH15] que les premiers filtres issus des premières couches de convolution des réseaux de neurones de traitement d'images détectent généralement les mêmes motifs indépendamment de la tâche finale accomplie par le réseau de neurones. Ces couches de convolution peuvent donc être vues comme accomplissant un prétraitement des données ayant pour but de représenter l'image initiale sous la forme d'un ensemble de motifs la composant.

Un parallèle peut ainsi être fait entre les opérations effectuées par ces premières couches et les algorithmes de compression d'images classiquement utilisés pour réduire l'espace nécessaire au stockage de fichiers d'images. Les algorithmes de compression classiques reposent justement sur la reconnaissance de motifs visuels dans une image permettant de reconstruire une image qui sera perçue comme identique à l'originale [ISO94 ; ISO19]. Ces algorithmes qui exploitent différentes transformations mathématiques de l'image sont optimisés pour nécessiter peu de ressources matérielles. Par ailleurs, en raison de leur utilisation très fréquente, les puces informatiques modernes embarquent généralement du matériel spécialisé rapide et peu consommateur de ressources pour exécuter ces algorithmes.

En raison du caractère embarqué des algorithmes utilisés dans le contexte de la navigation autonome, il peut être avantageux de remplacer les coûteuses couches de convolution en début de réseau de neurones par une méthode de compression optimisée. Dans ce contexte, les réseaux de neurones développés auraient comme entrée non pas une image au format matriciel, mais des données compressées issues de l'image originale.

Une première partie de ce chapitre traite de la méthode permettant d'extraire des informations des grilles d'occupation en une nouvelle représentation spectrale de faible dimension. La deuxième partie de ce chapitre explicite l'entraînement de modèles d'apprentissage profond utilisant ces représentations spectrales des grilles d'occupation pour segmenter les véhicules présents sur ces dernières.

5.1 Méthode de prétraitement des grilles d'occupation

Le prétraitement des grilles d'occupation a pour objectif de remplacer les coûteuses couches de convolution des méthodes présentées au chapitre 4 par des algorithmes de traitement d'images peu coûteux et capables de résumer l'information contenue dans les grilles d'occupation en un plus petit nombre de dimensions. Pour ces raisons, les transformations en cosinus discrètes (DCT) et en ondelettes discrètes (DWT) qui sont respectivement au cœur des formats de compression JPEG [ISO94] et JPEG2000 [ISO19] ont été considérées. Les bases théoriques derrière ces deux transformations sont résumées à la section 2.3.

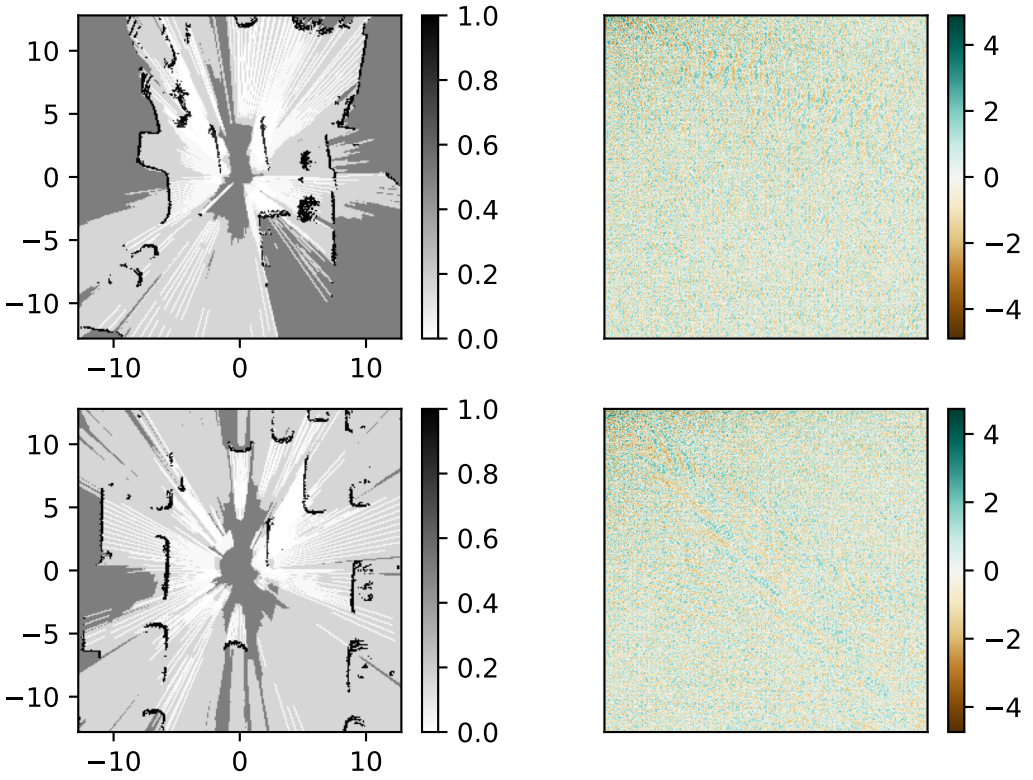

Figure 5.2. : Grilles d'occupation et spectre associé obtenu par la transformation en cosinus discrète. Les valeurs du spectre sont représentées par leur logarithme en base 10. La fréquence augmente depuis le coin supérieur gauche du spectre jusqu'au coin inférieur droit.

La transformée en cosinus discrète permet de passer du domaine spatial au domaine fréquentiel alors que la transformée en ondelettes discrète permet de conserver la localisation de différents motifs dans l'image via la convolution de l'ondelette choisie avec l'image. La transformée en ondelettes discrète peut ainsi être imitée par une couche de convolution avec un nombre de filtres égal au nombre de dilatations de l'ondelette mère considérée. Pour ces raisons, cette partie se concentre uniquement

sur la transformation en cosinus discrète qui ne peut pas être approchée par une couche de convolution.

5.1.1 Transformation en cosinus discrète appliquée aux grilles d'occupation

Une grille d'occupation $(x_{i,j})_{0\leq i,j\leq 255}$ issue du jeu de données présenté au chapitre 3 peut être transformée en un spectre $(y_{k,l})_{0\leq k,l\leq 255}$ de même taille par l'application de la transformée en cosinus discrète rapide [Mak80] :

$$y_{k,l} = 4\sum_{i=0}^{255}\sum_{j=0}^{255} x_{i,j}\cos(\frac{\pi k(4i+1)}{512})\cos(\frac{\pi l(4j+1)}{512})$$

Le spectre obtenu (des exemples sont donnés à la figure 5.2) contient l'amplitude de chaque fréquence sur deux dimensions. Bien que cette représentation duale soit de même taille que la grille d'occupation originale, elle présente pour principal avantage que les basses fréquences (situées en haut à gauche des spectres représentés à la figure 5.2) contiennent la plupart de l'information issue de la grille d'occupation originale.

5.1.2 Réduction de la dimensionnalité des spectres de grilles d'occupation

Les réseaux de neurones présentés au chapitre 4 sont composés de couches de convolution successives. L'usage de ce type de couches est justifié par le fait que les grilles d'occupation utilisées en entrée de ces réseaux sont une représentation de l'occupation de l'espace dans le domaine spatial. Ainsi, elles possèdent pour propriété la localité de l'information qui se traduit par le fait que deux cellules adjacentes dans la grille d'occupation sont plus susceptibles de correspondre au même objet physique, par exemple une voiture, que deux cellules éloignées dans la grille d'occupation. Cette propriété de localité de l'information justifie l'analyse des grilles d'occupation par un procédé de fenêtre glissante tel que l'opérateur de convolution.

Perte de localité du spectre et usage de couches denses

Le spectre d'une grille d'occupation obtenu par application de la transformée en cosinus discrète est, à la différence des grilles d'occupation, une représentation dans le domaine fréquentiel et non pas dans le domaine spatial. Ce changement de domaine implique la perte de la propriété de localité de l'information. La figure 5.3 illustre ce phénomène par l'exemple d'une grille d'occupation (synthétique dans ce cas) dont on enlève un véhicule. Ce changement localisé de la grille d'occupation amène à un changement global sur le spectre de la grille d'occupation qui touche l'ensemble des coefficients.

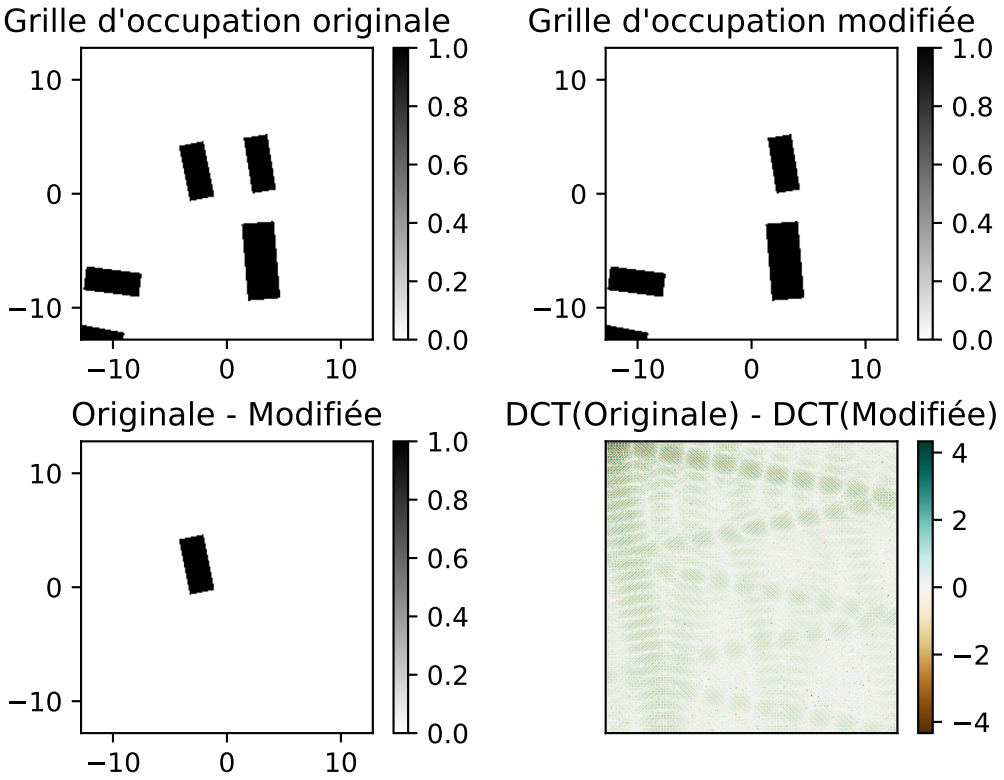

Figure 5.3. : Exemple de l'absence de propriété de localité dans le domaine fréquentiel. En haut à gauche : Une grille d'occupation synthétique contenant cinq véhicules. En haut à droite : La même grille d'occupation dont on a enlevé un véhicule. En bas à gauche : La différence entre la grille d'occupation originale et la grille d'occupation modifiée dans le domaine spatial. En bas à droite : La différence entre la grille d'occupation originale et la grille d'occupation modifiée dans le domaine fréquentiel. Contrairement au domaine spatial, la modification d'une portion restreinte de l'environnement ici matérialisée par la suppression d'un véhicule mène à une modification de l'ensemble des coefficients du spectre dans le domaine fréquentiel.

L'absence de propriété de localité dans le domaine fréquentiel semble rendre tout traitement ultérieur du spectre par des méthodes à convolution peu judicieuses. Les

réseaux de neurones présentés plus loin qui exploitent directement ces spectres sont donc composés de couches denses. Or, les couches denses ont un nombre de poids entraînables et donc un coût de stockage mémoire qui augmente linéairement avec la taille de leur entrée (Propriété 1). Par opposition, les couches de convolution ont un nombre de poids entraînables qui ne dépend pas de la taille de leur entrée (Propriété 2). Pour ces raisons, la taille des spectres traités doit être réduite au maximum pour s'assurer une faible empreinte mémoire des réseaux denses utilisés.

Propriété 1 (Nombre de poids d'une couche dense) *Soit $\mathcal{D}$ une couche dense à x entrées et y sorties. Alors le nombre de poids de la couche $\mathcal{D}$ est :*

$$\#\mathcal{D} = (x + 1)y$$

Propriété 2 (Nombre de poids d'une couche de convolution 2D) *Soit $\mathcal{C}$ une couche de convolution 2D à x entrées, f filtres de noyau de taille $w \times h$. Alors le nombre de poids de la couche $\mathcal{C}$ est :*

$$\#\mathcal{C} = (wh + 1)f$$

Réduction de la dimensionnalité des spectres de grille d'occupation

L'usage majoritaire de la transformée en cosinus discrète en informatique est la compression de fichiers images. En particulier, le format de compression JPEG [ISO94] exploite cette transformée pour séparer les basses fréquences des hautes fréquences. Par la suite, les hautes fréquences, considérées généralement comme du bruit de capteur, sont éliminées du spectre ce qui permet la compression effective des données.

L'approche considérée dans ce chapitre est similaire à l'algorithme utilisé par le format JPEG. L'objectif étant ici d'évaluer la manière la plus efficace de réduire la dimension effective des spectres de grilles d'occupation tout en conservant le maximum d'informations sur la grille d'occupation originale. Initialement, le spectre d'une grille d'occupation obtenu par transformée en cosinus discrète est de même dimension que la grille d'occupation, soit $256 \times 256 = 65536$ dimensions dans le cas des grilles d'occupation considérées dans ce manuscrit. La réduction de dimension peut être effectuée par l'application d'un masque sur le spectre original, mettant à zéro les coefficients de toutes les fréquences masquées.

La quantité d'information conservée dans le spectre masqué peut être facilement évaluée par la reconstruction d'une grille d'occupation à partir du spectre masqué à l'aide de la transformée en cosinus discrète inverse. Cette grille reconstruite peut ensuite être comparée à la grille d'occupation originale pour évaluer l'erreur de reconstruction due au masquage de certaines fréquences. Pour ces tests, la métrique de comparaison choisie entre grille d'occupation originale et grille d'occupation reconstruite est l'erreur quadratique moyenne.

Il est à noter que l'objectif de ces travaux est d'exploiter l'algorithme de transformée en cosinus discrète pour extraire des grilles d'occupation une représentation de faible dimension, obtenue à faible coût calculatoire, pour pouvoir ensuite l'utiliser en entrée de divers modèles d'apprentissage profond. Par conséquent, la reconstruction de grilles d'occupation en tant que telle ne présente pas d'intérêt autre que celui d'évaluer la quantité d'information perdue avec chaque masque utilisé.

Le format de compression JPEG exploite l'hypothèse que les informations les plus importantes pour la compression d'une image sont les coefficients des basses fréquences. Deux types de masque évalués dans ce chapitre reposent sur cette hypothèse en ne sélectionnant que les plus basses fréquences du spectre. Étant donné que les grilles d'occupation ne sont pas des photographies usuelles, un dernier type de masque appris à partir du jeu de données d'entraînement est proposé. La procédure permettant d'obtenir ces masques n'implique pas de sélectionner les basses fréquences du spectre. Elle est donc robuste au cas spécifique où l'hypothèse faite par le format JPEG se révèlerait fausse dans le cas des grilles d'occupation.

Trois familles de masques pouvant se décliner en divers masques en fonction de la dimension d résultante souhaitée ont été évaluées :

— Les masques carrés sont définis par :

$$m_{k,l} = 0 \iff k > c \vee l > c$$

avec c obtenu par la formule :

$$c = \lfloor \sqrt{d} \rfloor$$

— Les masques triangulaires sont définis par :

$$m_{k,l} = 0 \iff k + l - 1 > c$$

avec c obtenu par la formule :

$$c = \lfloor \sqrt{2d + 0.25} - 0.5 \rfloor$$

— Les masques appris sont définis à partir des grilles d'occupation contenues dans le jeu de données d'entraînement. Dans un premier temps, les spectres $y_{k,l}$ de chaque grille d'occupation sont calculés. Ensuite, la contribution de chaque coefficient à chaque grille d'occupation est défini comme :

$$\hat{y}_{k,l} = \frac{|y_{k,l}|}{\sum_{i,j=1}^{256} |y_{i,j}|}$$

Ensuite, ces contributions individuelles de chaque fréquence sont moyennées sur l'ensemble des grilles d'occupation du jeu de données. Finalement, pour une dimension de sortie d choisie, les d fréquences ayant la contribution la plus élevée sont conservées dans le masque tandis que les coefficients des autres fréquences sont mis à zéro. Une procédure de normalisation appelée normalisation par *contribution*, décrite à la sous-section 5.2.1, est appliquée sur chaque spectre de grille d'occupation avant le calcul de la contribution. Cette procédure permet de s'assurer que chaque coefficient représente effectivement la contribution de la fréquence à la grille d'occupation.

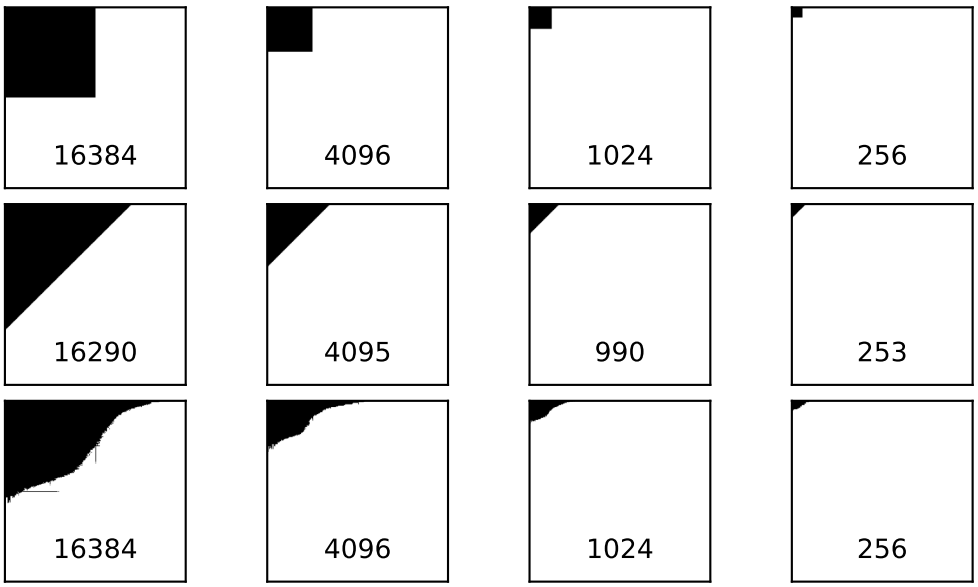

Figure 5.4. : Masques utilisés pour mettre à zéro certains coefficients du spectre. De haut en bas : Masques carrés ; Masques triangulaires ; Masques appris. Le nombre de coefficients conservés sur le spectre est indiqué dans chaque masque.

La figure 5.4 présente la forme des trois types de masques utilisés pour différentes valeurs de dimension souhaitées. On peut y observer que le nombre de dimensions

des masques triangulaires est généralement légèrement inférieur au nombre de dimensions des masques carrés et appris. Ce phénomène est dû au nombre de dimensions des masques triangulaires de la forme $d = c(c+1)/2$ qui restreint d à certaines valeurs possibles.

Les masques appris ont une forme particulière pouvant être interprétée. En effet, sur les spectres de grilles d'occupation, un déplacement vers la droite induit une fréquence horizontale plus élevée alors qu'un déplacement vers le bas induit une fréquence verticale plus élevée.

Le fait que ces masques appris soient plus étendus vers la droite que vers le bas peut s'expliquer par le fait qu'il existe plus de lignes verticales que de lignes horizontales sur les grilles d'occupation, nécessitant de plus hautes fréquences horizontalement que verticalement. Ce phénomène en particulier est dû à la présence d'une majorité de véhicules et de murs orientés parallèlement au véhicule autonome et donc verticalement sur les grilles d'occupation dans la majorité des situations de navigation. Cette particularité du jeu de données est visible dans l'histogramme polaire d'orientation des véhicules de la figure 3.4.

Une autre particularité des masques appris est que les coefficients des fréquences conservées sont un ensemble connexe de basses fréquences, similaire aux filtres carrés et triangulaires. L'algorithme de création de ces masques appris n'impose pourtant pas de contrainte particulière sur la connexité des fréquences conservées. Il semble donc que l'hypothèse de plus grande importance des basses fréquences que des hautes fréquences faite par le format de compression JPEG pour la compression des images est aussi vraie dans le cas des grilles d'occupation.

La figure 5.5 présente le spectre d'une grille d'occupation masqué à l'aide des trois types de masques, ainsi que les grilles d'occupation reconstruites, avec une dimension maximale choisie $d = 4096$.

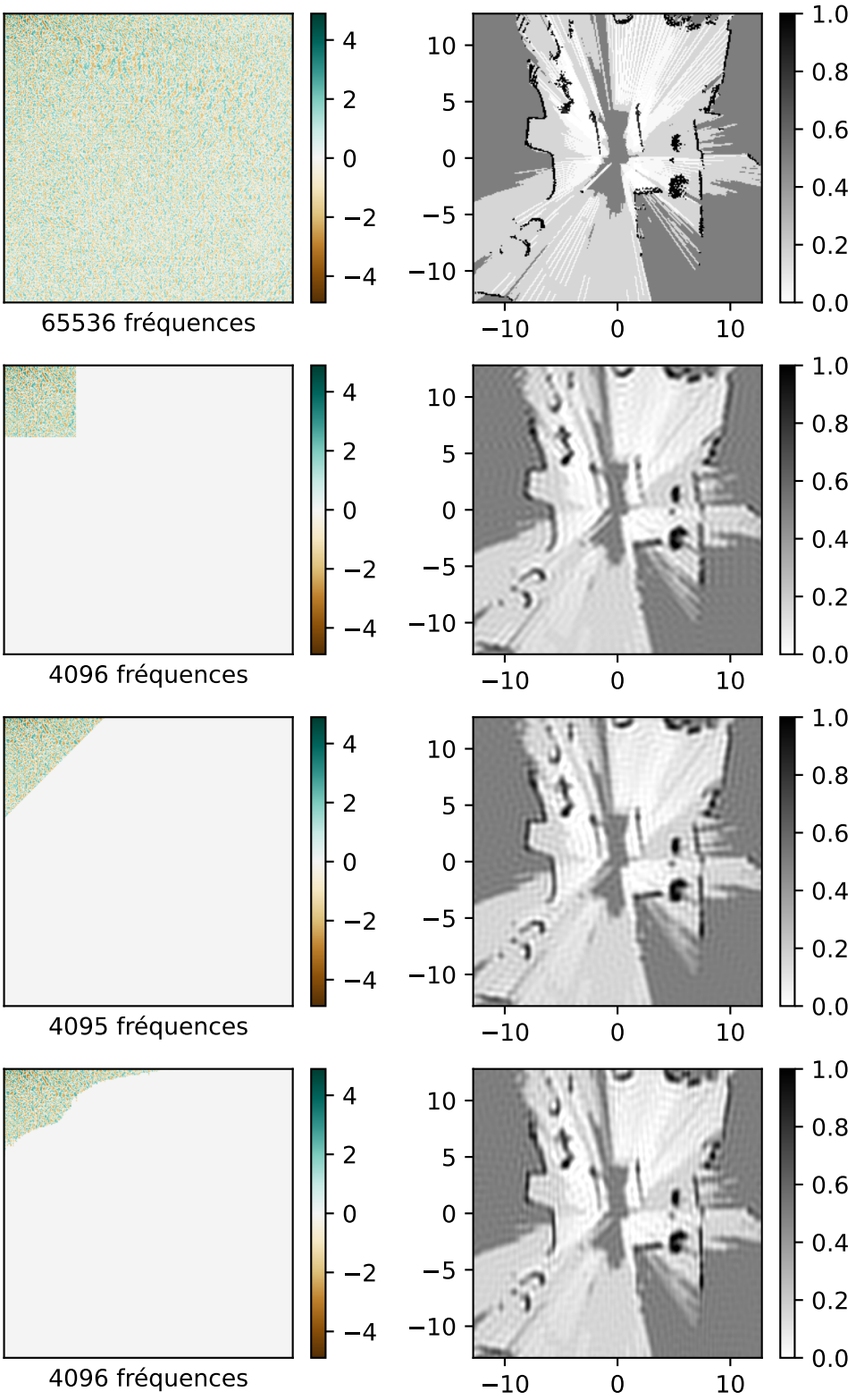

Figure 5.5. : Exemple de spectres et de reconstructions d'une grille d'occupation. De haut en bas : Le spectre complet ; Le spectre masqué par un masque carré conservant 4096 coefficients ; Le spectre masqué par un masque triangulaire conservant 4095 coefficients ; Le spectre masqué par un masque appris conservant 4096 coefficients.

Évaluation des différentes méthodes de masquage de spectre

Les reconstructions obtenues par l'intermédiaire des trois masques sont assez similaires entre elles et permettent encore d'identifier visuellement les véhicules présents sur la grille d'occupation malgré une dégradation perceptible de la qualité de cette dernière par rapport à l'originale. La possibilité d'identifier sur les reconstructions les différents obstacles composant la scène malgré un facteur de réduction de dimension de 16 illustre bien l'excellente capacité de compression de la transformée en cosinus discrète pour les grilles d'occupation. Visuellement, la différence entre les reconstructions obtenues à l'aide du masque carré et du masque triangulaire sont difficiles à discerner. En revanche, la reconstruction obtenue par l'usage du masque appris se reconnaît par la plus grande finesse des lignes horizontales obtenues dans la reconstruction, illustrant bien le biais de ce masque vers les hautes fréquences horizontales discuté ci-avant.

Le tableau 5.1 présente les erreurs quadratiques moyennes obtenues entre reconstructions et grilles d'occupation originales pour les trois types de masques différents et différentes valeurs de dimension du masque souhaitées. Les résultats obtenus permettent de classer les masques appris, les masques triangulaires puis les masques carrés de ceux offrant les meilleures reconstructions à ceux offrant les plus mauvaises reconstructions. Le bénéfice apporté par le masque appris par rapport aux masques carrés et triangulaires est en particulier plus marqué pour les dimensions de masque plus réduites, inférieures à 256. Au-delà de 256 dimensions, le masque appris offre toujours une meilleure reconstruction que les masques carrés et triangulaires, mais le bénéfice en termes de diminution de l'erreur de reconstruction est plus faible.

Masque	64d	128d	256d	512d	1024d	2048d	4096d	8192d	16384d
Carré	0.0233	0.0207	0.0182	0.0163	0.0142	0.0122	0.0101	0.0079	0.0054
Triangulaire	0.0229	0.0202	0.0179	0.0159	0.0139	0.0119	0.0098	0.0076	0.0052
Appris	**0.0217**	**0.0196**	**0.0175**	**0.0156**	**0.0136**	**0.0116**	**0.0096**	**0.0074**	**0.0051**

Tableau 5.1. : Erreur quadratique moyenne de reconstruction pour les masques carrés, triangulaires et appris avec des valeurs de dimensions du masque de 64 (55), 128 (120), 256 (253), 512 (496), 1024 (990), 2048 (2016), 4096 (4095), 8192 (8128) et 16384 (16290). Les valeurs entre parenthèses s'appliquent aux masques triangulaires. Ces erreurs ont été calculées sur le jeu de données d'entraînement.

Au vu de ces résultats, la méthode de masquage des fréquences utilisée dans la section suivante qui traite de l'entraînement de modèles de segmentation de véhicules

sera la méthode de masquage par masques appris. Les représentations spectrales masquées obtenues seront directement utilisées en entrée des modèles de segmentation.

5.2 Segmentation de véhicules à partir de spectres masqués de grilles d'occupation

La section précédente a mis en évidence la possibilité de représenter une grille d'occupation sous la forme d'un spectre, pouvant être tronqué à l'aide de masques de fréquences pour obtenir une représentation de l'occupation de l'espace qui soit de la dimension réduite souhaitée. Cette réduction de dimension se fait au prix d'une perte d'information qui peut être quantifiée en comparant la grille d'occupation reconstruite à partir du spectre tronqué à la grille d'occupation originale. L'objectif de cette section est de montrer le potentiel de cette nouvelle représentation de l'occupation de l'espace pour effectuer des tâches de perception, et en particulier de segmentation de véhicules, dans le contexte général de la navigation autonome.

L'état de l'art présenté à la section 2.3.2 présente les différents travaux identifiés dans le domaine particulier des tâches de perception à partir de données spectrales d'une image. Parmi ces travaux, un certain nombre [Tan+15 ; Zhu+23] exploitent la transformée en ondelettes, fondamentalement différente de la transformée en cosinus discrète car projetant dans un domaine particulier espace-fréquence (ou plus communément appelé temps-fréquence dans le cas unidimensionnel). Un grand nombre de travaux exploitant uniquement le domaine fréquentiel, par l'usage de transformations telles que la transformée de Fourier discrète ou la transformée en cosinus discrète, proposent des méthodes de traitement d'images ne rentrant pas dans le cadre de la perception, par exemple des méthodes de génération d'images [Cai+21] ou de reconnaissance d'images modifiées [Zhu+23].

Finalement, l'étude bibliographique menée n'a amené qu'à un très faible nombre de travaux ayant un rapport direct avec l'objectif des travaux présentés dans ce chapitre. Parmi ces derniers, deux articles [Zha+21 ; KS22] proposent d'utiliser le spectre d'une image obtenu par transformée de Fourier discrète pour identifier des fissures dans des ponts sur des images satellites. L'utilisation d'images satellites, similaires en termes de point de vue à des grilles d'occupation, ainsi que l'entraînement d'un modèle d'apprentissage profond à partir des spectres de ces images sont des caractéristiques de ces travaux assez similaires à l'approche recherchée. Néanmoins, les deux méthodes citées proposent des architectures convolutives unidimensionnelles

entraînées sur une mise à plat unidimensionnelle du spectre des images satellites utilisées. La perte de propriété de localité du domaine fréquentiel abordée à la sous-section 5.1.2 nous amène à penser qu'une approche à base de couches de convolution sur un spectre n'est pas justifiée. Pour cette raison, cette section aborde l'usage d'architectures à base de couches denses qui ne supposent pas de propriété de localité dans les données d'entrée. De tels réseaux denses utilisés pour des tâches de perception dans le domaine fréquentiel n'ayant pas été observés dans la littérature, les architectures présentées ici ne s'inspirent d'aucune d'architecture existante et constituent donc une contribution originale.

L'usage de couches denses dans un réseau de neurones amène rapidement à un coût calculatoire et mémoire important peu compatible avec les objectifs de légèreté des calculs inhérents à la navigation autonome sur plateforme embarquée. L'objectif principal de ces travaux va donc être la constitution d'architectures comportant le moins de couches denses possibles, mais néanmoins capables d'exploiter les spectres tronqués de grilles d'occupation pour identifier les véhicules présents dans la scène.

Comme dit précédemment, les entrées des réseaux de neurones présentés dans ce chapitre sont des spectres des grilles d'occupation tronqués à l'aide des filtres appris présentés à la section 5.1. Chaque spectre est mis à plat en un vecteur de coefficients unidimensionnel. L'ordre dans lequel les différents coefficients sont ordonnés dans le vecteur unidimensionnel n'a pas d'importance, étant donné que celui-ci sera utilisé en entrée d'une couche dense qui ne fait pas de supposition quant à l'ordre des caractéristiques reçues en entrée.

L'objectif étant la reconnaissance des autres véhicules présents dans la scène, les sorties des réseaux de neurones doivent encoder cette information. Dans le cas des réseaux convolutifs présentés au chapitre 4, les sorties encodaient directement les coordonnées de boîtes englobantes telles que la position, les dimensions et l'orientation de ces dernières. Pour des raisons d'uniformité du domaine de travail des réseaux de neurones, les sorties doivent ici encoder les positions des véhicules dans le domaine fréquentiel. Il est donc décidé pour cette partie d'encoder les véhicules sous la forme d'une segmentation de la grille d'occupation générée à partir des boîtes englobantes vérités terrain. Cette segmentation est ensuite projetée dans le domaine fréquentiel de la même manière que les grilles d'occupation et le réseau de neurones est entraîné à reproduire ces spectres de segmentation de grille d'occupation en sortie à partir des spectres de grilles d'occupation en entrée. La figure 5.6 illustre le fonctionnement de cette méthode de reconnaissance.

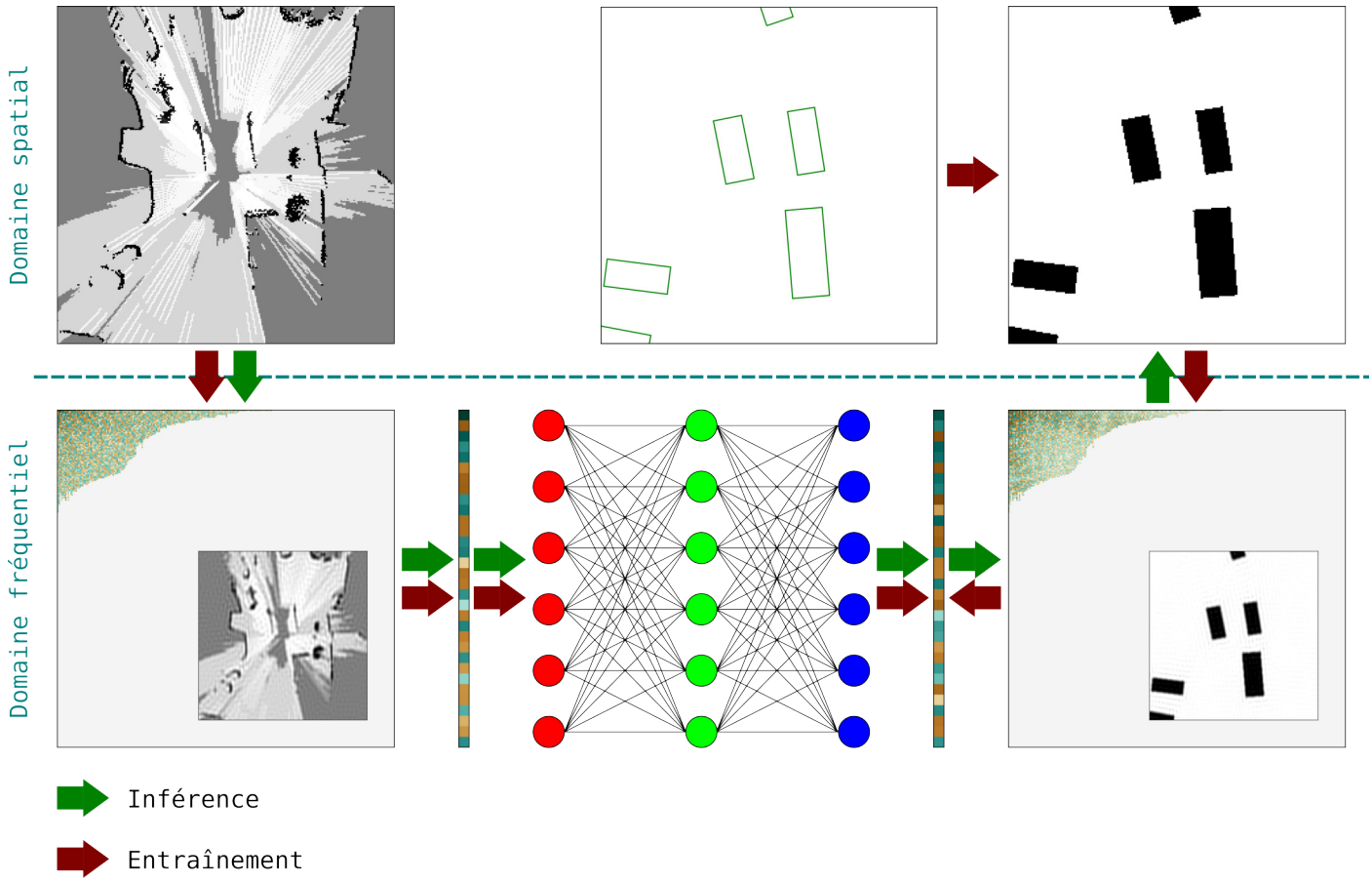

Figure 5.6. : Fonctionnement du système de reconnaissance de véhicules dans le domaine fréquentiel. Durant l'entraînement, la grille d'occupation est transformée dans le domaine spectral à l'aide de la transformée en cosinus discrète, puis le spectre est masqué à l'aide du masque appris sur les données d'entraînement et mis à plat en un vecteur unidimensionnel. Dans le même temps, les boîtes englobantes vérité terrain sont utilisées pour produire une segmentation de la grille d'occupation qui est ensuite transformée en vecteur unidimensionnel de coefficients de fréquence par le même procédé que la grille d'occupation. Le spectre de la grille d'occupation passe dans le réseau de neurones et le résultat est comparé avec le spectre de la segmentation pour calculer le gradient et mettre à jour les poids. Durant l'inférence, le spectre résultant du réseau de neurones est décodé en une segmentation de la grille d'occupation. Un exemple de ce que donnerait une reconstruction du spectre tronqué est présenté dans chaque spectre pour illustrer la perte d'informations due au masquage des spectres.

5.2.1 Méthodologie de recherche d'architecture

Le processus de recherche d'architecture vise à comprendre l'interaction des différents choix architecturaux entre eux et leur influence sur les résultats obtenus. Afin

de sélectionner les meilleurs paramètres, le nombre de coefficients de spectre conservés a été fixé à 2048 et différentes architectures ont été élaborées pour détecter des véhicules sur ces spectres. Par la suite, le choix du nombre de coefficients de spectre conservés sera relaxé et des modèles exploitant d'autres tailles de représentation seront présentés. Chaque architecture est donc composée d'un certain nombre de couches denses de tailles d'entrée et de sortie égales à 2048. Ce choix de ne pas faire augmenter la taille des vecteurs de caractéristiques intermédiaires est justifié par l'aspect embarqué de la navigation autonome qui a lui-même mené à la compression des grilles d'occupation en une représentation spectrale. La métrique permettant d'évaluer chaque architecture est le calcul de l'erreur quadratique moyenne entre la segmentation de grilles d'occupation obtenue à l'aide des vérités terrain et la segmentation obtenue à l'aide de la transformée en cosinus discrète inverse sur le spectre récupéré en sortie du réseau de neurones.

Le tableau 5.2 présente les différentes expériences réalisées. Les architectures des modèles évalués varient selon différents axes :

— Le nombre de couches denses composant le modèle varie entre 3 et 6. Chaque couche a 2048 entrées et 2048 sorties. Ces couches sont catégorisées en *couche de sortie* pour la dernière couche et en *couches internes* pour toutes les autres couches du modèle. Ces couches sont organisées séquentiellement dans le réseau donc la sortie d'une couche devient l'entrée de la suivante.

— Les fonctions d'activation utilisées en sortie des couches internes et de la couche de sortie peuvent être la fonction ReLU, la fonction LeakyReLU ou la fonction identité (équivalente à une absence de fonction d'activation). Ce choix limité de fonctions d'activation provient du très faible coût calculatoire de ces fonctions linéaires par morceaux.

— La présence de raccourcis (ou connexions sautées/résiduelles) locaux traduit le fait que l'entrée d'une couche est additionnée à sa sortie. La présence d'un raccourci global signifie que l'entrée du réseau de neurones est ajouté à sa sortie. Ces raccourcis permettent dans certains cas de faciliter l'entraînement de réseaux très profonds [He+16].

Différentes méthodes de normalisation des données d'entrée ont également été testées :

— L'absence de normalisation consiste à utiliser en entrée du réseau de neurones les coefficients du spectre de la grille d'occupation directement obtenus par application de la transformée en cosinus discrète.

— La normalisation par *maximum global* consiste en la division de l'ensemble des coefficients du spectre par le coefficient maximal en valeur absolue de

l'ensemble des spectres correspondant aux grilles d'occupation du jeu de données d'entraînement. Ce choix garantit que tous les coefficients utilisés en entrée sont dans l'intervalle $[-1, 1]$.

— La normalisation par *maximum par fréquence* consiste en la division de chaque coefficient du spectre par la valeur maximale en valeur absolue du coefficient de cette fréquence sur tous les spectres correspondant aux grilles d'occupation du jeu de données d'entraînement. Il s'agit de la même méthode que la normalisation par maximum global, mais avec un maximum par fréquence.

— La normalisation par *contribution* consiste en la multiplication de chaque coefficient $y_{k,l}$ par sa contribution unitaire $\alpha_{k,l}$ à la grille d'occupation obtenue par la formule suivante :

$$\alpha_{k,l} = \sum_{i,j=1}^{256} |\text{IDCT}((\delta_{i,j}^{k,l})_{1 \leq i,j \leq 256})|$$

$$\begin{aligned} \delta_{i,j}^{k,l} &= 1 \text{ si } k = i \wedge l = j \\ &= 0 \text{ sinon} \end{aligned}$$

$\text{IDCT}((\delta_{i,j}^{k,l})_{1 \leq i,j \leq 256})$ représente le vecteur de la base de fréquences associé à la fréquence de coordonnées (k, l) (un exemple d'une telle base est visible sur la figure 2.10). La somme de ce vecteur en valeur absolue est la contribution unitaire de cette fréquence au résultat de la DCT inverse. Le spectre obtenu par ce processus permet d'assurer que deux coefficients de même valeur dans le spectre auront la même quantité d'influence sur la grille d'occupation reconstruite par DCT inverse.

Finalement, deux techniques permettant d'aider l'entraînement des modèles sont évaluées :

— Des couches de décrochage (*dropout* en anglais) qui durant l'entraînement forcent à zéro un neurone sur huit de chaque couche interne du modèle sont présentes dans certains modèles. Ces couches sont désactivées durant l'inférence. Celles-ci permettent de limiter le surapprentissage en introduisant de l'instabilité dans les neurones ce qui les empêche d'encoder parfaitement les données d'apprentissage [Sri+14].

— L'usage d'un ordonnanceur de taux d'apprentissage basé sur la méthode de *Cosine Annealing* [LH17] permet de réduire graduellement le taux d'apprentissage afin de permettre une convergence rapide en début d'entraînement et une amélioration plus fine des résultats en fin d'entraînement.

Chacune des architectures est entraînée sur le jeu de données d'entraînement à l'aide de l'algorithme d'optimisation Adam [KB17], avec calcul de la fonction de coût sur le jeu de données de validation à la fin de chaque époque d'entraînement et arrêt anticipé de l'entraînement à partir du moment où cette dernière est supérieure aux précédentes. Par ailleurs, l'entraînement se fait sur des lots de 64 spectres à chaque itération. La figure 5.7 illustre une architecture à trois couches comportant des couches de décrochage et des connexions résiduelles locales et globales.

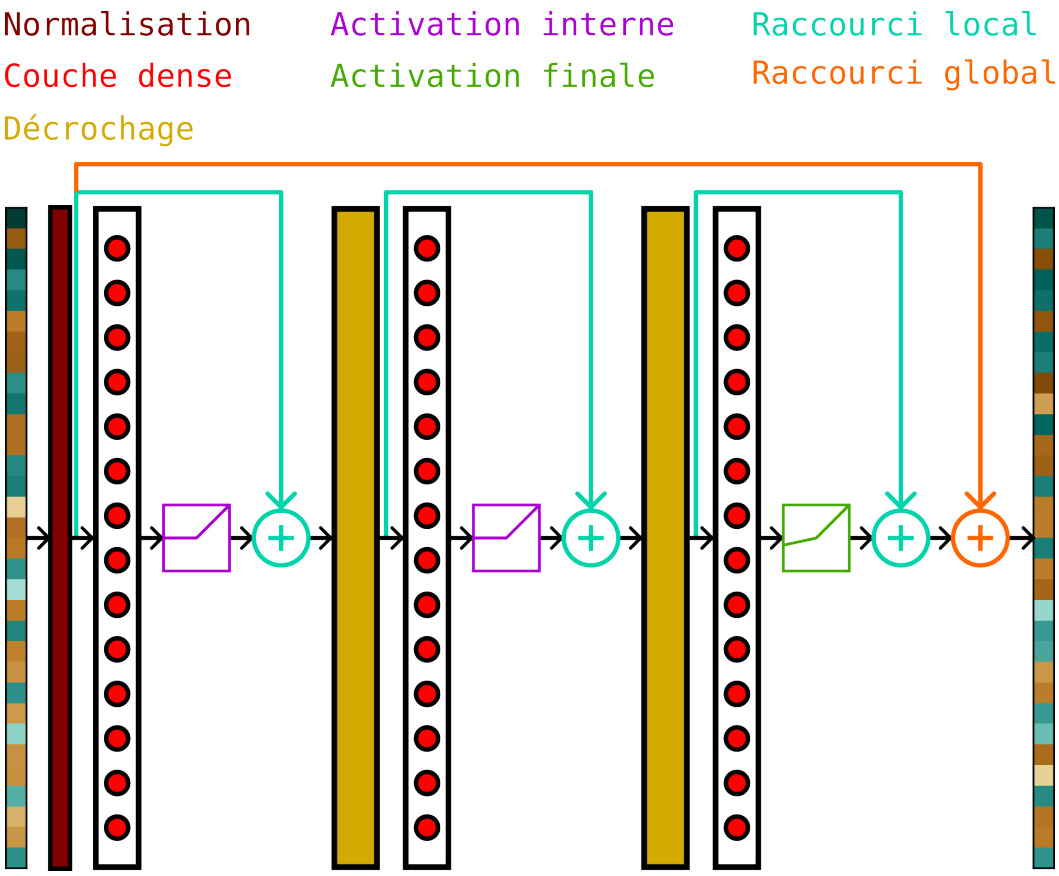

Figure 5.7. : Exemple d'une architecture à trois couches utilisée durant la recherche d'architecture.

Modèle	Architecture					Normalisation	Entraînement		MSE Test
	Couches	Activations		Raccourcis			Décrochage	Taux d'apprentissage	
		Internes	Sortie	Locaux	Global				
1	3	LeakyReLU	LeakyReLU	Oui	-	-	-	0.001	0.0354
2	4	LeakyReLU	LeakyReLU	Oui	-	-	-	0.001	0.0349
3	4	LeakyReLU	LeakyReLU	Oui	Oui	-	-	0.001	0.0368
4	3	LeakyReLU	LeakyReLU	Oui	-	Maximum global	-	0.001	0.0335
5	3	LeakyReLU	LeakyReLU	Oui	-	Maximum par fréquence	-	0.001	0.0377
6	3	LeakyReLU	LeakyReLU	-	-	Maximum global	-	0.001	0.0313
7	3	LeakyReLU	LeakyReLU	-	-	Contribution	-	0.001	0.0318
8	4	LeakyReLU	LeakyReLU	-	-	Contribution	-	0.001	0.0332
9	5	LeakyReLU	LeakyReLU	-	-	Contribution	-	0.001	0.0362
10	3	ReLU	LeakyReLU	Oui	-	Contribution	-	0.001	0.0358
11	3	ReLU	LeakyReLU	-	-	Contribution	-	0.001	0.0319
12	3	ReLU	-	-	-	Contribution	-	0.001	0.0272
13	4	LeakyReLU	-	Couches internes	-	Contribution	-	0.001	0.0267
14	5	LeakyReLU	-	Couches internes	-	Contribution	-	0.001	0.0264
15	6	LeakyReLU	-	Couches internes	-	Contribution	-	0.001	0.0263
16	6	LeakyReLU	-	Couches internes	-	Contribution	p=0.125	0.001 avec CA:(4e, lr>0)	0.0261
17	6	LeakyReLU	-	Couches internes	-	Contribution	p=0.125	0.001 avec CA:(6e, lr>1e-6)	**0.0259**

Tableau 5.2. : Expériences réalisées sur l'architecture d'un détecteur de véhicules sur spectre. La dimension du spectre choisie en entrée est 2048. Chaque couche dense des réseaux de neurones a 2048 entrées et 2048 sorties. Les couches sont considérées internes à l'exception de la dernière couche qui est considérée comme couche de sortie. Les raccourcis locaux sont un moyen d'accélérer l'entraînement en additionnant l'entrée d'une couche dense à sa sortie. Un raccourci global ajoute l'entrée du réseau à la sortie du réseau. La normalisation par maximum global est la division de l'ensemble du spectre d'entrée par le coefficient maximal en valeur absolue de toutes les fréquences de toutes les grilles d'occupation du jeu de données d'entraînement. La normalisation par maximum par fréquence réalise la même opération, mais indépendamment pour chaque fréquence. La normalisation par contribution multiplie chaque coefficient par la contribution de sa fréquence à la grille d'occupation mesurée comme l'intégrale de la valeur absolue de la grille d'occupation générée quand cette fréquence seule vaut 1. Les couches de décrochage sont insérées entre chaque paire de couches. Un taux d'apprentissage noté CA:(Ae, lr>B) signifie que l'ordonnanceur de taux d'apprentissage Cosine Annealing a été utilisé avec A le nombre d'époques pendant lesquelles le taux d'apprentissage décroit et B la limite inférieure du taux d'apprentissage. Tous ces modèles sont entraînés avec l'algorithme d'optimisation Adam, une procédure d'arrêt anticipé avec une patience égale à 1, et une taille de lot pour chaque itération de 64.

Différentes leçons sont tirées des résultats présentés dans le tableau 5.2 :

— La présence d'une fonction d'activation finale pénalise beaucoup les résultats, même dans le cas d'utilisation d'une fonction d'activation LeakyReLU dont l'image est l'ensemble des nombres réels. Ce résultat peut être observé par la différence de 0.005 points entre les modèles 11 et 12.

— L'augmentation du nombre de couches au-delà de trois a un effet faiblement positif dans le meilleur des cas, voire négatif en raison de difficultés d'entraînement, même avec usage de connexions résiduelles censées faciliter l'entraînement (voir les couples de modèles 1 et 2 ; 7, 8 et 9 ; ou 13, 14 et 15).

— L'utilisation de connexions résiduelles locales comme globales dégrade de manière importante les performances des modèles (voir les couples de modèles 4 et 6 ; 10 et 11).

— Les méthodes de normalisation par *maximum global* ou *contribution* aident l'apprentissage des réseaux, mais pas celle par *maximum par fréquence* qui dégrade les performances (voir les modèles 1, 4, 5 et 7).

— L'utilisation de couches de décrochage et d'un ordonnanceur de taux d'apprentissage ne permet pas d'éliminer le surapprentissage qui se manifeste dans la figure 5.8 par un écart important entre l'erreur d'entraînement et l'erreur de validation. Des tests préliminaires utilisant des couches de décrochage avec une plus grande probabilité $p \in [0.25, 0.5]$ de mise à zéro des neurones des couches denses ont provoqué des instabilités d'apprentissage trop importantes pour être utilisables.

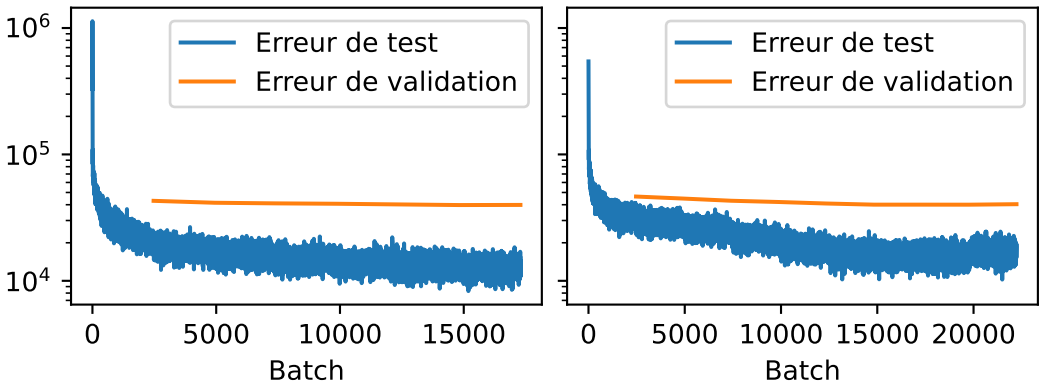

Figure 5.8. : Fonctions d'erreurs sur le jeu d'entraînement à chaque lot d'entraînement et sur le jeu de validation à la fin de chaque époque de modèles du tableau 5.2. Gauche : Modèle 15, Droite : Modèle 17. On y remarque que l'utilisation de couche de décrochage et d'un ordonnanceur de taux d'apprentissage pour le modèle 17 n'a pas permis de réduire l'écart entre l'erreur d'entraînement et l'erreur de validation.

5.2.2 Analyse de modèles de segmentation de véhicules

Les résultats obtenus dans la sous-section 5.2.1 permettent de sélectionner les paramètres d'architecture optimaux pour la conception d'un réseau de neurone de segmentation des véhicules sur spectre de grilles d'occupation. Dans ce qui suit, l'intérêt est porté à l'influence de la taille du spectre masqué considéré. Le tableau 5.3 donne les différentes métriques afférentes à différents modèles entraînés sur différentes dimensions de spectre. Tous les réseaux présentés comportent deux ou trois couches denses successives à d entrées et d sorties avec d la dimension du spectre d'entrée du réseau de neurones. Des réseaux à une seule couche dense sans fonction d'activation ont été entraînés, mais les résultats de segmentation obtenus visuellement montraient une incapacité totale de ces réseaux à segmenter une grille d'occupation. Ces mauvais résultats sont donc omis des comparatifs présentés ici.

Chacun des réseaux a des fonctions d'activation ReLU pour les couches internes et pas d'activation pour la couche finale. Les architectures ne comportent ni connexions résiduelles, ni couches de décrochage. La normalisation des spectres d'entrée est la méthode de *contribution*. Finalement, le taux d'apprentissage est fixé à 0.001 durant tout l'entraînement.

Comme montré sur la figure 5.6, le spectre produit par le réseau de neurones est utilisé pour reconstruire à l'aide de la transformée en cosinus discrète inverse une grille pouvant être utilisée pour segmenter les cellules de la grille d'occupation originale correspondant à des véhicules. Par ailleurs, le jeu de données de test donne aussi accès à des segmentations vérités terrain obtenues par projection des boîtes englobantes des véhicules sur une grille vide. Les différentes métriques de qualité des prédictions permettent de comparer ces deux grilles. La première métrique déjà utilisée dans le tableau 5.2 est l'erreur quadratique moyenne entre les deux grilles.

Une deuxième métrique utilisée ici est l'AP_{cell}, ou précision moyenne sur les cellules, qui est similaire à celle utilisée à la section 4.3.1 du chapitre précédent pour mesurer la performance des détecteurs. Les deux métriques évoluent en sens inverse, avec l'AP_{cell} augmentant quand les prédictions sont meilleures alors que l'erreur quadratique moyenne diminuera. Dans le chapitre précédent, cette métrique était calculée sur des boîtes englobantes classées en vrais positifs, faux positifs ou faux négatifs à partir d'une association entre boîtes prédites et boîtes vérités terrain obtenues en vérifiant que deux boîtes avaient un ratio d'aire de l'intersection sur l'aire de l'union (IoU) supérieur à un certain seuil (une illustration de ce processus est présente à l'annexe A.1). Dans ce chapitre, cette métrique est calculée sur les cellules des grilles comparées. Ainsi, une cellule de la grille reconstruite depuis le spectre prédit est

Dimensions	Couches	Qualité des prédictions		Coûts matériels	
		MSE IDCT	AP_{cell}	Temps total	Poids
64	2	0.0353	0.272	**0.76 ms**	**8 320 (35 Ko)**
64	3	0.0346	0.280	0.77 ms	12 480 (52 Ko)
128	2	0.0337	0.337	0.84 ms	33 024 (132 Ko)
128	3	0.0311	0.405	0.91 ms	49 536 (197 Ko)
256	2	0.0302	0.456	0.83 ms	131 584 (517 Ko)
256	3	0.0287	0.484	0.88 ms	197 376 (775 Ko)
512	2	0.0277	0.531	0.85 ms	525 312 (2 Mo)
512	3	0.0273	0.530	0.92 ms	787 968 (3 Mo)
1024	2	0.0266	0.561	0.90 ms	2 099 200 (8 Mo)
1024	3	0.0269	0.536	1.04 ms	3 148 800 (12 Mo)
2048	2	**0.0258**	**0.574**	1.62 ms	8 392 704 (32 Mo)
2048	3	0.0272	0.527	2.16 ms	12 589 056 (48 Mo)
4096	2	0.0262	0.558	4.57 ms	33 562 624 (128 Mo)

Tableau 5.3. : Métriques de qualité et de coût des réseaux de prédiction de spectre de segmentation. Chaque réseau comporte deux ou trois couches denses de taille d'entrée et de sortie égales à la dimension du spectre. Les couches internes ont une fonction d'activation ReLU et la couche de sortie n'a pas de fonction d'activation. Il n'y a pas de connexions résiduelles ni de couches de décrochage dans les réseaux. La méthode de normalisation par *contribution* est utilisée pour tous les réseaux. Finalement, le taux d'apprentissage est fixé à 0.001 durant tout l'entraînement. La taille de spectre 4096 n'a été entraînée que pour un réseau à deux couches étant donné qu'avec trois couches, les GPU utilisé pour l'entraînement n'avait pas assez de mémoire. Les temps totaux de calcul par grille qui regroupent la somme du temps de transformation de la grille d'occupation, d'inférence du réseau de neurones et de transformation inverse de la sortie du réseau de neurones ont été obtenus par calcul sur CPU et non pas GPU car pour d'aussi petits réseaux, le coût de transfert mémoire entre le CPU et le GPU est plus important que le gain de temps obtenu par l'utilisation de GPU à l'architecture parallèle plus efficace qu'un CPU.

classée en vrai positif uniquement si elle correspond à une cellule positive de la segmentation vérité terrain. Par conséquent, cette métrique est plus stricte que celle du chapitre précédent, car les cellules n'ont pas de tolérance particulière contrairement à la précision moyenne sur boîtes englobantes qui a une tolérance de placement matérialisée par l'usage d'un seuil de ratio d'IoU. Les deux métriques ne peuvent donc pas être comparées.

La figure 5.9 affiche les courbes de Précision par rapport au Rappel et une représentation graphique de l'AP_{cell} pour chaque détecteur. On peut y observer que les résultats de prédictions augmentent avec la dimension de spectre considéré jusqu'à 512 dimensions. Au-delà de 512, les scores d'AP_{cell} obtenus sont similaires. Par ailleurs, de 64 à 256 dimensions, les réseaux à trois couches sont plus efficaces que ceux à deux couches. Cette propriété s'inverse à partir de 512 dimensions où les réseaux à deux couches deviennent plus efficaces que ceux à trois couches.

Les métriques de coût matériel sont au nombre de deux :

— Le temps d'inférence est la somme de l'étape de conversion d'une grille d'occupation vers le domaine spectral, de prédiction par le réseau de neurones, puis de reconstruction d'une grille de segmentation à partir du spectre de réseau de neurones. Ce temps est calculé via la prédiction de lots de taille 1, qui correspondent à une utilisation en ligne où chaque grille d'occupation doit être prédite immédiatement et non pas mise en lots de grande taille pour lesquels l'inférence est plus rapide. Par ailleurs, ces résultats sont obtenus par inférence sur CPU et non pas sur GPU, car pour de si petits réseaux, les temps de transferts entre CPU et GPU sont plus importants que les gains potentiels obtenus par l'architecture hautement parallèle des GPU.

— Le nombre de poids, ainsi que la taille du fichier de stockage de ces poids (pour une précision de 32 bits par poids) permet d'évaluer le coût mémoire de l'exécution de tels modèles.

Les temps d'inférence de tous les réseaux avec une dimension de spectre de 512 ou moins sont inférieurs à une milliseconde. Les deux réseaux de dimension de spectre 1024 ont un temps d'inférence d'environ une milliseconde. Finalement, les réseaux à dimension de spectre 2048 ou 4096 ont un temps d'inférence variant entre 1 et 5 millisecondes. Ces temps d'inférence très faibles, largement inférieurs aux détecteurs de véhicules convolutifs obtenus au chapitre 4 (voir tableau 4.2), semblent avoir un potentiel d'utilisation embarquée intéressant et qui devrait être évalué via l'analyse du coût de calcul précis de l'algorithme en *FLOPs* et son intégration sur cible embarquée pour validation.

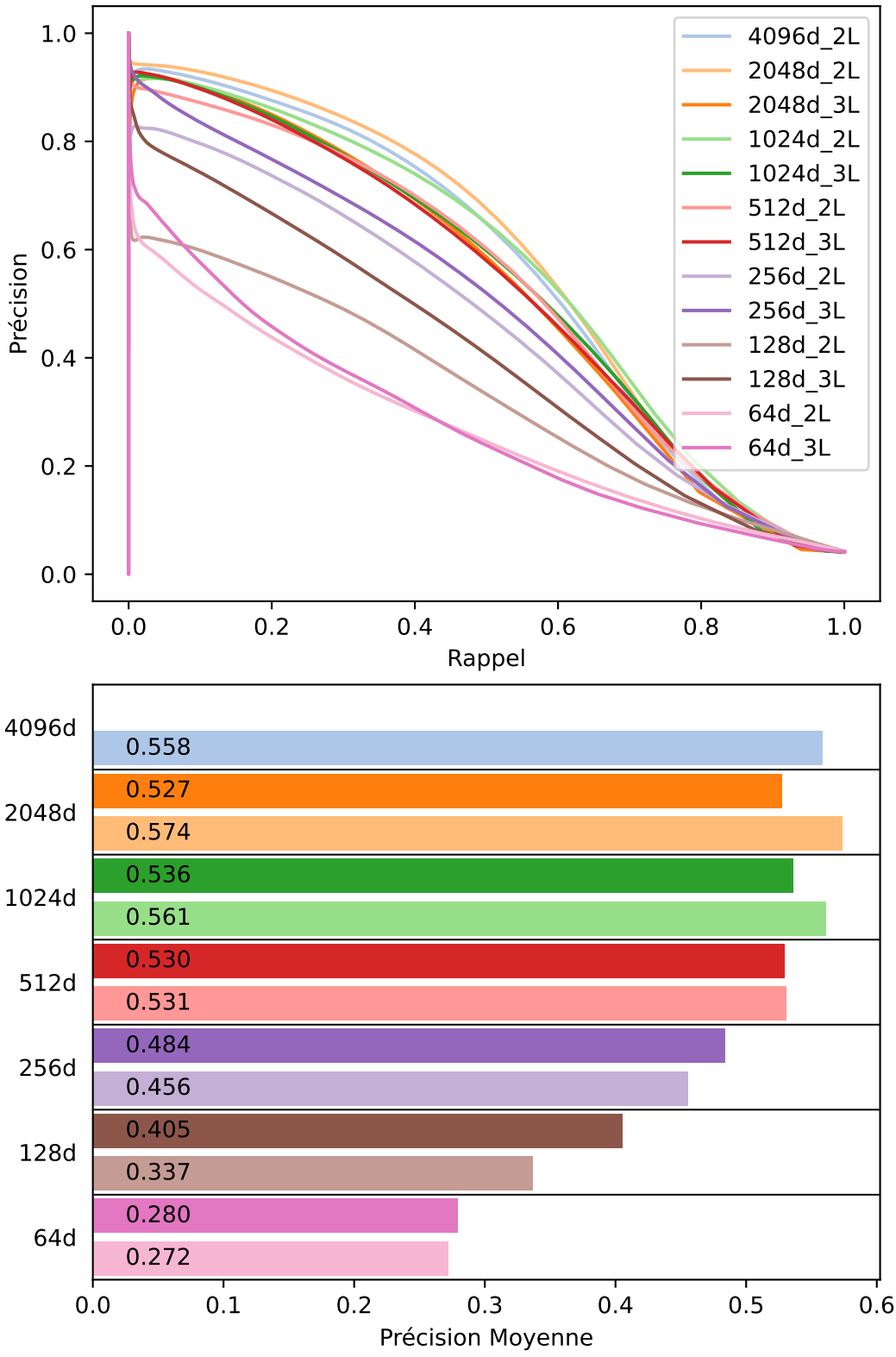

Figure 5.9. : Haut : Courbes de précision/rappel. Bas : Représentation graphique de l'AP_{cell} de chaque détecteur, qui correspond à l'aire sous les courbes de la figure du haut.

Les figures 5.10, 5.11, 5.12 et 5.13 montrent les résultats obtenus sur quatre grilles d'occupation du jeu de test pour différentes dimensions de spectre et réseaux de neurones associés. Pour chaque dimension de spectre, le réseau choisi est celui ayant la plus grande valeur de précision moyenne. Ces figures présentent la grille d'occupation originale, une grille reconstruite à partir du spectre tronqué utilisé en entrée du réseau de neurones qui permet de visualiser l'information utilisée effectivement pour la prédiction, la grille obtenue par reconstruction depuis le spectre prédit par le réseau de neurones, et finalement une version binarisée de cette grille reconstruite obtenue par seuillage de cette dernière qui permet une meilleure visualisation de la sortie du réseau de neurones. Le seuil de binarisation optimal diffère pour chaque réseau. Il est donc calculé par discrétisation de l'intervalle dans lequel le réseau de neurones produit des valeurs et par le choix parmi ces valeurs discrétisées du seuil qui minimise l'erreur moyenne entre la grille binarisée et la grille vérité terrain.

La figure 5.10 permet d'observer clairement les gains de précision de prédiction obtenus par augmentation de la dimension de spectre. Le réseau à 64 dimensions de spectre ne détecte aucun véhicule. Le réseau à 256 dimensions est quant à lui capable de détecter le véhicule du bas, mais avec une large sous-estimation de la dimension de ce dernier. Le réseau à 1024 dimensions détecte correctement les deux véhicules entièrement présents dans la grille, mais pas celui en haut à droite à moitié dans la grille. Finalement, le réseau à 2048 dimensions détecte tous les véhicules et évalue correctement leurs dimensions. Ces résultats sont remarquables par la capacité de ces réseaux de neurones à identifier des véhicules à partir d'un spectre tronqué dont il est très difficile visuellement d'identifier les véhicules sur la reconstruction par transformée inverse.

La figure 5.11 montre la capacité de ces réseaux de neurones à produire des segmentations de qualité y compris dans un contexte où de nombreux véhicules sont présents dans l'environnement.

Une limitation importante de ces réseaux est leur grande difficulté à détecter des véhicules orientés horizontalement, tels que ceux présentés sur la figure 5.12. Cette limitation semble liée au déséquilibre du jeu de données utilisé en termes d'orientation des véhicules comme visible sur la figure 3.4. Une solution permettant de résoudre ce problème serait de procéder à une augmentation de données par une rotation aléatoire du nuage de points LiDAR avant sa transformation en grilles d'occupation. Par ailleurs, la présence de murs parallèles au véhicule autonome amène à une fausse reconnaissance de véhicules comme visible sur la figure 5.13.

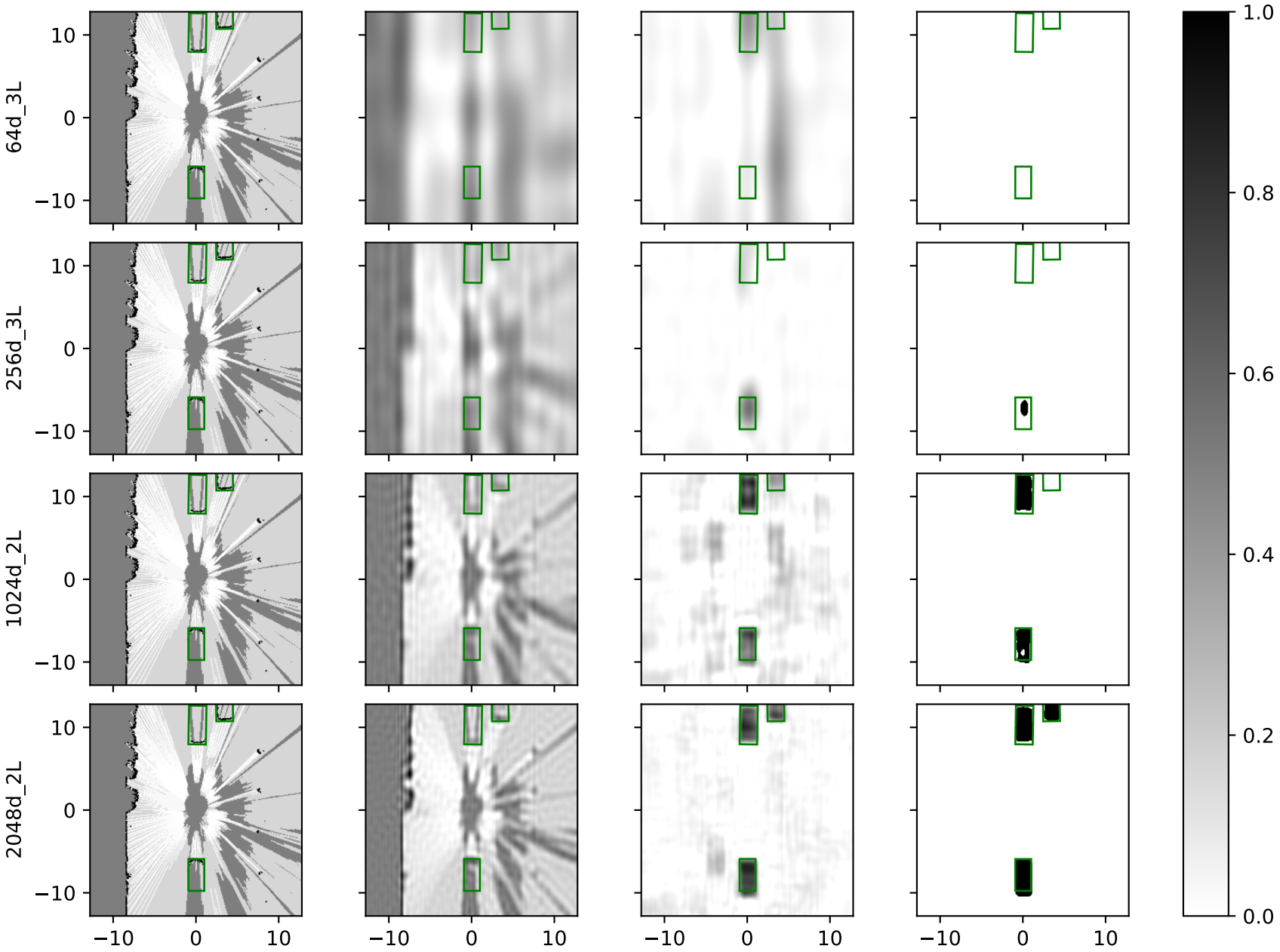

Figure 5.10. : Segmentations produites par quatre réseaux de neurones sur spectre à partir d'une grille d'occupation issue du jeu de test. De gauche à droite : La grille d'occupation originale, la reconstruction d'une grille d'occupation à partir du spectre utilisé en entrée du réseau de neurones, la segmentation reconstruite à partir du spectre en sortie du réseau de neurones, une binarisation de la reconstruction précédente avec une valeur de seuil optimisée pour garantir la meilleure correspondance possible avec les vérités terrain du jeu de données d'entraînement. De haut en bas : Le réseau à 64 dimensions de spectre et trois couches denses, à 256 dimensions de spectre et trois couches denses, à 1024 dimensions de spectre et deux couches denses et à 2048 dimensions de spectre et deux couches denses.

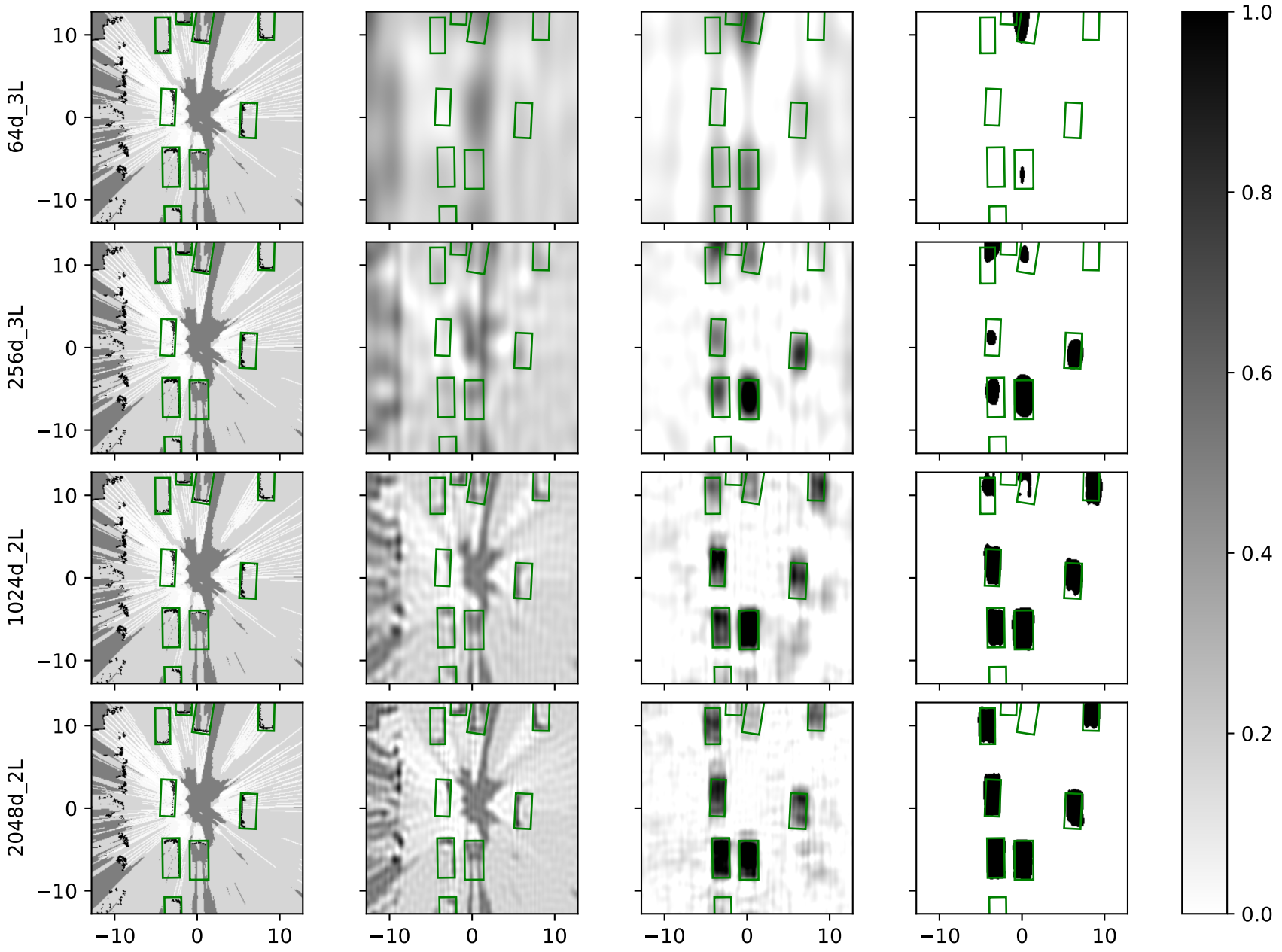

Figure 5.11. : Segmentations produites par quatre réseaux de neurones sur spectre à partir d'une grille d'occupation issue du jeu de test. De gauche à droite : La grille d'occupation originale, la reconstruction d'une grille d'occupation à partir du spectre utilisé en entrée du réseau de neurones, la segmentation reconstruite à partir du spectre en sortie du réseau de neurones, une binarisation de la reconstruction précédente avec une valeur de seuil optimisée pour garantir la meilleure correspondance possible avec les vérités terrain du jeu de données d'entraînement. De haut en bas : Le réseau à 64 dimensions de spectre et trois couches denses, à 256 dimensions de spectre et trois couches denses, à 1024 dimensions de spectre et deux couches denses et à 2048 dimensions de spectre et deux couches denses.

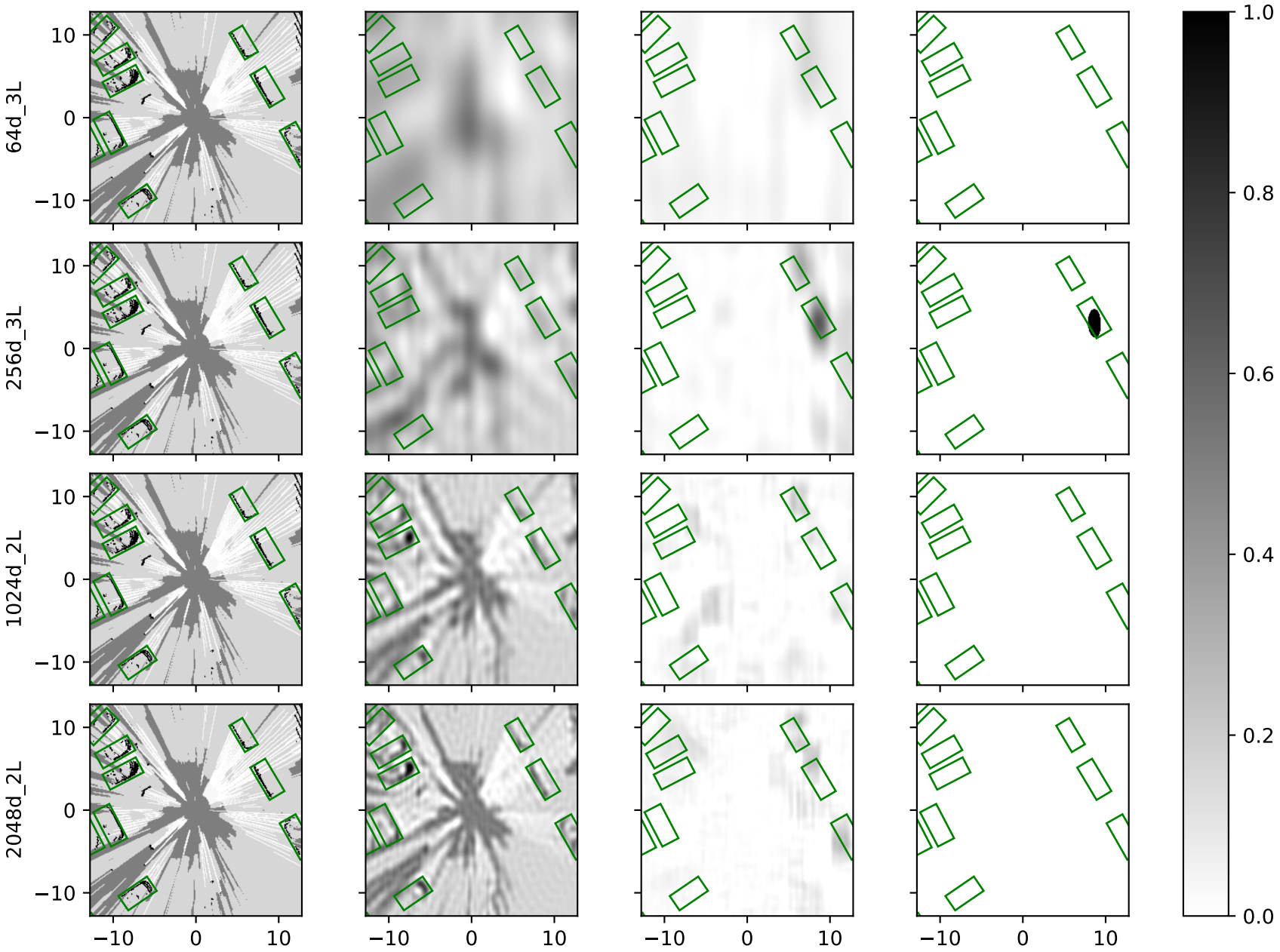

Figure 5.12. : Segmentations produites par quatre réseaux de neurones sur spectre à partir d'une grille d'occupation issue du jeu de test. De gauche à droite : La grille d'occupation originale, la reconstruction d'une grille d'occupation à partir du spectre utilisé en entrée du réseau de neurones, la segmentation reconstruite à partir du spectre en sortie du réseau de neurones, une binarisation de la reconstruction précédente avec une valeur de seuil optimisée pour garantir la meilleure correspondance possible avec les vérités terrain du jeu de données d'entraînement. De haut en bas : Le réseau à 64 dimensions de spectre et trois couches denses, à 256 dimensions de spectre et trois couches denses, à 1024 dimensions de spectre et deux couches denses et à 2048 dimensions de spectre et deux couches denses.

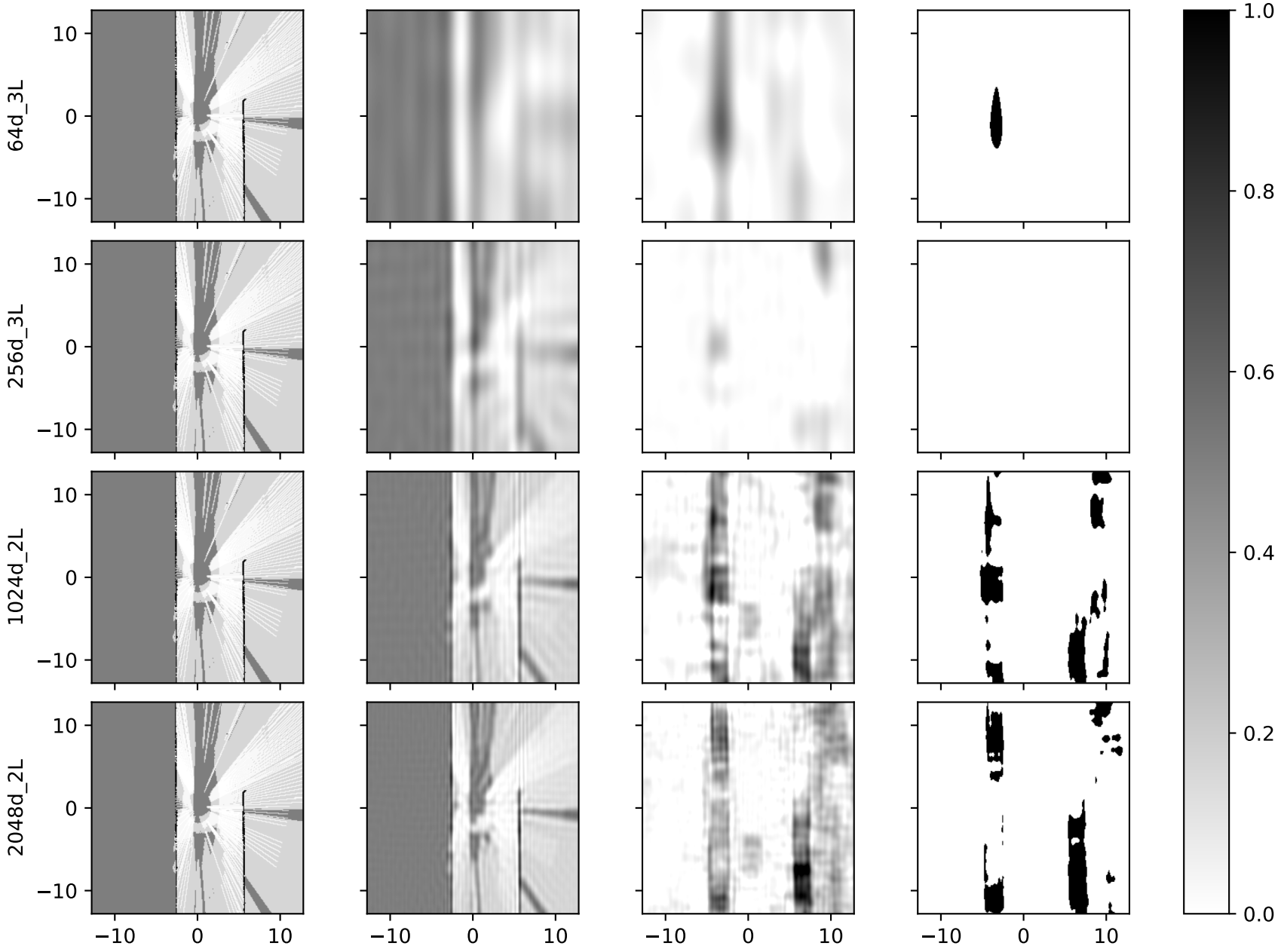

Figure 5.13. : Segmentations produites par quatre réseaux de neurones sur spectre à partir d'une grille d'occupation issue du jeu de test. De gauche à droite : La grille d'occupation originale, la reconstruction d'une grille d'occupation à partir du spectre utilisé en entrée du réseau de neurones, la segmentation reconstruite à partir du spectre en sortie du réseau de neurones, une binarisation de la reconstruction précédente avec une valeur de seuil optimisée pour garantir la meilleure correspondance possible avec les vérités terrain du jeu de données d'entraînement. De haut en bas : Le réseau à 64 dimensions de spectre et trois couches denses, à 256 dimensions de spectre et trois couches denses, à 1024 dimensions de spectre et deux couches denses et à 2048 dimensions de spectre et deux couches denses.

Des figures de segmentation de grilles d'occupation similaires avec en surimpression les boîtes englobantes prédites par le réseau convolutif **D2** proposé au chapitre 4 sont présentées à l'annexe A.2.

Finalement, l'ensemble des réseaux de neurones à deux et trois couches denses présentés dans cette section présentent un intérêt qui dépend des modalités d'usage souhaitées. Dans des contextes de très forte contrainte en termes de puissance de calcul, les réseaux à moins de 1024 dimensions de spectre offrent des temps d'inférence très faibles, inférieurs à une milliseconde sur notre ordinateur d'expérimentations équipé d'un processeur Intel Core i7-10850H. Ces temps, par ailleurs obtenus par du code Python soumis à une contrainte d'absence de parallélisation, pourraient être atteints sur des machines d'équipement inférieur, à l'aide d'optimisations informatiques telles que l'utilisation de code compilé, la parallélisation ou la quantification des poids du réseau. En cas de contrainte mémoire importante, les réseaux de dimension inférieure ou égale à 256 permettent un stockage des poids inférieur à 1 Mo. Ce coût mémoire pourrait même être réduit par la quantification des poids du réseau, qui sont dans ces expériences en précision standard sur 32 bits. Finalement, les meilleurs résultats peuvent être obtenus avec un réseau à deux couches et 2048 dimensions de spectre.

5.3 Conclusion

Comparativement aux réseaux convolutifs présentés au chapitre 4, cette nouvelle famille de réseaux de neurones très simples offre des temps d'inférence et des coûts mémoires classiquement utilisés pour des applications embarquées. Concernant la qualité des résultats obtenus, il est difficile de quantifier la différence entre les réseaux du chapitre 4 et ceux présentés ici en raison d'une différence fondamentale de nature des prédictions : boîtes englobantes avec les réseaux convolutifs, et segmentations de grilles avec ces réseaux fréquentiels. L'élaboration d'une métrique commune serait nécessaire pour compléter l'analyse. Néanmoins, et malgré certaines limitations par exemple illustrées par les figures 5.12 et 5.13, les résultats montrent que ces réseaux fréquentiels pourraient permettre la segmentation de l'espace occupé par d'autres véhicules et ainsi trouver leur place dans des applications de navigation avec de très importantes contraintes matérielles.

Par ailleurs, l'usage de spectres de grilles d'occupation et plus généralement de spectres d'images dans des applications de perception à l'aide d'apprentissage profond est un champ de recherche d'intérêt peu exploré à l'heure actuelle. En particulier,

le fait pour le réseau de neurones d'opérer entièrement dans le domaine fréquentiel de ses entrées à ses sorties permet d'obtenir des résultats de qualité pour un coût calculatoire très faible comparativement aux solutions convolutives présentées dans le chapitre 4.

6 Conclusion et perspectives

Ce chapitre conclut ce manuscrit de thèse par un résumé du travail présenté suivi d'une conclusion et de perspectives de recherche envisageables pour étendre la portée des résultats.

6.1 Résumé des travaux et conclusion

L'objectif de cette thèse de doctorat a été l'étude, la conception et la comparaison de méthodes de reconnaissance de véhicules sur grilles d'occupation avec une priorité donnée à la légèreté des calculs en vue de répondre au contexte embarqué propre aux applications de navigation autonome. Une première étude bibliographique de méthodes de détection d'objets sur images présentée au chapitre 2 a permis de retenir les méthodes d'apprentissage automatique et en particulier d'apprentissage profond à base de réseaux de neurones convolutifs comme méthodes de référence auxquelles comparer les différents modèles développés.

Une première contribution de ce travail de recherche, présentée au chapitre 3, a consisté en la conception d'un jeu de données de grilles d'occupation annoté avec des boîtes englobantes. Cette étape constitue un prérequis à tout entraînement de réseau de neurone. Pour ce faire, une première analyse quantitative et qualitative des principaux jeux de données automobiles a mené au choix de Waymo Open [Sun+20] comme jeu de données de base constitué de nuages de points LiDAR et de boîtes englobantes 3D correspondant aux véhicules, piétons et cyclistes présents dans la scène. Ce jeu de données de nuages de points a ensuite été transformé en jeu de données de grilles d'occupation par l'usage d'un outil développé par le laboratoire d'accueil [Rak17]. Finalement, une étude qualitative et quantitative des grilles d'occupation générées ainsi que des caractéristiques géométriques des boîtes englobantes a permis de mettre en lumière les principales caractéristiques et limites de ce jeu de données.

La deuxième contribution de ce travail de recherche, présentée au chapitre 4, a consisté en la conception de modèles d'apprentissage profond à base de réseaux de

neurones convolutifs, inspirés de l'état de l'art, pour la prédiction de boîtes englobantes correspondant aux véhicules sur des grilles d'occupation. Au total, quatre détecteurs ont été implémentés puis comparés en termes de qualité des prédictions et de temps d'inférence. Les résultats obtenus démontrent la pertinence d'utiliser de telles architectures pour la détection de véhicules sur grilles d'occupation, mais aussi leur coût calculatoire relativement important à l'origine d'un temps d'exécution de l'ordre de 10 à 50 millisecondes par grille d'occupation sur un GPU d'ordinateur portable non soumis à des critères embarqués particuliers. Ces travaux ont donné lieu à la publication de deux articles, l'un à la conférence GRETSI'22 [Def+22] et l'autre dans le journal MDPI Sensors [Def+23]. Par ailleurs, un brevet a été déposé au sujet d'un procédé de détection de véhicules sur grilles d'occupation à l'aide de ces travaux.

La troisième et dernière contribution de ce travail de recherche, présentée au chapitre 5, a visé à l'élaboration de nouvelles méthodes de reconnaissance d'objets sur grilles d'occupation présentant des coûts de calcul moindres que celles à base de réseaux de neurones convolutifs du chapitre 4. À cette fin, une méthode de prétraitement destinée à transformer les grilles d'occupation dans le domaine fréquentiel et à en supprimer les coefficients des fréquences les moins importantes afin d'en réduire la dimension a été proposée. Différentes méthodes de réduction de la dimension des spectres obtenus par la transformée en cosinus discrète à l'aide de masques de différentes formes ont été comparées et la méthode la plus efficace spécifiquement optimisée pour les grilles d'occupation a été conservée. Suite à cette compression, différents réseaux de neurones denses dans le domaine fréquentiel ont été entraînés à prédire le spectre d'une segmentation de grille d'occupation à partir du spectre de ladite grille. Les résultats obtenus démontrent une capacité de ces réseaux à identifier les véhicules dans le domaine spectral. Un fait particulièrement intéressant à noter est le coût calculatoire extrêmement faible de ces méthodes dont la plupart ont un temps d'inférence sur CPU par grille d'occupation inférieur à la milliseconde et un nombre de poids entraînable à stocker compris entre 8 320 et 33 562 624. Aucun travail comparable n'a pu être identifié dans l'état de l'art, aussi nous considérons cette contribution innovante. Elle donnera lieu au dépôt d'un brevet d'invention et à la soumission d'un article pour la conférence EUSIPCO'24.

Les travaux présentés dans ce manuscrit ne visent pas à atteindre la meilleure et la plus riche compréhension de la scène possible, car cela nécessiterait l'utilisation de modalités bien plus riches et complexes que des grilles d'occupation, par exemple des images RGB, des nuages de points LiDAR ou des scans RaDAR. De plus, les réseaux de neurones permettant l'exploitation de telles modalités ont un coup calculatoire réputé plus important. L'objectif des travaux présentés ici a été l'exploration

de méthodes de reconnaissance de véhicules permettant de minimiser les coûts calculatoire et mémoire tout en conservant des détections de qualité. En ce sens, les travaux du chapitre 4 exploitant des architectures convolutives inspirées de détecteurs d'objets sur images légers ont permis de valider cette approche de détection sur les grilles d'occupation et posent une base de comparaison. Les travaux du chapitre 5 présentent quant à eux une nouvelle approche de reconnaissance des véhicules présents sur une grille d'occupation en utilisant un prétraitement permettant de compresser fortement les grilles d'occupation dans le domaine fréquentiel et des architectures neuronales denses capables de segmenter les grilles d'occupation compressées dans le domaine fréquentiel. En dépit d'erreurs de prédiction observables sur certaines grilles d'occupation, ces nouvelles méthodes inconnues de la littérature, grâce à des temps d'exécution compatibles avec un processeur embarqué et des contraintes temps réel, et grâce à diverses possibilités d'améliorations détaillées dans les perspectives ci-après, appellent selon nous à de nouveaux développements et approfondissements.

6.2 Perspectives

Les travaux présentés dans ce manuscrit fournissent des premiers résultats d'intérêt pour la problématique de la reconnaissance de véhicules sur grilles d'occupation en gardant en mémoire la contrainte forte de solution embarquable. Néanmoins, un certain nombre d'améliorations et d'approfondissements pourraient être apportés à ces travaux afin d'en explorer plus en détail les tenants et aboutissants. Cette section présente ces travaux futurs dans l'ordre dans lequel ils devraient être menés selon nous.

6.2.1 Perspectives à court terme

— Le jeu de données utilisé tout au long de ce travail souffre de certaines limitations. Les grilles d'occupation d'une part ont été produites à l'aide d'un outil ne permettant pas de supprimer les points correspondant au sol dans les nuages de points LiDAR. Cette limitation amène à la présence de nombreux artefacts matérialisés par des arcs de cercles concentriques sur les grilles d'occupation qui peuvent gêner la reconnaissance des véhicules. Par ailleurs, la distribution très hétérogène des angles d'orientations des véhicules dans les scènes considérées (voir figure 3.4) amènent à des difficultés de certains détecteurs à détecter les véhicules non orientés parallèlement au véhicule

autonome. Ces deux limites pourraient être éliminées par la recréation d'un jeu de données de grilles d'occupation à l'aide d'un outil capable de supprimer les points correspondant au sol et par une rotation aléatoire des nuages de points LiDAR avant conversion en grilles d'occupation afin de rendre uniforme la distribution des angles d'orientations des véhicules.

— Les travaux de détection présentés se concentrent sur la reconnaissance de véhicules alors que le jeu de données utilisé contient aussi des boîtes englobantes correspondant aux piétons (les boîtes englobantes correspondant aux cyclistes, bien que présentes dans le jeu de données, sont très probablement en nombre insuffisant pour permettre un entraînement de qualité). La détection de piétons étant une tâche plus ardue que la détection de véhicules en raison de leur faible taille, il est difficile de prédire si les méthodes présentées dans ce travail fonctionneraient sur les piétons. Néanmoins, en raison de sa facilité, il serait avantageux d'entraîner à nouveau les modèles présentés dans ce travail à détecter cette fois les piétons pour obtenir un premier aperçu de la capacité de ces méthodes à détecter de petits objets.

— Les métriques de qualité des détections utilisées pour quantifier la performance des méthodes de détection proposées souffrent de multiples limites. D'une part, la précision moyenne sur les boîtes englobantes présentée au chapitre 4 ne tient pas compte du sens, mais uniquement de la direction du vecteur d'orientation du véhicule (ici, *direction* et *sens* sont entendus au sens de leur définition mathématique dans le cadre des vecteurs). Il serait intéressant de calculer une métrique permettant de tenir compte de ce sens pour affiner les résultats. D'autre part, la métrique de précision moyenne sur les cellules AP_{cell} utilisée au chapitre 5 est différente des métriques de précision moyenne sur les boîtes englobantes $\mathrm{AP}_{0.5}$ et $\mathrm{AP}_{0.7}$ utilisées au chapitre 4. En effet, la précision moyenne sur les boîtes englobantes autorise une certaine marge d'erreur entre la prédiction et la vérité terrain pour considérer une boîte comme un vrai positif et est donc moins stricte que la précision moyenne sur les cellules qui n'autorise pas de telles erreurs. Pour permettre une comparaison quantitative des méthodes du chapitre 4 et de celles du chapitre 5 il serait donc utile de développer une métrique s'appliquant à la fois aux boîtes englobantes et aux segmentations.

6.2.2 Perspectives à moyen terme

— Toutes les mesures de temps d'exécution des différentes méthodes présentées dans ce manuscrit l'ont été sur un ordinateur de bureau moyennement puis-

sant, mais néanmoins très différent architecturalement des cibles embarquées utilisées dans l'industrie automobile. Dans l'optique d'une exploration plus fine du potentiel d'utilisation embarquée des modèles proposés, il faudrait procéder à l'implémentation de ces méthodes directement sur une telle cible embarquée.

— Les réseaux de neurones utilisés au long de ce travail encodent les poids comme des nombres décimaux encodés sur 32 bits avec virgule flottante. Il serait possible, par une procédure de quantification des poids, de réduire cette taille d'encodage ce qui permettrait de réduire le temps d'inférence et le besoin mémoire de ces modèles au prix d'une légère perte de précision qui reste à quantifier.

6.2.3 Perspectives à long terme

— Les réseaux récurrents sont notoirement connus pour leur coût calculatoire important et la difficulté à les entraîner, mais permettent en échange d'intégrer de l'information issue du passé dans leur prédiction. En raison de la faible taille de la représentation compressée des grilles d'occupation, il devrait être possible d'entraîner un réseau récurrent à prédire la position des véhicules en utilisant toute l'information présente et passée de la scène dans le domaine fréquentiel.

— Les travaux présentés dans ce manuscrit se concentrent sur la reconnaissance de véhicules. Il devrait être possible de réaliser d'autres tâches de perception telles que la prédiction de grilles d'occupation futures à la condition qu'il soit possible de représenter les sorties desdites tâches dans le domaine fréquentiel.

La figure 6.1 illustre ces deux points de perspectives à long terme par la mise au point d'un système d'agrégation de représentations fréquentielles d'une série temporelle de grilles d'occupation. Diverses tâches de perception pourraient être apprises à partir de ces représentations fréquentielles agrégées.

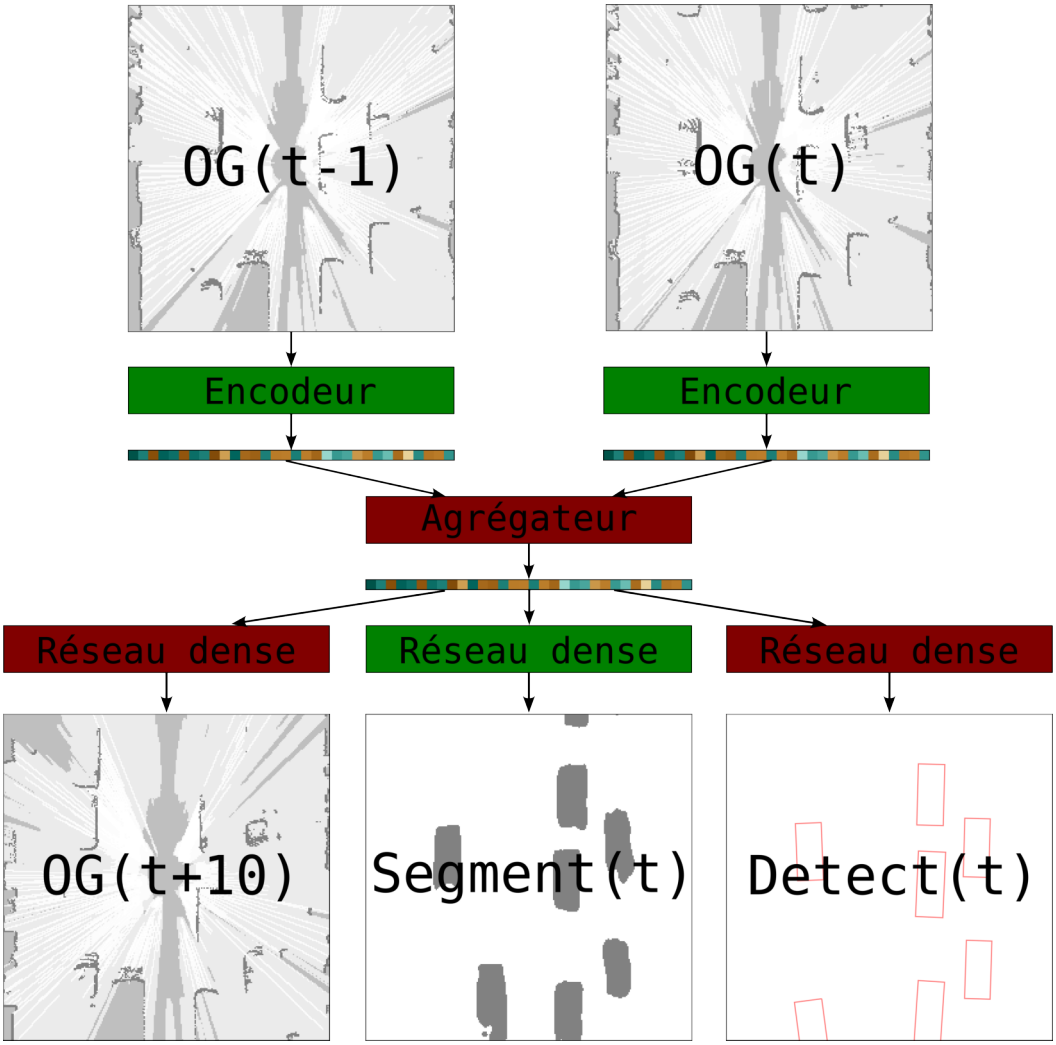

Figure 6.1. : Schéma d'un module de représentation fréquentielle d'une série temporelle de grilles d'occupation à l'aide des modules d'encodage dans le domaine fréquentiel développés au chapitre 5 et d'un module d'agrégation de multiples représentations fréquentielles. Par la suite, différentes tâches telles que la prédiction d'une grille d'occupation future pourraient être implémentées à l'aide de réseaux de neurones denses. Les boîtes vertes représentent les éléments déjà proposés dans ce manuscrit, les boîtes rouges représentent des modules non étudiés dans ce manuscrit.

Bibliographie

[Aba+16] Martín ABADI, Paul BARHAM, Jianmin CHEN et al. “TensorFlow : a system for large-scale machine learning”. In : *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*. OSDI’16. USA : USENIX Association, 2 nov. 2016, p. 265-283 (cf. p. 79).

[AGS23] Abdulrahman ALHARIQI, Ziyuan GU et Meead SABERI. “Impact of vehicle arrangement in mixed autonomy traffic on emissions”. In : *Transportation Research Part D : Transport and Environment* 125 (1er déc. 2023), p. 103964 (cf. p. 7).

[Ali+19] Waleed ALI, Sherif ABDELKARIM, Mahmoud ZIDAN, Mohamed ZAHRAN et Ahmad El SALLAB. “YOLO3D : End-to-End Real-Time 3D Oriented Object Bounding Box Detection from LiDAR Point Cloud”. In : *Computer Vision ECCV 2018 Workshops*. Sous la dir. de Laura LEAL-TAIXÉ et Stefan ROTH. Lecture Notes in Computer Science. Cham : Springer International Publishing, 2019, p. 716-728 (cf. p. 46, 68).

[Asg+22] Rabbia ASGHAR, Lukas RUMMELHARD, Anne SPALANZANI et Christian LAUGIER. “Allo-centric Occupancy Grid Prediction for Urban Traffic Scene Using Video Prediction Networks”. In : *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV). Déc. 2022, p. 255-260 (cf. p. 46).

[Bad+08] Hernan BADINO, Rudolf MESTER, Tobi VAUDREY, Uwe FRANKE et AG DAIMLER. “Stereo-based Free Space Computation in Complex Traffic Scenarios”. In : *2008 IEEE Southwest Symposium on Image Analysis and Interpretation*. 2008 IEEE Southwest Symposium on Image Analysis and Interpretation. Mars 2008, p. 189-192 (cf. p. 24).

[Bad+21] Claudine BADUE, Rânik GUIDOLINI, Raphael Vivacqua CARNEIRO et al. “Self-driving cars : A survey”. In : *Expert Systems with Applications* 165 (1er mars 2021), p. 113816 (cf. p. 1, 4, 9, 10, 14).

[Bal+23] R BALAJI V, S SANJAY, N SAKTHISHREE G et T SUSHRUTHAN. “Implementation of a Pre-Crash Detection System using Node MCU”. In : *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*. 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT). ISSN : 2832-3017. Jan. 2023, p. 84-87 (cf. p. 11).

[Bar+21] Alejandro BARRERA, Jorge BELTRAN, Carlos GUINDEL, Jose Antonio IGLESIAS et Fernando GARCIA. “BirdNet+ : Two-Stage 3D Object Detection in LiDAR Through a Sparsity-Invariant Birds Eye View”. In : *IEEE Access* 9 (2021), p. 160299-160316 (cf. p. 46).

[Bel+18] Jorge BELTRÁN, Carlos GUINDEL, Francisco Miguel MORENO et al. “BirdNet : A 3D Object Detection Framework from LiDAR Information”. In : *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018 21st International Conference on Intelligent Transportation Systems (ITSC). ISSN : 2153-0017. Nov. 2018, p. 3517-3523 (cf. p. 46).

[Bez+17] Jeff BEZANSON, Alan EDELMAN, Stefan KARPINSKI et Viral B. SHAH. “Julia : A Fresh Approach to Numerical Computing”. In : *SIAM Review* 59.1 (jan. 2017). Publisher : Society for Industrial and Applied Mathematics, p. 65-98 (cf. p. 79).

[BWL20] Alexey BOCHKOVSKIY, Chien-Yao WANG et Hong-Yuan Mark LIAO. *YOLOv4 : Optimal Speed and Accuracy of Object Detection*. 22 avr. 2020. arXiv : 2004.10934[cs,eess] (cf. p. 44, 69).

[Bot10] Léon BOTTOU. “Large-Scale Machine Learning with Stochastic Gradient Descent”. In : *Proceedings of COMPSTAT’2010*. Sous la dir. d’Yves LECHEVALLIER et Gilbert SAPORTA. Heidelberg : Physica-Verlag HD, 2010, p. 177-186 (cf. p. 37).

[Bra00] Ronald BRACEWELL. *The Fourier Transform and its Applications - 3rd ed.* McGraw Hill, 2000 (cf. p. 48, 49, 87).

[Bro+20] Tom BROWN, Benjamin MANN, Nick RYDER et al. “Language Models are Few-Shot Learners”. In : *Advances in Neural Information Processing Systems*. T. 33. Curran Associates, Inc., 2020, p. 1877-1901 (cf. p. 28).

[Cae+20] Holger CAESAR, Varun BANKITI, Alex H. LANG et al. “nuScenes : A Multimodal Dataset for Autonomous Driving”. In : *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA : IEEE, juin 2020, p. 11618-11628 (cf. p. 57, 58).

[Cai+21] Mu CAI, Hong ZHANG, Huijuan HUANG et al. “Frequency Domain Image Translation : More Photo-realistic, Better Identity-preserving”. In : *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada : IEEE, oct. 2021, p. 13910-13920 (cf. p. 53, 98).

[Cha+19] Ming-Fang CHANG, Deva RAMANAN, James HAYS et al. “Argoverse : 3D Tracking and Forecasting With Rich Maps”. In : *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA : IEEE, juin 2019, p. 8740-8749 (cf. p. 57, 58).

[Che+18] Xiaozhi CHEN, Kaustav KUNDU, Yukun ZHU et al. “3D Object Proposals Using Stereo Imagery for Accurate Object Class Detection”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.5 (1er mai 2018), p. 1259-1272 (cf. p. 45).

[Che+17] Xiaozhi CHEN, Huimin MA, Ji WAN, Bo LI et Tian XIA. “Multi-view 3D Object Detection Network for Autonomous Driving”. In : *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI : IEEE, juill. 2017, p. 6526-6534 (cf. p. 2, 14, 45, 46).

[CT65] James W. COOLEY et John W. TUKEY. “An algorithm for the machine calculation of complex Fourier series”. In : *Mathematics of Computation* 19.90 (1965), p. 297-301 (cf. p. 49).

[Cyb89] G. CYBENKO. “Approximation by superpositions of a sigmoidal function”. In : *Mathematics of Control, Signals and Systems* 2.4 (1er déc. 1989), p. 303-314 (cf. p. 32).

[DHS16] Jifeng DAI, Kaiming HE et Jian SUN. “Instance-Aware Semantic Segmentation via Multi-task Network Cascades”. In : *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). ISSN : 1063-6919. Juin 2016, p. 3150-3158 (cf. p. 20).

[Def+22] Nils DEFAUW, Olivier ANTONI, Marielle MALFANTE, Tiana RAKOTOVAO et Suzanne LESECQ. “Détection de véhicules en temps réel sur grilles doccupation par des méthodes dapprentissage profond”. In : *28ř Colloque sur le traitement du signal et des images*. Issue : 001-0298. Nancy : GRETSI - Groupe de Recherche en Traitement du Signal et des Images, 6 sept. 2022, p. 1193-1196 (cf. p. 22, 67, 118).

[Def+23] Nils DEFAUW, Marielle MALFANTE, Olivier ANTONI, Tiana RAKOTOVAO et Suzanne LESECQ. “Vehicle Detection on Occupancy Grid Maps : Comparison of Five Detectors Regarding Real-Time Performance”. In : *Sensors* 23.3 (jan. 2023). Number : 3 Publisher : Multidisciplinary Digital Publishing Institute, p. 1613 (cf. p. 22, 67, 118).

[Dub+14] Renaud DUBE, Markus HAHN, Markus SCHUTZ, Jurgen DICKMANN et Denis GINGRAS. “Detection of parked vehicles from a radar based occupancy grid”. In : *2014 IEEE Intelligent Vehicles Symposium Proceedings*. 2014 IEEE Intelligent Vehicles Symposium (IV). MI : IEEE, juin 2014, p. 1415-1420 (cf. p. 46).

[Elf89] A. ELFES. “Using occupancy grids for mobile robot perception and navigation”. In : *Computer* 22.6 (juin 1989). Conference Name : Computer, p. 46-57 (cf. p. 24).

[Eng+17] Martin ENGELCKE, Dushyant RAO, Dominic Zeng WANG, Chi Hay TONG et Ingmar POSNER. “Vote3Deep : Fast object detection in 3D point clouds using efficient convolutional neural networks”. In : *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017 IEEE International Conference on Robotics and Automation (ICRA). Singapore, Singapore : IEEE, mai 2017, p. 1355-1361 (cf. p. 20, 45).

[FK15] Daniel J. FAGNANT et Kara KOCKELMAN. “Preparing a nation for autonomous vehicles : opportunities, barriers and policy recommendations”. In : *Transportation Research Part A : Policy and Practice* 77 (1er juill. 2015), p. 167-181 (cf. p. 1, 6).

[FK18] Daniel J. FAGNANT et Kara M. KOCKELMAN. “Dynamic ride-sharing and fleet sizing for a system of shared autonomous vehicles in Austin, Texas”. In : *Transportation* 45.1 (1er jan. 2018), p. 143-158 (cf. p. 6).

[FRD18] Di FENG, Lars ROSENBAUM et Klaus DIETMAYER. “Towards Safe Autonomous Driving : Capture Uncertainty in the Deep Neural Network For Lidar 3D Vehicle Detection”. In : *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018 21st International Conference on Intelligent Transportation Systems (ITSC). Maui, HI : IEEE, nov. 2018, p. 3266-3273 (cf. p. 46).

[Fen+19] Di FENG, Lars ROSENBAUM, Fabian TIMM et Klaus DIETMAYER. “Leveraging Heteroscedastic Aleatoric Uncertainties for Robust Real-Time LiDAR 3D Object Detection”. In : *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019 IEEE Intelligent Vehicles Symposium (IV). ISSN : 2642-7214. Juin 2019, p. 1280-1287 (cf. p. 46).

[Fen+22] Di FENG, Zining WANG, Yiyang ZHOU et al. “Labels are Not Perfect : Inferring Spatial Uncertainty in Object Detection”. In : *IEEE Transactions on Intelligent Transportation Systems* 23.8 (août 2022). Conference Name : IEEE Transactions on Intelligent Transportation Systems, p. 9981-9994 (cf. p. 46).

[Fuk75] Kunihiko FUKUSHIMA. “Cognitron : A self-organizing multilayered neural network”. In : *Biological Cybernetics* 20.3 (1er sept. 1975), p. 121-136 (cf. p. 40).

[Fuk80] Kunihiko FUKUSHIMA. “Neocognitron : A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In : *Biological Cybernetics* 36.4 (1er avr. 1980), p. 193-202 (cf. p. 40).

[GLU12] A. GEIGER, P. LENZ et R. URTASUN. “Are we ready for autonomous driving ? The KITTI vision benchmark suite”. In : *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Providence, RI : IEEE, juin 2012, p. 3354-3361 (cf. p. 57, 58).

[Gey+20] Jakob GEYER, Yohannes KASSAHUN, Mentar MAHMUDI et al. *A2D2 : Audi Autonomous Driving Dataset*. 14 avr. 2020. arXiv : 2004.06320[cs,eess] (cf. p. 57, 58).

[Gir15] Ross GIRSHICK. “Fast R-CNN”. In : *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015 IEEE International Conference on Computer Vision (ICCV). Santiago, Chile : IEEE, déc. 2015, p. 1440-1448 (cf. p. 43).

[Gir+14] Ross GIRSHICK, Jeff DONAHUE, Trevor DARRELL et Jitendra MALIK. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”. In : *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Columbus, OH, USA : IEEE, juin 2014, p. 580-587 (cf. p. 19, 43, 44).

[Gir+16] Ross GIRSHICK, Jeff DONAHUE, Trevor DARRELL et Jitendra MALIK. “Region-Based Convolutional Networks for Accurate Object Detection and Segmentation”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.1 (1er jan. 2016), p. 142-158 (cf. p. 43).

[GBC16] Ian GOODFELLOW, Yoshua BENGIO et Aaron COURVILLE. *Deep Learning*. MIT Press, 2016 (cf. p. 28, 37, 74).

[He+17] Kaiming HE, Georgia GKIOXARI, Piotr DOLLAR et Ross GIRSHICK. “Mask R-CNN”. In : *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017 IEEE International Conference on Computer Vision (ICCV). Venice : IEEE, oct. 2017, p. 2980-2988 (cf. p. 43).

[He+15] Kaiming HE, Xiangyu ZHANG, Shaoqing REN et Jian SUN. “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.9 (1er sept. 2015), p. 1904-1916 (cf. p. 43).

[He+16] Kaiming HE, Xiangyu ZHANG, Shaoqing REN et Jian SUN. “Deep Residual Learning for Image Recognition”. In : Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016, p. 770-778 (cf. p. 101).

[Hin+12] Geoffrey HINTON, Li DENG, Dong YU et al. “Deep Neural Networks for Acoustic Modeling in Speech Recognition : The Shared Views of Four Research Groups”. In : *IEEE Signal Processing Magazine* 29.6 (nov. 2012). Conference Name : IEEE Signal Processing Magazine, p. 82-97 (cf. p. 28).

[HBD18] Stefan HOERMANN, Martin BACH et Klaus DIETMAYER. “Dynamic Occupancy Grid Prediction for Urban Autonomous Driving : A Deep Learning Approach with Fully Automatic Labeling”. In : *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018 IEEE International Conference on Robotics and Automation (ICRA). Brisbane, QLD : IEEE, mai 2018, p. 2056-2063 (cf. p. 46).

[Hoe+18] Stefan HOERMANN, Philipp HENZLER, Martin BACH et Klaus DIETMAYER. “Object Detection on Dynamic Occupancy Grid Maps Using Deep Learning and Automatic Label Generation”. In : *2018 IEEE Intelligent Vehicles Symposium (IV)*. 2018 IEEE Intelligent Vehicles Symposium (IV). ISSN : 1931-0587. Juin 2018, p. 826-833 (cf. p. 16, 47).

[HSW89] Kurt HORNIK, Maxwell STINCHCOMBE et Halbert WHITE. “Multilayer feedforward networks are universal approximators”. In : *Neural Networks* 2.5 (1er jan. 1989), p. 359-366 (cf. p. 32).

[Hou+20] John HOUSTON, Guido ZUIDHOF, Luca BERGAMINI et al. *One Thousand and One Hours : Self-driving Motion Prediction Dataset*. 16 nov. 2020. arXiv : 2006.14480[cs] (cf. p. 57).

[ISO94] ISO. *ISO/IEC 10918-1:1994 : Information technology - Digital compression and coding of continuous-tone still images : Requirements and guidelines*. Geneva, Switzerland : International Organization for Standardization, 1994. 182 p. (cf. p. 50, 88, 89, 92).

[ISO18] ISO. *ISO 15622:2018 : Systèmes intelligents de transports - Systèmes stabilisateurs de vitesse adaptés - Exigences de performance et modes opératoires*. Geneva, Switzerland : International Organization for Standardization, 2018 (cf. p. 5).

[ISO19] ISO. *ISO/IEC 15444-1:2019 : Information technology - JPEG 2000 image coding system - Part 1 : Core coding system*. Geneva, Switzerland : International Organization for Standardization, 2019. 234 p. (cf. p. 51, 88, 89).

[IDK19] Masha ITKINA, Katherine DRIGGS-CAMPBELL et Mykel J. KOCHENDERFER. “Dynamic Environment Prediction in Urban Scenes using Recurrent Representation Learning”. In : *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019 IEEE Intelligent Transportation Systems Conference (ITSC). Oct. 2019, p. 2052-2059 (cf. p. 46).

[Joy21] James JOYCE. *Bayes Theorem*. In : *The Stanford Encyclopedia of Philosophy*. Edward N. Zalta. Metaphysics Research Lab, Stanford University, 2021 (cf. p. 26).

[Jum+21] John JUMPER, Richard EVANS, Alexander PRITZEL et al. “Highly accurate protein structure prediction with AlphaFold”. In : *Nature* 596.7873 (août 2021). Number : 7873 Publisher : Nature Publishing Group, p. 583-589 (cf. p. 28).

[Kan+11] Yousun KANG, Koichiro YAMAGUCHI, Takashi NAITO et Yoshiki NINOMIYA. “Multiband Image Segmentation and Object Recognition for Understanding Road Scenes”. In : *IEEE Transactions on Intelligent Transportation Systems* 12.4 (déc. 2011). Conference Name : IEEE Transactions on Intelligent Transportation Systems, p. 1423-1433 (cf. p. 13).

[Kar+14] Andrej KARPATHY, George TODERICI, Sanketh SHETTY et al. “Large-Scale Video Classification with Convolutional Neural Networks”. In : *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014 IEEE Conference on Computer Vision and Pattern Recognition. ISSN : 1063-6919. Juin 2014, p. 1725-1732 (cf. p. 28).

[Kaw16] Kenji KAWAGUCHI. “Deep learning without poor local minima”. In : *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS’16. Red Hook, NY, USA : Curran Associates Inc., 5 déc. 2016, p. 586-594 (cf. p. 37).

[KL20] Patrick KIDGER et Terry LYONS. “Universal Approximation with Deep Narrow Networks”. In : *Proceedings of Thirty Third Conference on Learning Theory*. Conference on Learning Theory. ISSN : 2640-3498. PMLR, 15 juill. 2020, p. 2306-2327 (cf. p. 33).

[KB17] Diederik P. KINGMA et Jimmy BA. *Adam : A Method for Stochastic Optimization*. 29 jan. 2017. arXiv : 1412.6980[cs] (cf. p. 38, 73, 103).

[KS22] Ganesh KOLAPPAN GEETHA et Sung-Han SIM. “Fast identification of concrete cracks using 1D deep learning and explainable artificial intelligence-based analysis”. In : *Automation in Construction* 143 (1er nov. 2022), p. 104572 (cf. p. 53, 98).

[KSH17] Alex KRIZHEVSKY, Ilya SUTSKEVER et Geoffrey E. HINTON. “ImageNet classification with deep convolutional neural networks”. In : *Communications of the ACM* 60.6 (24 mai 2017), p. 84-90 (cf. p. 28, 36, 48).

[Ku+18] Jason KU, Melissa MOZIFIAN, Jungwook LEE, Ali HARAKEH et Steven L. WASLANDER. “Joint 3D Proposal Generation and Object Detection from View Aggregation”. In : *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Madrid : IEEE, oct. 2018, p. 1-8 (cf. p. 46).

[Lan+19] Alex H. LANG, Sourabh VORA, Holger CAESAR et al. “PointPillars : Fast Encoders for Object Detection From Point Clouds”. In : *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA : IEEE, juin 2019, p. 12689-12697 (cf. p. 46, 68).

[LIK21] Bernard LANGE, Masha ITKINA et Mykel J. KOCHENDERFER. “Attention Augmented ConvLSTM for Environment Prediction”. In : *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). ISSN : 2153-0866. Sept. 2021, p. 1346-1353 (cf. p. 46).

[LeC+89] Y. LECUN, B. BOSER, J. S. DENKER et al. “Handwritten digit recognition with a back-propagation network”. In : *Proceedings of the 2nd International Conference on Neural Information Processing Systems*. NIPS’89. Cambridge, MA, USA : MIT Press, 1er jan. 1989, p. 396-404 (cf. p. 36).

[LeC+98] Y. LECUN, L. BOTTOU, Y. BENGIO et P. HAFFNER. “Gradient-based learning applied to document recognition”. In : *Proceedings of the IEEE* 86.11 (nov. 1998). Conference Name : Proceedings of the IEEE, p. 2278-2324 (cf. p. 40, 42).

[LBH15] Yann LeCun, Yoshua Bengio et Geoffrey Hinton. “Deep learning”. In : *Nature* 521.7553 (mai 2015). Number : 7553 Publisher : Nature Publishing Group, p. 436-444 (cf. p. 28, 87, 88).

[Lee+20] Kuan-Hui Lee, Matthew Kliemann, Adrien Gaidon et al. “PillarFlow : End-to-end Birds-eye-view Flow Estimation for Autonomous Driving”. In : *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). ISSN : 2153-0866. Oct. 2020, p. 2007-2013 (cf. p. 46).

[Li17] Bo Li. “3D fully convolutional network for vehicle detection in point cloud”. In : *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Vancouver, BC : IEEE, sept. 2017, p. 1513-1518 (cf. p. 45).

[LZX16] Bo Li, Tianlei Zhang et Tian Xia. “Vehicle Detection from 3D Lidar Using Fully Convolutional Network”. In : *Robotics : Science and Systems XII*. Robotics : Science and Systems 2016. Robotics : Science et Systems Foundation, 2016 (cf. p. 46).

[Li+22] Chuyi Li, Lulu Li, Hongliang Jiang et al. “YOLOv6 : A Single-Stage Object Detection Framework for Industrial Applications”. In : (2022). Publisher : arXiv Version Number : 1 (cf. p. 44, 69).

[Li+23] Shuaibo Li, Shibiao Xu, Wei Ma et Qiu Zong. “Image Manipulation Localization Using Attentional Cross-Domain CNN Features”. In : *IEEE Transactions on Neural Networks and Learning Systems* 34.9 (sept. 2023). Conference Name : IEEE Transactions on Neural Networks and Learning Systems, p. 5614-5628 (cf. p. 53).

[LWW21] Zhichao Li, Feng Wang et Naiyan Wang. “LiDAR R-CNN : An Efficient and Universal 3D Object Detector”. In : *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, TN, USA : IEEE, juin 2021, p. 7542-7551 (cf. p. 46).

[LXG22] Yiyi Liao, Jun Xie et Andreas Geiger. “KITTI-360 : A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022), p. 1-1 (cf. p. 57).

[Liu+16] Wei Liu, Dragomir Anguelov, Dumitru Erhan et al. “SSD : Single Shot MultiBox Detector”. In : *Computer Vision ECCV 2016*. Sous la dir. de Bastian Leibe, Jiri Matas, Nicu Sebe et Max Welling. Lecture Notes in Computer Science. Cham : Springer International Publishing, 2016, p. 21-37 (cf. p. 44, 46, 68).

[Lom+15] Jakob Lombacher, Markus Hahn, Jurgen Dickmann et Christian Wohler. “Detection of arbitrarily rotated parked cars based on radar sensors”. In : *2015 16th International Radar Symposium (IRS)*. 2015 16th International Radar Symposium (IRS). Dresden, Germany : IEEE, juin 2015, p. 180-185 (cf. p. 46).

[Lom+16] Jakob LOMBACHER, Markus HAHN, Jurgen DICKMANN et Christian WOHLER. "Potential of radar for static object classification using deep learning methods". In : *2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*. 2016 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM). San Diego, CA, USA : IEEE, mai 2016, p. 1-4 (cf. p. 46).

[Lom+17a] Jakob LOMBACHER, Markus HAHN, Jurgen DICKMANN et Christian WOHLER. "Object classification in radar using ensemble methods". In : *2017 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*. 2017 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM). Nagoya, Japan : IEEE, mars 2017, p. 87-90 (cf. p. 46).

[Lom+17b] Jakob LOMBACHER, Kilian LAUDT, Markus HAHN, Jurgen DICKMANN et Christian WOHLER. "Semantic radar grids". In : *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017 IEEE Intelligent Vehicles Symposium (IV). Los Angeles, CA, USA : IEEE, juin 2017, p. 1170-1175 (cf. p. 46).

[LSD15] Jonathan LONG, Evan SHELHAMER et Trevor DARRELL. "Fully convolutional networks for semantic segmentation". In : *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). ISSN : 1063-6919. Juin 2015, p. 3431-3440 (cf. p. 20, 42).

[LH17] Ilya LOSHCHILOV et Frank HUTTER. *SGDR : Stochastic Gradient Descent with Warm Restarts*. 3 mai 2017. arXiv : `1608.03983[cs,math]` (cf. p. 102).

[LYU18] Wenjie LUO, Bin YANG et Raquel URTASUN. "Fast and Furious : Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net". In : *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. ISSN : 2575-7075. Juin 2018, p. 3569-3577 (cf. p. 46).

[Mak80] J. MAKHOUL. "A fast cosine transform in one and two dimensions". In : *IEEE Transactions on Acoustics, Speech, and Signal Processing* 28.1 (fév. 1980). Conference Name : IEEE Transactions on Acoustics, Speech, and Signal Processing, p. 27-34 (cf. p. 50, 90).

[Man+22] Khushdeep S. MANN, Abhishek TOMY, Anshul PAIGWAR, Alessandro RENZAGLIA et Christian LAUGIER. "Predicting Future Occupancy Grids in Dynamic Environment with Spatio-Temporal Learning". In : *2022 IEEE Intelligent Vehicles Symposium (IV)*. 2022 IEEE Intelligent Vehicles Symposium (IV). Juin 2022, p. 1121-1126 (cf. p. 46).

[Mao+21a] Jiageng MAO, Minzhe NIU, Chenhan JIANG et al. "One Million Scenes for Autonomous Driving : ONCE Dataset". In : (2021). Publisher : arXiv Version Number : 3 (cf. p. 57).

[Mao+21b] Jiageng MAO, Yujing XUE, Minzhe NIU et al. "Voxel Transformer for 3D Object Detection". In : *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada : IEEE, oct. 2021, p. 3144-3153 (cf. p. 45).

[MS15] Daniel MATURANA et Sebastian SCHERER. “VoxNet : A 3D Convolutional Neural Network for real-time object recognition”. In : *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Sept. 2015, p. 922-928 (cf. p. 24).

[MP43] Warren S. MCCULLOCH et Walter PITTS. “A logical calculus of the ideas immanent in nervous activity”. In : *The bulletin of mathematical biophysics* 5.4 (1er déc. 1943), p. 115-133 (cf. p. 31).

[Mil+19] Andres MILIOTO, Ignacio VIZZO, Jens BEHLEY et Cyrill STACHNISS. “RangeNet ++ : Fast and Accurate LiDAR Semantic Segmentation”. In : *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). ISSN : 2153-0866. Nov. 2019, p. 4213-4220 (cf. p. 20).

[ME85] H. MORAVEC et A. ELFES. “High resolution maps from wide angle sonar”. In : *1985 IEEE International Conference on Robotics and Automation Proceedings*. 1985 IEEE International Conference on Robotics and Automation Proceedings. T. 2. Mars 1985, p. 116-121 (cf. p. 2, 16, 18, 23, 67).

[Ona+23] Nuri C. ONAT, Jafar MANDOURI, Murat KUCUKVAR et al. “Rebound effects undermine carbon footprint reduction potential of autonomous electric vehicles”. In : *Nature Communications* 14.1 (6 oct. 2023). Number : 1 Publisher : Nature Publishing Group, p. 6258 (cf. p. 7).

[Pan+20] Hujie PAN, Zining WANG, Wei ZHAN et Masayoshi TOMIZUKA. “Towards Better Performance and More Explainable Uncertainty for 3D Object Detection of Autonomous Vehicles”. In : *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC). Sept. 2020, p. 1-7 (cf. p. 46).

[Pat+07] Kaustubh PATHAK, Andreas BIRK, Jann POPPINGA et Soren SCHWERTFEGER. “3D forward sensor modeling and application to occupancy grid based sensor fusion”. In : *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. ISSN : 2153-0866. Oct. 2007, p. 2059-2064 (cf. p. 24).

[Pat+19] Abhishek PATIL, Srikanth MALLA, Haiming GANG et Yi-Ting CHEN. “The H3D Dataset for Full-Surround 3D Multi-Object Detection and Tracking in Crowded Urban Scenes”. In : *2019 International Conference on Robotics and Automation (ICRA)*. 2019 International Conference on Robotics and Automation (ICRA). Montreal, QC, Canada : IEEE, mai 2019, p. 9552-9557 (cf. p. 57, 58).

[PS15] Jonathan PETIT et Steven E. SHLADOVER. “Potential Cyberattacks on Automated Vehicles”. In : *IEEE Transactions on Intelligent Transportation Systems* 16.2 (avr. 2015). Conference Name : IEEE Transactions on Intelligent Transportation Systems, p. 546-556 (cf. p. 7).

[Pha+20] Quang-Hieu PHAM, Pierre SEVESTRE, Ramanpreet Singh PAHWA et al. “A*3D Dataset : Towards Autonomous Driving in Challenging Environments”. In : *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020 IEEE International Conference on Robotics and Automation (ICRA). Paris, France : IEEE, mai 2020, p. 2267-2273 (cf. p. 57, 58).

[Pie+21] Aj PIERGIOVANNI, Vincent CASSER, Michael S. RYOO et Anelia ANGELOVA. “4D-Net for Learned Multi-Modal Alignment”. In : *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada : IEEE, oct. 2021, p. 15415-15425 (cf. p. 46).

[Qi+18] Charles R. QI, Wei LIU, Chenxia WU, Hao SU et Leonidas J. GUIBAS. “Frustum PointNets for 3D Object Detection from RGB-D Data”. In : *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, UT, USA : IEEE, juin 2018, p. 918-927 (cf. p. 46).

[Qia+20] Rui QIAN, Divyansh GARG, Yan WANG et al. “End-to-End Pseudo-LiDAR for Image-Based 3D Object Detection”. In : *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA : IEEE, juin 2020, p. 5880-5889 (cf. p. 46).

[Rak17] Tiana RAKOTOVAO. “Integer Occupancy Grids : a probabilistic multi-sensor fusion framework for embedded perception”. Thèse de doct. Université Grenoble Alpes, 21 fév. 2017 (cf. p. 2, 18, 24, 27, 117).

[Red+16] Joseph REDMON, Santosh DIVVALA, Ross GIRSHICK et Ali FARHADI. “You Only Look Once : Unified, Real-Time Object Detection”. In : *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA : IEEE, juin 2016, p. 779-788 (cf. p. 20, 44, 69).

[RF17] Joseph REDMON et Ali FARHADI. “YOLO9000 : Better, Faster, Stronger”. In : *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI : IEEE, juill. 2017, p. 6517-6525 (cf. p. 42, 44, 46, 48, 68, 69, 74, 84).

[RF18] Joseph REDMON et Ali FARHADI. *YOLOv3 : An Incremental Improvement*. 8 avr. 2018. arXiv : 1804.02767[cs] (cf. p. 44, 48, 69, 76, 84).

[Ren+17] Shaoqing REN, Kaiming HE, Ross GIRSHICK et Jian SUN. “Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6 (1er juin 2017), p. 1137-1149 (cf. p. 43).

[RFB15] Olaf RONNEBERGER, Philipp FISCHER et Thomas BROX. “U-Net : Convolutional Networks for Biomedical Image Segmentation”. In : *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Sous la dir. de Nassir NAVAB, Joachim HORNEGGER, William M. WELLS et Alejandro F. FRANGI. Lecture Notes in Computer Science. Cham : Springer International Publishing, 2015, p. 234-241 (cf. p. 77).

[Ros58] F. ROSENBLATT. “The perceptron : A probabilistic model for information storage and organization in the brain”. In : *Psychological Review* 65.6 (1958). Place : US Publisher : American Psychological Association, p. 386-408 (cf. p. 40).

[RK82] Azriel ROSENFELD et Avinash KAK. *Digital Picture Processing*. Academic Press, 1982 (cf. p. 72).

[Rus+15] Olga RUSSAKOVSKY, Jia DENG, Hao SU et al. “ImageNet Large Scale Visual Recognition Challenge”. In : *International Journal of Computer Vision* 115.3 (1er déc. 2015), p. 211-252 (cf. p. 78).

[SAE21] SAE INTERNATIONAL. *J3016_202104 : Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. Warrendale, PA, USA : Society of Automotive Engineers International, 30 avr. 2021 (cf. p. 4).

[SAS23] Maher SAID, Spencer AESCHLIMAN et Amanda STATHOPOULOS. “Robots at your doorstep : acceptance of near-future technologies for automated parcel delivery”. In : *Scientific Reports* 13.1 (29 oct. 2023). Number : 1 Publisher : Nature Publishing Group, p. 18556 (cf. p. 6).

[Sch+20] Marcel SCHREIBER, Vasileios BELAGIANNIS, Claudius GLÄSER et Klaus DIETMAYER. “Motion Estimation in Occupancy Grid Maps in Stationary Settings Using Recurrent Neural Networks”. In : *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020 IEEE International Conference on Robotics and Automation (ICRA). ISSN : 2577-087X. Mai 2020, p. 8587-8593 (cf. p. 47).

[Sch+21] Marcel SCHREIBER, Vasileios BELAGIANNIS, Claudius GLÄSER et Klaus DIETMAYER. “Dynamic Occupancy Grid Mapping with Recurrent Neural Networks”. In : *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021 IEEE International Conference on Robotics and Automation (ICRA). ISSN : 2577-087X. Mai 2021, p. 6717-6724 (cf. p. 47).

[SHD19] Marcel SCHREIBER, Stefan HOERMANN et Klaus DIETMAYER. “Long-Term Occupancy Grid Prediction Using Recurrent Neural Networks”. In : *2019 International Conference on Robotics and Automation (ICRA)*. 2019 International Conference on Robotics and Automation (ICRA). Montreal, QC, Canada : IEEE, mai 2019, p. 9299-9305 (cf. p. 46).

[Shi+23] Shaoshuai SHI, Li JIANG, Jiajun DENG et al. “PV-RCNN++ : Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection”. In : *International Journal of Computer Vision* 131.2 (1er fév. 2023), p. 531-551 (cf. p. 46).

[SWL19] Shaoshuai SHI, Xiaogang WANG et Hongsheng LI. “PointRCNN : 3D Object Proposal Generation and Detection From Point Cloud”. In : *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA : IEEE, juin 2019, p. 770-779 (cf. p. 46).

[Shi+20] Shaoshuai SHI, Zhe WANG, Jianping SHI, Xiaogang WANG et Hongsheng LI. “From Points to Parts : 3D Object Detection from Point Cloud with Part-aware and Part-aggregation Network”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), p. 1-1 (cf. p. 45).

[Shi+17] Weijing SHI, Mohamed Baker ALAWIEH, Xin LI et Huafeng YU. “Algorithm and hardware implementation for visual perception system in autonomous vehicle : A survey”. In : *Integration* 59 (1er sept. 2017), p. 148-156 (cf. p. 1, 10).

[SSL12] Steven E. SHLADOVER, Dongyan SU et Xiao-Yun LU. “Impacts of Cooperative Adaptive Cruise Control on Freeway Traffic Flow”. In : *Transportation Research Record* 2324.1 (1er jan. 2012). Publisher : SAGE Publications Inc, p. 63-70 (cf. p. 6).

[Sil+16] David SILVER, Aja HUANG, Chris J. MADDISON et al. “Mastering the game of Go with deep neural networks and tree search”. In : *Nature* 529.7587 (jan. 2016). Number : 7587 Publisher : Nature Publishing Group, p. 484-489 (cf. p. 28).

[SDC24] Fulvio SILVESTRI, Francesco DE FABIIS et Pierluigi COPPOLA. “Consumers expectations and attitudes towards owning, sharing, and riding autonomous vehicles”. In : *Case Studies on Transport Policy* 15 (1er mars 2024), p. 101112 (cf. p. 6).

[Sim+19] Martin SIMON, Stefan MILZ, Karl AMENDE et Horst-Michael GROSS. “Complex-YOLO : An Euler-Region-Proposal for Real-Time 3D Object Detection on Point Clouds”. In : *Computer Vision ECCV 2018 Workshops*. Sous la dir. de Laura LEAL-TAIXÉ et Stefan ROTH. Lecture Notes in Computer Science. Cham : Springer International Publishing, 2019, p. 197-209 (cf. p. 46, 68).

[Sri+22] SRINIVAS RAO P, ROHAN GUDLA, VIJAY SHANKAR TELIDEVULAPALLI, JAYASREE SARADA KOTA et GAYATHRI MANDHA. “Review on self-driving cars using neural network architectures”. In : *World Journal of Advanced Research and Reviews* 16.2 (30 nov. 2022), p. 736-746 (cf. p. 7).

[Sri+14] Nitish SRIVASTAVA, Geoffrey HINTON, Alex KRIZHEVSKY, Ilya SUTSKEVER et Ruslan SALAKHUTDINOV. “Dropout : a simple way to prevent neural networks from overfitting”. In : *The Journal of Machine Learning Research* 15.1 (1er jan. 2014), p. 1929-1958 (cf. p. 102).

[Ste+20] Sascha STEYER, Christian LENK, Dominik KELLNER, Georg TANZMEISTER et Dirk WOLLHERR. “Grid-Based Object Tracking With Nonlinear Dynamic State and Shape Estimation”. In : *IEEE Transactions on Intelligent Transportation Systems* 21.7 (juill. 2020). Conference Name : IEEE Transactions on Intelligent Transportation Systems, p. 2874-2893 (cf. p. 46).

[STW17] Sascha STEYER, Georg TANZMEISTER et Dirk WOLLHERR. “Object tracking based on evidential dynamic occupancy grids in urban environments”. In : *2017 IEEE Intelligent Vehicles Symposium (IV)*. 2017 IEEE Intelligent Vehicles Symposium (IV). Los Angeles, CA, USA : IEEE, juin 2017, p. 1064-1070 (cf. p. 46).

[Str23] Gilbert STRANG. *Calculus Online Textbook*. MIT OpenCourseWare, 2023 (cf. p. 35).

[Sun+20] Pei SUN, Henrik KRETZSCHMAR, Xerxes DOTIWALLA et al. “Scalability in Perception for Autonomous Driving : Waymo Open Dataset”. In : *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle, WA, USA : IEEE, juin 2020, p. 2443-2451 (cf. p. 57, 58, 65, 117).

[Sze22] Richard SZELISKI. *Computer Vision : Algorithms and Applications*. Springer, 2022 (cf. p. 48).

[Tan+23] Bohuan TAN, Bangji ZHANG, Nong ZHANG, Yuanchang CHEN et An QIN. “Integrated control of electronic stability program and active suspension system using a priority-weighting mechanism”. In : *Proceedings of the Institution of Mechanical Engineers, Part D : Journal of Automobile Engineering* 237.12 (1er oct. 2023). Publisher : IMECHE, p. 2857-2871 (cf. p. 11).

[Tan+15] Jiexiong TANG, Chenwei DENG, Guang-Bin HUANG et Baojun ZHAO. “Compressed-Domain Ship Detection on Spaceborne Optical Image Using Deep Neural Network and Extreme Learning Machine”. In : *IEEE Transactions on Geoscience and Remote Sensing* 53.3 (mars 2015). Conference Name : IEEE Transactions on Geoscience and Remote Sensing, p. 1174-1185 (cf. p. 98).

[Tho20] Richmond THOMASON. *Logic and Artificial Intelligence*. In : *The Stanford Encyclopedia of Philosophy*. Edward N. Zalta. Metaphysics Research Lab, Stanford University, 2020 (cf. p. 27).

[TBF05] Sebastian THRUN, Wolfram BURGARD et Dieter FOX. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents. The MIT Press, 19 août 2005. 672 p. (cf. p. 24, 25).

[Uni20] UNITED NATIONS GENERAL ASSEMBLY. *A/RES/74/299 : Improving global road safety*. 2 sept. 2020 (cf. p. 5).

[Van+18] Jessica VAN BRUMMELEN, Marie OBRIEN, Dominique GRUYER et Homayoun NAJJARAN. “Autonomous vehicle perception : The technology of today and tomorrow”. In : *Transportation Research Part C : Emerging Technologies* 89 (1er avr. 2018), p. 384-406 (cf. p. 1, 9, 12, 25).

[Van+23] Jenna A. VAN FOSSEN, Chu-Hsiang CHANG, J. Kevin FORD, Elizabeth A. MACK et Shelia R. COTTEN. “Identifying Alternative Occupations for Truck Drivers Displaced Due to Autonomous Vehicles by Leveraging the O*NET Database”. In : *American Behavioral Scientist* 67.14 (1er déc. 2023). Publisher : SAGE Publications Inc, p. 1693-1715 (cf. p. 6, 7).

[WZZ23] Rui WAN, Tianyun ZHAO et Wei ZHAO. “PTA-Det : Point Transformer Associating Point Cloud and Image for 3D Object Detection”. In : *Sensors* 23.6 (17 mars 2023), p. 3229 (cf. p. 46).

[Wan+21a] Bo WANG, Ming ZHU, Ying LU et al. “Real-Time 3D Object Detection From Point Cloud Through Foreground Segmentation”. In : *IEEE Access* 9 (2021), p. 84886-84898 (cf. p. 46).

[WBL22] Chien-Yao WANG, Alexey BOCHKOVSKIY et Hong-Yuan Mark LIAO. “YOLOv7 : Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”. In : (2022). Publisher : arXiv Version Number : 1 (cf. p. 44, 69).

[Wan+21b] Ze WANG, Sihao DING, Ying LI et al. “Cirrus : A Long-range Bi-pattern LiDAR Dataset”. In : *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021 IEEE International Conference on Robotics and Automation (ICRA). Xi’an, China : IEEE, 30 mai 2021, p. 5744-5750 (cf. p. 57).

[Wan+20] Zining WANG, Di FENG, Yiyang ZHOU et al. “Inferring Spatial Uncertainty in Object Detection”. In : *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). ISSN : 2153-0866. Oct. 2020, p. 5792-5799 (cf. p. 46).

[Wen+21] Xinshuo WENG, Yunze MAN, Jinhyung PARK et al. “All-In-One Drive : A Large-Scale Comprehensive Perception Dataset with High-Density Long-Range Point Clouds”. In : *arXiv* (2021) (cf. p. 57).

[Wik23] WIKIPEDIA. *Speedometer*. In : *Wikipedia*. Page Version ID : 1185609282. 17 nov. 2023 (cf. p. 11).

[Wil+21] Benjamin WILSON, William QI, Tanmay AGARWAL et al. “Argoverse 2 : Next Generation Datasets for Self-Driving Perception and Forecasting”. In : *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks* 1 (6 déc. 2021) (cf. p. 57).

[Wir+18] Sascha WIRGES, Tom FISCHER, Christoph STILLER et Jesus Balado FRIAS. “Object Detection and Classification in Occupancy Grid Maps Using Deep Convolutional Networks”. In : *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018 21st International Conference on Intelligent Transportation Systems (ITSC). ISSN : 2153-0017. Nov. 2018, p. 3530-3535 (cf. p. 16, 47).

[Wir+19a] Sascha WIRGES, Johannes GRÄTER, Qiuhao ZHANG et Christoph STILLER. “Self-Supervised Flow Estimation using Geometric Regularization with Applications to Camera Image and Grid Map Sequences”. In : *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019 IEEE Intelligent Transportation Systems Conference (ITSC). Oct. 2019, p. 1782-1787 (cf. p. 46).

[Wir+19b] Sascha WIRGES, Marcel REITH-BRAUN, Martin LAUER et Christoph STILLER. “Capturing Object Detection Uncertainty in Multi-Layer Grid Maps”. In : *2019 IEEE Intelligent Vehicles Symposium (IV)*. 2019 IEEE Intelligent Vehicles Symposium (IV). ISSN : 2642-7214. Juin 2019, p. 1520-1526 (cf. p. 47).

[Xia+21] Pengchuan XIAO, Zhenlei SHAO, Steven HAO et al. “PandaSet : Advanced Sensor Suite Dataset for Autonomous Driving”. In : *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). Indianapolis, IN, USA : IEEE, 19 sept. 2021, p. 3095-3101 (cf. p. 57).

[Xie+93] M. XIE, L. TRASSOUDAINE, J. ALIZON, M. THONNAT et J. GALLICE. “Active and intelligent sensing of road obstacles : Application to the European Eureka-PROMETHEUS project”. In : 1993 (4th) International Conference on Computer Vision. IEEE Computer Society, 1er jan. 1993, p. 616, 617, 618, 619, 620, 621, 622, 623-616, 617, 618, 619, 620, 621, 622, 623 (cf. p. 7).

[YML18] Yan YAN, Yuxing MAO et Bo LI. “SECOND : Sparsely Embedded Convolutional Detection”. In : *Sensors* 18.10 (6 oct. 2018), p. 3337 (cf. p. 46).

[YLU18] Bin YANG, Wenjie LUO et Raquel URTASUN. “PIXOR : Real-time 3D Object Detection from Point Clouds”. In : *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, UT, USA : IEEE, juin 2018, p. 7652-7660 (cf. p. 46, 47, 68, 69, 71, 77, 84).

[Yog+21] Senthil YOGAMANI, Ciaran HUGHES, Jonathan HORGAN et al. *WoodScape : A multi-task, multi-camera fisheye dataset for autonomous driving*. 2 juill. 2021. arXiv : 1905.01489[cs,stat] (cf. p. 57).

[ZP15] Dominic ZENG WANG et Ingmar POSNER. “Voting for Voting in Online Point Cloud Object Detection”. In : *Robotics : Science and Systems XI*. Robotics : Science and Systems 2015. Robotics : Science et Systems Foundation, 13 juill. 2015 (cf. p. 45).

[Zha+21] Qianyun ZHANG, Kaveh BARRI, Saeed K. BABANAJAD et Amir H. ALAVI. “Real-Time Detection of Cracks on Concrete Bridge Decks Using Deep Learning in the Frequency Domain”. In : *Engineering* 7.12 (1er déc. 2021), p. 1786-1796 (cf. p. 52, 53, 98).

[Zha+22] Yifan ZHANG, Qingyong HU, Guoquan XU et al. “Not All Points Are Equal : Learning Highly Efficient Point-based Detectors for 3D LiDAR Point Clouds”. In : *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). ISSN : 2575-7075. Juin 2022, p. 18931-18940 (cf. p. 46).

[Zhe+21] Wu ZHENG, Weiliang TANG, Li JIANG et Chi-Wing FU. “SE-SSD : Self-Ensembling Single-Stage Object Detector From Point Cloud”. In : *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Nashville, TN, USA : IEEE, juin 2021, p. 14489-14498 (cf. p. 46, 68).

[ZT18] Yin ZHOU et Oncel TUZEL. “VoxelNet : End-to-End Learning for Point Cloud Based 3D Object Detection”. In : *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, UT, USA : IEEE, juin 2018, p. 4490-4499 (cf. p. 14, 45).

[Zhu+23] Qing ZHU, Xiumei LI, Junmei SUN et Huang BAI. “WDIG : a wavelet domain image generation framework based on frequency domain optimization”. In : *EURASIP Journal on Advances in Signal Processing* 2023.1 (19 juin 2023), p. 66 (cf. p. 98).

[ZL17] Barret ZOPH et Quoc V. LE. *Neural Architecture Search with Reinforcement Learning*. 15 fév. 2017. arXiv : 1611.01578[cs] (cf. p. 33).

Webpages

[@Car] CARNEGIE MELLON UNIVERSITY. *The Robot Hall of Fame*. URL : http://www.robothalloffame.org/inductees/08inductees/navlab.html (visité le 27 nov. 2023) (cf. p. 7).

[@Cru22] CRUISE. *Welcome Riders To A Fully Driverless Car Future*. 1er fév. 2022. URL : https://getcruise.com/news/blog/2022/welcome-riders/ (visité le 27 nov. 2023) (cf. p. 8).

[@Cru] CRUISE. *Cruise Self Driving Cars | Autonomous Vehicles | Driverless Rides*. URL : https://getcruise.com/ (visité le 27 nov. 2023) (cf. p. 6).

[@Def] DEFENSE ADVANCED RESEARCH PROJECTS AGENCY. *The Grand Challenge*. URL : https://www.darpa.mil/about-us/timeline/-grand-challenge-for-autonomous-vehicles (visité le 27 nov. 2023) (cf. p. 8).

[@Hon21] HONDA GLOBAL. *Honda to Begin Sales of Legend with New Honda SENSING Elite*. 4 mars 2021. URL : https://global.honda/en/newsroom/news/2021/4210304eng-legend.html (visité le 27 nov. 2023) (cf. p. 8).

[@IEE14] IEEE SPECTRUM. *The Unknown Start-up That Built Googles First Self-Driving Car*. 19 nov. 2014. URL : https://spectrum.ieee.org/the-unknown-startup-that-built-googles-first-selfdriving-car (visité le 27 nov. 2023) (cf. p. 8).

[@Kan22] Ali KANI. *DRIVE Thor Unites AV and Cockpit on a Single SoC*. NVIDIA Blog. 20 sept. 2022. URL : https://blogs.nvidia.com/blog/drive-thor/ (visité le 21 déc. 2023) (cf. p. 20).

[@Mas07] MASSACHUSETTS INSTITUTE OF TECHNOLOGY. *MIT finishes fourth in DARPA challenge for robotic vehicles*. MIT News | Massachusetts Institute of Technology. 5 nov. 2007. URL : https://news.mit.edu/2007/urban-1105 (visité le 27 nov. 2023) (cf. p. 8).

[@NXP] NXP. *i.MX Applications Processors | Multicore based on 32-bit and 64-bit Arm | NXP Semiconductors*. URL : http://www.nxp.com/pages/:IMX_HOME (visité le 21 déc. 2023) (cf. p. 20).

[@Ren] RENESAS. *R-Car Automotive System-on-Chips (SoCs) | Renesas*. URL : https://www.renesas.com/us/en/products/automotive-products/automotive-system-chips-socs (visité le 21 déc. 2023) (cf. p. 20).

[@Ste17] Taylor STEWART. *263 Self-Driving Car Startups to Watch*. Medium. 10 mai 2017. URL : https://blog.cometlabs.io/263-self-driving-car-startups-to-watch-8a9976dc62b0 (visité le 20 déc. 2023) (cf. p. 8).

[@The] THE ROBOTICS INSTITUTE, CARNEGIE MELLON UNIVERSITY. *Navlab*. URL : https://www.ri.cmu.edu/robotics-groups/navlab/ (visité le 27 nov. 2023) (cf. p. 7).

[@Uni] UNIVERSITÉ CARNEGIE-MELLON. *Navlab 5 - The Robotics Institute Carnegie Mellon University*. URL : https://www.ri.cmu.edu/robot/navlab-5/ (visité le 21 déc. 2023) (cf. p. 4).

[@Way18] WAYMO. *Waymo One : The next step on our self-driving journey*. Waypoint The official Waymo blog. 5 déc. 2018. URL : https://waymo.com/blog/2019/08/waymo-one-next-step-on-our-self-driving.html (visité le 27 nov. 2023) (cf. p. 8).

[@Way] WAYMO. *Waymo - Self-Driving Cars - Autonomous Vehicles - Ride-Hail*. Waymo. URL : https://waymo.com/ (visité le 27 nov. 2023) (cf. p. 6).

[@Wor22] WORLD HEALTH ORGANIZATION. *Road traffic injuries*. 20 juin 2022. URL : https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries (visité le 27 nov. 2023) (cf. p. 5).

Table des figures

Liste des tableaux

Annexe

A

A.1 Calcul de la précision moyenne pour un détecteur d'objets

La métrique de précision moyenne est utilisée pour évaluer la performance des détecteurs d'objets. Ces derniers prédisent des boîtes englobantes avec un score de confiance pour chacune dans l'intervalle $[0, 1]$. À partir de ces boîtes englobantes prédites et des boîtes englobantes vérités terrain la précision moyenne se calcule de la manière suivante (une illustration du processus est visible sur la figure A.1) :

1. Premièrement, plusieurs seuils de score de confiance sont utilisés afin de sélectionner différents ensembles de boîtes englobantes prédites qui seront considérées comme positives. Plus ce seuil est bas, plus le nombre de boîtes englobantes prédites ayant un score de confiance supérieur à ce seuil est élevé.
2. Ensuite, pour chacun de ces ensembles de boîtes englobantes positives, les nombres de vrais positifs (une boîte positive correspondant à une boîte vérité terrain), faux positifs (une boîte positive ne correspondant à aucune boîte vérité terrain) et faux négatifs (une boîte vérité terrain ne correspondant à aucune boîte positive) sont calculés. L'association entre boîtes positives et boîtes vérités terrain est faite par le calcul du ratio de l'aire de l'intersection sur l'aire de l'union de deux boîtes. Si ce ratio est supérieur à un seuil choisi (généralement 0,5 ou 0,7), les deux boîtes englobantes sont considérées comme associées.
3. Ensuite, pour chaque ensemble de boîtes englobantes considérées positives, la précision (définie comme le ratio du nombre de vrais positifs sur le nombre total de boîtes positives) et le rappel (défini comme le ratio du nombre de vrais positifs sur le nombre total de boîtes vérités terrain) sont calculés. Ces différents couples (Précision, Rappel) permettent de tracer une courbe de la précision en fonction du rappel.
4. Finalement, la précision moyenne est définie comme l'aire sous cette courbe.

Un détecteur est généralement évalué sur plusieurs images, auquel cas les nombres de vrais positifs, faux positifs et faux négatifs sont sommés sur l'ensemble des images

du jeu de données de test. Par la suite les différentes valeurs de précision et de rappel sont calculées à l'aide de ces nombres sommés.

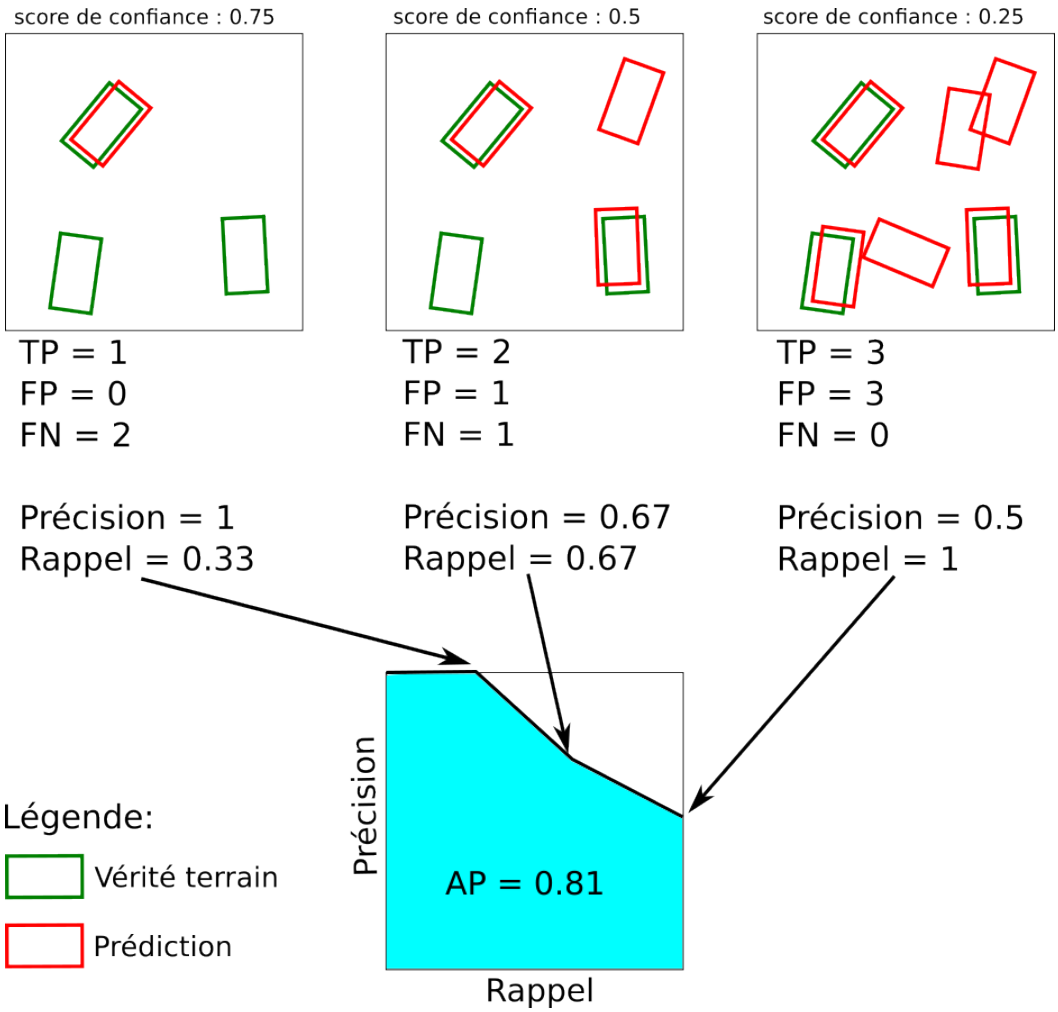

Figure A.1. : Illustration de la méthode de calcul de la précision moyenne d'un détecteur d'objets. Les associations entre boîtes englobantes normalement calculées à l'aide du ratio de l'aire de l'intersection sur l'aire de l'union ne sont pas représentées ici.

A.2 Comparaisons entre segmentations dans le domaine fréquentiel et détection dans le domaine spatial

Cette annexe présente comparativement des résultats de reconnaissance de véhicules produits par les réseaux de segmentation dans le domaine fréquentiel compressé du chapitre 5 et par le réseau de détection de véhicules **D2** proposé au chapitre 4. Les figures A.2, A.3, A.4 et A.5 représentent de gauche à droite :

— La grille d'occupation originale

— La grille d'occupation reconstruite par *DCT* inverse à partir du spectre utilisé en entrée du réseau de segmentation

— La segmentation reconstruite à partir du spectre produit en sortie du réseau de segmentation

— Une binarisation de la segmentation

De haut en bas :

— Le réseau de segmentation à 64 dimensions et 3 couches denses

— Le réseau de segmentation à 256 dimensions et 3 couches denses

— Le réseau de segmentation à 1024 dimensions et 2 couches denses

— Le réseau de segmentation à 2048 dimensions et 2 couches denses

En parallèle, des boîtes englobantes vertes pour les vérités terrain et rouges pour les détections produites par le réseau convolutif **D2** sur les grilles d'occupation originales sont représentées.

Pour rappel :

— le réseau **D2** comporte 4 961 463 poids entraînables et a un temps d'exécution par grille sur GPU de 11ms

— Les quatre réseaux de segmentation comportent entre 12 480 et 8 392 704 poids entraînables et ont un temps d'exécution compris entre 0.77ms et 1.62ms.

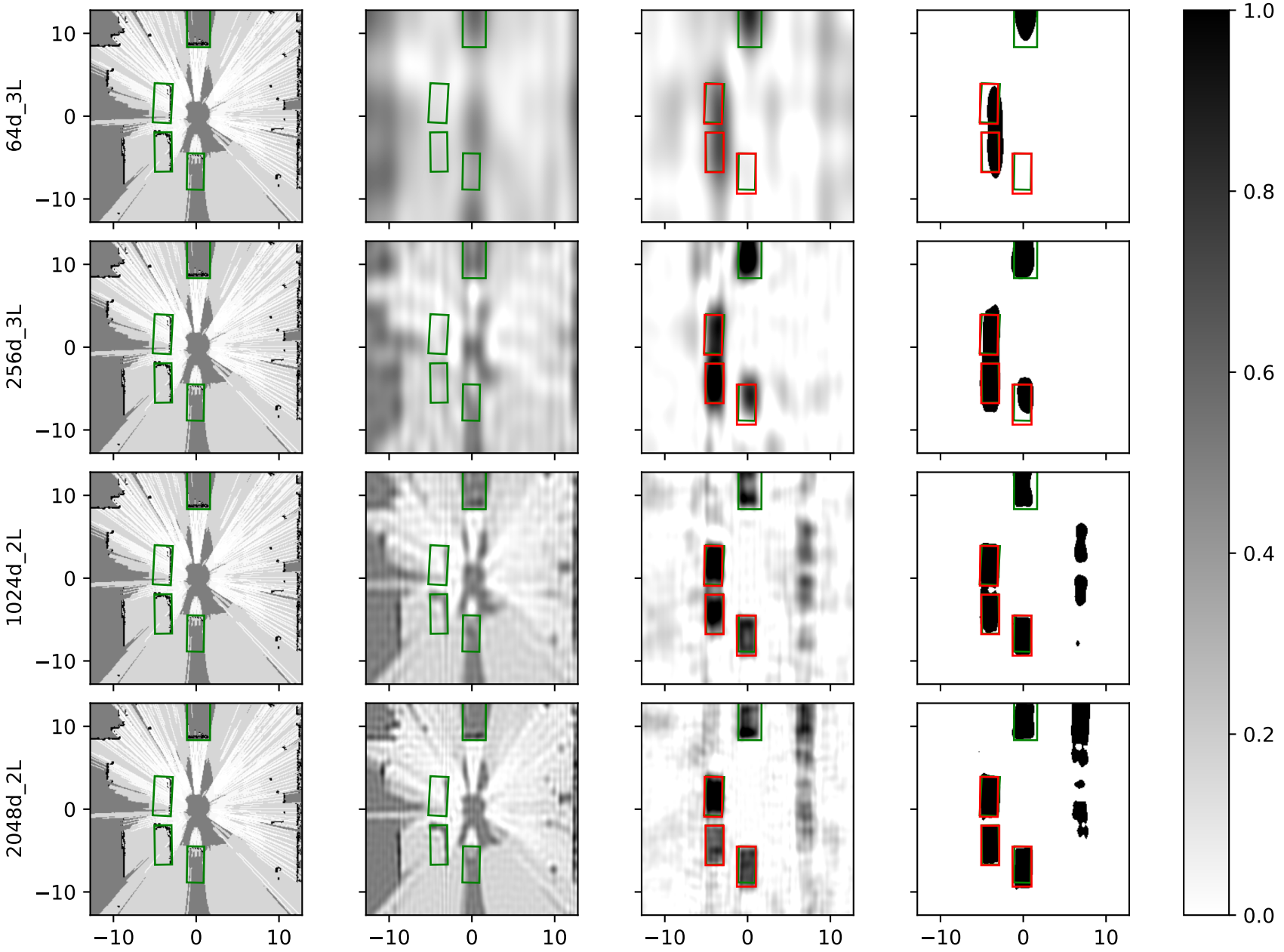

Figure A.2. : De gauche à droite : La grille d'occupation originale, la reconstruction d'une grille d'occupation à partir du spectre utilisé en entrée du réseau de neurones, la segmentation reconstruite à partir du spectre en sortie du réseau de neurones, une binarisation de la reconstruction précédente. De haut en bas : Le réseau à 64 dimensions de spectre et trois couches denses, à 256 dimensions de spectre et trois couches denses, à 1024 dimensions de spectre et deux couches denses et à 2048 dimensions de spectre et deux couches denses. Boîtes englobantes vertes pour les vérités terrain et rouges pour les détections produites par **D2**.

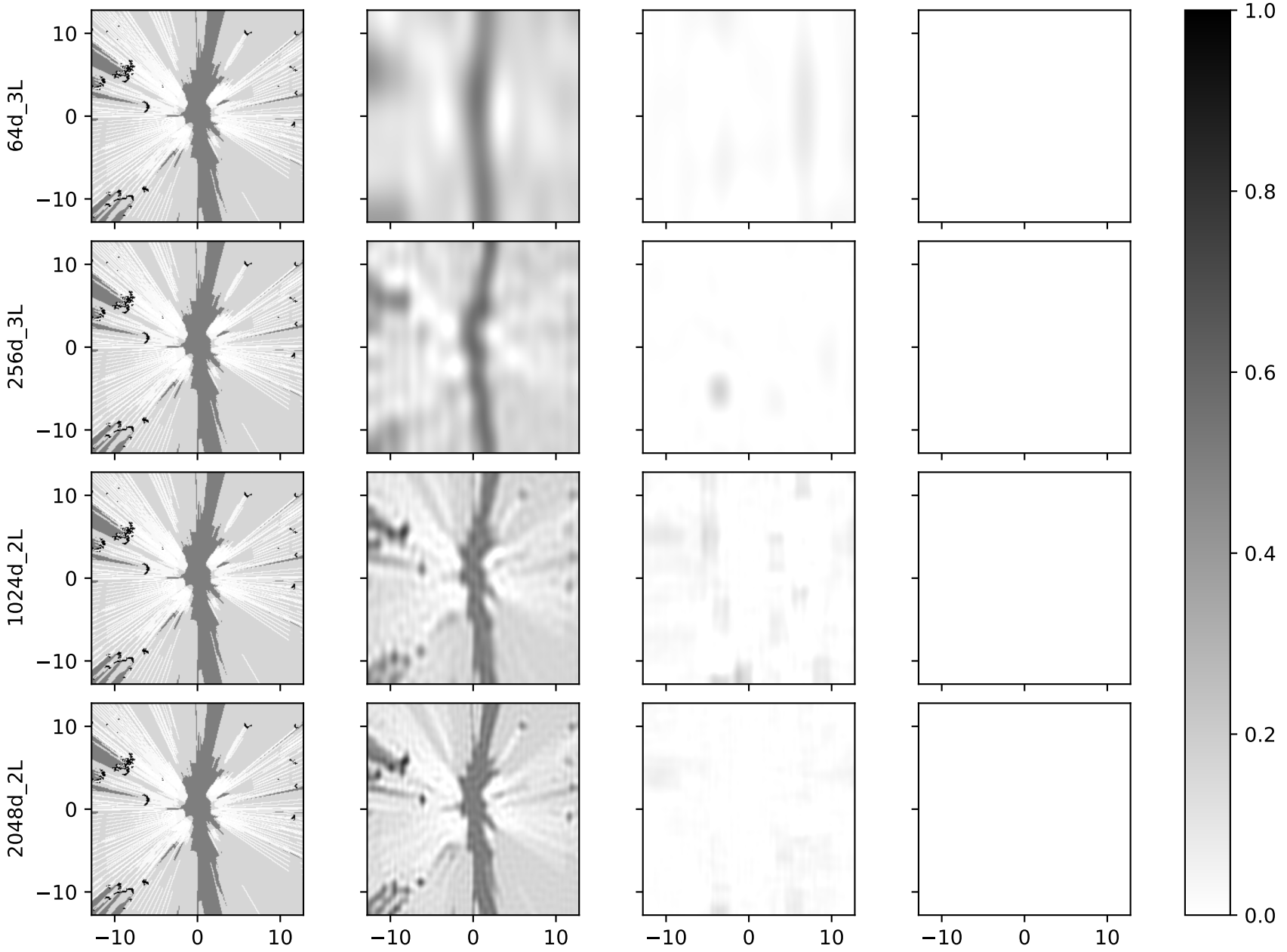

Figure A.3. : De gauche à droite : La grille d'occupation originale, la reconstruction d'une grille d'occupation à partir du spectre utilisé en entrée du réseau de neurones, la segmentation reconstruite à partir du spectre en sortie du réseau de neurones, une binarisation de la reconstruction précédente. De haut en bas : Le réseau à 64 dimensions de spectre et trois couches denses, à 256 dimensions de spectre et trois couches denses, à 1024 dimensions de spectre et deux couches denses et à 2048 dimensions de spectre et deux couches denses. Boîtes englobantes vertes pour les vérités terrain et rouges pour les détections produites par **D2**.

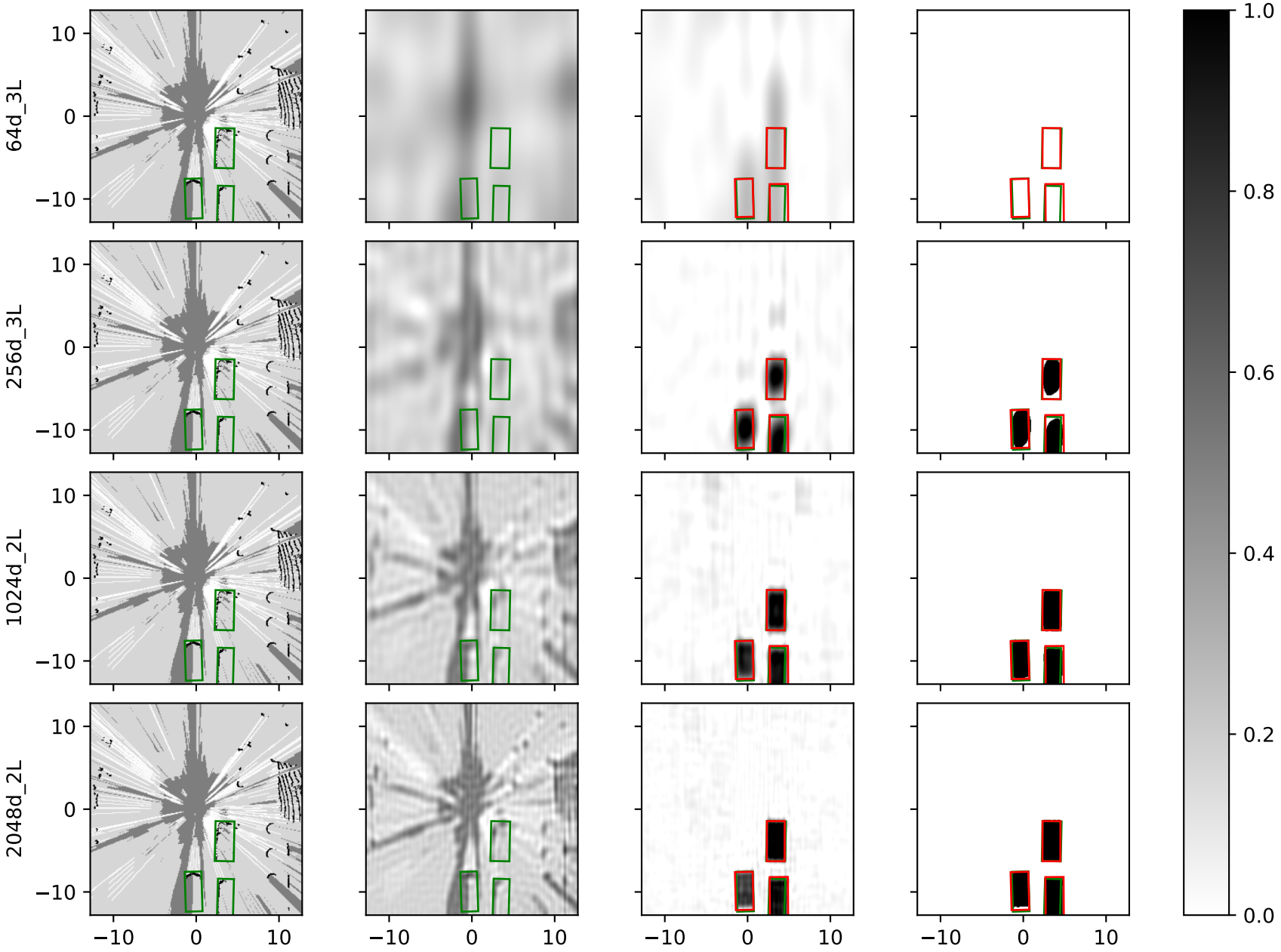

Figure A.4. : De gauche à droite : La grille d'occupation originale, la reconstruction d'une grille d'occupation à partir du spectre utilisé en entrée du réseau de neurones, la segmentation reconstruite à partir du spectre en sortie du réseau de neurones, une binarisation de la reconstruction précédente. De haut en bas : Le réseau à 64 dimensions de spectre et trois couches denses, à 256 dimensions de spectre et trois couches denses, à 1024 dimensions de spectre et deux couches denses et à 2048 dimensions de spectre et deux couches denses. Boîtes englobantes vertes pour les vérités terrain et rouges pour les détections produites par **D2**.

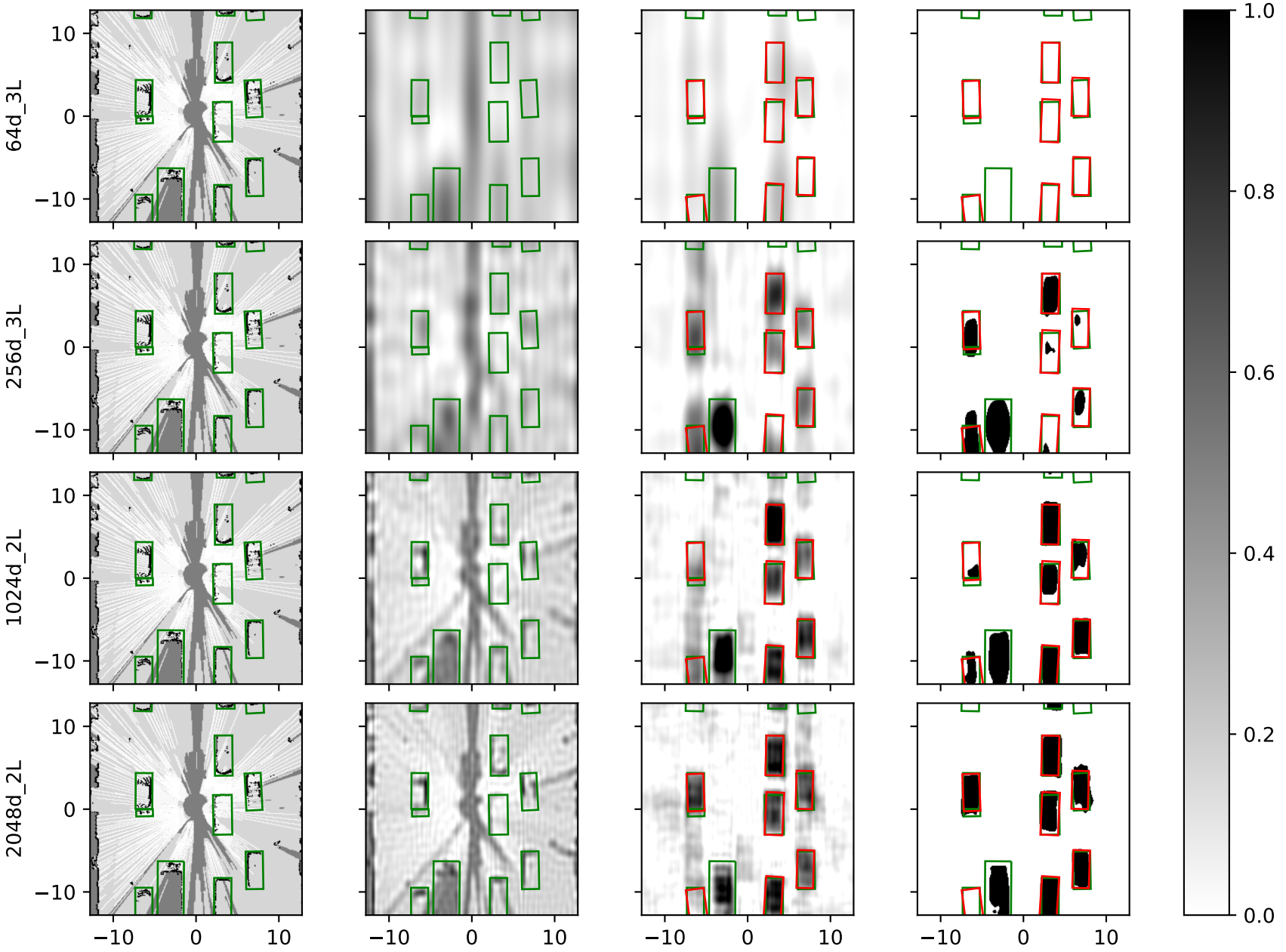

Figure A.5. : De gauche à droite : La grille d'occupation originale, la reconstruction d'une grille d'occupation à partir du spectre utilisé en entrée du réseau de neurones, la segmentation reconstruite à partir du spectre en sortie du réseau de neurones, une binarisation de la reconstruction précédente. De haut en bas : Le réseau à 64 dimensions de spectre et trois couches denses, à 256 dimensions de spectre et trois couches denses, à 1024 dimensions de spectre et deux couches denses et à 2048 dimensions de spectre et deux couches denses. Boîtes englobantes vertes pour les vérités terrain et rouges pour les détections produites par **D2**.

Achevé d'imprimer

Ce manuscrit de thèse de doctorat intitulé *Contributions à la reconnaissance de véhicules dans des grilles d'occupation pour un système de perception embarqué* a été produit à l'aide du système de composition de documents LaTeX à partir du modèle *Clean Thesis* hébergé à l'adresse `http://cleanthesis.der-ric.de/`. La plupart des figures présentes dans ce manuscrit ont été réalisées à l'aide de la bibliothèque logicielle *Matplotlib* ou à l'aide des logiciels de dessin vectoriel en source ouverte *draw.io* et *Inkscape*.

Serment

En présence de mes pairs, parvenu à l'issue de mon doctorat en Automatique - Productique, et ayant ainsi pratiqué, dans ma quête du savoir, l'exercice d'une recherche scientifique exigeante, en cultivant la rigueur intellectuelle, la réflexivité éthique et dans le respect des principes de l'intégrité scientifique, je m'engage, pour ce qui dépendra de moi, dans la suite de ma carrière professionnelle quel qu'en soit le secteur ou le domaine d'activité, à maintenir une conduite intègre dans mon rapport au savoir, mes méthodes et mes résultats.

Grenoble, 8 mars 2024

Nils Defauw