République Algérienne Démocratique et Populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Larbi Tébessi - Tébessa
Faculté des Sciences Exactes et des Sciences de la Nature et de la Vie
Département : Mathématiques et Informatique

Mémoire de fin d'étude

Pour l'obtention du diplôme de MASTER

Domaine : Mathématiques et Informatique
Filière : Informatique
Option : système et multimédia

Thème

Présenté Par

Prédiction de la distance entre les objets dans le domaine de la conduite autonome des véhicules

AISSAOUI Nada

Devant le jury

Dr. Djeddi Chawki	*MCA*	*Université Larbi Tébessi*	*Président*
Dr. Taher Mekhaznia	*MCA*	*Université Larbi Tébessi*	*Examinateur*
Dr. Ahmed Zaggari	*MCA*	*Université Larbi Tébessi*	*Encadreur*
Dr. Taher Gherbi	*MCB*	*Université Larbi Tébessi*	*Co-Encadreur*

Date de soutenance : 25/ 06 / 2023


بسم الله الرحمن الرحيم

Remerciements

Au terme de ce travail, j'aimerais exprimer ma gratitude au bon dieu qui ma donner de force et du courage, du volanté et la patience pour mener à bien ce travail.

Je tiens également à exprimer mon immense reconnaissance envers mon encadrant Ahmed ZEGGARI et Co-encadrant Tahar GHERBI pour leur soutient et ses conseils avisés et sa présence.

Je remercie également les membres des jurys DJEDDI Chawki et Tahar MEKHAZNIA pour leurs temps, leurs expertises et leur évaluation minutieuse de mon travail.

Enfin j'ai exprimé ma gratitude envers toutes les personnes qui en contribué de près ou de loin à l'élaboration de ce travail votre soutient, vos connaissances partagées ont été essentiels à ma réussite.

Dédicace

Je dédie ce travail à mes adorables parents (Djanet et Taoufik) merci chers parents sans lesquels je n'en serais pas là aujourd'hui. Vous êtes toujours été présents pour moi, vous m'avez écoutée, soutenue et encouragée durant toutes ses années. Merci pour les sacrifices que vous avez faits afin que je réalise mes rêves. Vous êtes mes repères. Merci à ma jolie sœur NOUR et adorable frère Hichem que dieu vous protèges

Un grand remercîment ma grande mère que dieu la protège pour tes daawat. Merci à mes oncles (Abderrahman, Mouhamed Chaib, Tahar, Nadjib, Youcef) ainsi leurs épouses

Merci ma très chère tante Saida ses enfants Sabrina, Aicha et Ilyes.

Merci mes tantes Meriem, Hafiza, Zakia, Nassira, Nadjia, Assia, Moufida, et ma cousine Bouchra

Je tiens a remercié mes amis qui en beaucoup sacrifier pour moi (Salsabil, Abdelkarim, Abderazek, Zaineb)

Enfin je remercie chaque personne de près ou de loin qui m'a donné un coup de main merci à tous

Table des matières

Liste des abréviations

IA : Intelligence Artificielle

CNN : Réseaux de neurones convolutifs

RNN : Réseaux de neurones récurrents

GAN : Réseaux de neurones adversariaux génératifs

YOLO: You Only Look Once

ML: Machine Learning

DL: Machine Learning

Liste des tableaux

Liste des figures

Résumé

La conduite autonome est un domaine en pleine expansion, offrant des avantages considérables en termes de sécurité, de commodité et d'efficacité. Les véhicules autonomes sont capables de comprendre leur environnement, de prendre des décisions et d'effectuer des actions de conduite sans intervention humaine directe. Ils peuvent contribuer à réduire les accidents de la route, améliorer la gestion du trafic et faciliter la mobilité pour les personnes à mobilité réduite.

Cependant, la conduite autonome dans des conditions de visibilité réduite présente des défis particuliers. Dans des situations de conduite sombres ou floues, la détection et l'estimation précise de la distance entre les objets deviennent cruciales pour assurer une conduite sûre.

Le projet de fin d'étude se propose donc de développer des méthodes de prédiction de distance robustes et précises dans des conditions de vue sombre ou floue. Pour ce faire, il explore des techniques avancées telles que l'apprentissage profond et les réseaux de neurones convolutifs (CNN) pour améliorer la capacité des véhicules autonomes à percevoir leur environnement dans de telles conditions. L'objectif ultime est d'améliorer la sécurité et la fiabilité des véhicules autonomes dans des situations de conduite difficiles.

Mots clés : YOLO, détection d’objet, estimation de la distance, réseaux de neurones, CNN

Abstract

Autonomous driving is a rapidly expanding field, offering significant benefits in terms of safety, convenience, and efficiency. Autonomous vehicles are capable of understanding their environment, making decisions, and performing driving actions without direct human intervention. They can help reduce road accidents, improve traffic management, and facilitate mobility for people with limited mobility.

However, autonomous driving in low visibility conditions presents particular challenges. In dark or blurry driving situations, accurate detection and estimation of distance between objects become crucial to ensure safe driving.

Therefore, the final year project aims to develop robust and accurate distance prediction methods in dark or blurry viewing conditions. To achieve this, it explores advanced techniques such as deep learning and convolutional neural networks (CNN) to enhance the ability of autonomous vehicles to perceive their environment in such conditions. The ultimate goal is to improve the safety and reliability of autonomous vehicles in challenging driving situations.

Keywords: YOLO, object detection, distance estimation, neural networks, CNN

تلخيص

القيادة الذاتية هي مجال متنامٍ بسرعة، وتوفر فوائد هامة من حيث الأمان والراحة والكفاءة. تستطيع المركبات الذاتية الفهم المحيط بها واتخاذ القرارات وتنفيذ إجراءات القيادة دون تدخل بشري مباشر. يمكن أن تساهم في تقليل حوادث الطرق، وتحسين إدارة حركة المرور، وتسهيل التنقل للأشخاص ذوي القدرات المحدودة.

ومع ذلك، فإن القيادة الذاتية في ظروف رؤية منخفضة تواجه تحديات خاصة. في حالات القيادة المظلمة أو الغير واضحة، يصبح الكشف الدقيق وتقدير المسافة بين الكائنات أمرًا حاسمًا لضمان القيادة الآمنة.

لذلك، يهدف مشروع التخرج إلى تطوير طرق قوية ودقيقة لتوقع المسافة في ظروف رؤية مظلمة أو غير واضحة. ولتحقيق ذلك، يستكشف تقنيات متقدمة مثل التعلم العميق والشبكات العصبية التابعة للتحسين قدرة المركبات الذاتية على إدراك بيئتها في مثل هذه الظروف. الهدف النهائي هو تحسين الأمان والموثوقية للمركبات الذاتية في حالات القيادة التحديّة.

الكلمات المفتاحية: YOLO، كشف الكائنات، تقدير المسافة، شبكات الأعصاب، شبكات الأعصاب التكنولوجية المكتبية

INTRODUCTION GÉNÉRALE

Introduction générale

La conduite autonome est un domaine en constante évolution qui suscite un intérêt croissant de la part des chercheurs, des ingénieurs et des constructeurs automobiles. L'idée de permettre aux véhicules de se déplacer de manière autonome, sans l'intervention humaine, ouvre la voie à de nombreuses possibilités passionnantes en termes de mobilité, de sécurité routière et d'efficacité énergétique. Cependant, malgré les avancées significatives réalisées jusqu'à présent, la conduite autonome présente encore certains défis à relever.

L'un de ces défis concerne la prédiction précise de la distance entre les objets, en particulier dans des conditions de vision sombre et floue. La capacité d'un véhicule autonome à estimer avec précision les distances par rapport aux autres objets sur la route est essentielle pour garantir une conduite sûre et réactive. La détection des obstacles et la prédiction de leur distance sont des éléments clés de la prise de décision autonome du véhicule, lui permettant de s'adapter rapidement aux changements du trafic et de prévenir les collisions.

Pour relever ce défi, l'intelligence artificielle (IA) et l'apprentissage automatique (machine learning) jouent un rôle crucial. L'IA est la discipline qui vise à développer des systèmes capables de réaliser des tâches qui nécessitent normalement l'intelligence humaine, telles que la perception visuelle, la prise de décision et l'apprentissage à partir de l'expérience. L'apprentissage automatique, une sous-branche de l'IA, se concentre spécifiquement sur le développement de modèles et d'algorithmes qui permettent aux machines d'apprendre et de s'améliorer de manière autonome à partir des données.

Dans le contexte de la conduite autonome, l'IA et l'apprentissage automatique sont utilisés pour analyser et interpréter les données provenant de capteurs tels que les caméras, les radars et les lidars. Ces données sont ensuite utilisées pour prédire les distances entre les objets environnants et le véhicule autonome. Les algorithmes d'apprentissage automatique permettent au système de reconnaître des schémas et des caractéristiques spécifiques dans les données, ce qui facilite la prédiction précise des distances.

Dans cette étude, nous nous concentrons spécifiquement sur la prédiction de la distance entre les objets dans des conditions de vision sombre et floue. Ces conditions représentent des scénarios réalistes rencontrés par les conducteurs dans des situations de conduite nocturne, de brouillard dense ou de mauvaise visibilité. L'objectif est de développer un modèle d'apprentissage automatique capable d'estimer de manière fiable la distance entre les objets, en utilisant des données provenant de capteurs spécialement conçus pour des conditions de vision défavorables.

En combinant les avancées technologiques de la conduite autonome, l'IA et l'apprentissage automatique, notre projet de fin d'études vise à contribuer à l'amélioration de la sécurité et de la performance des véhicules autonomes, en particulier dans des conditions de conduite difficiles. En développant un modèle de prédiction de distance précis et fiable, nous espérons ouvrir la voie à une conduite autonome plus sûre et plus efficace, tout en rendant possible une adoption plus large de cette technologie prometteuse dans un avenir proche.

Ce mémoire est structuré en trois chapitres. Le premier chapitre aborde le sujet de traitement d'images et de vision par ordinateur, en soulignant son importance et ses applications variées. Elle met en avant l'importance de l'apprentissage automatique et de l'apprentissage profond dans le développement de techniques avancées de traitement d'images.

Le deuxième chapitre se concentre sur état de l'art. Ce chapitre donne un aperçu des défis liés à la détection d'objets et à l'estimation de la distance dans la conduite autonome, en mettant l'accent sur l'utilisation de l'apprentissage profond. L'objectif est de concevoir des systèmes de conduite autonomes performants et sûrs en combinant ces connaissances.

Le troisième chapitre présente la conception de notre système, en mettant une approche complète pour développer un système de prédiction de distance dans la conduite autonome. Nous décrivons également la collecte des données, l'utilisation du modèle YOLO, l'estimation de distance par un réseau neuronal. Enfin, nous exposons les résultats que nous avons obtenus dans le cadre de notre recherche.

CHAPITRE I

Traitement d'images et de vision par ordinateur

Chapitre 1

Traitement d'images et de vision par ordinateur

Introduction

Le domaine du traitement d'images est en perpétuelle progression, connaissant une croissance remarquable ces dernières années. Son objectif est d'appliquer diverses techniques aux images numériques afin de les améliorer ou d'en extraire des informations pertinentes

Dans la première section de ce chapitre, nous allons examiner la notion d'image numérique et ses diverses caractéristiques, ainsi que les principales méthodes de traitement des images.

La deuxième section explorera les domaines de la vision par ordinateur, de l'apprentissage automatique et de l'apprentissage profond, en mettant l'accent sur des concepts tels que les réseaux de neurones convolutifs (CNN), la classification d'images et la détection d'objets.

1. Le traitement d’image

1.1 L’image numérique

Une image numérique est définie comme une fonction à valeurs discrètes, qui est associée à un pavage rectangulaire de l'espace en pixels. Ce pavage est multidimensionnel, typiquement en 2D ou 3D, et les valeurs peuvent être scalaires, représentant des niveaux de gris, ou vectorielles, pour l'imagerie en couleur ou multi composante. La numérisation d'une image implique une conversion de l'image analogique en une matrice bidimensionnelle de valeurs numériques f (x, y), où chaque pixel est caractérisé par ses propriétés à un niveau de gris ou de couleur. [1]

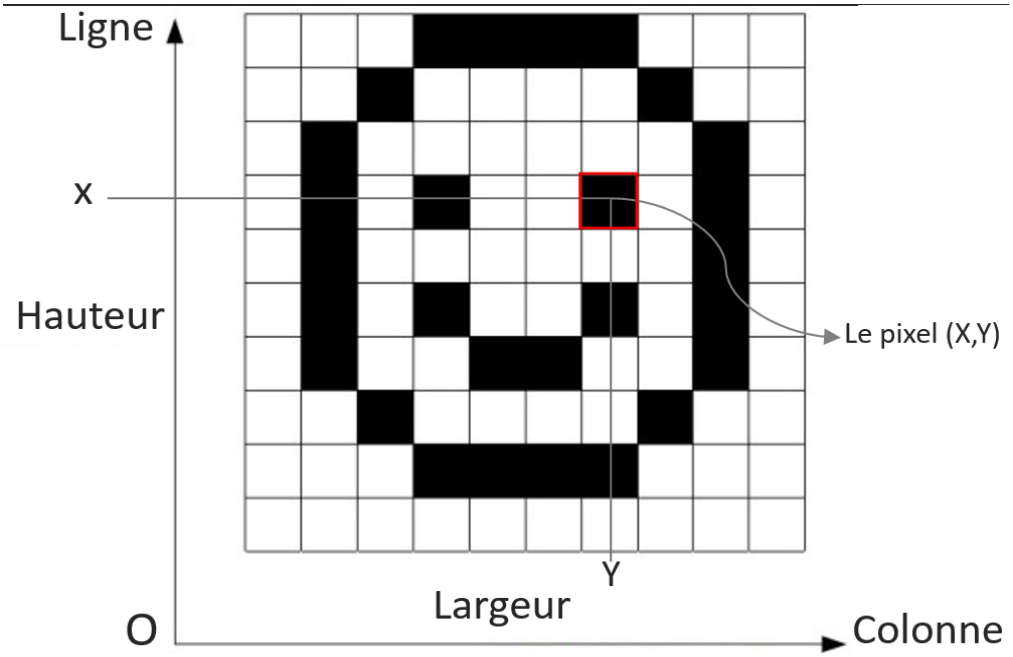

Figure 1: Image numérique

1.2 Les caractéristiques des images numériques

Les paramètres suivants caractérisent un ensemble structuré d'informations qui compose une image :

1.2.1 Dimension

La taille d'une image numérique est déterminée par le nombre total de pixels qu'elle contient. Les pixels sont stockés sous forme de valeurs numériques dans une matrice représentant les intensités lumineuses de chaque pixel. Le nombre de pixels peut être calculé en multipliant le nombre de lignes par le nombre de colonnes de cette matrice. La dimension d'une image numérique, mesurée en pixels, correspond à sa hauteur et sa largeur, également stockées dans une matrice de valeurs numériques représentant les intensités lumineuses de chaque pixel. Cette information est obtenue en multipliant le nombre de colonnes par le nombre de lignes de la matrice. [1]

1.2.2 Résolution

La résolution d'une image numérique correspond à la concentration de pixels présents par unité de surface. Elle est généralement mesurée en points par pouce (PPP), ou DPI en anglais (Dots Per Inch). La qualité et la précision de l'image sont directement influencées par sa résolution. En effet, une résolution plus élevée (avec plus de pixels dans une surface d'un

pouce) permet d'obtenir une image plus nette et de meilleure qualité. La résolution d'une image est mesurée en PPP, c'est-à-dire le nombre de pixels par pouce. Un point est considéré comme un pixel, et la résolution dépend de la taille de l'image. Une résolution plus élevée signifie une image plus fine et plus détaillée, car elle contient plus d'informations (plus de points ou pixels par pouce). [1]

1.2.3 Profondeur

La profondeur de l'image, également appelée la quantité d'informations chromatiques par pixel, est déterminée par le nombre de bits qui y sont affectés. Cette valeur reflète le nombre de niveaux de gris ou de couleurs présents dans l'image. En effet, le nombre de couleurs disponibles et la précision de la représentation des couleurs dans une image sont proportionnels au nombre de bits d'informations par pixel. Ainsi, une image d'une profondeur de 1 bit par pixel ne peut afficher que deux valeurs possibles (noir et blanc), tandis qu'une image avec une profondeur de 8 bits (soit 1 octet) par pixel peut avoir jusqu'à 256 niveaux de gris différents. [3]

1.2.4 Luminance

La luminance est un paramètre qui mesure la luminosité des points d'une image. Elle est déterminée en divisant l'intensité lumineuse d'une surface par sa taille apparente. La brillance est un terme synonyme utilisé pour décrire l'éclat d'un objet, en particulier lorsqu'il est observé de loin. Afin d'obtenir une luminance optimale, il est nécessaire d'avoir des images lumineuses et brillantes, avec un contraste adéquat. Il est crucial d'éviter les images présentant un contraste trop faible ou trop élevé, car cela peut entraîner une perte de détails dans les zones sombres ou lumineuses. En outre, il est essentiel d'éliminer les parasites afin d'obtenir une luminance de qualité. [1]

1.2.5 Histogramme

Un histogramme est une représentation graphique statistique qui illustre la répartition des intensités des pixels d'une image en fonction du nombre de pixels pour chaque intensité lumineuse. Traditionnellement, l'axe horizontal de l'histogramme indique les niveaux d'intensité, allant des tons les plus sombres (à gauche) aux tons les plus clairs (à droite). En utilisant l'histogramme des niveaux de gris ou des couleurs d'une image, on peut obtenir des informations précieuses sur la répartition des niveaux de gris ou de couleur dans une image, ainsi que sur les bornes des valeurs les plus représentatives. Cela permet d'améliorer la qualité

de l'image en introduisant des modifications (rehaussement d'image), ou d'extraire des informations utiles de l'image.[3]

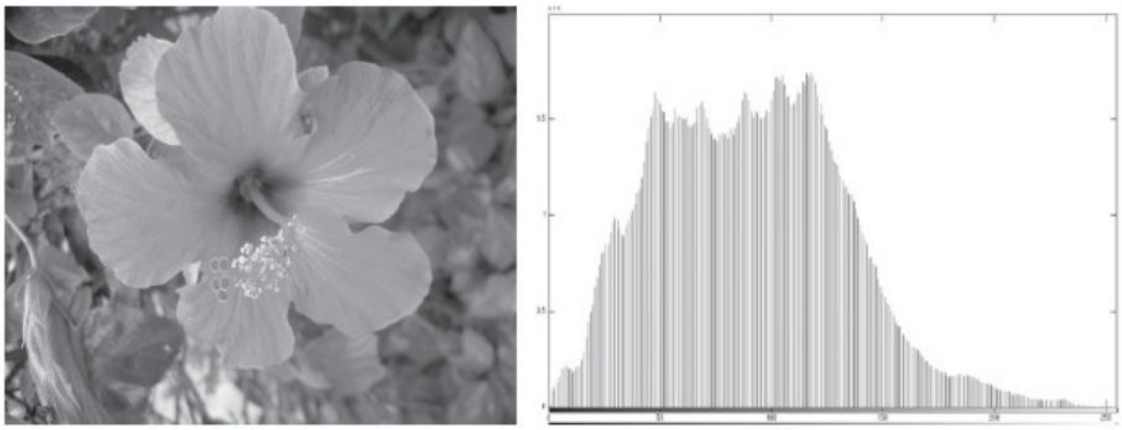

Figure 2: Histogramme d'une image sous Matlab. [4]

1.2.6 Le contracte

Le contraste dans une image est défini comme la différence significative entre les zones sombres et les zones claires, ce qui est basé sur les niveaux de luminosité de deux zones spécifiques de l'image. En somme, le contraste est une mesure de la distinction remarquable entre deux zones distinctes d'une image en fonction de leur niveau de luminosité. [5]

1.3 Codage de couleur

1.3.1 Noir et blanc

Les images binaires, également connues sous le nom d'images noir et blanc, sont des images constituées de pixels qui ne peuvent prendre que deux valeurs distinctes : 0 pour le noir et 1 pour le blanc. Cela signifie que tous les pixels de l'image ne peuvent être que noirs ou blancs. [6]

1.3.2 Niveaux de gris

Le niveau de gris d'une image correspond à l'intensité lumineuse d'un point donné. Les pixels peuvent varier en couleur, allant du noir au blanc en passant par un nombre limité de niveaux intermédiaires. Pour représenter une image en niveau de gris, il est possible d'attribuer

à chaque pixel une valeur qui représente la quantité de lumière reflétée, habituellement située entre 0 et 255. Cette représentation utilise un octet pour chaque pixel, plutôt qu'un seul bit. Le matériel d'affichage doit être capable de produire les différents niveaux de gris nécessaires pour afficher correctement une image en niveau de gris. Les images en niveaux de gris sont couramment utilisées dans des contextes tels que la photographie, les images médicales et les applications de traitement d'image. [6]

1.3.3 Couleur

Une image couleur est créée en combinant plusieurs images en niveaux de gris, appelées composantes, qui sont chacune associées à une couleur primaire : rouge, vert et bleu. Chaque composante représente la quantité de lumière reflétée dans sa couleur respective pour chaque pixel de l'image. La couleur finale d'un pixel est alors obtenue par une synthèse additive de ces trois composantes. Par exemple, si un pixel a des valeurs élevées pour les composantes rouge et vert, mais une valeur faible pour la composante bleue, alors la couleur finale du pixel sera une nuance de jaune.

1.4 Techniques de traitement d'images

La manipulation d'images comprend diverses techniques et méthodes utilisées pour améliorer l'esthétique visuelle d'une image ou extraire des informations pertinentes. Elle regroupe un ensemble de tâches visant à obtenir des données de qualité, à la fois qualitatives et quantitatives, à partir de l'image. [1]

1.4.1 Acquisition

Avant de pouvoir effectuer des opérations sur une image à l'aide d'un système informatique, il est essentiel de la convertir de manière à ce qu'elle puisse être interprétée et modifiée par le système. Cette conversion est réalisée par un processus de numérisation qui implique l'échantillonnage et la quantification de l'image d'origine afin de la représenter de manière interne dans l'unité de traitement. Les caméras vidéo et les appareils photo numériques sont couramment utilisés pour numériser des images. En médecine, des imageurs d'échodoppler, d'échographie ou de scintigraphie sont utilisés pour numériser les images médicales et les rendre accessibles aux systèmes informatiques pour une analyse plus poussée.[1]

1.4.2 Filtrage

Le filtrage d'une image est une méthode permettant de supprimer le bruit présent dans l'image. Cette technique utilise des algorithmes mathématiques tels que l'interpolation ou la morphologie mathématique pour ajuster chaque pixel de l'image en fonction des informations locales extraites de son environnement. Pour ce faire, l'image est convoluée avec un noyau prédéfini qui contient un modèle de transformation linéaire ou non linéaire. Le résultat est une nouvelle image filtrée qui a une apparence améliorée. En somme, cette technique consiste à créer une nouvelle image en modifiant les valeurs des pixels de l'image d'origine.[7]

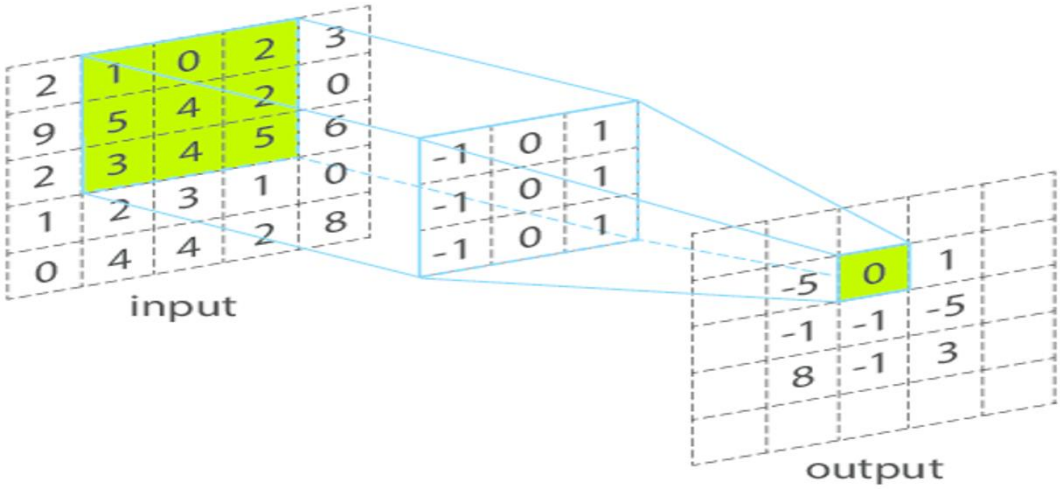

Figure 3: Filtrage d'une image. [9]

1.4.3 Segmentation

La segmentation est une technique de traitement d'image qui consiste à répartir les pixels de l'image en groupes homogènes, formant ainsi des régions de l'image. L'objectif est de séparer de manière précise les différents objets présents dans l'image pour extraire de l'information utile. Cette méthode permet de diviser l'image en plusieurs zones d'intérêt distinctes, où chaque zone est composée de pixels connectés partageant des caractéristiques similaires telles que l'intensité ou la texture. Il existe différentes formes de segmentation, regroupées en trois catégories principales : la segmentation basée sur les propriétés des pixels, la segmentation basée sur les caractéristiques régionales et la segmentation basée sur les contours. [7]

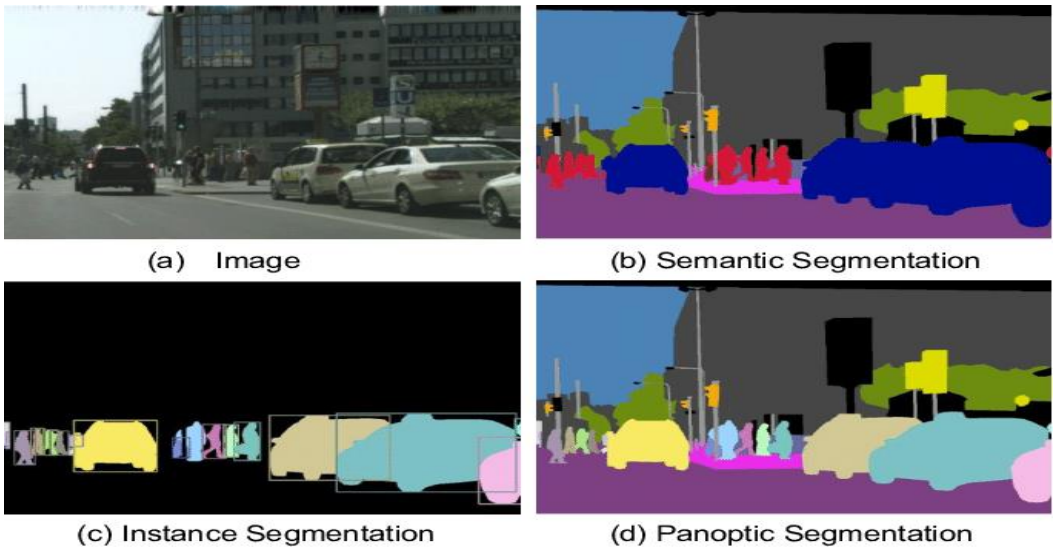

Figure 4: La segmentation d'une image. [10]

2. La vision par l'ordinateur

2.1 Définition

La vision par ordinateur, également connue sous le nom de "computer vision" en anglais, est un domaine pluridisciplinaire de l'informatique qui se focalise sur la capacité des ordinateurs à percevoir, identifier et traiter les images de manière similaire à la vision humaine. Pour ce faire, il fait appel à des capteurs tels que des caméras, des ordinateurs et des algorithmes d'apprentissage automatique. L'objectif de la vision par ordinateur est d'extraire des informations complexes à partir des images afin de les utiliser dans d'autres processus. Ce domaine englobe des spécialités telles que la classification d'images et la détection d'objets, et vise à automatiser des tâches qui reproduisent les capacités de vision et de compréhension humaines. [8]

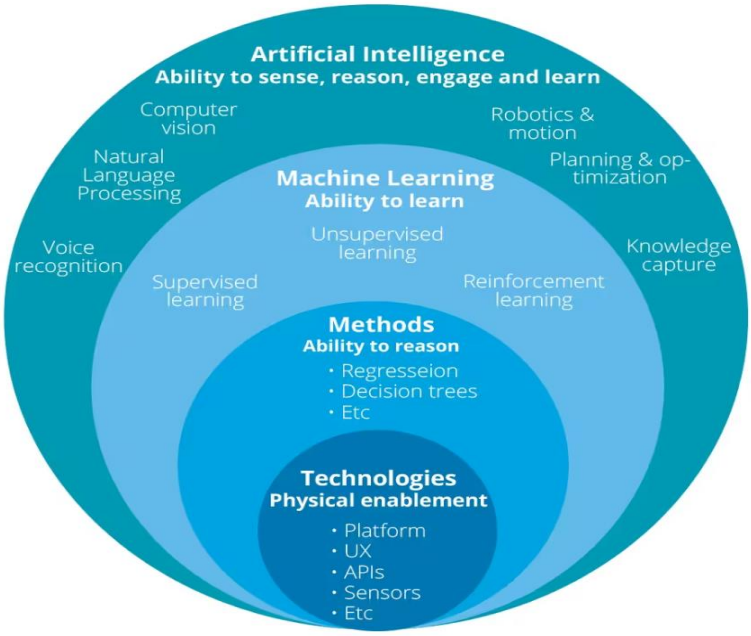

Figure 5: Présentation de l'intelligence artificielle. [9]

2.2 Apprentissage automatique

L'apprentissage automatique, également connu sous le nom de "Machine Learning", est un domaine de l'intelligence artificielle qui permet aux ordinateurs d'apprendre à partir de données sans être explicitement programmés pour chaque tâche. Il se compose de deux phases : la première phase utilise des données d'apprentissage pour calculer un modèle, tandis que la seconde phase consiste à utiliser ce modèle pour prédire des résultats sur de nouvelles données.[1]

L'apprentissage automatique permet aux machines de s'améliorer progressivement en apprenant à partir de données fournies en entraînement et de prendre des décisions ou de fournir des prédictions sur des données similaires à celles qu'elles ont apprises auparavant. Il existe différents types d'apprentissage automatique, tels que l'apprentissage supervisé, non supervisé et par renforcement, chacun utilisant des techniques et des algorithmes spécifiques. [18]

L'apprentissage automatique est utilisé dans divers domaines tels que la reconnaissance vocale, la détection de fraudes, la recommandation de produits, la classification d'images, etc. Dans le champ de la classification d'images, les descripteurs caractéristiques sont employés comme une base de données permettant au modèle d'acquérir la capacité de classer le contenu des images. Un exemple serait la reconnaissance de la présence d'un chat dans une image. [1]

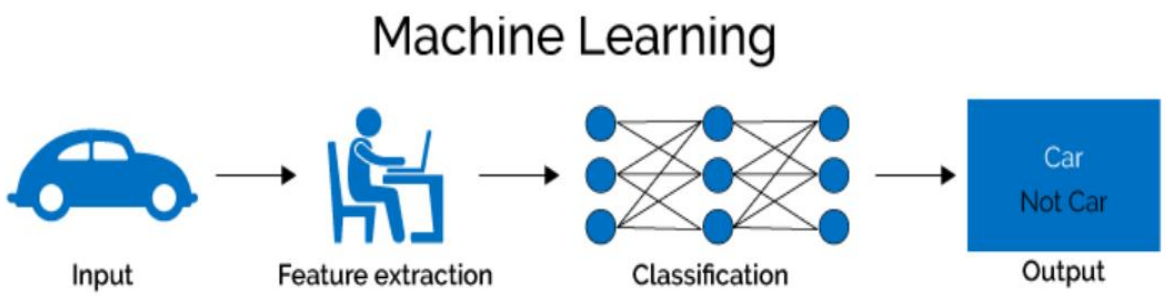

Figure 6: Présentation de ML. [13]

2.3 Apprentissage profond

L'apprentissage profond (ou "Deep Learning" en anglais) est une technique d'apprentissage automatique qui utilise des réseaux de neurones artificiels pour résoudre des problèmes complexes en extrayant des caractéristiques de plus en plus abstraites et complexes à partir des données d'entrée. Contrairement aux modèles linéaires ou peu profonds utilisés dans l'apprentissage automatique traditionnel, les architectures de réseaux de neurones profonds utilisées dans l'apprentissage profond comportent plusieurs couches cachées. Ces couches cachées permettent de modéliser des données de manière hautement abstraite, permettant ainsi aux modèles d'apprentissage profond d'obtenir des performances remarquables dans des domaines tels que la reconnaissance d'image, la reconnaissance vocale et la traduction automatique. Les couches d'un réseau de neurones profonds peuvent inclure des couches de convolution, des couches de mise en commun, des couches de normalisation, des couches entièrement connectées et des couches de perte. [15]

Le domaine du Deep Learning englobe une variété d'algorithmes, dont les réseaux de neurones profonds qui se distinguent par leur structure à plusieurs couches cachées, contrairement aux réseaux de neurones artificiels traditionnels. Parmi ces algorithmes figurent également les réseaux de neurones récurrents et les réseaux de neurones convolutifs. [19]

Notre recherche se concentre spécifiquement sur l'étude des réseaux de neurones convolutifs, une technique largement utilisée dans le domaine de la classification d'images et de la détection d'objets en apprentissage profond.

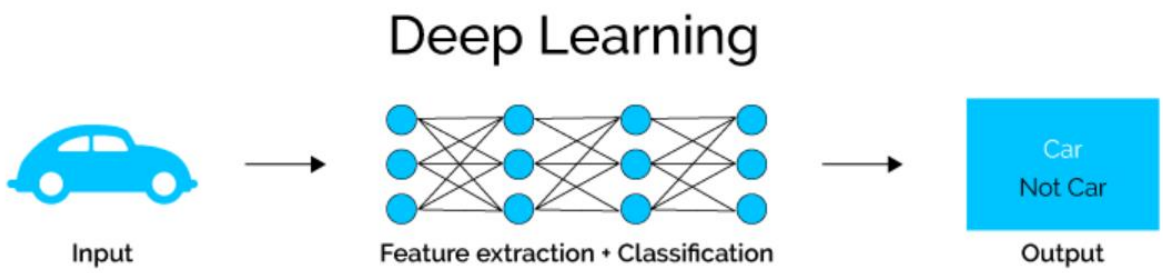

Figure 7: Présentation de DL. [13]

2.3.1. Les réseaux de neurones convolutifs

Les réseaux de neurones à convolution (CNN) sont une structure couramment employée en vision par ordinateur dans le but de réaliser différentes tâches, notamment la classification, la segmentation et la détection d'objets, au sein des réseaux de neurones profonds. Les CNN sont inspirés du système visuel biologique des animaux et ont été conçus pour détecter des motifs dans les données d'entrée. Les réseaux de neurones convolutifs sont spécifiquement conçus pour le traitement des signaux bidimensionnels. Ils ont été initialement proposés par Yann LeCun et ses collègues en 1989. [1]

Un CNN est généralement divisé en deux parties principales. La partie convolutive, qui constitue la première étape, sert d'extracteur de caractéristiques pour les images. En passant par une série de filtres, ou noyaux de convolution, l'image est transformée en de nouvelles images appelées cartes de convolutions. Ces filtres permettent de détecter des motifs tels que des bords, des coins et des contours dans l'image. La couche de mise en commun est utilisée pour réduire la taille spatiale de la représentation de l'image et ainsi réduire le nombre de paramètres dans le modèle, ce qui rend le modèle plus robuste aux variations mineures dans l'image. La couche de normalisation est utilisée pour normaliser les activations de la couche précédente afin de réduire la variance et d'augmenter la stabilité du modèle.

Le CNN comporte une deuxième partie composée de couches de perceptrons multicouches entièrement connectées. Cette étape consiste à fusionner les caractéristiques extraites par le code CNN afin de procéder à la classification de l'image. Le résultat du code CNN provenant de la section convolutive est fusionné en un vecteur de caractéristiques, puis transmis en tant qu'entrée à la deuxième partie du réseau. La couche de sortie est constituée d'un neurone par catégorie, et les valeurs numériques obtenues sont fréquemment normalisées entre 0 et 1 afin de former une distribution de probabilités pour les différentes catégories. Cette normalisation garantit que la somme des valeurs soit égale à 1. Les réseaux de neurones

convolutifs (CNN) ont apporté une transformation majeure dans le domaine de la vision par ordinateur, ouvrant la voie à des progrès importants dans des domaines tels que la reconnaissance d'objets, la classification d'images et la segmentation d'images. [16]

I. Les couches d'un Réseau Neurones Convolutives

A. La couche de convolution (CONV)

La couche de convolution, également appelée couche CONV, est une composante clé des réseaux de neurones convolutifs (CNN) utilisés en vision par ordinateur. Elle applique une opération de convolution entre une fenêtre de la matrice d'entrée (généralement une image) et une matrice de poids (filtre) pour détecter des motifs tels que des bords, des coins et des contours dans l'image. En appliquant une série de filtres de convolution sur l'image d'entrée, la couche CONV extrait des caractéristiques de niveau supérieur qui sont utiles pour la classification d'images et la détection d'objets. [1]

B. La couche de correction (ReLU)

La couche de correction (ReLU) est une fonction d'activation souvent utilisée dans les réseaux de neurones profonds, y compris les réseaux de neurones convolutifs (CNN) utilisés en vision par ordinateur. La fonction ReLU est une fonction non linéaire qui prend en entrée la somme pondérée des entrées d'une couche de neurones et renvoie la valeur maximale entre 0 et l'entrée. [1]

Plus précisément, la fonction ReLU retourne 0 si l'entrée est négative et renvoie l'entrée elle-même si elle est positive ou nulle. Cela signifie que la fonction ReLU ne modifie pas les entrées positives et supprime simplement les entrées négatives, ce qui lui permet d'agir comme un seuil qui filtre les entrées négatives. [17]

La fonction ReLU est souvent préférée à d'autres fonctions d'activation en raison de sa simplicité et de son efficacité. Elle permet également de résoudre le problème de la disparition du gradient, qui peut se produire avec d'autres fonctions d'activation telles que la fonction sigmoïde ou la fonction tangente hyperbolique. Le problème de la disparition du gradient peut rendre l'apprentissage plus difficile dans les réseaux de neurones profonds en réduisant la mise à jour des poids dans les couches précédentes.

C. La couche de pooling (POOL)

La couche de pooling (ou couche de mise en commun en français) est une composante clé des réseaux de neurones convolutifs (CNN) utilisés en vision par ordinateur. Cette couche intervient généralement après la couche de convolution et a pour objectif de réduire la taille spatiale de la représentation de l'image en conservant les informations les plus importantes. Cette réduction de taille permet de diminuer le nombre de paramètres dans le modèle, de rendre le modèle plus léger et plus rapide, et d'éviter le surapprentissage. [1]

La couche de pooling peut être implémentée de différentes manières, mais la plus courante est la mise en commun maximale (max pooling). Cette opération consiste à diviser l'image en zones de pooling et à extraire le maximum de chaque zone pour créer une nouvelle carte d'activation réduite. Par exemple, si la zone de pooling est de 2x2, chaque groupe de 4 pixels de l'image initiale est remplacé par la valeur maximale de ces 4 pixels. Cette opération permet de conserver les caractéristiques les plus importantes de l'image, tout en réduisant la taille de la représentation.

Il existe également d'autres méthodes de pooling, comme la mise en commun moyenne (average pooling), qui calcule la moyenne des valeurs dans chaque zone de pooling, ou la mise en commun L2, qui calcule la racine carrée de la somme des carrés des valeurs dans chaque zone. Cependant, la mise en commun maximale est la plus couramment utilisée car elle donne de meilleurs résultats dans la plupart des cas.

D. La couche de entièrement connectée (FC)

La couche entièrement connectée (FC) est une couche dans les réseaux de neurones artificiels où chaque nœud ou neurone de la couche est connecté à tous les nœuds de la couche précédente. Dans cette couche, chaque entrée est considérée comme un vecteur à une dimension et est connectée à chaque neurone de la couche FC. Les couches entièrement connectées sont souvent utilisées dans les dernières couches des réseaux de neurones pour effectuer des classifications ou des prédictions. Elles permettent d'extraire des caractéristiques complexes à partir des couches précédentes et de les combiner pour obtenir une sortie finale. Cependant, les couches entièrement connectées peuvent également entraîner une forte complexité de modèle et un risque de surapprentissage. Les architectures modernes de réseaux de neurones, telles que les réseaux de neurones convolutifs, utilisent moins de couches entièrement connectées pour éviter ces problèmes. [12]

E. La couche de pert (LOSS, Softmax)

La couche de perte (ou "couche de coût") est une couche qui calcule l'écart entre la prédiction du modèle et la vérité terrain (ou "ground truth") pour une tâche donnée. Cette couche est souvent utilisée dans les réseaux de neurones pour entraîner le modèle et ajuster les poids des différentes couches. [1]

Dans le contexte de la classification, la couche de perte est souvent associée à la fonction Softmax, qui calcule une distribution de probabilités sur les différentes classes d'objets. La fonction Softmax prend en entrée un vecteur de scores (ou "logits") et produit en sortie une distribution de probabilités normalisée sur les différentes classes. La couche de perte va alors comparer cette distribution de probabilités prédite par le modèle à la distribution de probabilités "vérité terrain" et calculer l'écart entre les deux.

Il existe différentes fonctions de perte qui peuvent être utilisées en fonction de la tâche à accomplir et de la nature des données. Par exemple, pour la classification binaire, la fonction de perte la plus courante est l'entropie croisée binaire, tandis que pour la classification multi-classes, on utilise généralement l'entropie croisée catégorielle.

II. Les architectures de CNN

A. GoogLeNet

GoogLeNet est une architecture de CNN utilisée par Google pour remporter la tâche de classification ILSVRC 2014. Elle a été développée par une équipe dirigée par Jeff Dean, Christian Szegedy et Alexandro Szegedy. Cette architecture a montré une réduction significative du taux d'erreur par rapport aux vainqueurs précédents, AlexNet (vainqueur ILSVRC 2012) et ZF-Net (vainqueur ILSVRC 2013), et son taux d'erreur est nettement inférieur à celui de VGG (finaliste 2014). GoogLeNet atteint une architecture plus profonde en utilisant plusieurs techniques distinctes, telles que la convolution 1x1 et le pooling global moyen. Cependant, cette architecture est coûteuse en termes de calcul. [21]

Afin de réduire les paramètres qui doivent être appris, des couches de déconvolution importantes sont utilisées sur les CNN pour supprimer la redondance spatiale pendant l'entraînement, ainsi que des connexions raccourcies entre les deux premières couches convolutionnelles avant d'ajouter de nouveaux filtres dans les couches CNN ultérieures. Des exemples réels d'applications de l'architecture de CNN de GoogLeNet incluent la

reconnaissance de chiffres Street View House Number (SVHN), qui est souvent utilisée comme proxy pour la détection d'objets en bordure de route. Ci-dessous se trouve le schéma de bloc simplifié représentant l'architecture du réseau de neurones convolutifs GoogLeNet.

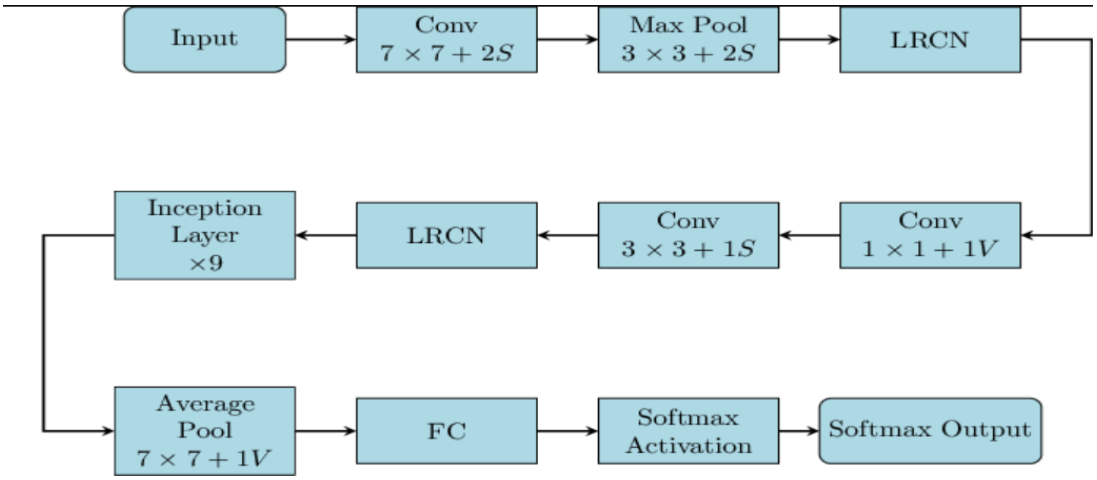

Figure 8: Architecture du réseau de neurones convolutifs GoogLeNet. [14]

B. VGGNet

VGGNet est une architecture CNN de 16 couches développée par Karen Simonyan et Andrew Zisserman à l'université d'Oxford. Avec jusqu'à 95 millions de paramètres et entraîné sur plus d'un milliard d'images, VGGNet peut prendre des images d'entrée de taille 224 x 224 pixels avec 4096 caractéristiques convolutionnelles. Cependant, les CNN avec des filtres aussi grands sont coûteux à entraîner et nécessitent beaucoup de données, ce qui explique pourquoi des architectures comme GoogLeNet fonctionnent mieux pour la plupart des tâches de classification d'images. Malgré cela, le modèle VGGNet reste une base solide pour de nombreuses applications en vision par ordinateur, telles que la détection d'objets, et ses représentations de caractéristiques profondes sont utilisées dans plusieurs architectures de réseaux neuronaux. [21] Le diagramme ci-dessous représente le schéma d'architecture standard du réseau VGG16 :

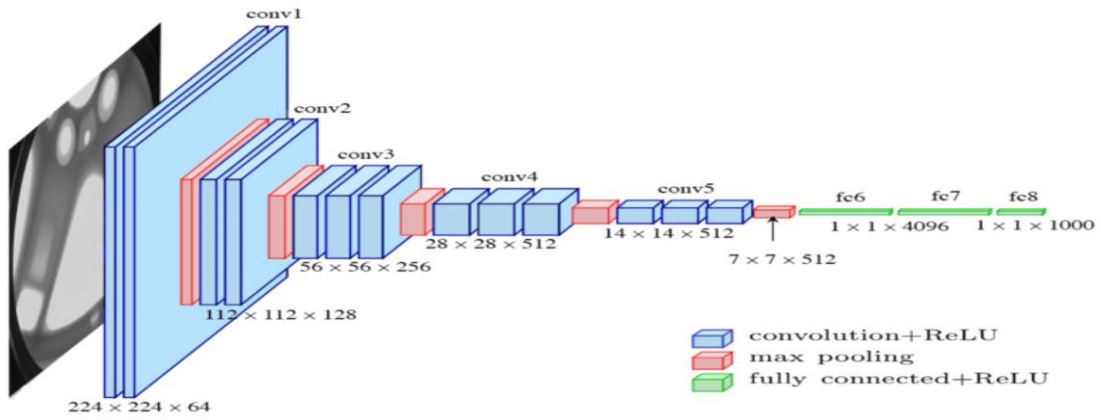

Figure 9: Architecture standard du réseau VGG16

Conclusion

Ce chapitre a couvert différents aspects clés du traitement d'images et de la vision par ordinateur, en mettant l'accent sur la détection d'objets et l'estimation de la distance dans le contexte des véhicules autonomes.

Nous avons commencé par explorer les bases de l'image numérique et des techniques de traitement d'images. Cela nous a permis de comprendre les fondements nécessaires pour aborder les concepts avancés de la vision par ordinateur et de l'apprentissage automatique.

Ensuite, nous avons plongé dans les domaines de la vision par ordinateur, de l'apprentissage automatique et de l'apprentissage profond, en nous concentrant sur des concepts tels que les réseaux de neurones convolutifs (CNN), la classification d'images et la détection d'objets. Ces techniques sont essentielles pour permettre aux véhicules autonomes de comprendre leur environnement et de prendre des décisions en conséquence.

Enfin, nous avons souligné les multiples domaines d'application de la vision par ordinateur et des techniques de traitement d'images, tels que la gestion du trafic, la santé, la surveillance et les véhicules autonomes. Ces applications démontrent l'ampleur des possibilités offertes par ces technologies.

Dans les prochains chapitres, nous approfondirons ces concepts et techniques, en explorant des sujets tels que la détection d'objets, l'estimation de la distance et d'autres aspects clés de la conduite autonome. En renforçant notre compréhension de ces domaines, nous contribuerons à

l'avancement des technologies de conduite autonome et à la création d'un avenir où les véhicules pourront fonctionner de manière autonome et en toute sécurité.

CHAPITRE II

Etat de l'art

Chapitre 2

Etat de l’art

Introduction

Ce chapitre examine les aspects fondamentaux de la conduite autonome, en se concentrant sur la détection des objets dans l'environnement et l'estimation précise de leur distance. Il met particulièrement l'accent sur l'utilisation de l'apprentissage profond dans ce domaine. La détection d'objets en conduite autonome révolutionne notre compréhension de l'environnement et l'anticipation des actions à prendre. L'estimation précise de la distance est cruciale et les méthodes basées sur l'apprentissage profond utilisent les caractéristiques visuelles des objets pour prédire cette distance avec précision.

Ce chapitre traite des sources de données pour l'apprentissage de la détection d'objets et de l'estimation de la distance dans les véhicules autonomes, ainsi que des différents capteurs utilisés. Il met l'accent sur l'importance de comprendre les avantages et les limitations de chaque source de données pour concevoir des systèmes performants. De plus, il explore les caractéristiques pertinentes pour la prédiction de distance et l'importance de séparer les ensembles d'entraînement, de validation et de test pour évaluer les performances des modèles. Ensuite, nous nous pencherons sur la sélection d'une architecture de réseau de neurones adaptée à la prédiction de distance, en explorant les avantages et les limitations des architectures telles que les CNN et les RNN. Nous couvrirons les bases de la conduite autonome et de l'apprentissage profond. De plus, nous examinerons les travaux antérieurs sur la prédiction de la distance entre les objets pour situer notre recherche dans le contexte existant. Nous couvrirons les bases de la conduite autonome et de l'apprentissage profond. De plus, nous examinerons les travaux antérieurs sur la prédiction de la distance entre les objets pour situer notre recherche dans le contexte existant.

1. La détection d'objets

La détection d'objets est une méthode avancée de vision par ordinateur qui intègre le traitement d'images, l'intelligence artificielle et l'apprentissage automatique afin d'identifier, localiser et encadrer des objets spécifiques dans des images ou des vidéos. Cette technologie permet de repérer les objets présents dans une scène donnée et de comprendre leur mouvement. Contrairement à la reconnaissance d'images, la détection d'objets recourt à des approches plus sophistiquées pour analyser et interpréter les données visuelles. [1]

Dans le contexte de l'analyse d'une image spécifique, notre objectif est de localiser les zones susceptibles de contenir un objet. Une fois que ces régions ont été identifiées, nous les isolons individuellement et les soumettons à un modèle de classification d'images. Les régions de l'image d'origine qui obtiennent de bons résultats de classification sont conservées, tandis que les autres sont écartées. Ainsi, pour parvenir à une méthode efficace de détection d'objets, il est crucial de disposer à la fois d'un algorithme robuste de détection des régions et d'un algorithme performant de classification des images. [23]

Divers types d'objets en mouvement peuvent être extraits des régions de mouvement, tels que des êtres humains, des véhicules, des animaux et d'autres objets. Ci-dessous, vous trouverez un tableau présentant quelques points de comparaison entre diverses méthodes de classification d'objets :

Tableau 1: Comparaison entre des méthodes de classification d'objets

Méthodes	Un niveau de précision	Temps de calcule	Commentaires
Basé sur la couleur	Elevé	Elevé	• En raison de sa faible précision, il n'est pas toujours approprié. • Les algorithmes ont un coût de calcul réduit.
Basé sur la texture	Elevé	Elevé	• Besoin de temps de calcul supplémentaire. • Permet d'obtenir une qualité supérieure.
Basé sur la forme	Raisonnable	Faible	• Cette méthode manque d'efficacité dans les situations dynamiques et ne peut pas détecter les mouvements internes. • Il peut être mis en œuvre en utilisant des modèles adaptés.
Basé sur le mouvement	Raisonnable	Elevé	• Il est difficile d'identifier un être humain qui ne bouge pas. • Il n'est pas nécessaire d'avoir des modèles de motifs prédéfinis.

Les méthodes de détection d'objets nécessitent toutes un mécanisme permettant d'identifier les objets d'intérêt dans une séquence vidéo, que ce soit dans chaque image ou lors

de la première apparition de l'objet. Cette étape essentielle du processus de suivi d'objet consiste à repérer et à regrouper les pixels correspondant à ces objets. Étant donné que les objets en mouvement fournissent généralement la principale source d'informations, la plupart méthodes se concentrent sur la détection de ces objets.

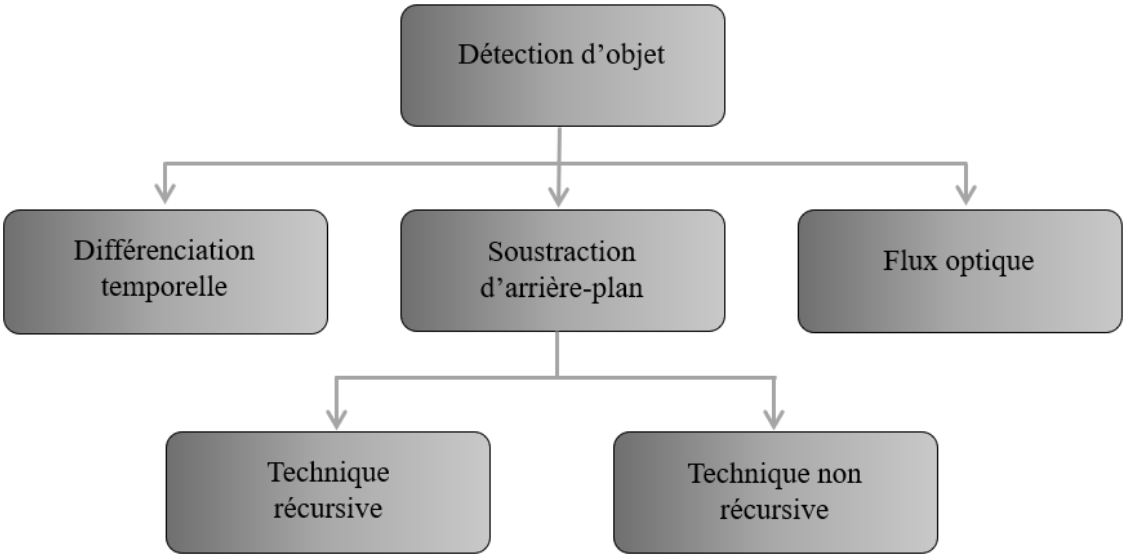

Figure 10: Les techniques de détections

2. Estimation de la distance

L'estimation de la distance, quant à elle, vise à déterminer la distance réelle entre la caméra et les objets détectés. Cette information est cruciale pour de nombreuses applications, notamment pour éviter les collisions dans la conduite autonome ou pour calculer la profondeur dans la réalité augmentée. Il existe plusieurs approches pour estimer la distance dans le cadre de la vision par ordinateur.

L'une des méthodes couramment utilisées pour l'estimation de la distance est la stéréovision. La stéréovision repose sur le principe de triangulation, où deux caméras sont utilisées pour capturer des images d'une scène à partir de points de vue légèrement différents. En comparant les disparités entre les images, il est possible de calculer la profondeur des objets dans la scène. Cette méthode est largement utilisée dans les systèmes de vision stéréoscopique et a prouvé son efficacité pour estimer les distances avec précision.

Une autre approche populaire est l'utilisation de capteurs de profondeur tels que les caméras à temps de vol (Time-of-Flight) ou les capteurs LiDAR (Light Detection and Ranging).

Ces capteurs émettent des signaux lumineux et mesurent le temps qu'il faut pour que ces signaux rebondissent sur les objets et reviennent au capteur. En utilisant cette mesure de temps, il est possible de déterminer la distance entre le capteur et les objets. Les capteurs de profondeur sont largement utilisés dans les véhicules autonomes et les systèmes de réalité virtuelle pour obtenir des estimations précises de la distance.

En outre, certaines techniques d'apprentissage en profondeur peuvent également être utilisées pour l'estimation de la distance. Par exemple, certaines architectures de réseaux de neurones, telles que les réseaux de neurones à convolution profonde (DCNN) ou les réseaux de neurones récurrents (RNN), peuvent être entraînées à prédire la distance en se basant sur des images d'entraînement annotées avec des informations de profondeur. Ces réseaux peuvent apprendre des caractéristiques visuelles et spatiales pour estimer la distance des objets dans de nouvelles images.

3. Concepts de base de la conduite autonome et de l'apprentissage profond

3.1. Capteurs

Les véhicules autonomes sont équipés de capteurs tels que des caméras, des lidars, des radars et des capteurs ultrasoniques pour percevoir leur environnement.

3.2. Perception

Les données des capteurs sont utilisées pour détecter et reconnaître les objets et les obstacles dans l'environnement routier, tels que les véhicules, les piétons, les panneaux de signalisation, les feux de circulation, etc.

3.3. Localisation

Les véhicules autonomes doivent être capables de déterminer leur position précise sur la route à l'aide de systèmes de positionnement global (GPS), de centrales inertielles (IMU), de balises ou de cartes haute résolution.

3.4. Planification

Une fois que le véhicule a perçu son environnement et s'est localisé, il doit planifier une trajectoire sûre et efficace pour atteindre sa destination. Cela implique la prise de décisions en

temps réel, la gestion des obstacles et des contraintes de la route, ainsi que la navigation dans le trafic.

3.5. Contrôle

Une fois que la trajectoire est planifiée, le véhicule autonome utilise des systèmes de contrôle pour ajuster sa vitesse, son accélération, sa direction et d'autres paramètres afin de suivre la trajectoire prévue et de réagir aux conditions changeantes de la route.

4. Les jeux de données

Dans cette partie on va parler sur les dataset disponible dans le domaine de détection des objets et l'estimation de la distance et petite comparaison entre eux :

4.1. PASCAL Dataset

Le jeu de données Pascal est une collection populaire utilisée dans le domaine de la vision par ordinateur. Il tire son nom du concours annuel PASCAL Visual Object Classes, qui vise à détecter et à classifier les objets présents dans des images. Le jeu de données Pascal comprend des milliers d'images annotées, provenant d'une variété de scènes et contenant différentes catégories d'objets tels que des voitures, des personnes, des animaux, des meubles, etc. [24]

4.2. COCO Dataset

L'ensemble de données COCO (Common Objects in Context) est une référence largement utilisée dans la recherche en vision par ordinateur pour les tâches de détection, de segmentation et de sous-titrage d'objets. Il fournit une collection à grande échelle d'images qui ont été annotées avec des informations détaillées sur les objets qui y sont présents. [25]

4.3. Cityscapes Dataset

Le Cityscapes Dataset est une collection de données pour la segmentation sémantique dans les scènes urbaines. Il comprend des images haute résolution annotées pour différents types d'objets et de classes de la route, ce qui est utile pour l'analyse de l'environnement urbain dans la conduite autonome. [26]

4.4. CamVid Dataset

Le jeu de données CamVid est une ressource populaire utilisée dans le domaine de la vision par ordinateur pour la tâche de segmentation sémantique. Il s'agit d'un ensemble de données annotées qui fournit des images et des masques de segmentation correspondants pour différents scénarios de conduite. [27]

4.5. KITTI Dataset

Le KITTI Dataset est un ensemble de données largement utilisé pour la perception et la vision par ordinateur dans la conduite autonome. Il comprend des séquences de vidéos, des données lidar, des données GPS et des annotations pour des tâches telles que la détection d'objets, la reconnaissance de panneaux de signalisation, la localisation, etc. [28]

4.6. Comparaison entre les dataset

Tableau 2: comparaison entre les Datasets

Nom	Usage	Année	Classe	Data	Résolution	Echantillons (entrainement)	Echantillons (validation)	Echantillons (test)
PASCAL	Générique	2012	21	2D	Variable	1464	1449	Privée
COCO	Générique	2014	+80	2D	Variable	82783	40504	81434
Cityscapes	Urban	2015	30 (8)	2D	2048x1024	2975	500	1525
CamVid	Urban (Driving)	2009	32	2D	960x720	367	101	233
KITTI	Urban (Driving)	2012	3	2D	Variable	7481	7518	7521

5. Extraction des caractéristiques

Dans cette partie on va parler de choix des caractéristiques pertinentes pour la prédiction de distance :

5.1. Données sensorielles

Les données provenant de différents capteurs tels que les caméras, les lidars, les radars et les capteurs ultrasoniques fournissent des informations essentielles pour la prédiction de distance. Les caractéristiques spécifiques extraites de ces données peuvent inclure les coordonnées spatiales des objets, leurs dimensions, leur vitesse, leur accélération, leur orientation, etc.

5.2. Profondeur et distance relative

La perception de la distance peut être basée sur des informations de profondeur fournies par des capteurs tels que les lidars et les radars. La distance relative entre le véhicule autonome et les objets environnants est une caractéristique importante pour prédire la distance.

5.3. Caractéristiques visuelles

Les caractéristiques visuelles extraites à partir d'images, telles que les couleurs, les textures, les formes et les contours des objets, peuvent être utilisées pour prédire la distance. Des techniques d'apprentissage profond telles que les réseaux de neurones convolutifs peuvent être utilisées pour extraire ces caractéristiques à partir des images

5.4. Mouvement et trajectoire

L'analyse du mouvement des objets et la prédiction de leur trajectoire future sont des aspects importants pour estimer la distance. Les caractéristiques telles que la vitesse, l'accélération, la direction de déplacement et les modèles de mouvement peuvent être utilisées pour prédire la distance entre les objets.

5.5. Contexte environnemental

Les informations contextuelles sur l'environnement routier peuvent également être prises en compte. Cela peut inclure des caractéristiques telles que le type de route (autoroute, route urbaine, intersection, etc.), les panneaux de signalisation, les feux de circulation, les marquages au sol, les obstacles statiques (bâtiments, arbres, etc.) qui peuvent influencer la distance entre les objets.

6. Division de données

La séparation des ensembles d'entraînement, de validation et de test :

6.1. Séparation aléatoire

Dans cette approche, l'ensemble de données se divise aléatoirement en ensembles d'entraînement, de validation et de test. Par exemple, une répartition de 70% peut être assignée à l'entraînement, 15% à la validation et 15% au test. Cependant, la répartition précise dépend de la taille de l'ensemble de données et de la complexité du problème.

6.2. Séparation temporelle

Lorsque les données sont collectées dans le temps, il est avisé de segmenter les ensembles d'entraînement, de validation et de test selon l'ordre chronologique des données. Les données les plus anciennes peuvent être utilisées pour l'entraînement, les plus récentes pour la validation et les plus récentes encore pour le test. Cette approche simule des conditions réelles où le modèle doit être évalué sur des données inédites.

6.3. Validation croisée

Dans cette approche, l'ensemble de données se divise en plusieurs partitions de taille similaire. Une de ces partitions sert de jeu de test, une autre de jeu de validation, et les autres constituent le jeu d'entraînement. On répète ce processus en changeant l'ordre des partitions à plusieurs reprises, puis on fait la moyenne des performances pour obtenir une évaluation plus fiable du modèle. La validation croisée se révèle particulièrement utile lorsque les données sont limitées.

6.4. Méthodes spécifiques au domaine

Dans le domaine de la conduite autonome, des considérations spécifiques se posent quant à la séparation des ensembles de données. Par exemple, les données peuvent être divisées en fonction de différents types de scènes de conduite, de conditions météorologiques, de périodes de la journée, etc. Cette pratique vise à garantir que le modèle puisse généraliser dans diverses situations, indépendamment de sa dimensionnalité.

7. Model de réseau de neurones

Dans cette partie on va parles de la sélection d'une architecture de réseau de neurones adaptée

7.1. Couches d'entrée

Pour les données d'image, l'utilisation d'une couche de convolution permet l'extraction des caractéristiques visuelles, suivie d'une couche de mise en commun (pooling) pour réduire la dimensionnalité.

7.2. Couches intermédiaires

Plusieurs couches de convolution et de normalisation peuvent être utilisées pour capturer des informations hiérarchiques et des motifs complexes. L'utilisation de blocs résiduels (residual blocks) peut également faciliter l'apprentissage en profondeur.

7.3. Couches de sortie

Pour la prédiction de distance, l'utilisation de une ou plusieurs couches entièrement connectées est possible afin d'estimer la distance entre le véhicule autonome et les objets environnants.

7.4. Fonctions d'activation

Les fonctions d'activation non linéaires telles que ReLU (Rectified Linear Unit) peuvent être employées entre les couches pour incorporer des non-linéarités dans le modèle. Des couches supplémentaires peuvent aussi être ajoutées pour prédire d'autres informations pertinentes, comme la vitesse relative ou l'estimation de la trajectoire.

7.5. Rétropropagation et optimisation

Les techniques de rétropropagation du gradient peuvent être employées pour l'entraînement du réseau de neurones, avec la minimisation d'une fonction de perte appropriée comme l'erreur quadratique moyenne (MSE) ou l'erreur absolue moyenne (MAE). Pour l'optimisation, des algorithmes tels que la descente de gradient stochastique (SGD), des variantes de l'optimisation basée sur le moment (comme Adam), ou d'autres algorithmes d'optimisation avancés peuvent être utilisés.

8. Les travaux connexes

Dans cette partie on va donner les travaux antérieurs sur la prédiction de la distance entre objets:

8.1. "Deep Multi-modal Object Detection and Semantic Segmentation for Autonomous Driving

Datasets, Methods, and Challenges" (2018) par Xiaozhi Chen et al. : Ce travail propose une approche combinant la détection d'objets et la segmentation sémantique en utilisant des réseaux de neurones profonds pour la prédiction de la distance entre objets dans des scènes de conduite autonomes. [29]

8.2. "Deep Driving: Convolutional Neural Networks for Autonomous Driving" (2015)

Deep Driving (2015) par Mariusz Bojarski et al.: Les auteurs utilisent un réseau de neurones convolutifs profond pour prédire la distance entre le véhicule autonome et les objets environnants en utilisant des données de caméras embarquées. Ils montrent que leur approche permet une prédiction précise des distances dans différentes situations de conduite. [30]

8.3. "Monocular Distance Estimation with Hierarchical Multi-Scale Deep Networks"

Ce travail propose une méthode de prédiction de distance basée sur des réseaux de neurones profonds à plusieurs échelles hiérarchiques par Zhihui Wang et al (2017). Ils utilisent des images monoculaires pour estimer la distance entre le véhicule autonome et les objets environnants, en prenant en compte les informations à différentes résolutions spatiales. [31]

9. Apprentissage profond pour la conduite autonome

9.1. Réseaux de neurones convolutifs (CNN)

Les CNN sont largement utilisés pour la perception visuelle dans la conduite autonome. Ils sont efficaces pour extraire des caractéristiques pertinentes à partir d'images ou de données spatiales, tels que la détection d'objets, la segmentation sémantique, la reconnaissance de panneaux de signalisation, etc. [32]

9.2. Réseaux de neurones récurrents (RNN)

Les RNN sont utilisés pour modéliser des dépendances temporelles dans les données, ce qui est important pour des tâches telles que la prédiction de trajectoire, la planification de mouvement et la prédiction de comportement des autres véhicules. Les architectures de RNN, telles que les réseaux LSTM (Long Short-Term Memory) et GRU (Gated Recurrent Unit), sont souvent utilisées dans ce contexte. [32]

9.3. Réseaux de neurones adversariaux génératifs (GAN)

Les GAN sont utilisés pour générer des données synthétiques réalistes et diversifiées qui peuvent être utilisées pour enrichir les ensembles de données d'entraînement et améliorer les performances des modèles. Les GAN peuvent être utilisés pour générer des images de scènes

de conduite, des données lidar ou radar, ou pour augmenter la variabilité des données existantes. [32]

9.4. Apprentissage par renforcement

L'apprentissage par renforcement est utilisé pour apprendre des politiques de contrôle de conduite en utilisant des récompenses et des pénalités. Les agents d'apprentissage par renforcement peuvent apprendre à naviguer dans un environnement routier en prenant des décisions basées sur l'état actuel du véhicule et les informations sensorielles. [33]

9.5. Comparaison entre CNN et RNN

Tableau 3: Comparaison entre CNN et RNN

	CNN	RNN
Les Avantages	• Efficaces pour la reconnaissance spatiale et la détection de motifs. • Partage des paramètres pour une efficacité computationnelle. • Invariance de translation pour une reconnaissance position indépendante. • Apprentissage hiérarchique des caractéristiques. • Possibilité d'utiliser des modèles pré-entraînés et le transfert d'apprentissage.	• Adaptés à la modélisation de données séquentielles. • Gestion des entrées de longueur variable. • Compréhension du contexte et des dépendances à long terme. • Utilisés pour la modélisation et la génération de langage. • Traitement en ligne et en temps réel.
Les Inconvénients	• Inadaptés au traitement de données séquentielles. • Exigent des entrées de taille fixe. • Compréhension limitée du contexte à longue portée. • Coût computationnel élevé pour les grands modèles. • Difficulté à traiter les données irrégulières.	• Coût computationnel élevé pour les longues séquences. • Problème de disparition et d'explosion des gradients. • Difficulté à capturer des dépendances très longues. • Limitations de la parallélisation des calculs. • Difficulté à traiter les sorties de longueur variable.

Conclusion

Dans ce chapitre, nous avons exploré plusieurs aspects clés de la détection d'objets et de l'estimation de la distance dans le contexte de la conduite autonome et de l'apprentissage profond. Nous avons vu comment la détection d'objets constitue une étape fondamentale pour percevoir l'environnement et prendre des décisions en conséquence. De plus, l'estimation précise de la distance entre les objets et le véhicule autonome est essentielle pour assurer une conduite sûre et fiable.

Nous avons également examiné les différentes sources de données disponibles, telles que les capteurs lidar, les caméras et les radars, ainsi que les techniques de fusion de données pour obtenir une perception plus complète de l'environnement.

En conclusion, ce chapitre a jeté les bases nécessaires pour aborder la détection d'objets et l'estimation de la distance dans le domaine de la conduite autonome. En comprenant les concepts et les techniques clés, nous sommes prêts à poursuivre notre exploration des travaux antérieurs sur la prédiction de la distance entre les objets, ainsi que des techniques d'apprentissage profond utilisées dans ce domaine. Ces connaissances nous permettront d'approfondir notre compréhension et de développer des solutions plus avancées pour la conduite autonome.

CHAPITRE III

Résultat et expérimentations

Chapitre 3

Résultat et expérimentations

Introduction

Ce chapitre explore les différentes étapes nécessaires à la mise en place de ce système, depuis la collecte des données jusqu'à l'évaluation des résultats obtenus.

La première étape essentielle est la collecte de données. Un ensemble de données de haute qualité est nécessaire pour former le modèle de détection d'objet et d'estimation de distance. Ensuite, nous examinons le prétraitement des données, où nous appliquons des techniques telles que le redimensionnement, la normalisation et la suppression du bruit pour améliorer la qualité des données et faciliter l'apprentissage.

Le modèle de détection d'objet est ensuite présenté, mettant en évidence l'utilisation du modèle YOLO (You Only Look Once). Nous discutons des raisons derrière ce choix et explorons en détail l'architecture de ce modèle, en soulignant ses avantages en termes d'efficacité et de précision.

L'estimation de la distance est un aspect clé de notre système. Nous décrivons les techniques utilisées, y compris l'utilisation d'un réseau neuronal, pour prédire avec précision la distance entre la caméra et les objets détectés.

Enfin, nous discuterons des résultats d'estimation de distance obtenus grâce à notre système intelligent de détection d'objet. Nous analyserons les performances du modèle et les défis rencontrés, ainsi que les perspectives d'amélioration pour de futures recherches.

Dans l'ensemble, ce dernier chapitre présente une approche complète et détaillée pour développer un système intelligent de détection d'objet et d'estimation de distance. Les résultats obtenus démontrent l'efficacité de notre approche et ouvrent la voie à de nombreuses applications potentielles dans des domaines tels que la conduite autonome, la sécurité routière et la surveillance vidéo.

1. Système intelligent de la détection d'objet et l'estimation distance

1.1. Collecte de données

Tout d'abord, vous devez collecter un ensemble de données contenant des images d'objets avec des étiquettes de classe (par exemple, voiture, personne, etc.) et des informations de distance associées à chaque objet dans l'image.

1.2. Prétraitement des données

Vous devrez effectuer un prétraitement sur les images et les informations de distance pour les préparer à l'entraînement du modèle. Cela peut inclure le redimensionnement des images, la normalisation des valeurs de distance et l'encodage des étiquettes de classe.

1.3. Modèle de détection d'objet

Utilisez un réseau de neurones convolutifs (CNN) pour effectuer la détection d'objet dans les images. Vous pouvez utiliser des architectures populaires comme YOLO (You Only Look Once) ou SSD (Single Shot MultiBox Detector) qui sont connues pour leur précision et leur efficacité en termes de temps d'exécution. Ces modèles vous permettent de détecter les objets dans une image et de prédire leur classe et leur position approximative.

1.4. Estimation de la distance

Une fois que vous avez détecté les objets dans l'image, vous pouvez utiliser les informations de distance associées à chaque objet pour estimer leur distance réelle par rapport à la caméra. Vous pouvez utiliser différentes techniques, telles que la stéréovision, la triangulation ou l'apprentissage par régression, pour estimer la distance en fonction des caractéristiques des objets détectés.

1.5. Entraînement du modèle

Entraînez votre modèle en utilisant l'ensemble de données préparé dans les étapes précédentes. Vous devrez définir une fonction de perte appropriée qui prend en compte à la fois la détection d'objet et l'estimation de la distance. Vous pouvez utiliser des techniques d'apprentissage supervisé pour entraîner votre modèle.

1.6. Évaluation du modèle

Une fois que votre modèle est entraîné, évaluez sa performance en utilisant un ensemble de données de test. Mesurez la précision de détection des objets ainsi que l'erreur d'estimation de distance.

1.7. Optimisation et ajustement du modèle

Selon les résultats de l'évaluation, vous pouvez ajuster les hyperparamètres de votre modèle, tels que la taille de l'architecture du réseau, les fonctions de perte, les taux d'apprentissage, etc., afin d'améliorer les performances du modèle.

2. Contribution

L'architecture suivante représente notre contribution

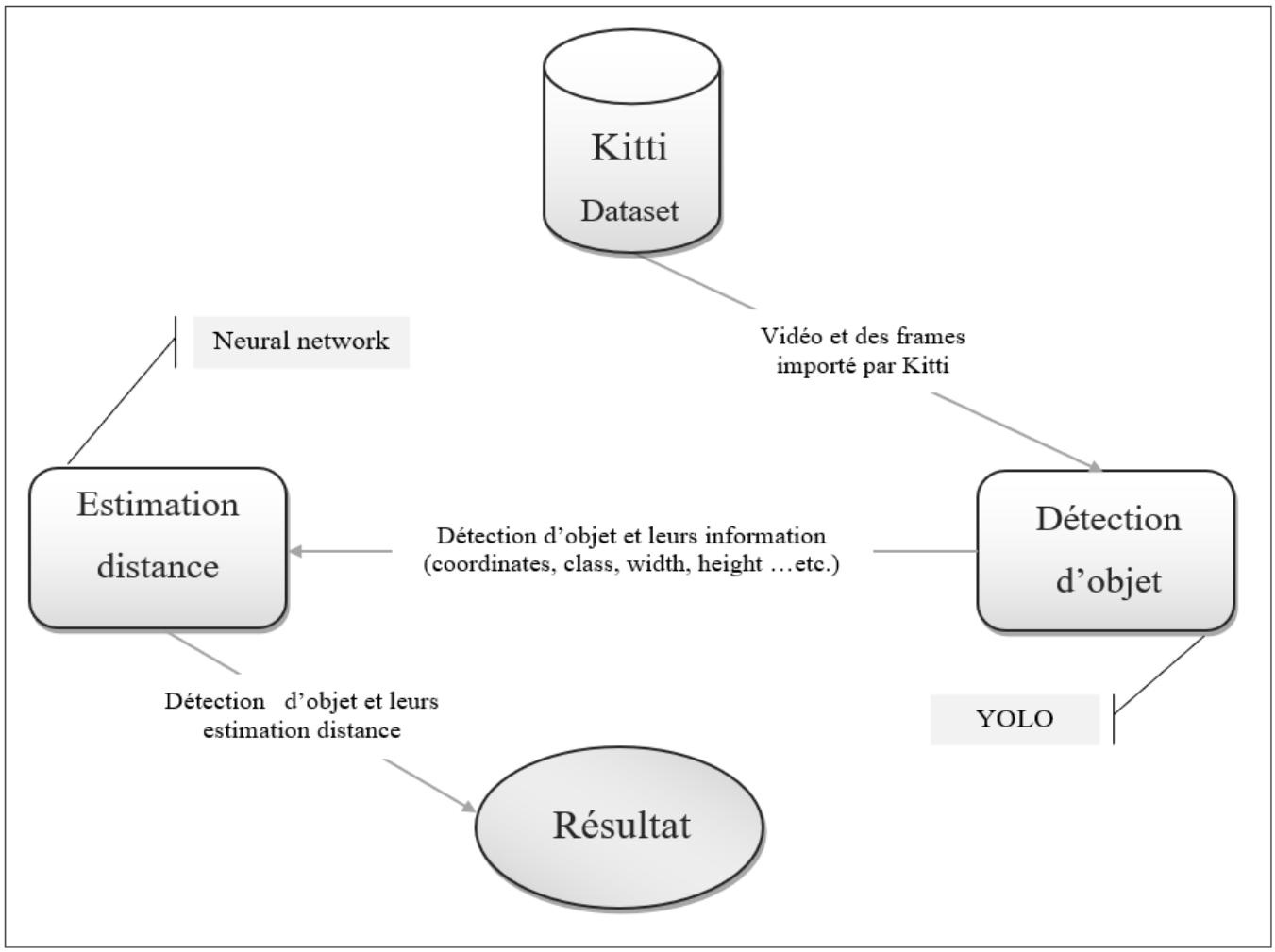

Figure 11 Architecture

L'architecture de notre contribution se compose de deux étapes principales : la détection d'objets à l'aide de l'algorithme YOLO (You Only Look Once) et l'estimation de la distance en utilisant un réseau neuronal.

Dans la première étape, nous utilisons YOLO pour détecter et localiser les objets présents dans une image. YOLO divise l'image en une grille et attribue à chaque cellule de la grille la responsabilité de prédire les boîtes englobantes des objets détectés. Cette approche nous permet d'obtenir une détection rapide et précise des objets, ce qui est essentiel pour notre architecture.

Une fois les objets détectés, nous passons à la deuxième étape qui consiste à estimer leur distance. Nous avons développé un réseau neuronal spécifique à cette tâche. Ce réseau est entraîné à partir d'un ensemble de données préalablement annotées, où les distances réelles entre les objets et la caméra sont connues. En utilisant ces données, le réseau peut apprendre des relations complexes entre les caractéristiques des objets détectés et leurs distances réelles. Ainsi, lors de l'exécution en temps réel, le réseau peut estimer la distance des objets à partir des caractéristiques extraites lors de la détection d'objets.

Cette approche combinant la détection d'objets avec YOLO et l'estimation de la distance à l'aide de notre réseau neuronal offre plusieurs avantages. Elle permet une détection rapide et précise des objets dans une scène, ainsi qu'une estimation précise de leur distance. Nous pouvons envisager d'appliquer cette architecture dans divers domaines tels que la conduite autonome, la surveillance vidéo ou la robotique.

3. Détection des objets

3.1 Choix du model

3.1.1 YOLO

YOLO est l'abréviation du terme "You Only Look Once". Il s'agit d'un algorithme qui détecte et reconnaît divers objets dans une image (en temps réel). La détection d'objets dans YOLO se fait comme un problème de régression et fournit les probabilités de classe des images détectées. L'algorithme YOLO utilise des réseaux de neurones convolutionnelles (CNN) pour détecter des objets en temps réel. Comme son nom l'indique, l'algorithme ne nécessite qu'une seule propagation vers l'avant à travers un réseau de neurones pour détecter des objets. Cela

signifie que la prédiction dans l'image entière est effectuée en une seule exécution d'algorithme. Le CNN est utilisé pour prédire simultanément diverses probabilités de classe et boîtes englobantes.

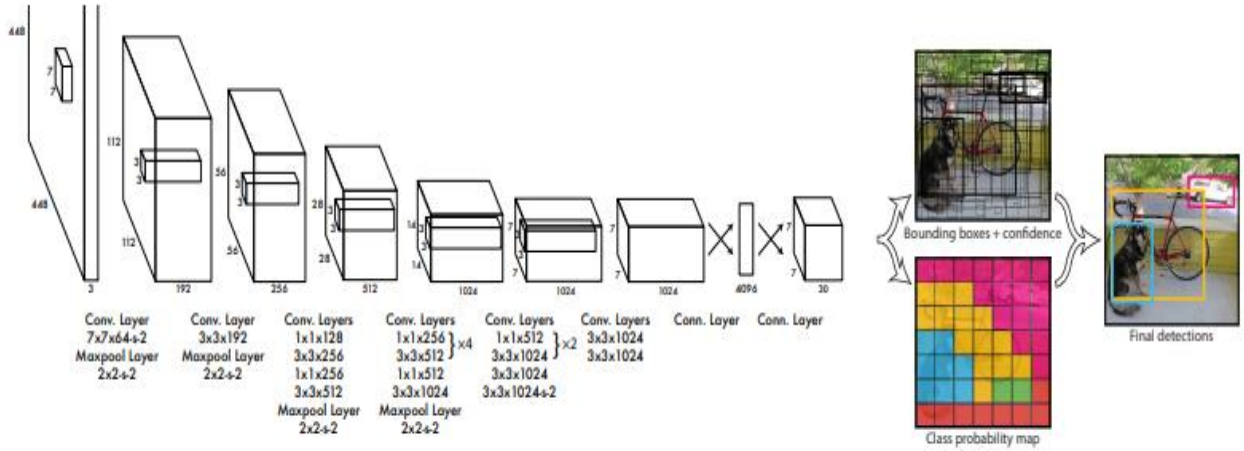

Figure 12: Architecture YOLO. [34]

L'algorithme YOLO est caractérisé par : [35]

- **Vitesse :** cet algorithme améliore la vitesse de détection car il peut prédire les objets en temps réel.
- **Haute précision :** YOLO est une technique prédictive qui fournit des résultats précis avec un minimum d'erreurs de fond.
- **Capacités d'apprentissage :** l'algorithme possède d'excellentes capacités d'apprentissage qui lui permettent d'apprendre les représentations d'objets et de les appliquer à la détection d'objets.

3.1.2 Fonctionnement de YOLO

YOLO suit les étapes suivant dans leur fonctionnement : [35]

A. Blocs résiduels

Tout d'abord, l'image est divisée en différentes grilles. Chaque grille a une dimension de S x S. L'image suivante montre comment une image d'entrée est divisée en grilles.

Dans l'image ci-dessus, il existe de nombreuses cellules de grille de dimension égale. Chaque cellule de la grille détectera les objets qui y apparaissent. Par exemple, si un centre d'objet apparaît dans une certaine cellule de la grille, cette cellule sera responsable de sa détection.

Figure 13: Blocs résiduels. [36]

B. Régression de boîte englobante

Une boîte englobante est un contour qui met en évidence un objet dans une image.

Chaque cadre de délimitation de l'image se compose des attributs suivants :

- Largeur (pc)
- Hauteur (bh)
- Classe (par exemple, personne, voiture, feu de circulation, etc.) - Ceci est représenté par la lettre c.
- Centre de la boîte englobante (bx, by)

L'image suivante montre un exemple de zone de délimitation. La boîte englobante a été représentée par un contour jaune.

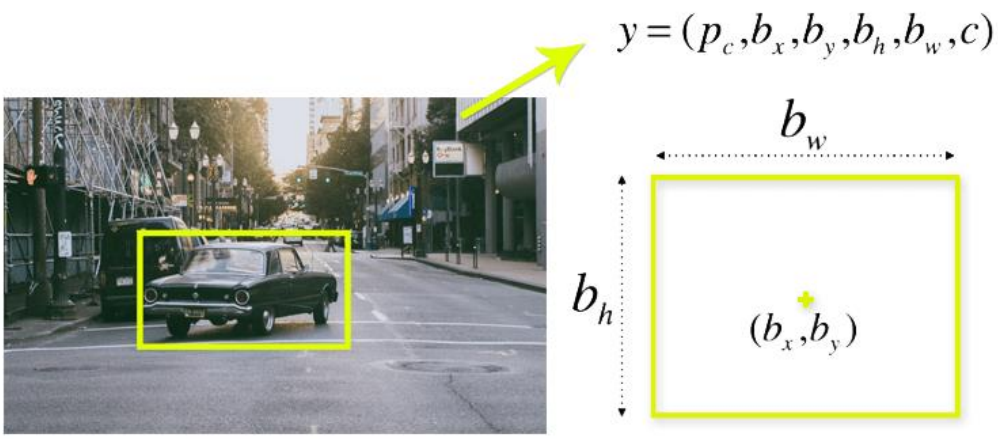

Figure 14: Régression de boîte englobante. [37]

YOLO utilise une seule régression de boîte englobante pour prédire la hauteur, la largeur, le centre et la classe des objets. Dans l'image ci-dessus, représente la probabilité qu'un objet apparaisse dans la boîte englobante.

C. Intersection sur l'union (IOU)

L'intersection sur l'union (IOU) est un phénomène de détection d'objet qui décrit comment les boîtes se chevauchent. YOLO utilise IOU pour fournir une boîte de sortie qui entoure parfaitement les objets.

Chaque cellule de la grille est chargée de prédire les boîtes englobantes et leurs scores de confiance. L'IOU est égal à 1 si la boîte englobante prédite est la même que la boîte réelle. Ce mécanisme élimine les boîtes englobantes qui ne sont pas égales à la boîte réelle.

L'image suivante fournit un exemple simple du fonctionnement de l'IOU.

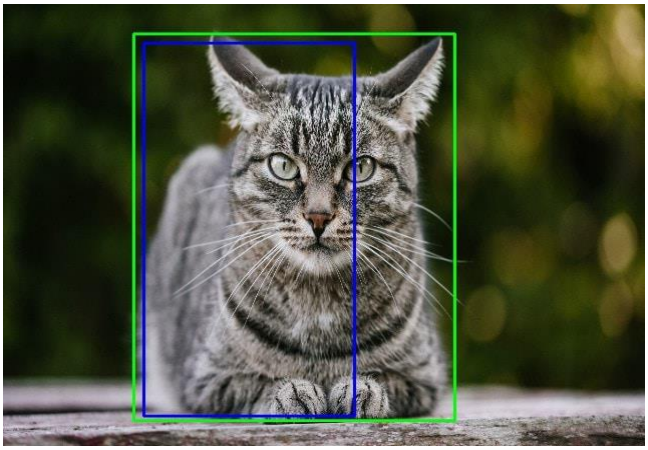

Figure 15: Intersection sur l'union. [38]

A partir de YOLOv5, cette version est adaptée pour fonctionner avec des images thermiques et peuvent détecter des objets dans ces images. L'architecture YOLO n'est pas spécifiquement conçue pour un type d'image particulier, mais plutôt pour la détection d'objets en général.

YOLO, un algorithme de détection d'objets, a connu des évolutions majeures avec les versions populaires YOLOv5 et YOLOv8. YOLOv5 est recommandé pour une détection précise des petits objets et un déploiement sur des appareils sans GPU, tandis que YOLOv8 offre une vitesse optimale sur des dispositifs avec GPU.

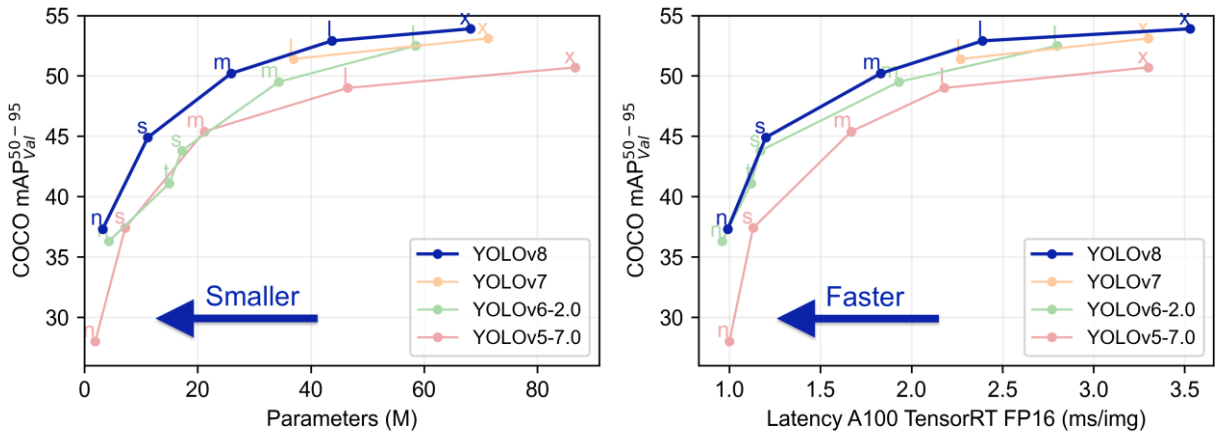

Figure 16: comparaison entre les versions de YOLO. [40]

Les modèles YOLOv5 sont principalement entraînés sur des images RVB. Pour détecter des objets dans des images thermiques, il faut reprendre l'entraînement avec des ensembles de données d'images thermiques annotées, en ajustant les paramètres en conséquence.

Pour réentraîner le modèle YOLO pour la détection d'objets dans des images thermiques, il faut collecter et annoter un ensemble de données d'images thermiques, puis utiliser la dataset FLIR pour entraîner le modèle à détecter spécifiquement les objets d'intérêt, tels que les véhicules, en utilisant des boîtes englobantes.

3.1.3 FLIR dataset

Le FLIR Thermal Starter Dataset est un ensemble de données qui fournit un ensemble d'images thermiques annotées et d'images RVB non annotées pour l'entraînement et la validation de réseaux neuronaux de détection d'objets. [20]

Cet ensemble de données a été acquis à l'aide d'une caméra RVB et thermique montée sur un véhicule. Il contient un total de 14 452 images thermiques annotées, dont 10 228 images extraites de courtes vidéos et 4 224 images provenant d'une vidéo continue de 144 secondes.

Toutes les vidéos ont été prises dans les rues et les autoroutes de Santa Barbara, en Californie, aux États-Unis, de novembre à mai. Les vidéos ont été enregistrées dans des conditions généralement dégagées, de jour comme de nuit. [20]

Des annotateurs humains ont réalisé l'annotation en plaçant des boîtes englobantes autour de trois catégories d'objets. La numérotation des classes a été effectuée en utilisant le vecteur d'étiquettes MSCOCO.

• Catégorie 1 : Personnes
• Catégorie 2 : Vélos
• Catégorie 3 : Voitures - véhicules personnels et certains petits véhicules commerciaux.

Cet ensemble de données fournit donc des images thermiques annotées avec des boîtes englobantes autour des personnes, des vélos et des voitures. Les images RVB correspondantes ne sont pas annotées.

3.1.4 Entrainement de yolov5 avec la dataset FLIR

A. F1 score

La courbe F1 représente les performances d'un modèle de classification multi-classes pour quatre classes différentes : voiture, vélo, personne et autres. La courbe représente la relation entre le score F1 et le seuil de confiance utilisé pour la classification. La formule pour le calcul du score F1 peut être exprimée comme suit : [41]

$$F1\ Score\ = \frac{2 * (Précision * Rappel)}{(Précision + Rappel)}$$

Le score F1 est une mesure couramment utilisée dans les tâches de classification qui combine précision et rappel. Il fournit une mesure équilibrée de la précision d'un modèle en tenant compte à la fois de la capacité d'identifier correctement les échantillons positifs (précision) et de la capacité de capturer tous les échantillons positifs (rappel). Le score F1 varie de 0 à 1, une valeur plus élevée indiquant une meilleure performance. Le seuil de confiance est une limite de décision qui détermine si une prédiction est considérée comme positive ou

négative. En faisant varier le seuil de confiance, vous pouvez ajuster le compromis entre précision et rappel. Un seuil de confiance plus bas conduit à des prédictions plus positives, augmentant potentiellement le rappel mais sacrifiant la précision. À l'inverse, un seuil de confiance plus élevé entraîne moins de prédictions positives, augmentant potentiellement la précision mais sacrifiant le rappel.

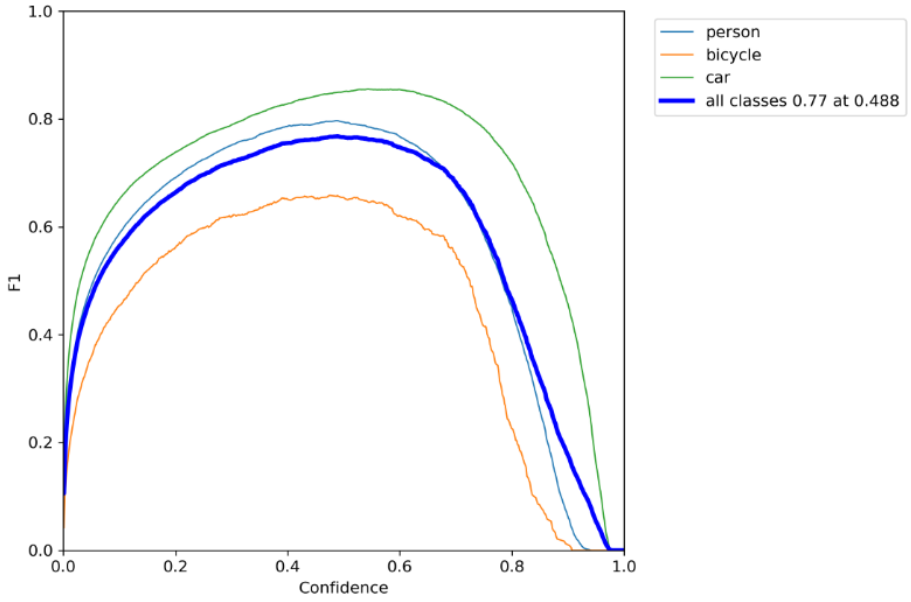

Figure 17: La courbe F1

B. Précision

La courbe de précision est utilisée pour évaluer la performance d'un modèle de classification multi-classes en fonction du seuil de confiance. Elle montre la relation entre la précision et le seuil de confiance, qui mesure l'exactitude du modèle en minimisant les erreurs de faux positifs. La formule pour la précision est la suivante : [41]

$$Précision = \frac{Vrai\ Positif}{(Vrai\ Positif + Faux\ Positif)}$$

La précision est une mesure de l'exactitude d'un modèle de classification. Elle représente le rapport entre le nombre de vrais positifs (samples correctement identifiés comme positifs) et le nombre total de prédictions positives (vrais positifs + faux positifs). La précision mesure la capacité du modèle à minimiser les erreurs de type "faux positifs", c'est-à-dire les cas où il prédit à tort une classe comme positive.

Le seuil de confiance est la limite qui détermine si une prédiction est considérée comme positive ou négative. En faisant varier ce seuil, on peut ajuster le compromis entre la précision et le rappel (recall) du modèle. Un seuil de confiance plus bas entraîne plus de prédictions positives, ce qui peut augmenter le rappel mais réduire la précision. À l'inverse, un seuil de confiance plus élevé entraîne moins de prédictions positives, ce qui peut augmenter la précision mais réduire le rappel.

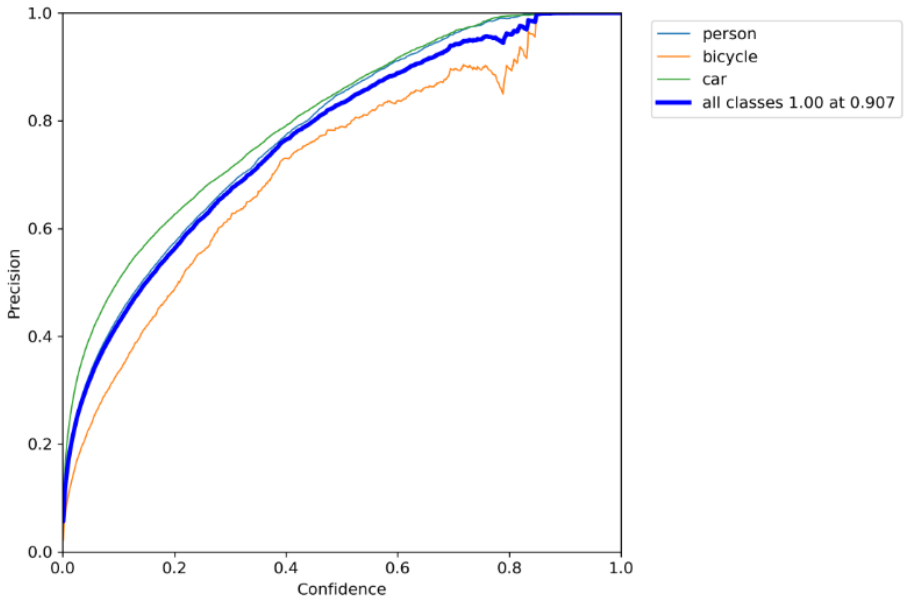

Figure 18: La courbe précision

C. Rappel

La courbe de rappel (recall curve) représente la performance d'un modèle de classification multi-classes en fonction du seuil de confiance utilisé pour la classification. Elle permet d'analyser la relation entre le rappel et le seuil de confiance. La formule du rappel est décrite comme suit :[41]

$$Rappel = \frac{Vrai\ Positif}{(Vrai\ Positif + Faux\ Négatif)}$$

Le rappel, également appelé sensibilité ou taux de vrais positifs, est une mesure de la capacité d'un modèle à identifier correctement les échantillons positifs parmi tous les échantillons réellement positifs. Il représente le rapport entre le nombre de vrais positifs et le nombre total de vrais positifs et de faux négatifs (vrais positifs + faux négatifs). Le rappel

mesure la capacité du modèle à minimiser les erreurs de type "faux négatifs", c'est-à-dire les cas où il prédit à tort une classe comme négative.

Le seuil de confiance est la limite qui détermine si une prédiction est considérée comme positive ou négative. En faisant varier ce seuil, on peut ajuster le compromis entre le rappel et la précision du modèle. Un seuil de confiance plus bas entraîne plus de prédictions positives, ce qui peut augmenter le rappel mais réduire la précision. À l'inverse, un seuil de confiance plus élevé entraîne moins de prédictions positives, ce qui peut augmenter la précision mais réduire le rappel.

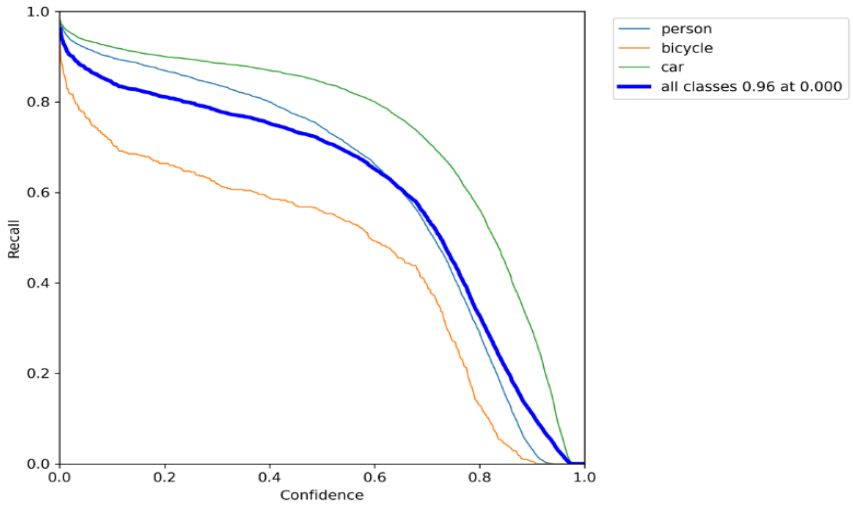

Figure 19: La courbe recall

D. Matrice de confusion

La matrice de confusion est une représentation tabulaire utilisée pour évaluer les performances d'un modèle de classification. Elle permet de visualiser le nombre de prédictions correctes et incorrectes effectuées par le modèle pour chaque classe.

La matrice de confusion est généralement présentée sous la forme d'un tableau carré où les lignes représentent les classes réelles et les colonnes représentent les classes prédites par le modèle. Chaque cellule de la matrice indique le nombre d'instances appartenant à une classe réelle donnée et prédites comme appartenant à une classe spécifique.

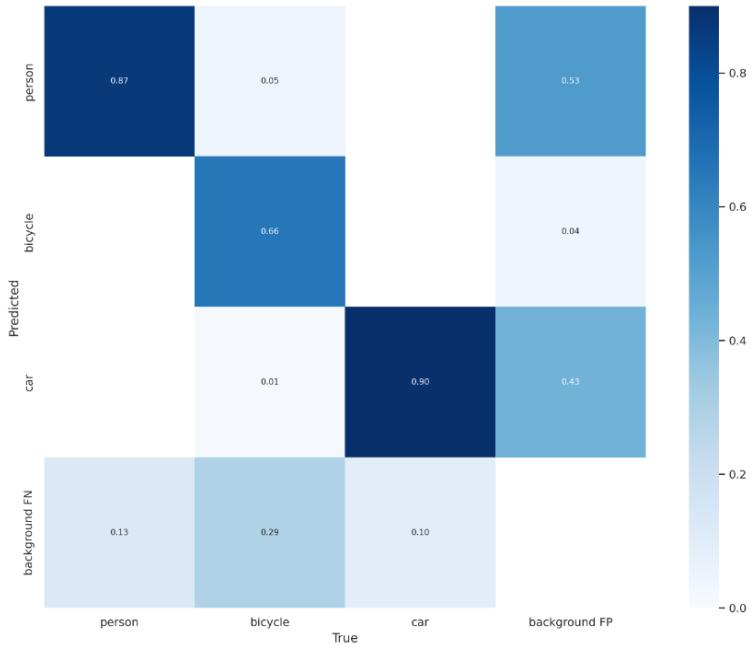

Figure 20: Matrice de confusion [43]

3.1.5 Résultat de détection d'objet

- **Dans une vision sombre**

Figure 21: Détection dans une vision sombre

- **Dans une vision floue**

Figure 22: Détection dans une vision floue

3.1.6 Les coordinations de détection

Tableau 4: Coordinations de détection

frame	xmin	ymin	xmax	ymax	scaled_xmin	scaled_ymin	scaled_xmax	scaled_ymax
1	656	234	862	330	636.525	121.875	836.4093750000001	171.875
2	627	234	870	332	608.3859375000001	121.875	844.171875	172.91666666666669
3	626	233	869	332	607.415625	121.35416666666667	843.2015625	172.91666666666669
4	627	232	869	330	608.3859375000001	120.83333333333334	843.2015625	171.875
5	623	230	867	329	604.5046874999999	119.79166666666666	841.2609375000001	171.35416666666666
6	624	229	868	327	605.475	119.27083333333333	842.2312499999999	170.3125
7	626	229	870	325	607.415625	119.27083333333333	844.171875	169.27083333333334
8	626	244	748	323	607.415625	127.08333333333334	725.7937499999999	168.22916666666669
8	628	227	872	323	609.3562499999999	118.22916666666666	846.1125000000001	168.22916666666669
9	645	226	872	321	625.8515625	117.70833333333333	846.1125000000001	167.1875

4. Estimation de distance

Pour l'estimation de la distance, nous avons entraîné un modèle d'apprentissage en profondeur qui prend les coordonnées de la boîte englobante de l'objet détecté et estime la distance à l'objet.

4.1.Dataset Kitti

Le dataset Kitti est l'un des ensembles de données les plus largement utilisés dans le domaine de la vision par ordinateur et de la perception des véhicules autonomes. Il est spécialement conçu pour faciliter la recherche et le développement de systèmes de conduite autonome en fournissant des données réelles et précises de scènes urbaines. [28]

Le dataset Kitti comprend une vaste collection de médias, comprenant des images, des vidéos et des données de lidar, tous capturés dans des environnements de conduite réels. Les images sont prises à partir de caméras embarquées sur un véhicule, tandis que les données lidar sont obtenues à l'aide d'un capteur lidar monté sur le toit du véhicule. Ces données sont synchronisées temporellement pour permettre une analyse précise des scènes et des mouvements des objets.

En ce qui concerne les chiffres, le dataset Kitti contient plus de 4000 séquences d'images de conduite, couvrant environ 40 kilomètres de trajets urbains. Ces séquences sont capturées dans différentes villes et présentent une variété de scénarios de conduite, y compris des rues étroites, des autoroutes, des intersections, des changements de voie, etc. Chaque séquence d'images est accompagnée de données de lidar correspondantes, permettant ainsi une compréhension tridimensionnelle complète de l'environnement. [28]

De plus, le dataset Kitti comprend également des annotations précises pour les objets présents dans les images, tels que les voitures, les piétons, les cyclistes, les panneaux de signalisation, etc. Ces annotations sont fournies pour une partie des séquences et sont utiles pour l'entraînement et l'évaluation de modèles de détection et de suivi d'objets.

En résumé, le dataset Kitti offre un ensemble de données complet et diversifié pour la recherche et le développement de systèmes de conduite autonome. Avec plus de 4000 séquences d'images et des données lidars correspondants, il constitue une ressource précieuse pour l'apprentissage automatique et la vision par ordinateur dans le contexte de la conduite autonome.

L'ensemble de données KITTI est un ensemble de données largement utilisé pour la recherche sur la conduite autonome et comprend diverses données de capteurs telles que des images, des nuages de points et des annotations. Les annotations fournies dans le jeu de données KITTI incluent non seulement les coordonnées de la boîte englobante (xmin, ymin, xmax, ymax) pour les objets, mais également des informations supplémentaires telles que

l'emplacement 3D (x, y, z) des objets dans la scène. La distance (z) dans le jeu de données KITTI fait référence à la profondeur ou à la distance entre la caméra et les objets de la scène. Ces informations de profondeur sont cruciales pour des tâches telles que la détection d'objets, le suivi et la compréhension de scènes dans les applications de conduite autonome.

4.2. Choix du model

```
WARNING:tensorflow:`epsilon` argument is deprecated and will be removed, use `min_delta` instead.
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_8 (Dense)             (None, 6)                 30

 dense_9 (Dense)             (None, 5)                 35

 dense_10 (Dense)            (None, 2)                 12

 dense_11 (Dense)            (None, 1)                 3

=================================================================
Total params: 80
Trainable params: 80
Non-trainable params: 0
_________________________________________________________________
```

Figure 23: Neural network model

4.3. Résultats d'estimation de distance

Tableau 5: Résultats d'estimation de distance

frame	xmin	ymin	xmax	ymax	scaled_xmin	scaled_ymin	scaled_xmax	scaled_ymax	distance
1	656	234	862	330	636.525	121.875	836.4093750000001	171.875	27.684844970703125
2	627	234	870	332	608.3859375000001	121.875	844.171875	172.91666666666669	27.111783981323242
3	626	233	869	332	607.415625	121.35416666666669	843.2015625	172.91666666666669	27.000965118408203
4	627	232	869	330	608.3859375000001	120.83333333333334	843.2015625	171.875	26.9412899017334
5	623	230	867	329	604.5046874999999	119.79166666666666	841.2609375000001	171.35416666666666	26.704425811767578
6	624	229	868	327	605.475	119.27083333333331	842.2312499999999	170.3125	26.633277893066406
7	626	229	870	325	607.415625	119.27083333333331	844.171875	169.27083333333334	26.67296028137207
8	626	244	748	323	607.415625	127.08333333333334	725.7937499999999	168.22916666666669	29.706480026245117
8	628	227	872	323	609.3562499999999	118.22916666666666	846.1125000000001	168.22916666666669	26.499046325683594
9	645	226	872	321	625.8515625	117.70833333333331	846.1125000000001	167.1875	26.68744468688965

- **Dans une vision sombre**

Figure 24: Estimation de distance dans une vision sombre

- **Dans une vision floue**

Figure 25: Estimation de distance dans une vision floue

Conclusion

En conclusion, ce chapitre a présenté en détail le système intelligent de détection d'objet et d'estimation de distance. Nous avons abordé les différentes étapes clés, de la collecte des données au modèle final, en passant par le prétraitement des données, l'entraînement et l'évaluation du modèle. Nous avons choisi l'architecture YOLO pour la détection d'objets et utilisé le dataset Kitti pour l'entraînement.

Les résultats obtenus ont démontré l'efficacité de notre approche, avec une détection précise des objets et une estimation fiable de la distance. Ces résultats ouvrent la voie à de nombreuses applications potentielles dans des domaines tels que la conduite autonome, la sécurité routière et la surveillance vidéo.

En somme, ce chapitre constitue une avancée significative dans le domaine de la vision par ordinateur, en combinant la détection d'objet et l'estimation de distance dans un système intelligent. Les résultats obtenus sont prometteurs et ouvrent la voie à de nouvelles possibilités pour des applications pratiques et innovantes.

CONCLUSION GÉNÉRALE

Conclusion générale et perspectives

Dans le cadre de ce projet de fin d'études, nous nous sommes intéressés à la détection d'objets et à l'estimation de distances dans le contexte de la conduite autonome. Nous avons proposé une architecture combinant YOLOv5 pour la détection d'objets dans des images floues et la vision thermique, ainsi qu'un réseau neuronal simple pour estimer la distance des objets détectés en utilisant la base de données KITTI.

Les résultats obtenus ont démontré l'efficacité de notre approche. En utilisant YOLOv5 pré-entrainé sur la base de données FLIR, nous avons pu détecter avec précision les objets dans des conditions de vision floue et thermique. Ensuite, en passant les résultats de détection à notre réseau neuronal dédié à l'estimation de distances, nous avons pu obtenir des estimations fiables et cohérentes.

Ces résultats montrent le potentiel de notre approche pour améliorer la perception des systèmes de conduite autonome dans des conditions difficiles, telles que la nuit, le brouillard ou les environnements à faible visibilité. En fournissant des informations précises sur les objets détectés et leur distance par rapport au véhicule autonome, notre architecture pourrait contribuer à renforcer la sécurité et la fiabilité des systèmes de conduite autonome.

Cependant, il convient de noter quelques limites et perspectives d'amélioration. Tout d'abord, bien que notre architecture ait donné des résultats prometteurs, il reste encore des opportunités d'amélioration en termes de précision et de vitesse de traitement. En explorant des techniques d'apprentissage profond plus avancées ou en affinant les hyperparamètres, il serait possible d'obtenir des performances encore meilleures.

De plus, notre étude s'est principalement concentrée sur la détection d'objets et l'estimation de distances, mais il existe d'autres aspects importants à prendre en compte dans la conduite autonome, tels que la prédiction de trajectoire, la planification des mouvements ou la prise de décision. En intégrant notre architecture dans un système global de conduite autonome et en le combinant avec d'autres modules, il serait possible d'obtenir une solution plus complète et performante.

En conclusion, notre projet a démontré que l'utilisation de YOLOv5 pour la détection d'objets et d'un réseau neuronal simple pour l'estimation de distances peut constituer une approche prometteuse dans le domaine de la conduite autonome. Tout en reconnaissant les limites de notre travail, nous pensons que ces résultats ouvrent la voie à de nouvelles recherches et à des améliorations futures pour rendre les systèmes de conduite autonome plus sûrs et plus efficaces.

Bibliographie

[1] Beye, P. D. (2021). Faisabilité technique des systèmes avancés d'aide à la conduite (ADAS) pour la sécurité routière (Thèse de doctorat). Université du Québec à Trois-Rivières.

[2] Noura, R. B. C., & Ben Bezziane, R. (2011). La recherche d'images par la sémantique (Mémoire de master). Université Kasdi Merbah Ouargla Algérie.

[3] https://www.exoco-lmd.com/traitement-dimages/generalites-sur-le-traitement-dimages/?action=dlattach;attach=13114

[4] http://staff.univ-batna2.dz/sites/default/files/behloul_ali/files/chap1fti.pdf

[5] Munguakonkwa Biringanine, J. (2008). La liaison automatique des plusieurs images perçues sur un scanner (Rapport de recherche). Institut Supérieur Pédagogique de Bukavu.

[6] http://dspace.univ-setif.dz:8888/jspui/bitstream/123456789/1806/1/TT%20m%C3%A9moire.pdf

[7] Crouspeyre, C. (2017). Comment les Réseaux de neurones à convolution fonctionnent [Blog post]. Récupéré de https://medium.com/@CharlesCrouspeyre/comment-les-réseaux-de-neurones-àconvolutionfonctionnent-b288519dbcf8

[8] https://azure.microsoft.com/fr-fr/resources/cloud-computing-dictionary/what-is-computer-vision

[10] https://stax.strath.ac.uk/downloads/db78tc523

[11] https://www2.deloitte.com/content/dam/Deloitte/se/Images/inline_images/articles/ai/3-artificial-intelligence-capabilities.png

[12] Mhammedi, A., Yakoub, I., & Ouahab, A. (2021). La détection de Covid-19 par l'apprentissage profonde (Deep Learning) (Thèse de doctorat). UNIVERSITE AHMED DRAIA-ADRAR.

[13] https://blog.isc2.org/.a/6a00e54f109b67883402af1c8f0725200d-pi

[14] https://www.researchgate.net/profile/Sanjay-Singh-22/publication/292722442/figure/fig7/AS:668646217093129@1536429089352/A-simplified-block-diagram-of-the-GoogLeNet-Architecture.png

[15] Lindgren Walter, et al. (2008). Requirements for the Design of Advanced Driver Assistance Systems - The Differences between Swedish and Chinese Drivers. International Journal of Design, 2, 41-54.

[16] https://www.ibm.com/fr-fr/topics/convolutional-neural-networks

[17] Lorrain, V. (Thèse de doctorat). Etude et conception de circuits innovants exploitant les caractéristiques des nouvelles technologies mémoires résistives, Université Paris-Saclay, Université Paris-Sud.

[18] Harsha, A. (2018, 8 juin). AI vs Machine Learning vs Deep Learning. Edureka. Récupéré de https://www.edureka.co/blog/ai-vs-machine-learning-vs-deep-learning/ (consulté le 11 mai 2023).

[19] Ismaili, Z. (2019, 28 janvier). Apprentissage Supervisé Vs. Non Supervisé. BrightCape. Récupéré de https://brightcape.co/apprentissage-supervise-vs-non-supervise/ (consulté le 10 mai 2023).

[20] https://www.flir.com/oem/adas/adas-dataset-form/

[21] https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/

[22] Kumar, B. V., Abirami, S., Lakshmi, R. B., Lohitha, R., & Udhaya, R. B. (2019, octobre). Detection and Content Retrieval of Object in an Image using YOLO. IOP Conference Series: Materials Science and Engineering, 590(1), 012062.

[23] Kang, S., Byun, H., & Lee, S. W. (2002). Real-Time Pedestrian Detection Using Support Vector Machines. Volume 2388.

[24] https://docs.ultralytics.com/datasets/detect/voc/

[25] https://viso.ai/computer-vision/coco-dataset/

[26] https://www.cityscapes-dataset.com

[27] https://paperswithcode.com/dataset/camvid

[28] https://www.cvlibs.net/datasets/kitti/

[29] Chen, X., et al. (2018). Deep Multi-modal Object Detection and Semantic Segmentation for Autonomous Driving Datasets, Methods, and Challenges.

[30] Bojarski, M., et al. (2015). FIGURE 3 Deep Driving: Convolutional Neural Networks for Autonomous Driving.

[31] Wang, Z., et al. (2017). Monocular Distance Estimation with Hierarchical Multi-Scale Deep Networks.

[32] https://biblio.univ-annaba.dz/ingeniorat/wp-content/uploads/2022/02/Feriel-Zehar.pdf

[33] Lali, A. J., Juton, V. N., & Rida. (Publication Year). Introduction à l'apprentissage par renforcement.

[34] https://rubikscode.net/wp-content/uploads/2021/04/yolo1.png

[35] http://bib.univ-oeb.dz:8080/jspui/bitstream/123456789/14297/1/M%C3%A9moir.pdf

[36] https://khaleejaffairs.com/wp-content/uploads/2022/08/grids.png

[37] https://media.licdn.com/dms/image/C4D12AQHFrN8S6Knrjg/article-inline_image-shrink_1000_1488/0/1651217876023?e=1692230400&v=beta&t=IDhtevH7F3Bd6YyFlP3bCtSWP3XginoyAFz-h7mxaZA

[38] https://miro.medium.com/v2/resize:fit:832/format:webp/1*4SYXQozCdfWcEXDFMIMovQ.jpeg

[39] https://miro.medium.com/v2/resize:fit:828/0*SXAG0c3y4LlKBH9R.png

[40] https://miro.medium.com/v2/resize:fit:828/0*SXAG0c3y4LlKBH9R.png

[41] https://inside-machinelearning.com/recall-precision-f1-score/

[42] https://www.jedha.co/formation-ia/matrice-confusion

[43] https://www.mdpi.com/jsan/jsan-11-00015/article_deploy/html/images/jsan-11-00015-g008-550.jpg