République Tunisienne
Ministère de l'Enseignement Supérieur et
de la Recherche Scientifique

Université de Gafsa
Institut Supérieur des Sciences
Appliquées et de la Technologie de Gafsa

Cycle de Formation en Mastère
Professionnelle dans la Discipline
sciences des données

Mémoire de MASTERE
Sciences des données

MEMOIRE

Présenté à

L'Institut Supérieur des Sciences Appliquées

et de Technologie de Gafsa

(*Département informatique et télécommunication*)

En vue de l'obtention diplôme en

MASTERE

Dans la discipline science des données

Par

Maher KHODMI

Nouvelle approche de segmentation sémantique basée sur le modèle U-net appliquée à la conduite autonome

Soutenu devant le jury composé de :

Dr. Moez MIRAOUI : *Président*

Dr. Rabaa IBRAHMI : *Rapporteur*

Dr. Haifa CHARFI : *Encadreur*

M. Ahmed KHLIFI : *Co-Encadreur*

ᔓ Année Universitaire : 2022-2023ᔕ

Dédicaces 

Au nom d'ALLAH,

Je dédie ce mémoire aux êtres qui me sont les plus chers au monde, mes parents, à qui je ne saurai exprimer ma profonde reconnaissance et gratitude… mais je sais bien que ma réussite est le plus beau et le plus cher cadeau à leurs offrir.

***

A vous ma mère, pour ton amour, ta patience et ta présence à mes côtés dans les moments les plus difficiles que j'ai vécus.

A vous mon père pour ton amour, ta patience et surtouts les sacrifices que tu as fait pour moi.

***

A vous mes chers frères pour vos amours et surtout vos soutiens illimités

***

Pour n'oublier personne, tous mes amis, qui de près ou de loin m'ont encouragé tout au long de mon cursus.

Maher KHODMI

Remerciements

Avant d'entamer ce rapport de fin d'études, nous tenons à exprimer notre profonde gratitude et nos vifs remerciements à tous nos enseignements pour leurs suivies tout le long de ces années d'étude à l'Institut supérieur des sciences appliquées et de la technologie de Gafsa et pour leur disponibilité et aide technique.

Nous tenons encore à exprimer nos remerciements à Dr. Haifa CHARFI *pour avoir bien voulu encadrer ce travail ainsi que pour sa riche contribution et ses précieux conseils, pour suivre mes travaux et pour son soutien, ses encouragements, sa disponibilité et sa patience.*

Un grand remerciement à mon Co 'encadreur M. Ahmed KHLIFI, *pour ses qualités scientifiques, sa rigueur et son encadrement régulier tout au long de ce travail.*

Maher KHODMI

Table des matières

Liste des figures

Liste des abréviations

CNN : Convolution Neural Network

FCN : les réseaux de neurones entièrement convolutifs

U-Net : réseau de neurones à convolution développé pour la segmentation d’images

R-CNN : Region-based Convolution Neural Networks

ReLU : Unités Rectifié linéaires

ROI : régions d'intérêt

SVM : Support Vector Machine

YOLO : You Only Look Once

Introduction générale

Nous vivons dans un monde numérique où les systèmes informatiques jouent un rôle crucial dans le stockage, le traitement, l'indexation et la recherche d'informations. Cette évolution a ouvert la voie à de nombreuses avancées dans le domaine de la vision par ordinateur, notamment dans la catégorisation et la division des images appliquées à la conduite autonome.

Ces dernières années, d'importants progrès ont été réalisés dans le domaine de la division des images en conduite autonome grâce à des recherches approfondies menées dans ce domaine et à la disponibilité de vastes bases de données internationales d'images. Ces bases de données ont permis aux chercheurs de présenter de manière crédible les performances de leurs approches en matière de division d'images et de les comparer à d'autres méthodes utilisant les mêmes ensembles de données.

Dans ce contexte, les réseaux neuronaux convolutifs ont joué un rôle essentiel. À la fin des années 80, Yann LeCun a développé ce type de réseau neuronal, s'inspirant de l'architecture des connexions du cortex visuel des mammifères. Bien que les réseaux convolutifs aient montré leur potentiel à cette époque, leur utilisation a été limitée par les capacités des ordinateurs de l'époque, et ils ont été relativement négligés par la communauté de recherche en vision par ordinateur jusqu'en 2012.

Cependant, trois événements majeurs ont changé radicalement la situation. Tout d'abord, l'accessibilité des unités de traitement graphique (GPU), capables d'effectuer des milliards d'opérations par seconde, a révolutionné les calculs des réseaux neuronaux, les rendant plus puissants et efficaces. Ensuite, des entreprises renommées telles que Microsoft, Google et IBM, en collaboration avec le laboratoire de Geoff Hinton, ont mené des expériences démontrant la capacité des réseaux profonds à réduire considérablement les taux d'erreurs dans les tâches de reconnaissance vocale. Enfin, les réseaux neuronaux convolutifs ont établi de nouveaux records en matière de reconnaissance d'images, notamment lors de la compétition "ImageNet" remportée brillamment par l'équipe de Toronto.

Ces avancées ont suscité un enthousiasme réel pour les réseaux neuronaux convolutifs et d'autres formes de réseaux neuronaux en vision par ordinateur. En conséquence, de nombreuses équipes de recherche et l'industrie de la conduite autonome ont rapidement adopté ces techniques pour la division en temps réel des images dans le contexte de la

conduite autonome. Cela a entraîné d'importants investissements dans la recherche et le développement du Deep Learning, en mettant l'accent notamment sur les applications de division d'images pour la compréhension de l'environnement routier et la prise de décisions intelligentes en temps réel.

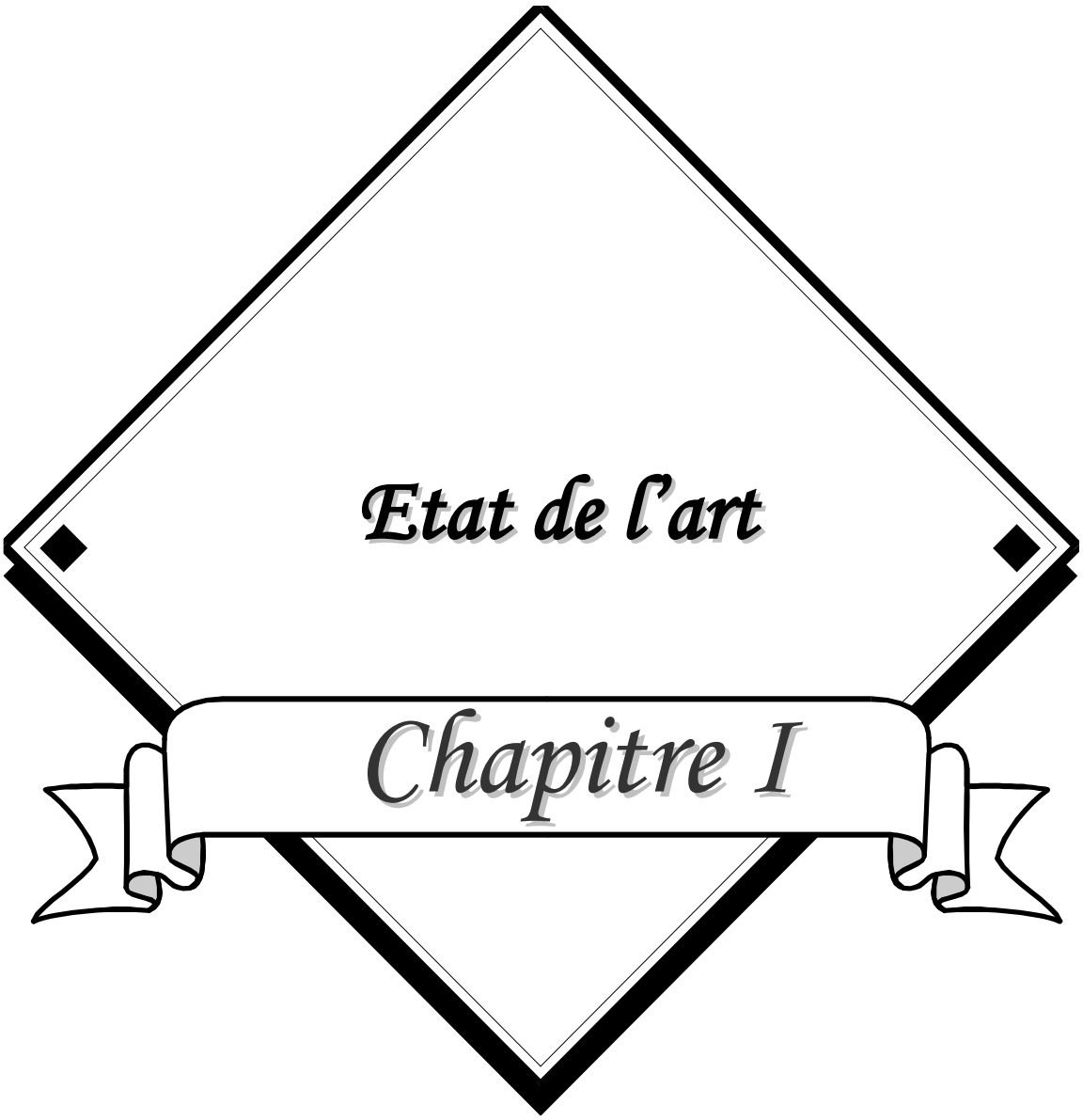
Dans notre projet on va utiliser le segmentation sémantique base sur le modèle UNET Les plus répandus pour faire une segmentation d'image entrainé avec la base d'images Cityscapes et par la suite on va tester le réseau formé avec d'autre images.

Pour ce faire, nous avons structuré notre mémoire en trois chapitres :

Dans le premier chapitre on va présenter les notions de base de la segmentation d'images et la détection d'objet.

Le deuxième chapitre est consacré à la nouvelle approche de segmentation sémantique basé sur l'UNET.

Dans le troisième chapitre, on va montrer la partie expérimentale de notre travail et on discute les résultats obtenus.


Etat de l'art
Chapitre I

Chapitre 1 : Etat de l’art

1. Introduction

Le champ du traitement d'images englobe les domaines de l'informatique et des mathématiques appliquées qui se focalisent sur l'étude des images numériques et de leurs modifications. L'objectif principal de cette discipline est d'améliorer la qualité des images et d'extraire des informations pertinentes à partir de celles-ci. Au sein de ce domaine, la segmentation d'images joue un rôle crucial et primordial. Elle représente une étape essentielle du processus de traitement et d'analyse d'images, visant à diviser une image en régions homogènes et à regrouper les pixels présentant des caractéristiques similaires selon des critères prédéfinis. Cette étape de segmentation permet de délimiter les différentes parties qui composent une image, facilitant ainsi leur compréhension et leur interprétation ultérieure.

2. Domaine de la vision par ordinateur

Le domaine de la vision par ordinateur est une branche de l'intelligence artificielle qui se concentre sur l'analyse et la compréhension des images et des vidéos par des ordinateurs. Cette discipline utilise des techniques de traitement d'images, de reconnaissance de formes, de traitement du signal et d'apprentissage automatique pour extraire des informations pertinentes à partir de ces médias visuels. Les applications de la vision par ordinateur sont diverses et nombreuses, allant de la reconnaissance faciale et vocale aux systèmes de surveillance et de sécurité, en passant par la conduite autonome, la réalité augmentée et la robotique. Ce domaine trouve également des applications dans des domaines scientifiques tels que la médecine, la biologie, la géologie et l'astronomie.

Les techniques de vision par ordinateur comprennent la segmentation d'image, la détection d'objets, la reconnaissance de formes, la reconnaissance de mouvements, la reconstruction 3D, la stéréovision, et bien d'autres. Grâce aux avancées récentes dans l'apprentissage profond et les réseaux de neurones convolutifs, de nombreuses améliorations significatives ont été réalisées dans différents domaines de la vision par ordinateur, en particulier dans la reconnaissance d'images et la classification d'objets.

2.1. Segmentation d'images

La segmentation d'image est une approche avancée pour détecter les objets, utilisant des masques pixel par pixel afin d'indiquer la présence de chaque objet dans une image. Cette méthode offre une précision plus fine que la simple délimitation par des boîtes englobantes, car elle permet de déterminer la forme exacte de chaque objet présent dans l'image. Cette précision détaillée est particulièrement utile dans des domaines tels que le traitement d'images médicales et l'imagerie de la conduite autonome, où des détails précis sont nécessaires pour une analyse approfondie.

Il existe principalement deux types de segmentation :

2.1.1. La segmentation sémantique

La segmentation sémantique est une technique de vision par ordinateur qui consiste à diviser une image en régions ou segments, et à attribuer à chaque segment une étiquette de classe correspondant à la signification sémantique de l'image. En d'autres termes, la segmentation sémantique associe chaque pixel d'une image à une classe d'objet spécifique, telle qu'un arbre, un bâtiment, une voiture, etc. La segmentation sémantique est utilisée dans de nombreux domaines de la vision par ordinateur, tels que la conduite autonome, la surveillance de la sécurité, la reconnaissance de gestes, la réalité augmentée, etc. Elle est souvent réalisée à l'aide de réseaux de neurones convolutifs (CNN) qui apprennent à identifier les motifs dans les images et à les associer à des étiquettes de classe.

La segmentation sémantique peut être réalisée de deux manières principales : la segmentation basée sur les régions et la segmentation basée sur les pixels. Dans la segmentation basée sur les régions, l'image est d'abord divisée en régions, puis chaque région est classée en fonction de ses caractéristiques visuelles. Dans la segmentation basée sur les pixels, chaque pixel est classé en fonction de ses caractéristiques visuelles et de sa position dans l'image.

La segmentation sémantique est une technique essentielle de vision par ordinateur qui permet d'identifier les objets et les régions d'intérêt dans une image en leur attribuant une étiquette de classe sémantique correspondante [1, 2].

2.1.2. La segmentation d'instance

La segmentation d'instance est une tâche de vision par ordinateur qui vise à identifier et isoler de manière individuelle chaque occurrence d'objet dans une image. Contrairement à la segmentation sémantique qui classe chaque pixel dans des catégories prédéfinies (comme voiture, chien, arbre, etc.), la segmentation d'instance permet de différencier les objets de la même classe en leur attribuant des étiquettes distinctes.

En d'autres termes, la segmentation d'instance divise une image en plusieurs masques, chacun représentant une occurrence spécifique de l'objet. Par exemple, si une image contient plusieurs voitures, la segmentation d'instance attribuera un masque différent à chaque voiture, permettant ainsi de les distinguer individuellement.

Cette technique est utilisée dans divers domaines tels que la détection d'objets, la robotique, la réalité augmentée, la conduite autonome, etc. Elle fournit des informations détaillées et précises sur les objets présents dans une scène, ce qui permet des analyses avancées et des applications plus sophistiquées [1,2].

2.2. Différentes approches de segmentation

En général, les méthodes de segmentation peuvent être classées en trois approches distinctes, chacune présentant ses propres avantages et domaines d'application spécifiques. Ces approches sont souvent complémentaires les unes aux autres.

Les trois approches de segmentation sont les suivantes :

- ✓ Segmentation basée sur les contours
- ✓ Segmentation basée sur les régions
- ✓ Segmentation basée sur l'apprentissage automatique

2.2.1. Segmentation basée sur les contours

La segmentation basée sur les contours est une approche de segmentation d'image qui se concentre sur la détection et la segmentation des contours des objets. Son objectif principal est d'identifier les discontinuités et les transitions entre les régions d'une image afin de délimiter les objets présents.

L'objectif de la segmentation basée sur les contours est de trouver les limites des objets en se basant sur les variations des niveaux de gris, des couleurs ou d'autres caractéristiques visuelles. Elle permet de créer une séparation claire entre les objets et leur environnement en détectant les zones où il y a des changements significatifs dans les valeurs des pixels.

Cette approche utilise généralement des techniques de traitement d'image telles que la détection de contours par gradient, où les gradients des niveaux de gris sont calculés pour identifier les transitions abruptes. Des algorithmes tels que le filtre de Canny, le filtre de Sobel ou le filtre de Laplace sont souvent utilisés pour détecter les contours.

La segmentation basée sur les contours est largement utilisée dans des applications telles que la détection d'objets, la reconnaissance de formes, la surveillance vidéo, la vision par ordinateur et l'analyse d'images médicales. Elle peut être utilisée comme étape préliminaire pour d'autres tâches de traitement d'image telles que la reconnaissance d'objets ou la mesure de formes [3]

2.2.2. Segmentation basée sur les régions

La segmentation basée sur les régions est une approche de segmentation d'image qui vise à diviser une image en régions homogènes en termes de couleur, de texture, de luminosité ou d'autres caractéristiques visuelles. Son objectif principal est de regrouper les pixels ou les superpixels similaires pour former des régions cohérentes représentant des objets ou des parties de l'image.

L'objectif de la segmentation basée sur les régions est de créer des groupes de pixels ayant des caractéristiques similaires afin de former des régions distinctes dans l'image. Cela permet de regrouper les pixels appartenant à un même objet et de séparer les objets les uns des autres. Les régions ainsi formées sont généralement cohérentes en termes de couleur, de texture ou d'autres attributs visuels, ce qui facilite l'identification et la caractérisation des objets présents dans l'image.

Cette approche de segmentation utilise différentes techniques telles que la croissance de région, où des régions sont développées à partir de graines initiales en ajoutant progressivement des pixels similaires. La segmentation par seuillage est également couramment utilisée, où les pixels sont assignés à différentes régions en fonction de leur similitude avec un seuil prédéfini.

La segmentation basée sur les régions est utilisée dans de nombreux domaines tels que la reconnaissance d'objets, la segmentation d'images médicales, l'analyse de scènes, la cartographie, etc. Elle permet de séparer les objets d'intérêt du reste de l'image et peut être utilisée pour des tâches ultérieures telles que la classification, la détection ou la mesure de formes [3].

2.2.3. Segmentation basée sur l'apprentissage automatique

La segmentation basée sur l'apprentissage automatique est une approche de segmentation d'image qui utilise des algorithmes d'apprentissage automatique pour apprendre à segmenter les objets dans une image. Au lieu de se fier à des règles ou à des critères prédéfinis, cette approche permet au modèle d'apprendre automatiquement à partir de données annotées pour identifier et segmenter les objets.

L'objectif de la segmentation basée sur l'apprentissage automatique est d'utiliser des techniques d'apprentissage automatique, telles que les réseaux de neurones convolutifs (CNN) ou les méthodes de classification, pour apprendre à reconnaître les objets et à les segmenter avec précision. Le modèle est entraîné sur un ensemble de données d'entraînement contenant des images annotées où les objets sont étiquetés avec leurs contours ou leurs masques de segmentation.

Le modèle d'apprentissage automatique apprend à extraire des caractéristiques discriminantes des images pour distinguer les différents objets présents. Il utilise ces caractéristiques pour prédire la présence et la localisation des objets dans une image, ainsi que pour générer des masques de segmentation précis qui délimitent les objets de manière fine.

L'avantage de la segmentation basée sur l'apprentissage automatique est sa capacité à apprendre des modèles complexes et à généraliser à de nouvelles images. Une fois entraîné, le modèle peut être utilisé pour segmenter des objets dans des images non vues précédemment. Cela permet d'automatiser le processus de segmentation et d'obtenir des résultats précis et cohérents.

Des architectures spécifiques ont été développées pour la segmentation basée sur l'apprentissage automatique, telles que U-Net, Mask R-CNN et les réseaux de neurones entièrement convolutifs (FCN). Ces modèles ont montré de bonnes performances dans des tâches de segmentation d'objets, notamment dans des domaines tels que la vision par ordinateur, la médecine, la reconnaissance d'images et la robotique [3,4].

2.3. La détection des objets

La détection d'objets est une tâche cruciale en vision par ordinateur qui consiste à repérer et à identifier des objets dans une image ou une vidéo. Pour accomplir cette tâche, les réseaux de neurones convolutifs (CNN) sont souvent utilisés. Ces réseaux sont entraînés à détecter des objets spécifiques en analysant une vaste quantité d'images annotées.

La détection d'objets s'effectue généralement en deux étapes distinctes : la localisation des régions d'intérêt (ROI) et la classification de ces régions. La première étape utilise des

algorithmes de détection de ROI, tels que R-CNN, Fast R-CNN, Faster R-CNN, YOLO (You Only Look Once) ou SSD (Single Shot Detector), pour repérer les régions de l'image susceptibles de contenir des objets. La deuxième étape consiste à classifier ces régions en utilisant des algorithmes de classification tels que le SVM (Support Vector Machine), le CNN, etc.

Les applications de la détection d'objets sont vastes et diversifiées. Elle est utilisée dans des domaines tels que la surveillance de la sécurité, la conduite autonome, la reconnaissance de gestes, la réalité augmentée, la reconnaissance de plaques d'immatriculation, la reconnaissance faciale, la médecine, la biologie, et bien d'autres encore [5,6].

2.4. Domaine d'application

La détection d'objets joue un rôle essentiel en vision par ordinateur et trouve une grande variété d'applications dans divers domaines. Voici quelques exemples d'applications de la détection d'objets :

- ✓ Surveillance de sécurité : La détection d'objets est utilisée pour repérer les mouvements suspects ou les intrusions dans les zones sous surveillance, comme les bâtiments, les aéroports, les gares, etc.
- ✓ Véhicules autonomes : La détection d'objets est utilisée pour détecter les piétons, les autres véhicules, les feux de signalisation, les panneaux de signalisation, les marquages au sol, etc., afin d'assurer une conduite sûre et une navigation précise.
- ✓ Reconnaissance faciale : La détection d'objets est utilisée pour détecter les visages et les caractéristiques faciales, telles que les yeux, le nez, la bouche, etc., afin de faciliter la reconnaissance faciale et d'améliorer la sécurité.
- ✓ Analyse médicale : La détection d'objets est utilisée pour détecter les anomalies ou les structures d'intérêt dans les images médicales, comme les tumeurs, les nodules, les lésions, etc., pour faciliter le diagnostic et le traitement des maladies.
- ✓ Surveillance de la faune : La détection d'objets est utilisée pour repérer les animaux dans leur environnement naturel, tels que les oiseaux, les animaux sauvages, les poissons, etc., afin de surveiller leur comportement, leur migration ou leur présence pour des études écologiques.
- ✓ Contrôle qualité dans l'industrie : La détection d'objets est utilisée pour identifier les défauts ou les erreurs dans les produits manufacturés, tels que les produits électroniques, les pièces automobiles, les aliments, etc., afin d'assurer des normes de qualité élevées et d'éviter les produits défectueux.

- ✓ Détection de la fraude : La détection d'objets est utilisée pour repérer les activités frauduleuses, telles que la falsification de documents, le blanchiment d'argent, etc., afin de prévenir les fraudes et de maintenir l'intégrité des systèmes.

La détection d'objets est largement utilisée dans de nombreux domaines, tels que la sécurité, la reconnaissance faciale, les véhicules autonomes, l'analyse médicale, la surveillance de la faune, le contrôle qualité dans l'industrie et la détection de la fraude.

2.5. Les défis de la détection d'objets

La détection d'objets est une tâche complexe qui fait face à plusieurs défis nécessitant des améliorations constantes pour augmenter les performances des systèmes de détection. Voici quelques-uns des défis actuels de la détection d'objets :

- ✓ Précision : La précision est essentielle en détection d'objets, car toute erreur peut avoir des conséquences graves, notamment en matière de sécurité routière. Les systèmes de détection doivent être capables de détecter les objets avec une grande précision tout en évitant les fausses détections.
- ✓ Gestion de la variation : Les objets peuvent présenter des variations significatives en termes de taille, d'orientation, de position, de forme, de texture, d'éclairage, de fond, etc. Les systèmes de détection doivent être capables de gérer cette variation afin de détecter avec précision les objets dans toutes les conditions.
- ✓ Vitesse de traitement : Les systèmes de détection doivent être capables de traiter rapidement les images et les vidéos en temps réel. Cela est particulièrement important dans des applications telles que les voitures autonomes et la surveillance de la sécurité.
- ✓ Détection d'objets multiples : Les systèmes de détection doivent être capables de détecter plusieurs objets simultanément dans une image ou une vidéo. Cela peut être difficile lorsque les objets se chevauchent ou sont proches les uns des autres.
- ✓ Gestion des données volumineuses : Les systèmes de détection doivent être capables de gérer de grandes quantités de données, car les images et les vidéos peuvent être très volumineuses.
- ✓ Adaptabilité : Les systèmes de détection doivent être capables de s'adapter à de nouveaux types d'objets et à de nouvelles conditions environnementales. Cela peut être difficile, car de nouveaux types d'objets peuvent présenter des caractéristiques différentes de celles apprises par le modèle d'apprentissage automatique.

La détection d'objets est une tâche complexe qui rencontre plusieurs défis, tels que la précision, la gestion de la variation, la vitesse de traitement, la détection d'objets multiples, la gestion des données volumineuses et l'adaptabilité [6].

3. Étude de quelques approches

Dans le domaine de la détection d'objets, il est possible d'identifier plusieurs boîtes englobantes qui représentent différents objets d'intérêt dans une image. Le nombre de ces boîtes n'est pas prédéterminé à l'avance. Cependant, il est difficile de résoudre ce problème en utilisant simplement un réseau convolutif suivi d'une couche entièrement connectée, car la longueur de la couche de sortie varie en fonction du nombre d'occurrences des objets d'intérêt.

Une approche pour résoudre ce problème consiste à sélectionner différentes régions d'intérêt dans l'image et à utiliser un réseau neuronal convolutif (CNN) pour déterminer la présence de l'objet dans chaque région. Cependant, cette approche présente des défis, car les objets d'intérêt peuvent avoir des positions spatiales et des proportions différentes dans l'image. Par conséquent, il faudrait sélectionner un grand nombre de régions, ce qui pourrait entraîner une charge de calcul excessive.

Pour résoudre ce problème de manière plus efficace, des algorithmes tels que R-CNN (Region-based Convolutional Neural Networks), YOLO (You Only Look Once), etc. ont été développés. Ces algorithmes permettent de détecter rapidement les occurrences des objets en utilisant des techniques telles que la génération de propositions de régions d'intérêt, la prédiction de la présence d'objets et la localisation précise de ces objets.

3.1. RCNN

R-CNN (Réseau de Neurones Convolutifs basé sur les régions) est un algorithme largement utilisé dans le domaine de la vision par ordinateur pour détecter les objets dans les images. Son objectif principal est de localiser et classifier les objets d'intérêt présents dans une image.

L'approche R-CNN repose sur la sélection de différentes régions d'intérêt (boîtes englobantes) à l'intérieur de l'image, qui sont ensuite traitées individuellement à l'aide d'un réseau neuronal convolutif (CNN). Le CNN extrait des caractéristiques pertinentes de chaque région, qui sont ensuite utilisées par un classifieur pour déterminer la présence d'un objet ainsi que sa classe.

L'un des principaux avantages du R-CNN réside dans sa capacité à gérer des objets présentant différentes positions spatiales et rapports d'aspect au sein de l'image. En traitant

chaque région indépendamment, le R-CNN parvient à capturer efficacement la diversité des objets présents.

Cependant, le R-CNN présente également quelques limitations. Tout d'abord, il nécessite la sélection d'un grand nombre de régions d'intérêt, ce qui peut être coûteux en termes de puissance de calcul. De plus, l'algorithme original du R-CNN était relativement lent en raison du traitement individuel de chaque région.

Pour surmonter ces problèmes, des versions ultérieures du R-CNN ont été développées, telles que le Fast R-CNN et le Faster R-CNN. Ces itérations ont introduit des améliorations en termes de vitesse et de précision en intégrant des réseaux de proposition de régions (RPN) pour générer plus efficacement les régions d'intérêt.

Dans l'ensemble, le R-CNN et ses variantes ont considérablement contribué aux progrès de la détection d'objets en fournissant des solutions efficaces pour gérer les sorties de longueurs variables et détecter avec précision les objets dans des images complexes. Toutefois, le R-CNN présente certains problèmes et limitations, notamment sa complexité computationnelle, sa dépendance aux régions de proposition, son inefficacité d'entraînement et le manque de partage de fonctionnalités entre les régions d'intérêt. C'est pourquoi des versions améliorées ont été développées pour répondre à ces défis et améliorer les performances globales de la détection d'objets [7].

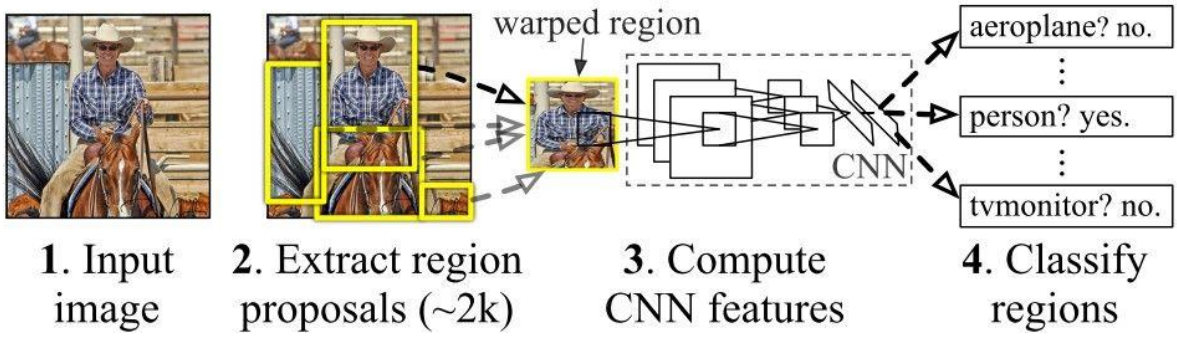


Figure 1: architecture RCNN

3.2. Fast R-CNN

Le Fast R-CNN est une amélioration du R-CNN (Réseau de Neurones Convolutifs basé sur les régions) qui vise à accélérer le processus de détection d'objets et à résoudre certaines limitations du R-CNN original.

Dans le Fast R-CNN, l'approche diffère de celle du R-CNN en utilisant l'image complète pour extraire les caractéristiques à l'aide d'un réseau neuronal convolutif (CNN). Au lieu de traiter chaque région d'intérêt individuellement, les caractéristiques extraites sont utilisées pour générer une carte de caractéristiques (feature map) qui capture les informations pertinentes de l'image.

Pour classifier les régions d'intérêt, le Fast R-CNN utilise une couche de pooling des régions d'intérêt (region of interest pooling layer) qui aligne spatialement les caractéristiques extraites avec la région d'intérêt correspondante. Cela permet de réduire la redondance de calcul et d'obtenir des représentations de taille fixe pour chaque région.

Enfin, les caractéristiques alignées sont fournies à un réseau de classification qui prédit la classe de l'objet et estime les coordonnées de la boîte englobante associée.

Le Fast R-CNN présente plusieurs avantages par rapport au R-CNN original. En éliminant le besoin de traiter chaque région d'intérêt séparément, il améliore considérablement l'efficacité et la vitesse de détection. De plus, en utilisant l'image complète pour extraire les caractéristiques, il facilite le partage des calculs, réduisant ainsi la redondance et améliorant l'efficacité globale du processus de détection d'objets [7].

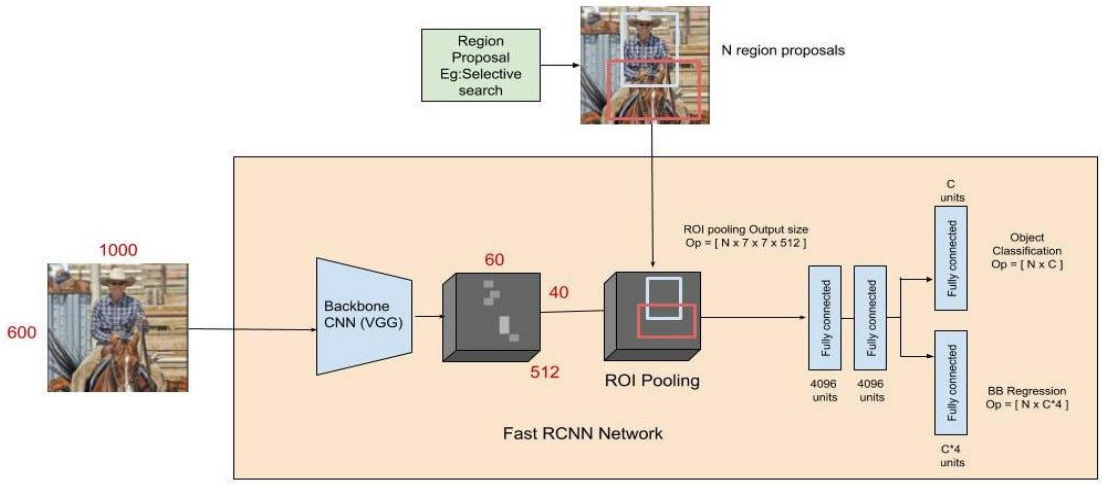


Figure 2: architecture Fast RCNN

3.3. Faster R-CNN

Le Faster R-CNN est une évolution du R-CNN (Réseau de Neurones Convolutifs basé sur les régions) et du Fast R-CNN. Il s'agit d'un algorithme de détection d'objets en vision par ordinateur qui vise à améliorer à la fois la précision et la vitesse du processus de détection.

Le rôle principal du Faster R-CNN est de localiser et classifier les objets d'intérêt dans une image de manière plus rapide et efficace. Pour cela, il introduit un module appelé "Region Proposal Network" (RPN) qui génère automatiquement les régions d'intérêt (bounding boxes) pour la détection, résolvant ainsi certaines limitations des approches précédentes.

Le fonctionnement du Faster R-CNN se décompose en plusieurs étapes. Tout d'abord, l'algorithme utilise un réseau neuronal convolutif (CNN) pour extraire les caractéristiques de l'image complète. Ensuite, le RPN est utilisé pour générer des régions d'intérêt potentielles en analysant la carte de caractéristiques. Ces régions d'intérêt proposées sont ensuite raffinées à l'aide d'une couche appelée "ROI Pooling", qui les aligne spatialement avec les caractéristiques extraites. Enfin, les caractéristiques alignées sont utilisées par des branches distinctes du réseau pour la classification des objets et l'estimation précise des coordonnées des bounding boxes.

Une des principales améliorations apportées par le Faster R-CNN est l'utilisation du RPN pour générer automatiquement les régions d'intérêt, éliminant ainsi le besoin d'une étape externe de génération de propositions de régions. Cela permet une approche end-to-end plus efficace et améliore considérablement la vitesse de détection.

Grâce à cette architecture, le Faster R-CNN parvient à combiner précision et rapidité, ce qui en fait l'un des algorithmes les plus performants pour la détection d'objets dans les images. Il est largement utilisé dans des domaines tels que la reconnaissance d'objets, la sécurité, la surveillance et l'analyse vidéo [7].

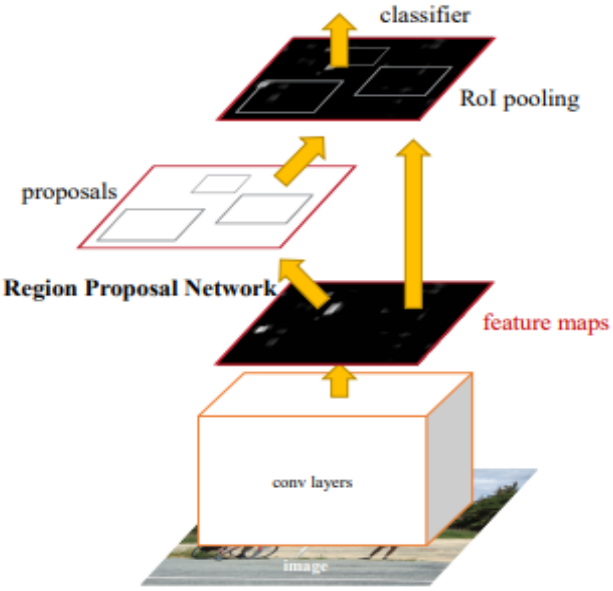


Figure 3: architecture Faster RCNN

3.4. YOLO

YOLO (You Only Look Once) est une méthode de détection d'objets en vision par ordinateur qui se distingue par sa rapidité et son efficacité. Contrairement à d'autres méthodes qui utilisent des régions de proposition, YOLO effectue la détection d'objets en une seule passe sur l'image complète.

Le principal objectif de YOLO est de localiser et classifier les objets d'intérêt dans une image en temps réel. Pour cela, YOLO divise l'image en une grille régulière et prédit les boîtes englobantes (bounding boxes) et les classes des objets pour chaque cellule de la grille. Ainsi, des prédictions simultanées sont obtenues pour tous les objets présents dans l'image.

Le fonctionnement de YOLO est le suivant : l'image est divisée en une grille de cellules régulières. Chaque cellule est responsable de la prédiction d'un ensemble de boîtes englobantes et de leurs classes correspondantes. Chaque boîte englobante contient des informations sur les coordonnées (position, taille) de l'objet détecté. YOLO calcule également un score de confiance pour chaque prédiction de boîte englobante, indiquant ainsi la précision et la fiabilité de la prédiction. Pour éliminer les boîtes englobantes redondantes, YOLO utilise une technique appelée "non-maximum suppression", qui conserve uniquement les prédictions les plus pertinentes et précises.

YOLO se distingue par sa rapidité, car il effectue la détection d'objets en une seule passe sur l'image complète. Cette approche permet de traiter les images en temps réel avec une latence minimale, ce qui est essentiel pour des applications telles que la surveillance vidéo et la conduite autonome. YOLO est également capable de détecter plusieurs objets simultanément et de traiter des scènes complexes avec une précision raisonnable. Cependant,

par rapport à certaines méthodes plus lentes et intensives en calcul, YOLO peut parfois sacrifier une partie de la précision au profit de la vitesse [7].

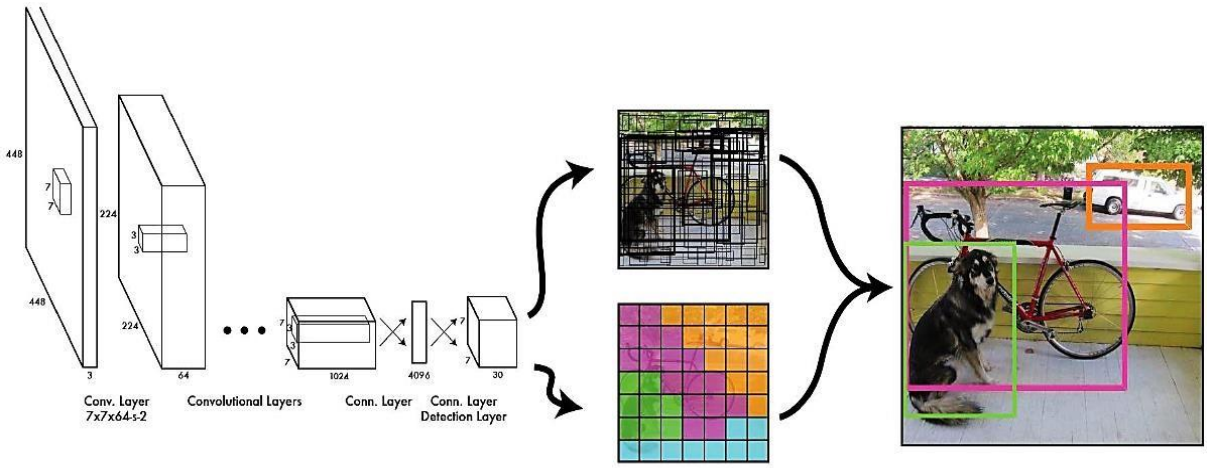


Figure 4: architecture YOLO

4. Discussion de l'état de l'art

L'état de l'art en matière de segmentation et de détection d'objets est dominé par plusieurs approches populaires, notamment U-Net, RCNN, Fast R-CNN et YOLO. U-Net est largement utilisé pour la segmentation sémantique et se distingue par son architecture en forme de U, qui combine des opérations de contraction et d'expansion pour capturer les caractéristiques à différentes échelles et les détails contextuels.

D'un autre côté, RCNN (Region Convolutional Neural Network) a été l'une des premières approches à introduire la détection d'objets basée sur des propositions de régions. Elle utilise des techniques de proposition de régions pour générer des régions d'intérêt potentielles, puis extrait les caractéristiques spécifiques de ces régions à l'aide d'un réseau de neurones convolutifs. RCNN a été amélioré par Fast R-CNN, qui combine les étapes de proposition de région et d'extraction de caractéristiques en une seule étape.

Enfin, YOLO (You Only Look Once) est une approche populaire qui se distingue par sa vitesse de traitement en temps réel. Contrairement à RCNN et Fast R-CNN, YOLO traite l'image complète en une seule passe et prédit directement les boîtes englobantes et les classes des objets à partir de la grille de sortie du réseau de neurones. Cela permet d'obtenir des performances rapides tout en maintenant une précision raisonnable.

Chacune de ces approches présente des avantages et des inconvénients. U-Net excelle dans la segmentation sémantique, RCNN et Fast R-CNN sont efficaces pour la détection précise des objets mais nécessitent des étapes de traitement plus complexes, tandis que YOLO se distingue par sa rapidité et sa simplicité, mais peut rencontrer des difficultés à détecter de

petits objets. Le choix entre ces approches dépendra des exigences spécifiques de la tâche et des contraintes de performance en termes de précision et de vitesse.

5. Conclusion

Ce chapitre a abordé la définition de la segmentation d'images en explorant différentes approches. Dans le prochain chapitre, nous allons examiner les nouvelles approches de segmentation sémantique basées sur U-Net qui seront utilisées pour réaliser la segmentation d'images dans le contexte de la conduite autonome.


Nouvelle approche
de segmentation sémantique
basé sur l'U-net
Chapitre II

Chapitre 2 : Nouvelle approche de segmentation sémantique base sur l'UNET

1. Introduction

Dans ce deuxième chapitre, nous nous focalisons sur une approche récente de segmentation sémantique basée sur U-Net, une tâche cruciale dans le domaine de la vision par ordinateur. La segmentation sémantique consiste à attribuer des étiquettes sémantiques à chaque pixel d'une image, permettant ainsi de différencier et d'identifier les différentes classes d'objets présentes.

Nous examinons spécifiquement l'utilisation du modèle U-Net, une architecture profonde extrêmement populaire et performante pour la segmentation sémantique. U-Net est largement reconnu pour sa capacité à produire des résultats précis et détaillés tout en traitant efficacement des images de différentes tailles.

2. Choix du modèle

Le modèle U-Net se démarque par ses performances exceptionnelles en matière de segmentation sémantique. Alors que des modèles tels que YOLO et Faster R-CNN sont plus adaptés à la détection d'objets, U-Net excelle dans la segmentation pixel par pixel, offrant une précision et une qualité de segmentation élevées.

2.1. U-Net

U-Net est une architecture de réseau de neurones convolutifs (CNN) largement utilisée pour la segmentation d'images, en particulier dans des domaines tels que la médecine et la conduite autonome. Cette architecture a été développée en 2015 par Olaf Ronneberger, Philipp Fischer et Thomas Brox. Le nom U-Net fait référence à la forme de son architecture, qui ressemble à la lettre "U".

L'architecture se compose d'un encodeur et d'un décodeur symétriques, reliés par un chemin de liaison. Ce chemin de liaison est un pont qui permet au décodeur d'accéder directement aux caractéristiques extraites par l'encodeur, facilitant ainsi l'alignement des caractéristiques à différents niveaux d'abstraction.

L'encodeur dans U-Net est similaire à celui des autres architectures CNN, avec des couches de convolution et de sous-échantillonnage qui extraient les caractéristiques de l'image

d'entrée. Le décodeur utilise des couches de déconvolution et d'upsampling pour produire une carte de segmentation de la même taille que l'image d'entrée.

La caractéristique la plus distinctive de l'architecture U-Net réside dans l'utilisation de connexions de saut (skip connections), permettant au décodeur d'accéder aux caractéristiques de l'encodeur à différents niveaux d'abstraction. Ces connexions de saut aident à aligner les caractéristiques à différents niveaux d'abstraction, améliorant ainsi la précision de la segmentation. Elles compensent également la perte d'informations causée par les couches de sous-échantillonnage de l'encodeur [11]

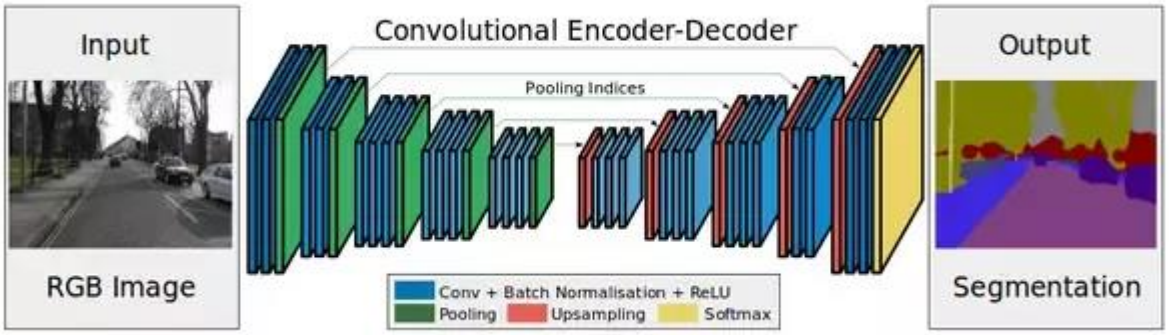


Figure 5: architecture générale d'U-net

2.2. Modèle basé sur U-Net

La segmentation sémantique revêt une importance cruciale dans le domaine de la conduite autonome, car elle permet de comprendre l'environnement de conduite en identifiant différentes classes d'objets tels que les voitures, les piétons, les panneaux de signalisation et les routes.

Le modèle U-Net est largement utilisé dans le contexte de la conduite autonome. Bien qu'il ait été initialement développé pour la segmentation d'images biomédicales, il a été adapté avec succès à la segmentation sémantique dans d'autres domaines.

Le modèle U-Net repose sur une architecture encodeur-décodeur, où l'encodeur extrait des caractéristiques à différentes échelles de l'image d'entrée, tandis que le décodeur restaure la résolution spatiale des cartes de caractéristiques et génère les prédictions de segmentation. Des connexions résiduelles sont également utilisées pour améliorer le flux d'informations et la stabilité de l'entraînement.

Voici un exemple de modèle de segmentation sémantique basé sur U-Net pour la conduite autonome :

- ✓ Entrée : Image provenant de la caméra embarquée.
- ✓ Couche de prétraitement : Normalisation des pixels de l'image
- ✓ Encodeur : Utilisation d'une série de couches de convolution pour extraire les caractéristiques de l'image. Chaque couche réduit la résolution spatiale de l'image d'entrée et augmente le nombre de canaux de caractéristiques.
- ✓ Connexions résiduelles : Ajout de connexions directes entre l'encodeur et le décodeur pour améliorer la stabilité de l'entraînement.
- ✓ Décodeur : Utilisation d'une série de couches de déconvolution pour restaurer la résolution spatiale de la carte de caractéristiques et générer les prédictions de segmentation. Chaque couche de déconvolution augmente la résolution spatiale de la carte de caractéristiques et réduit le nombre de canaux de caractéristiques.
- ✓ Sortie : Carte de segmentation identifiant les différentes classes d'objets dans l'image.

Pour entraîner ce modèle, il est nécessaire d'utiliser un ensemble de données d'apprentissage comprenant des images annotées avec des masques de segmentation pour chaque classe d'objet.

Ensuite, un algorithme d'optimisation tel que la descente de gradient stochastique peut être utilisé pour ajuster les poids du modèle afin de minimiser l'erreur de prédiction entre la carte de segmentation prédite et le masque de segmentation réel.

Une fois entraîné, ce modèle peut être utilisé pour la détection et la segmentation en temps réel des objets dans le contexte de la conduite autonome.

3. Architecture U-Net

L'architecture du modèle U-Net se compose de deux parties principales : l'encodeur (contraction) et le décodeur (expansion). Ces deux parties travaillent de concert pour réaliser une segmentation sémantique précise des images.

L'encodeur et le décodeur sont les éléments clés de l'architecture U-Net. L'encodeur extrait les caractéristiques de l'image, tandis que le décodeur reconstruit les détails des contours des objets. Cette combinaison permet au modèle U-Net d'effectuer une segmentation sémantique précise en capturant à la fois les informations contextuelles et les détails locaux.

3.1. Encodeur

L'encodeur est une composante essentielle des architectures de réseaux neuronaux profonds tels que les réseaux neuronaux convolutifs (CNN) et les réseaux neuronaux récurrents (RNN). L'encodeur prend une entrée et la transforme en une représentation latente, c'est-à-dire un vecteur de caractéristiques contenant des informations sur les principales caractéristiques de l'entrée.

Dans le contexte des CNN, l'encodeur est généralement la première partie du réseau, composée d'une série de couches de convolution et de sous-échantillonnage (également appelé "pooling"). Les couches de convolution extraient les caractéristiques à différents niveaux d'abstraction en appliquant des filtres de convolution à l'image d'entrée. Les couches de sous-échantillonnage réduisent la taille de l'image en agrégeant les valeurs voisines tout en préservant les caractéristiques les plus importantes.

Dans les architectures de segmentation sémantique, l'encodeur est souvent combiné avec un décodeur pour produire une segmentation précise de l'image d'entrée. L'encodeur extrait les caractéristiques de l'image, qui sont ensuite transmises au décodeur pour générer la carte de segmentation.

Dans le contexte des RNN, l'encodeur est utilisé pour transformer une séquence d'entrée en une représentation latente. Cette représentation peut ensuite être utilisée pour la classification, la génération de séquences ou toute autre tâche nécessitant une compréhension globale de la séquence d'entrée [13,14].

3.2. Décodeur

Le décodeur joue un rôle crucial dans les architectures de réseaux neuronaux profonds tels que les réseaux neuronaux convolutifs (CNN) et les réseaux neuronaux récurrents (RNN). Le décodeur prend une représentation latente de l'entrée (généralement produite par un encodeur) et la transforme en une sortie de même taille que l'entrée.

Dans le contexte des CNN, le décodeur est généralement la dernière partie du réseau. Il est composé d'une série de couches de déconvolution et d'upsampling. Les couches de déconvolution restaurent la résolution spatiale de la carte de caractéristiques produite par l'encodeur en inversant les effets des couches de convolution de l'encodeur. Les couches d'upsampling augmentent la taille de la carte de caractéristiques en répétant les valeurs voisines, créant ainsi une carte de caractéristiques plus grande.

Dans les architectures de segmentation sémantique, le décodeur est souvent utilisé en combinaison avec un encodeur pour produire une segmentation précise de l'image d'entrée.

L'encodeur extrait les caractéristiques de l'image, qui sont ensuite transmises au décodeur pour générer la carte de segmentation. Le décodeur peut également être utilisé pour générer des images à partir d'un vecteur latent, ce qui est utile dans les applications de génération d'images.

Dans le contexte des RNN, le décodeur est utilisé pour générer une séquence de sortie à partir de la représentation latente produite par l'encodeur. Le décodeur peut également être utilisé pour effectuer d'autres tâches telles que la traduction de texte, la génération de dialogues ou la prédiction de valeurs futures [13,14].

3.3. Bottleneck

Dans l'architecture U-Net, le terme "bottleneck" désigne la partie centrale du réseau qui joue un rôle de réduction ou de compression des informations. Cette section spécifique se trouve entre l'encodeur et le décodeur.

Le "bottleneck" de l'architecture U-Net est constitué de couches de convolution qui réduisent la résolution spatiale des caractéristiques tout en augmentant leur dimensionnalité. Cette étape permet de capturer les informations sémantiques de haut niveau.

Son objectif principal est d'extraire les caractéristiques les plus importantes et les plus abstraites de l'image en entrée, tout en réduisant la quantité de données à traiter. En agissant comme un lien entre l'encodeur et le décodeur, le "bottleneck" transmet les informations essentielles nécessaires pour réaliser des tâches telles que la segmentation précise de l'image.

3.4. Les couches de convolution du modèle U-Net

Les couches de convolution du modèle U-Net que vous mentionnez sont les éléments clés de l'architecture de convolutions descendantes et ascendantes.

- Couches de convolutions descendantes (contractantes) :

✓ Convolution 3x3 avec ReLU : Une couche de convolution est appliquée à l'image d'entrée avec un filtre de taille 3x3. Ensuite, une fonction d'activation ReLU est appliquée aux sorties de la convolution pour introduire la non-linéarité.

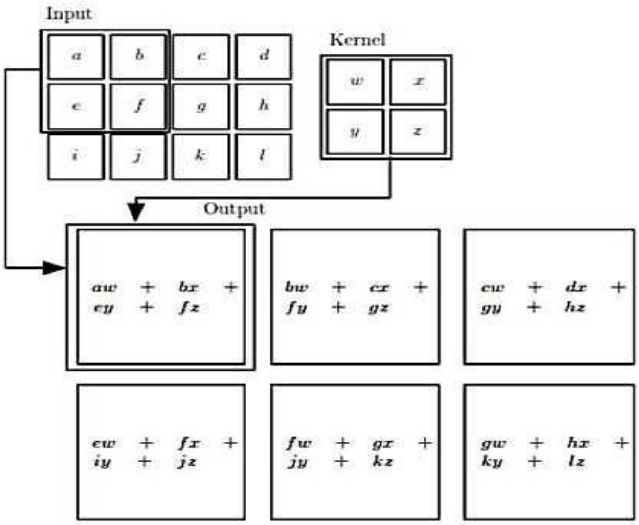


Figure 6: exemple d'une convolution ReLU

- ✓ Copy and Crop : Les activations en sortie de la couche précédente sont copiées et rognées pour avoir la même taille que les activations correspondantes des convolutions ascendantes. Cela permet de fusionner les informations spatiales.

- ✓ Max Pooling : Une opération de max pooling est effectuée pour réduire la dimension spatiale des activations. Cela permet de résumer les informations à une échelle plus élevée.

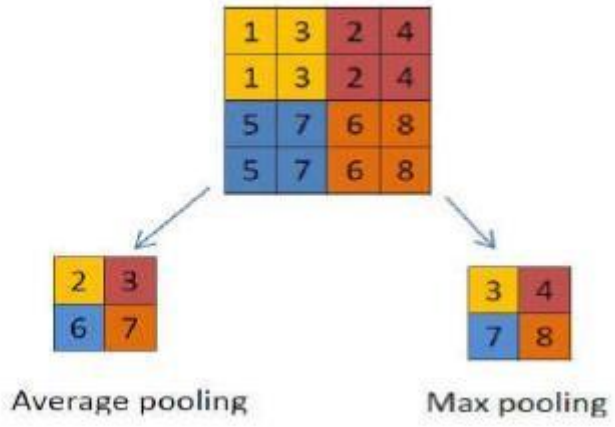


Figure 7: exemple d'une Max Pooling

- Couches de convolutions ascendantes (expansives) :
  - ✓ Up Convolution 2x2 : Une opération de transposée de convolution est utilisée pour augmenter la résolution spatiale des activations. Elle permet de restaurer la résolution après le pooling et d'expanser les pixels.
  - ✓ Concaténation : Les activations de la couche correspondante des convolutions descendantes sont concaténées avec les activations transposées de convolution. Cela permet de fusionner des informations de bas niveau avec des informations de haut niveau.
  - ✓ Convolution 1x1 : Une couche de convolution avec un filtre de taille 1x1 est utilisée pour réduire la dimensionnalité des activations, en réduisant le nombre de canaux.
  - ✓ Convolution 3x3 avec ReLU : Une autre couche de convolution avec un filtre de taille 3x3 est appliquée aux activations fusionnées. La fonction d'activation ReLU est ensuite appliquée.
  - ✓ Les connexions skip, également appelées connexions résiduelles, sont des liens directs entre les couches d'encodage et de décodage dans le modèle U-Net. Elles permettent de relier les informations à différentes échelles spatiales, en fusionnant les caractéristiques extraites à partir de l'encodeur avec celles du décodeur. Cela aide à améliorer la localisation précise des objets dans l'image segmentée en combinant des informations de résolution élevée avec des informations de contexte à plus grande échelle. Les connexions skip sont essentielles pour obtenir de bons résultats de segmentation sémantique dans U-Net.

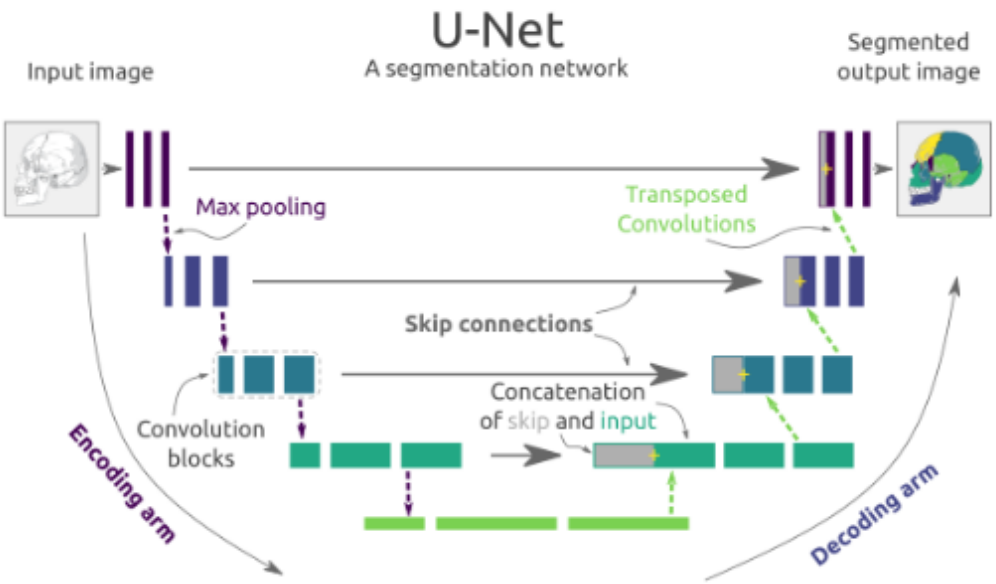


Figure 8: architecture générale encodeur-décodeur

4. Base d'apprentissage

4.1. Types de bases de données

Il existe trois types de bases de données : gratuites, payantes et nécessitant une autorisation spécifique pour leur utilisation.

- Les différences entre ces types sont les suivantes :

✓ Bases de données gratuites : Les bases de données gratuites sont accessibles sans frais. Elles peuvent être open source, ce qui signifie que leur code source est disponible et modifiable par la communauté, ou elles peuvent être mises à disposition gratuitement par une entreprise ou une organisation. Les bases de données gratuites sont souvent utilisées par les développeurs et les petites entreprises disposant de ressources limitées. Elles offrent une alternative économique pour stocker et gérer des données, mais elles peuvent avoir des fonctionnalités limitées ou une communauté de support moins développée par rapport aux bases de données payantes.

✓ Bases de données payantes : Les bases de données payantes nécessitent un achat ou un abonnement pour être utilisées. Elles sont généralement développées et entretenues par des entreprises spécialisées dans les systèmes de gestion de bases de données (SGBD). Les bases de données payantes offrent souvent des fonctionnalités avancées, des performances optimisées, une sécurité renforcée et un support technique professionnel. Elles sont couramment utilisées par les grandes entreprises et les

applications critiques où la fiabilité et les performances sont essentielles. Les bases de données payantes peuvent également inclure des services de support, des mises à jour régulières et des outils de gestion avancés.

- ✓ Bases de données nécessitant une autorisation : Certaines bases de données exigent une autorisation spécifique pour leur utilisation. Il peut s'agir de bases de données propriétaires dont l'accès est limité aux personnes ou organisations ayant obtenu une autorisation formelle de l'entité propriétaire. Ces bases de données peuvent contenir des données sensibles ou confidentielles, et leur accès est généralement restreint pour des raisons de sécurité ou de confidentialité. Les bases de données nécessitant une autorisation peuvent être gratuites ou payantes, mais elles nécessitent une autorisation explicite pour accéder et utiliser les données qu'elles contiennent.

4.2. Base d'apprentissage utilisée

Cityscapes est une base de données de grande envergure largement utilisée dans le domaine de la vision par ordinateur pour la segmentation sémantique et la compréhension des scènes urbaines. Elle a été spécifiquement créée pour fournir des annotations détaillées et précises sur des images de rues provenant de différentes villes européennes. La base de données Cityscapes comprend plus de 5 000 images haute résolution capturées dans des conditions réelles, couvrant une variété de scènes urbaines telles que des rues, des intersections, des trottoirs et des bâtiments.

Chaque image de Cityscapes est annotée avec des étiquettes de segmentation sémantique qui identifient et classifient différents objets et régions présents dans l'image, tels que les voitures, les piétons, les feux de signalisation, les panneaux de signalisation, les trottoirs, les routes, etc. Ces annotations sont réalisées avec une grande précision par des annotateurs humains experts, ce qui garantit la qualité et la fiabilité des données.

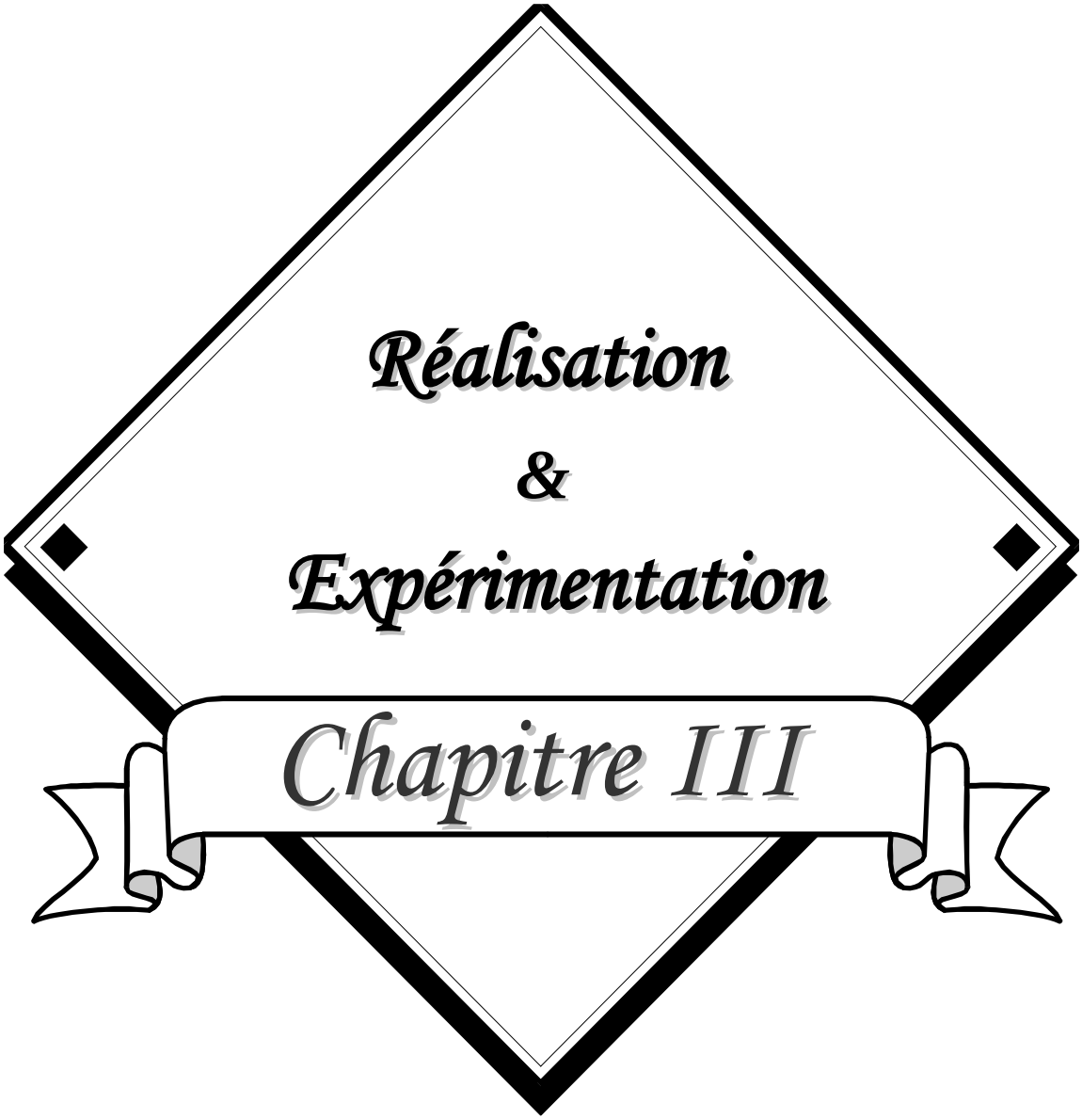
En plus des images et des annotations, Cityscapes propose également des données supplémentaires, telles que des données de profondeur, des images de détection d'instances et des images stéréoscopiques. Ces informations complémentaires permettent d'explorer des tâches plus avancées, comme la détection d'objets spécifiques ou la reconstruction 3D de scènes urbaines.

Grâce à sa richesse en données et à ses annotations détaillées, Cityscapes est devenue une référence dans le domaine de la vision par ordinateur pour l'évaluation et le développement d'algorithmes de segmentation sémantique et de détection d'objets dans des environnements

urbains complexes. Cette base de données a joué un rôle crucial dans l'avancement de la recherche dans ce domaine et continue d'être utilisée comme référence pour évaluer les performances des modèles et des méthodes de pointe.

5. Conclusion

Dans ce chapitre, nous avons présenté notre approche basée sur le modèle U-Net et la base de données Cityscapes pour la segmentation sémantique dans le contexte de la conduite autonome. Cette combinaison représente une avancée significative, offrant aux véhicules autonomes une compréhension précise de leur environnement et améliorant ainsi leur capacité à naviguer en toute sécurité de manière autonome. Des recherches futures et des améliorations continues sont nécessaires pour rendre cette technologie plus robuste et prête à être déployée dans des scénarios réels de conduite autonome.


Réalisation
&
Expérimentation
Chapitre III

Chapitre 3 : Réalisation et Expérimentation

1. Introduction

Ce chapitre présente le processus de développement de l'application. Dans la première partie, nous aborderons l'environnement matériel et logiciel de développement utilisé, ainsi que les choix techniques pris en compte pour sa réalisation. Dans la deuxième partie, nous décrirons le fonctionnement de l'application et ses différentes fonctionnalités.

2. Environnement de développement

Cette section présente la configuration matérielle et logicielle utilisée pour notre projet.

2.1. Environnement Matériel

Nous avons travaillé sur une machine TOSHIBA avec les spécifications suivantes :

- ✓ Processeur : Intel(R) Celeron(R) CPU B830
- ✓ Disque dur : 500 Go
- ✓ RAM : 4 Go
- ✓ Système d'exploitation : Windows 10

2.2. Environnement Logiciel

Pour la réalisation de ce mémoire, nous avons utilisé le langage de programmation Python et la plateforme Google Colab.

- Python est un langage de programmation interprété, polyvalent, de haut niveau et facile à apprendre. Créé par Guido van Rossum en 1991, Python se distingue par sa syntaxe claire et lisible, en en faisant un excellent choix pour les débutants en programmation. Il est largement utilisé dans divers domaines tels que le développement web, l'analyse de données, l'apprentissage automatique, l'automatisation des tâches, les scripts système, etc. Python dispose d'une vaste bibliothèque standard offrant de nombreuses fonctionnalités prêtes à l'emploi, ainsi que de nombreuses bibliothèques tierces développées par la communauté pour des tâches spécifiques.

Figure 9: logo de python

- Google Colab, également connu sous le nom de Google Colaboratory, est une plateforme de cloud computing développée par Google. Elle permet aux utilisateurs d'écrire, d'exécuter et de collaborer sur du code Python directement dans un navigateur web, sans nécessiter l'installation de logiciels supplémentaires sur l'ordinateur local. Google Colab offre une intégration transparente avec Google Drive et fournit des ressources de calcul gratuites pour des tâches de calcul intensif. C'est un outil populaire pour le développement et la recherche en science des données et en intelligence artificielle, offrant également un accès aux GPU pour l'apprentissage en profondeur et l'exécution de bibliothèques telles que Open CV, TensorFlow ou Keras.

Figure 10: Logo de google colaboraty

- Keras est une API (interface de programmation d'application) de haut niveau largement utilisée pour le développement rapide et l'expérimentation de réseaux neuronaux. Elle offre une compatibilité avec les frameworks TensorFlow, CNTK et Theano. Keras est particulièrement populaire dans les domaines du deep learning et de

la reconnaissance d'images. Cette bibliothèque open-source, développée par François Chollet, facilite la création rapide et aisée de modèles de réseaux neuronaux en tirant parti des fonctionnalités offertes par les principaux frameworks tels que TensorFlow.

Figure 11: logo de keras

TensorFlow, Keras présente les caractéristiques suivantes :

- ✓ Rapidité et facilité de prototypage : Keras offre une interface conviviale, modulaire et extensible qui permet de prototyper rapidement des modèles.
- ✓ Orientation vers l'expérience utilisateur : Keras met l'accent sur une expérience utilisateur agréable en proposant une interface intuitive pour la création de réseaux neuronaux.
- ✓ Facilité de production des modèles : Keras facilite également la mise en production des modèles, permettant aux utilisateurs de déployer facilement leurs modèles entraînés dans des environnements de production.
- ✓ Prise en charge des réseaux convolutifs et récurrents : Keras prend en charge à la fois les réseaux convolutifs et récurrents, ainsi que leur combinaison, offrant ainsi une flexibilité pour différents types de tâches d'apprentissage automatique.
- ✓ Fonctionnement transparent sur CPU et GPU : Keras fonctionne de manière transparente sur les architectures CPU et GPU, permettant une utilisation efficace des ressources matérielles disponibles pour accélérer l'entraînement des modèles.
- ✓ Prise en charge de plusieurs backends et plateformes : Keras est compatible avec plusieurs backends tels que Keras lui-même, Theano, CNTK, etc. Il est également multi-plateforme, fonctionnant sur des systèmes tels que Unix, Windows, etc.

- TensorFlow est un framework open source de machine learning développé par Google. Il fournit une plateforme puissante pour le développement et l'exécution d'applications de machine learning et de deep learning. En tant qu'outil polyvalent, TensorFlow permet de résoudre des problèmes mathématiques complexes de manière plus accessible. Il offre aux chercheurs la possibilité de créer et de transformer des architectures d'apprentissage expérimentales en logiciels fonctionnels.

Figure 12: logo de tensorflow

- Matplotlib est une bibliothèque Python largement utilisée pour la création de graphiques 2D. Elle permet de générer des figures de haute qualité et peut être utilisée dans tous les environnements de développement Python. Les graphiques produits avec matplotlib peuvent être enregistrés dans différents formats, ce qui facilite leur utilisation dans des publications scientifiques ou d'autres contextes.
- Numpy est une bibliothèque essentielle pour le calcul scientifique en Python. Elle offre des fonctionnalités pour manipuler des tableaux multidimensionnels et des structures de données matricielles. Numpy permet d'effectuer diverses opérations mathématiques sur ces tableaux, incluant des routines trigonométriques, statistiques et algébriques. La bibliothèque propose également une vaste gamme de fonctions mathématiques, algébriques et de transformation pour répondre aux besoins des utilisateurs.

3. Expérimentation

Dans cette section, nous détaillerons le processus de mise en œuvre de notre modèle de segmentation sémantique.

3.1. Importation de la dataset

- Nous avons commencé par l'importation la dataset.

Dans la base de données Cityscapes contient 6500 images, chaque image est composée de deux parties distinctes : l'image originale et le masque de segmentation associé.

Image originale : L'image originale représente la scène urbaine capturée par une caméra. Elle peut contenir des objets tels que des voitures, des piétons, des bâtiments, des panneaux de signalisation, etc. L'image est généralement au format RGB, ce qui signifie qu'elle est composée de trois canaux de couleur : rouge, vert et bleu. Chaque pixel de l'image contient des valeurs RVB qui définissent l'intensité des trois canaux pour représenter une couleur spécifique.

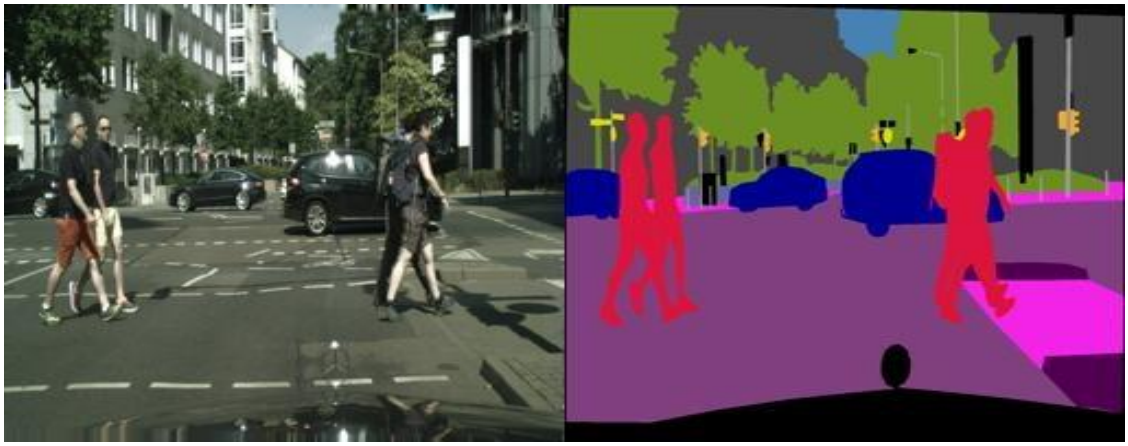
Masque de segmentation : Le masque de segmentation est une image binaire qui représente les différentes classes d'objets présentes dans l'image originale. Il est utilisé pour l'annotation et la segmentation des objets dans la scène urbaine. Chaque pixel du masque de segmentation est étiqueté avec une valeur spécifique qui correspond à une classe d'objet particulière. Par exemple, un pixel avec la valeur 0 peut représenter le fond (l'environnement urbain sans aucun objet), tandis qu'un pixel avec la valeur 1 peut représenter une voiture. Chaque classe d'objet est associée à une valeur entière unique dans le masque.

- ✓ L'image originale de 512 pixels est divisée en deux parties distinctes :

La première partie est une sous-image de 256 pixels de largeur et 256 pixels de hauteur, représentant la partie supérieure ou la partie gauche de l'image originale.

La deuxième partie est également une sous-image de 256 pixels de largeur et 256 pixels de hauteur, représentant la partie inférieure ou la partie droite de l'image originale.

Cette division en deux parties égales permet de créer des paires d'images et de masques de segmentation correspondants pour l'apprentissage supervisé, où chaque sous-image est associée à son masque de segmentation correspondant, permettant ainsi la segmentation précise des objets dans chaque partie de l'image originale.



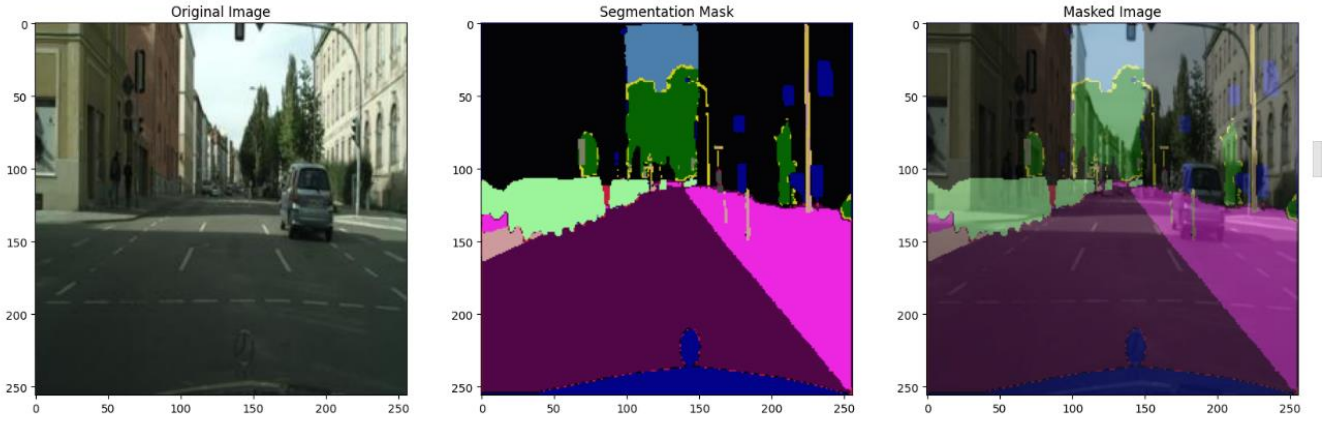


Figure 13: résultat d'importation dataset

3.2. Réalisation du modèle

Ce résultat de code représenter la réalisation et l'architecture de modèle basé sur l'UNET.

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_1 (InputLayer)           [(None, 256, 256, 3  0           []
                                )]

 conv2d (Conv2D)                (None, 256, 256, 64  1792        ['input_1[0][0]']
                                )

 batch_normalization (BatchNorm  (None, 256, 256, 64  256        ['conv2d[0][0]']
 alization)                     )

 conv2d_1 (Conv2D)              (None, 256, 256, 64  36928       ['batch_normalization[0][0]']
                                )

 batch_normalization_1 (BatchNo  (None, 256, 256, 64  256        ['conv2d_1[0][0]']
 rmalization)                   )

 conv2d_2 (Conv2D)              (None, 256, 256, 64  36928       ['batch_normalization_1[0][0]']
                                )

 batch_normalization_2 (BatchNo  (None, 256, 256, 64  256        ['conv2d_2[0][0]']
 rmalization)                   )

 max_pooling2d (MaxPooling2D)   (None, 128, 128, 64  0           ['batch_normalization_2[0][0]']
                                )

 conv2d_4 (Conv2D)              (None, 128, 128, 12  73856       ['max_pooling2d[0][0]']
                                8)

 batch_normalization_4 (BatchNo  (None, 128, 128, 12  512        ['conv2d_4[0][0]']
 rmalization)                   8)

 conv2d_5 (Conv2D)              (None, 128, 128, 12  147584      ['batch_normalization_4[0][0]']
                                8)

 batch_normalization_5 (BatchNo  (None, 128, 128, 12  512        ['conv2d_5[0][0]']
 rmalization)                   8)
__________________________________________________________________________________________________
==============================================================
Total params: 77,343,821
Trainable params: 77,321,037
Non-trainable params: 22,784
______________________________________________________________
```

Figure 14: réalisation du modèle

Le nombre total de paramètres reflète la complexité du modèle et sa capacité à apprendre à partir des données. Un grand nombre de paramètres entraînables permet au modèle de capturer des relations complexes et d'ajuster ses prédictions en conséquence. Les paramètres non entraînables, bien qu'en nombre moins élevé, peuvent également contribuer à la performance globale en introduisant des invariances ou des contraintes spécifiques.

3.3. L'architecture du modèle

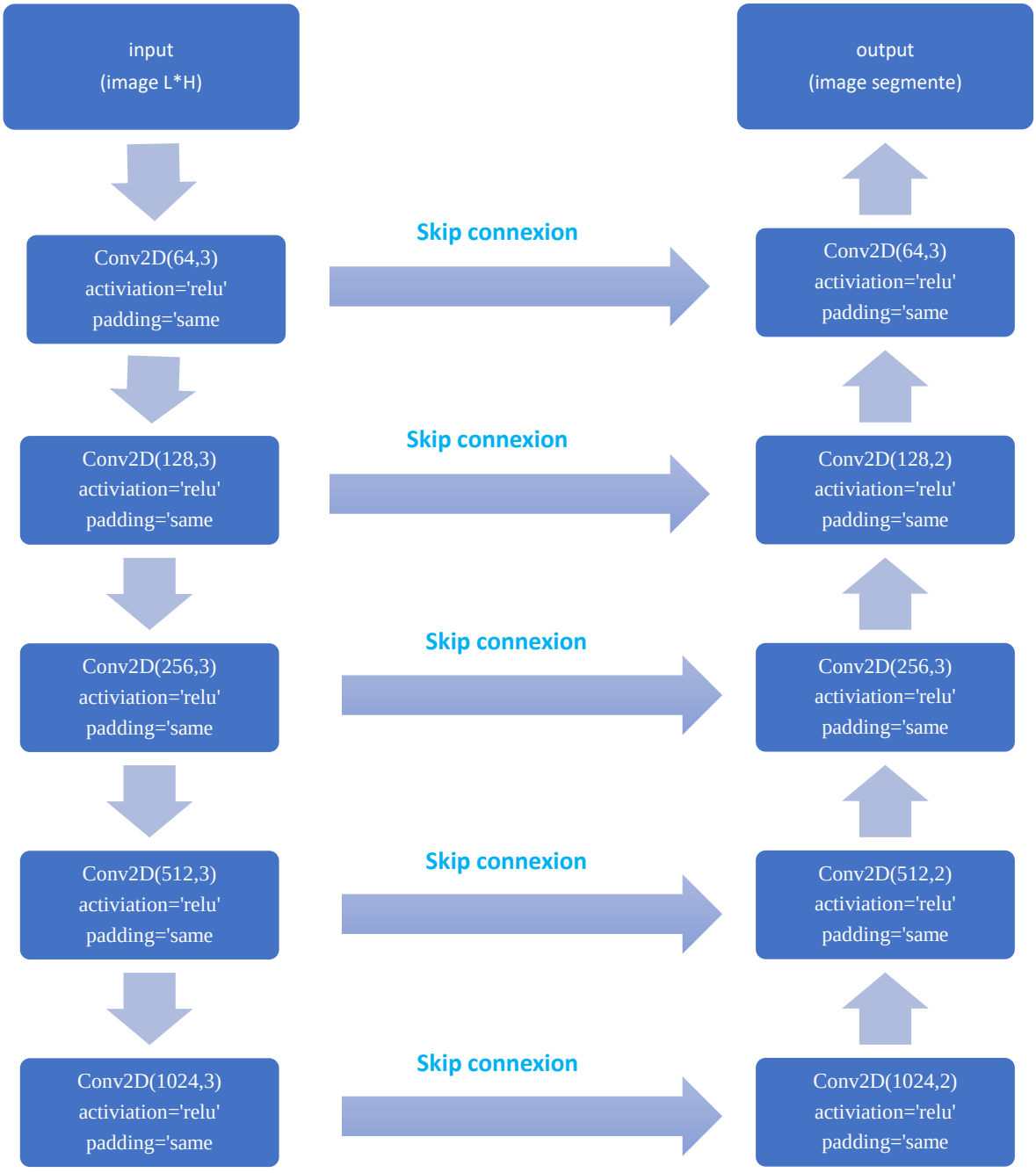


Figure 15: architecture du modèle crée

3.4. L’entrainement de modèle

Dans cette partie, nous évaluons notre modèle en termes de précision et de performance pour mettrons en évidence l’efficacité de notre approche.

```
Epoch 1/20
297/297 [==============================] - 298s 1s/step - loss: 0.2522 - accuracy: 0.9201 -
Epoch 2/20
297/297 [==============================] - 311s 1s/step - loss: 0.2364 - accuracy: 0.9247 -
Epoch 3/20
297/297 [==============================] - 311s 1s/step - loss: 0.2315 - accuracy: 0.9259 -
Epoch 4/20
297/297 [==============================] - 306s 1s/step - loss: 0.2282 - accuracy: 0.9266 -
Epoch 5/20
297/297 [==============================] - 306s 1s/step - loss: 0.2181 - accuracy: 0.9297 -
Epoch 6/20
297/297 [==============================] - 306s 1s/step - loss: 0.2119 - accuracy: 0.9313 -
Epoch 7/20
297/297 [==============================] - 310s 1s/step - loss: 0.2015 - accuracy: 0.9343 -
Epoch 8/20
297/297 [==============================] - 310s 1s/step - loss: 0.2023 - accuracy: 0.9341 -
Epoch 9/20
297/297 [==============================] - 306s 1s/step - loss: 0.2114 - accuracy: 0.9311 -
Epoch 10/20
297/297 [==============================] - 306s 1s/step - loss: 0.2232 - accuracy: 0.9274 -
Epoch 11/20
297/297 [==============================] - 306s 1s/step - loss: 0.2063 - accuracy: 0.9326 -
Epoch 12/20
297/297 [==============================] - 306s 1s/step - loss: 0.2122 - accuracy: 0.9307 -
Epoch 13/20
297/297 [==============================] - 306s 1s/step - loss: 0.1872 - accuracy: 0.9381 -
Epoch 14/20
297/297 [==============================] - 306s 1s/step - loss: 0.1741 - accuracy: 0.9420 -
Epoch 15/20
297/297 [==============================] - 310s 1s/step - loss: 0.1664 - accuracy: 0.9443 -
Epoch 16/20
297/297 [==============================] - 311s 1s/step - loss: 0.1629 - accuracy: 0.9452 -
Epoch 17/20
297/297 [==============================] - 310s 1s/step - loss: 0.1609 - accuracy: 0.9457 -
Epoch 18/20
297/297 [==============================] - 310s 1s/step - loss: 0.1593 - accuracy: 0.9460 -
Epoch 19/20
297/297 [==============================] - 310s 1s/step - loss: 0.1569 - accuracy: 0.9467 -
Epoch 20/20
297/297 [==============================] - 310s 1s/step - loss: 0.1547 - accuracy: 0.9473 -
```

Figure 16: apprentissage du modèle

Le code fournit les résultats de l'entraînement d'un modèle, affichant les informations pour chaque époque. Chaque ligne représente une époque avec le numéro de l'époque, la durée écoulée, la perte et l'exactitude du modèle sur les données d'entraînement et de validation. Le modèle a été entraîné pendant 20 époques, et on observe une diminution de la perte et une augmentation de l'exactitude au fil des époques, indiquant une amélioration des performances du modèle sur les données d'entraînement. La précision sur les données de validation peut varier légèrement d'une époque à l'autre, mais globalement, une tendance à l'amélioration de la précision est observée.

Ces résultats permettent de suivre l'apprentissage du modèle et d'évaluer ses performances au cours de l'entraînement.

3.5. Evaluation du modèle

La mesure de l'exactitude (accuracy) et de la perte (loss) peut être calculée de la manière suivante :

Exactitude (Accuracy) : L'exactitude est obtenue en comparant les prédictions du modèle avec les valeurs réelles des données. On compte le nombre d'échantillons pour lesquels la prédiction est correcte, puis on divise ce nombre par le total des échantillons.

Formule :

Accuracy = (Nombre d'échantillons correctement prédits) / (Nombre total d'échantillons) (1)

Perte (Loss) : La perte mesure l'erreur entre les prédictions du modèle et les valeurs réelles. Selon le type de problème, différentes fonctions de perte peuvent être utilisées. Par exemple, dans une tâche de classification binaire, on utilise souvent l'entropie croisée binaire (binary cross-entropy) comme fonction de perte.

Formule :

Loss = Somme des pertes individuelles / Nombre total d'échantillons (2)

Pour chaque échantillon individuel, la perte est calculée en utilisant la fonction de perte spécifique du problème. Par exemple, pour l'entropie croisée binaire, la perte pour un échantillon peut être calculée à l'aide de la formule suivante :

$$Loss_i = -[y_i * \log(p_i) + (1 - y_i) * \log(1 - p_i)] \quad (3)$$

Ici, y_i représente la valeur réelle de l'échantillon (0 ou 1) et p_i représente la prédiction du modèle pour cet échantillon (un nombre entre 0 et 1, représentant la probabilité d'appartenance à la classe 1).

La perte totale est ensuite calculée en sommant les pertes individuelles et en divisant le résultat par le nombre total d'échantillons.

Veuillez noter que les formules précises peuvent varier selon le type de modèle et la fonction de perte utilisée. Celles-ci sont des exemples couramment utilisés dans les problèmes de classification.

3.6. Résultat final du modèle

Nous avons obtenu le résultat suivant :

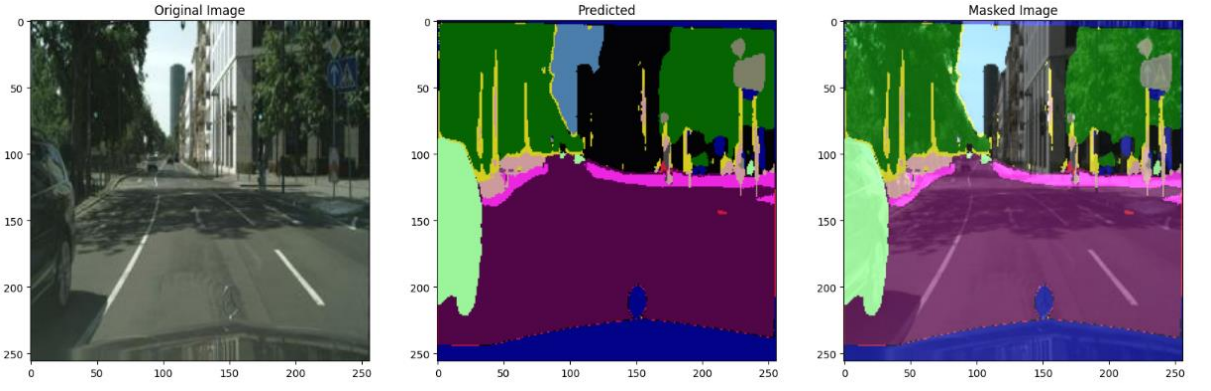
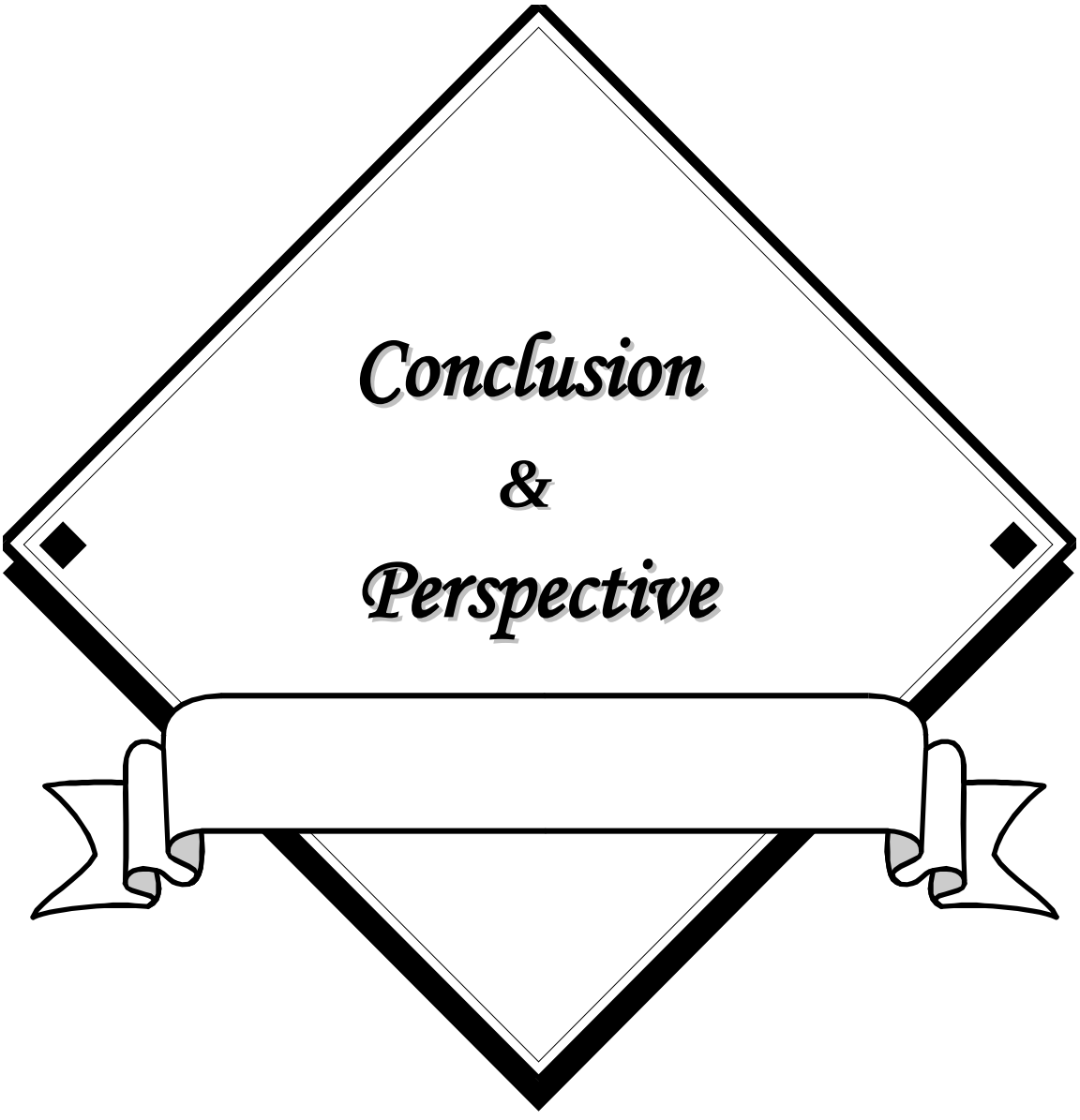


Figure 17: résultat final du modèle

Nous voyons bien que les résultats obtenus montrent que le modèle de segmentation sémantique a réussi à apprendre et à généraliser à partir des données d'entraînement, et il est capable de fournir des prédictions précises sur de nouvelles images.

4. Conclusion

Au cours de ce chapitre, nous avons commencé par présenter les choix technologique et l'environnement logiciel utilisé pour réaliser ce travail. Ensuite, nous avons présenté notre travail par quelque captures écran de structure de notre solution et le test réalisé.


Conclusion
&
Perspective

Conclusion et perspective

Durant l'élaboration de ce mémoire, qui a pour but la réalisation une nouvelle approche de segmentation sémantique basé sur l'UNET appliquée à la conduite autonome.

Cette nouvelle approche de segmentation sémantique basée sur le modèle U-Net représente une avancée significative dans le domaine de la conduite autonome. Elle ouvre la voie à des systèmes de perception plus performants, capables de reconnaître et de comprendre leur environnement avec une précision accrue. Cette avancée contribue ainsi à rendre la conduite autonome plus sûre et plus fiable, et rapproche la réalisation de la vision d'un futur où les véhicules autonomes sont largement déployés sur nos routes.

Comme perspectives, nous suggérons aux futures promotions de faire des études et des travaux de simulation sur d'autres model d'imagerie de segmentation en utilisant une autre technique de classification du CNN, et des autres algorithmes, en collaboration avec des centre d'imagerie de conduite autonome et des programme plus performante. Pour réaliser sa propre segmentation sémantique, il est nécessaire d'utiliser sa propre base étiquetée, Ceci est un travail fastidieux mais qui est nécessaire dans le cas d'un domaine d'utilisation bien défini au préalable.

Les systèmes de conduite autonome doivent être capables de prendre des décisions en temps réel et l'amélioration de la robustesse aux conditions.

Bibliographie

[1] : C. Houassine, segmentation d’images par une approche biomimétique hybride. université universite m’hamed bougara- boumerdes. 2012.

[2] : Sarah GHANDOUR. Segmentation d'images couleurs par morphologie mathématique : application aux images microscopiques. PhD thesis, Université de Toulouse III - Paul Sabatier, 2010.

[3] : Mr Cyril Meurie. Segmentation d'images couleur par classication pixellaire ethiérarchie de partitions. PhD thesis, Université de CAEN/BASSENORMANDIE, 2005.

[4] : J. Cocquerez and S. Philipp. Analyse d'images : ltrage et segmentation, 1995. Paris Masson

[5] :« Object Recognition vs Object Detection vs Image Segmentation | Data Science and Machine Learning ». https://www.kaggle.com/getting-started/169984 (consulté le 22 mai 2022).

[6] : R. B. Girshick, J. Donahue, T. Darrell, et J. Malik, « Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation », 2014 IEEE Conference on Computer Vision and Pattern Recognition, p. 580-587, 2014, doi : 10.1109/CVPR.2014.81.

[7] : R. Gandhi, « R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms », Medium, 9 juillet 2018. https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e (consulté le 22 mai 2022).

[8] : R. Tedrake, « Object detection and segmentation », 20 mai 2022. https://manipulation.csail.mit.edu/segmentation.html (consulté le 22 mai 2022).

[9] : H. Chehri, A. Chehri, L. Kiss, et A. Zimmermann, « Automatic Anode Rod Inspection in Aluminum Smelters using Deep-Learning Techniques : A Case Study », Procedia Computer Science, vol. 176, p. 3536-3544, janv. 2020, doi: 10.1016/j.procs.2020.09.033.

[10] : F. Chollet, Deep Learning with Python. Manning, 2017

[11] : A.N.Benaichouche, "Conception de métaheuristiques d'optimisation pour la segmentation d'images. Application aux images IRM du cerveau et aux images de Tomographie par Émission de positons ", thèse de doctorat université paris 12, 2012.

[12] : MOHAMMEDI Hanane BENBERNOU Nacera, Etude comparative entre les cartes de Kohonen et K-means (Application à la segmentation des images satellitaires), Mémoire de fin d'études pour l'obtention du diplôme d'ingénieur d'état en Informatique, université Dr. Tahar Moulay Saida 2016-2017

[13] : Automatical Segmentation : Application to 3D Angiograms of the Live, DELINGETTE Institut National de Recherche en Informatique et en Automatique Projet EPIDAUR E 2 004, route des Lucioles, BP 9 3 06 902 Sophia Antipolis Cedex, France, Luc SOLER, Grégoire MALANDAIN et Hervé, Traitement du Signal – Volume 15 - n°5 – Spécial 1998.

[14] : Peijun Hu, Fa Wu, Jialin Peng [et al], « Automatic abdominal multi-organ segmentation using deep convolutional neural network and time-implicitlevel sets»,3 November 2016

Résumé

La segmentation d'image est un sujet clé dans le traitement d'image et la vision par ordinateur avec des applications telles que la compréhension de scènes, l'analyse d'images médicales, la segmentation sémantique de conduite autonome, la vidéosurveillance, la réalité augmentée et la compression d'images, et autres. Divers algorithmes de segmentation d'images ont été développés dans le domaine scientifique. Dans ce travail nous avons proposé une nouvelle approche de segmentation sémantique base sur l'UNET pour segmenter les images de conduite autonome.

Mots clés : segmentation d'image, segmentation sémantique, conduite autonome, UNET

Abstract

Image segmentation is a key topic in image processing and computer vision with applications such as scene understanding, medical image analysis, semantic segmentation for autonomous driving, video surveillance, augmented reality and image compression, and others. Various image segmentation algorithms have been developed in the scientific field. In this work we proposed a new semantic segmentation approach based on UNET to segment driving images.

Keywords : Image segmentation, semantic segmentation, autonomous driving, UNET

الخلاصة

يعد تجزئة الصور موضوعا رئيسيا في معالجة الصور ورؤية الحاسوب مع تطبيقات مثل فهم المشهد وتحليل الصور الطبية والتجزئة الدلالية للقيادة الذاتية والمراقبة بالفيديو، والواقع المعزز وضغط الصور، وغيرها. تم تطوير خوارزميات مختلفة لتجزئة الصور في المجال العلمي. في هذا العمل اقترحنا نهجا جديدا للتجزئة الدلالية يعتمد على يونات لتقسيم صور القيادة المستقلة.

***المفاتيح**: تجزئة الصورة، التجزئة الدلالية، القيادة المستقلة، يونات.*