الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université d'Ain Temouchent - Belhadj Bouchaib
Faculté des Sciences et Technologie
Département de Mathématiques et Informatique

Projet de Fin d'Etudes
Pour l'obtention du diplôme de Master en : Informatique
Domaine : Mathématiques et Informatique
Filière : Informatique
Spécialité : Cyber Sécurité et Intelligence Artificielle

Thème

Approche d'apprentissage profond pour la segmentation sémantique d'images

Soutenu le : 24 / 06 / 2024

Présenté par :

- Melle : BAROUDI RANIA
- Melle : BENCHIHA ZAHRA

Devant le jury composé de :

Mr BOUAFIA.Z	**MAA**	UAT.B.B (Ain Temouchent)	Président
Mme BELGRANA F.Z	**MCA**	UAT.B.B (Ain Temouchent)	Examinatrice
Mme SAIDI S.	**MAA**	UAT.B.B (Ain Temouchent)	Encadrante

Année Universitaire : 2023/2024

REMERCIEMENTS

*Avant tout, nous exprimons notre gratitude à **Allah**, Le Tout-Puissant et Miséricordieux, qui nous a accordé la force, le courage et la patience nécessaires pour mener à bien ce modeste travail.*

*Nous tenons à remercier notre Encadrante, **Madame Saidi Samira** pour ses précieux conseils, son aide durant toute la période du travail et pour tout le soutien et l'orientation.*

Nous tenons aussi à remercier les membres du jury pour leur précieux temps accordé à l'étude de notre mémoire.

Nous remercions nos parents et nos proches pour l'amour et le soutien constant qu'ils nous ont témoigné tout au long de notre parcours.

Dédicace

De tout mon cœur, je dédie ce travail :

A mes chers parents, pour leur bienveillance, et leur soutien tout au long de ma formation. A ma sœur Amina pour tous les soutiens et l'aide.

A mon frère youcef,

et A tous mes enseignants de la première année licence jusqu'au master2

A mes très chères copines, ma second famille :

Belaggoun amina , Benchiha zahra, Bensalah soumia ,Djedoui khadija,Meggueni yousra,Dada narimen.

Rania

Dédicace

Je dédie ce mémoire à ma famille et mes amis qui m'ont soutenu tout au long de ce parcours académique. Leur amour, leur patience et leurs encouragements ont été une source inestimable de force et de motivation.

À mes parents, pour leur soutien inconditionnel et leur croyance en mes capacités. Vous avez été mes piliers et votre foi en moi m'a permis de continuer même dans les moments les plus difficiles.

A mes chers frères Imad, Azzedine, Yasser

A ma très chère sœur Amina

zahra

ملخص:

يمثل التجزئة الدلالية في التعلم العميق مجالًا مهمًا للبحث في رؤية الكمبيوتر، بهدف تعيين تسميات دلالية لكل بكسل من الصورة. أحدثت التطورات في التعلم العميق ثورة في هذا التخصص، حيث قدمت أبنية متطورة مثل الشبكات العصبية التلافيفية (CNNs) تركز هذه الدراسة على استخدام نموذج U–Net للتجزئة الدلالية، مع التركيز على المشاهد الحضرية من مجموعة بيانات Cityscapes .

لتحسين أداء نموذج U–Net في التجزئة الدلالية، تم دمج طبقات تلافيفية إضافية مع تعديلات محددة أخرى. مكنت هذه التحسينات النموذج من التقاط التفاصيل المعقدة بشكل أكثر كفاءة، مع تنظيم عملية التعلم بشكل فعال.

الكلمات المفتاحية: التعلم العميق، التجزئة الدلالية، بنية U-Net ، CNN، مناظر المدينة، صورة حضرية.

Résumé :

La segmentation sémantique dans l’apprentissage profond représente un domaine de recherche crucial en vision par ordinateur, visant à attribuer des étiquettes sémantiques à chaque pixel d'une image. Les avancées dans le domaine de l’apprentissage profond ont révolutionné cette discipline, en introduisant des architectures sophistiquées telles que les réseaux de neurones convolutifs (CNN).Cette étude se concentre sur l'utilisation du modèle U-Net pour la segmentation sémantique, en mettant l'accent sur les scènes urbaines de l'ensemble de données Cityscapes.

Pour améliorer les performances du modèle U-Net en segmentation sémantique, des couches convolutionelles supplémentaires ont été intégrées ainsi que d'autres ajustements spécifiques. Ces optimisations ont permis au modèle de capturer plus efficacement les détails complexes tout en régulant efficacement le processus d'apprentissage.

Mots clés : Apprentissage profond, la segmentation sémantique, architecture U-Net, CNN, Cityscapes , Image urbain.

Abstract:

Semantic segmentation in deep learning represents a crucial area of research in computer vision, aiming to assign semantic labels to each pixel of an image. Advances in deep learning have revolutionized this discipline, introducing sophisticated architectures such as convolutional neural networks (CNNs).This study focuses on the use of the U-Net model for semantic segmentation, with emphasis on urban scenes from the Cityscapes dataset.

To improve the U-Net model's performance in semantic segmentation, additional convolutional layers were integrated along with other specific adjustments. These optimizations enabled the model to capture complex details more efficiently, while effectively regulating the learning process.

Keywords: Deep learning, semantic segmentation, U-Net architecture, CNN, Cityscapes, Urban image.

Table des matières

CHAPITRE I: Segmentation des images

CHAPITRE II: Deep Learning

CHAPITRE III: Implémentation et discussion des résultats

Liste des figures

Liste des figures

Liste des tableaux

Liste des abréviations

AE : AutoEncoder neural network.

Adam : Adaptive moment estimation.

ANN : Artificial Neural Network.

CNN : Convolutional Neural Network.

GPU : Graphics Processing Unit.

IA : Intelligence Artificielle.

MLP : Multi Layer Perceptron.

Ng : Niveaux de gris.

Relu : Linear Rectification Unit.

RGB : Red Green Blue.

RNN : Recurrent Neural Network.

RVB : Rouge Vert Bleu.

SGD : Stochastic Gradient Descent.

Tanh : Tangent Hyperbolique.

TL : Transfer Learning.

VGG : Visual Geometry Group.

Introduction générale

La segmentation sémantique constitue un pilier fondamental de la vision par ordinateur, visant à comprendre le contenu visuel d'une image à un niveau pixel. Cette tâche complexe consiste à attribuer à chaque pixel une étiquette sémantique correspondant à la classe d'objet à laquelle il appartient. L'avènement des méthodes de deep learning, en particulier des réseaux de neurones convolutionnels (CNN) a radicalement transformé la manière dont la segmentation sémantique est abordée, permettant des avancées significatives en termes de la compréhension et l'analyse des scènes visuelles et de vitesse de traitement.

Dans ce contexte, la base de données Cityscapes s'est imposée comme une référence incontournable pour l'évaluation et le développement d'un modèle de segmentation sémantique dans des environnements urbains réels. Cette base de données offre une vaste collection d'images capturées dans des environnements urbains variés, et annotées avec une granularité élevée, fournissant des annotations détaillées pour chaque pixel de l'image. Chaque image est associée à des étiquettes sémantiques qui identifient et classifient différents éléments tels que les routes, les bâtiments, les personnes, les véhicules, les panneaux de signalisation, et bien d'autres,

Le recours au deep learning pour la segmentation sémantique sur la base de données Cityscapes présente un double intérêt. D'une part, les réseaux de neurones profonds ont démontré leur capacité à apprendre des représentations de haut niveau à partir de données brutes, ce qui est crucial pour la segmentation précise des scènes urbaines complexes. D'autre part, l'utilisation de données provenant de Cityscapes permet de former des modèles qui sont non seulement performants, mais aussi généralisables à une variété de contextes urbains réels.Cette capacité à segmenter et à comprendre les scènes visuelles revêt une importance cruciale dans de nombreux domaines, notamment la conduite autonome, la surveillance vidéo/urbaine et la réalité augmentée.

Cependant, malgré les progrès réalisés et les ensembles de données riches comme Cityscapes, la segmentation sémantique reste confrontée à plusieurs défis majeurs. L'un des défis principaux réside dans la diversité et la complexité des scènes urbaines avec ses conditions d'éclairage variables, où une multitude d'objets, de structures et de contextes interagissent de manière dynamique. De plus, la disponibilité de données annotées de haute qualité est essentielle

pour l'entraînement efficace des modèles de segmentation, mais la création de telles bases de données reste souvent coûteuse et fastidieuse.

Le défi principal de ce travail de recherche est donc de développer un modèle de segmentation sémantique robuste, spécifiquement adaptées aux environnements urbains complexes représentés dans la base de données Cityscapes. Alors, Comment concevoir des architectures de réseaux de neurones capables de capturer efficacement la variabilité et la complexité des scènes urbaines, tout en assurant une segmentation performante ?

Dans le cadre de ce projet de fin d'études, on essaye d'adopter une architecture de segmentation sémantique plus développée et innovante, spécifiquement conçue pour traiter les défis posés par la diversité des scènes urbaines. Cette architecture intègre des techniques de deep learning, en exploitant pleinement les informations spatiales et contextuelles pour améliorer la précision de la segmentation.

En outre, nous évaluons les performances de l'architecture proposées en utilisant la métrique accuracy et on compare les résultats avec un autre travail mettant en lumière les avantages de nos contributions.

Le présent mémoire s'articule autour de trois chapitres organisés comme Suits :

Une introduction où on situe notre projet de fin étude et son plan

- Dans le premier chapitre, Nous avons présenté un aperçu général sur les images et leurs caractéristiques, ensuite nous avons résumé les types de la segmentation des images et les différentes approches et méthodes utilisés dans ce domaine.
- Ensuite, le deuxième chapitre est un chapitre théorique nécessaire qui vise à assimiler les divers concepts de base de l'apprentissage profond (Deep Learning) avec un état de l'art vers la fin du chapitre.
- Puis, Dans le troisième chapitre, on va présenter la partie expérimentale de notre travail, en fournissant plus de détails sur le modèle de segmentation proposée et on discute les différents résultats obtenus.

Enfin, nous achevons ce mémoire par une conclusion générale sur le travail que nous avons effectué. En soulignant les limites rencontrées et les perspectives pour des recherches futures dans ce domaine en évolution rapide.

En résumé, ce travail de recherche vise à relever le défi de la segmentation sémantique dans les environnements urbains complexes, en proposant une architecture de segmentation sémantique plus développée en utilisant le Deep learning pour une compréhension meilleur des images.

Chapitre I:

Segmentation des images

1. Introduction :

La segmentation sémantique est une technique cruciale dans le domaine de la vision par ordinateur et de l'analyse d'images. Elle vise à diviser une image en différentes régions significatives et à attribuer des étiquettes sémantiques à chaque région, facilitant ainsi la compréhension et l'interprétation des images par les machines.

Ce chapitre explore les bases de l'image numérique et de la segmentation, en commençant par une définition et une analyse des images. Nous examinons les différents types d'images, qu'elles soient binaires, en niveaux de gris ou en couleurs, et leurs caractéristiques essentielles telles que le pixel, la dimension, la résolution, et d'autres attributs importants comme l'homogénéité, la luminance, le contraste, l'histogramme, le bruit, la région et le contour. Ensuite, nous introduisons la segmentation d'image en définissant ses différents types, notamment la segmentation sémantique, la segmentation d'instance et la segmentation panoptique. Nous comparons également la segmentation des instances à la segmentation sémantique pour mieux comprendre leurs différences et leurs applications respectives. Enfin, nous explorons les différentes approches et méthodes de segmentation, telles que la segmentation basée sur les contours, la segmentation basée sur les régions, et la segmentation basée sur la classification des pixels. Chaque méthode est détaillée avec ses techniques spécifiques.

Ce parcours nous permettra de comprendre les techniques et les algorithmes utilisés pour segmenter une image de manière précise et efficace, fournissant ainsi une base solide pour les applications avancées de la vision par ordinateur.

2. Image numérique :

2.1. Définition de l'image :

L'image est une représentation d'une personne ou d'un objet par la peinture, sculpture, le dessin, la photographie, le film...etc. C'est aussi un ensemble structuré d'informations qui, après l'affichage sur l'écran, ont une signification pour l'œil humain.

Elle peut être décrite sous la forme d'une fonction (x, y) de brillance analogique continue, définie dans un domaine borné, tel que x et y sont les coordonnées spatiales d'un point de l'image et I est une fonction d'intensité lumineuse et de couleur. Sous cet aspect, l'image est inexploitable par la machine, ce qui nécessite sa numérisation [1].

2.2. Définition de l'image numérique :

L'image numérique est l'image dont la surface est divisée en éléments de taille fixe appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleurs.

La numérisation d'une image est la conversion de celle-ci de son état analogique en une image numérique représentée par une matrice bidimensionnelle de valeurs numériques f(x,y), comme le montre la figure 01 suivante :

x,y : Coordonnées cartésiennes d'un point de l'image.

f(x, y) : Niveau d'intensité. La valeur en chaque point exprime la mesure d'intensité lumineuse perçue par le capteur. [2]

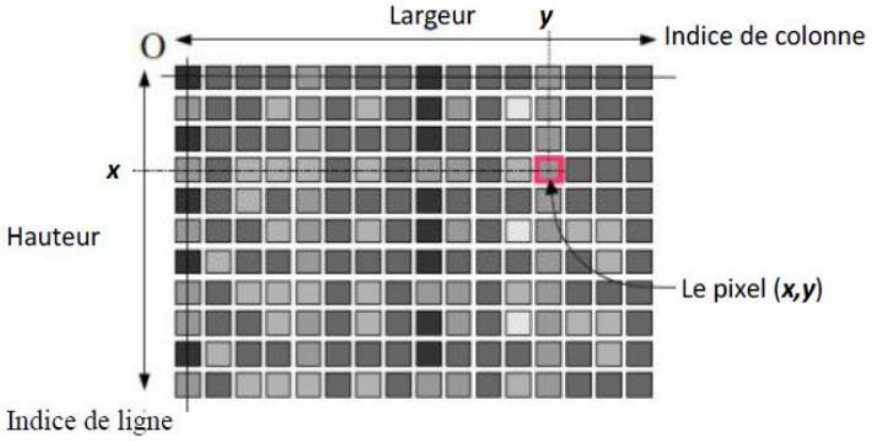


Figure I-01: Représentation d'une image numérique. [2]

2.3. Les Types des images :

On distingue trois types d'images :

- **Binaire :** 2 couleurs (arrière-plan et avant-plan).
- **Monochrome :** variations d'une même teinte.
- **Polychrome :** « vraies » couleurs. [2]

2.3.1. Image binaire (Noir et blanc) :

Une image binaire (ou image noir et blanc) est une image MxN où chaque point peut prendre uniquement la valeur 0 ou 1. Les pixels sont noirs (0) ou blancs (1). Le niveau de gris est codé sur un bit (Binary digIT). Avec Ng = 2 et la relation sur les niveaux de gris devient : p (i,j) = 0 ou p (i,j) = 1. [3]

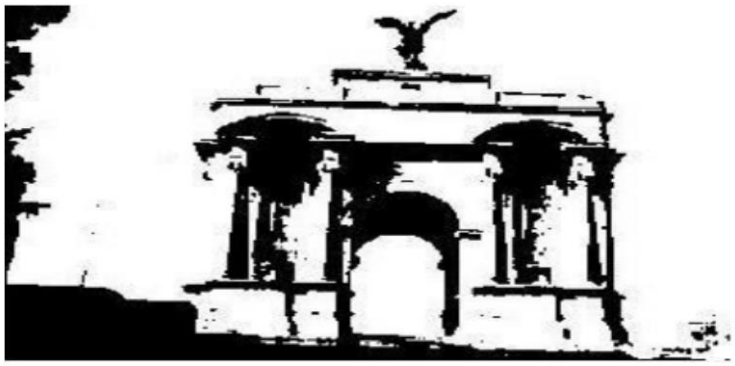


Figure I-02: Image binaire (noir et blanc). [2]

2.3.2. Image en niveaux de gris (Monochromes) :

En général, les images en niveaux de gris sont des images de profondeur 8 bits donc chaque pixel peut prendre l'une des valeurs de l'intervalle [0…255], où la valeur 0 représente la brillance minimale (le noir) et 255 la brillance maximale (le blanc). Ce type d'image est fréquemment utilisé pour reproduire des photos en noir et blanc ou du texte. [4]

Figure I-03: Image monochrome. [2]

2.3.3. Image en couleurs (Polychromes) :

L'espace couleur est basé sur la synthèse additive des couleurs, c'est-à-dire que le mélange entre différentes couleurs donne une nouvelle couleur. La plupart des images couleurs sont basées sur trois couleurs primaires : Rouge, Vert et Bleu (RVB) (RGB en anglais), et utilisent typiquement 8 $bits$ pour chaque composante de couleur, donc chaque pixel nécessite $3 * 8 = 24$ $bits$ pour coder les trois composantes, et chaque composante de couleur peut prendre l'une des valeurs de l'intervalle [0 … 255]. [4]

Figure I-04: Image couleur (polychrome). [2]

On peut convertir une image RVB en niveaux de gris selon plusieurs méthodes la plus simple est de faire $Gris = (Bleu + Vert + Rouge) / 3$ **(Équation I-01)** c'est équivalent d'affecter la couleur grise à chacune des trois composantes RVB. [4]

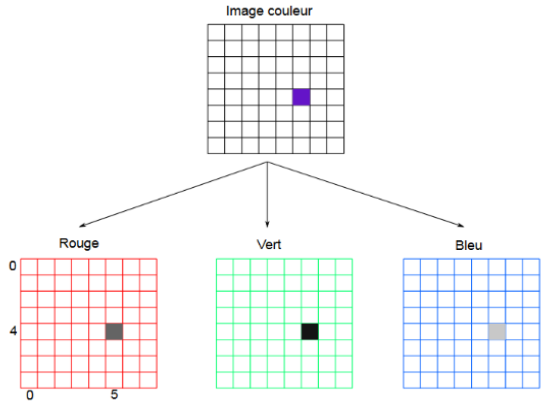


Figure I-05: Structure image en couleur. [4]

3. Les caractéristiques d'une image :

L'image est un ensemble structuré d'informations caractérisé par les paramètres suivants:

3.1. Le pixel :

Une image est constituée d'un ensemble de points appelés pixels (pixel est une abréviation de PICture ELement) Le pixel représente ainsi le plus petit élément constitutif d'une image numérique. L'ensemble de ces pixels est contenu dans un tableau à deux dimensions constituant l'image, et ils fournissent toute l'information qui constitue l'image dans son intégralité. [5]

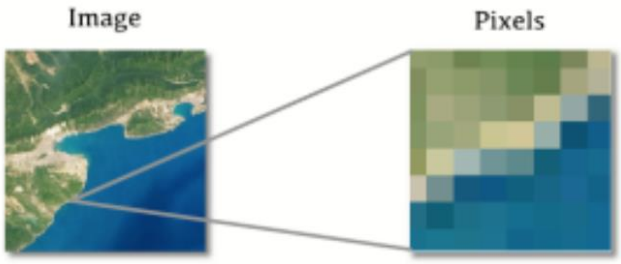


Figure I-06 : Représentation pixels d'une image. [6]

3.2. La dimension :

C'est la taille de l'image. Cette dernière se présente sous forme de matrice dont les éléments sont des valeurs numériques représentatives des intensités lumineuses (pixels). Le nombre de lignes de cette matrice multipliée par le nombre de colonnes nous donne le nombre total de pixels dans une image. [7]

3.3. La résolution :

La résolution est le nombre de pixel par unité de surface, elle s'exprime plus souvent en Points Par Pouce **(PPP, en anglais DPI pour Dots Per Inch)**.La résolution définit la précision et la qualité d'une image. Plus la résolution est grande (c'est-à-dire plus il y a de pixels dans une surface de 1 pouce), plus votre image est précise dans les détails. [8]

Remarque:

1 pouce = 2,54 cm.

1 pouce = 25,40 mm = 100 pixels.

1 inch = 2,54 cm = 1 pouce. [8]

3.4. L'homogénéité :

L'homogénéité est une information locale qui se correspond à l'uniformité d'une région. Une région dans une image est dite homogène si elle doit contenir un ensemble de pixels ayant des caractéristiques similaires ou uniformes, telles que la variance du niveau de gris, la couleur, la texture, etc. [9]

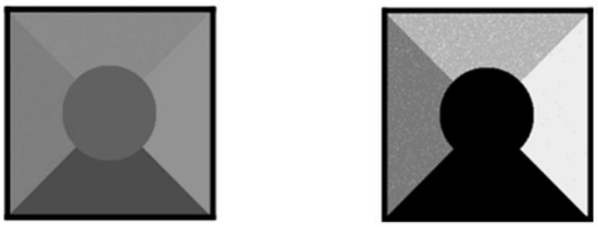
3.5. La luminance :

C'est le degré de luminosité des points de l'image. Elle est définie aussi comme étant le quotient de l'intensité lumineuse d'une surface par l'aire apparente de cette surface, pour un observateur lointain, le mot luminance est substitué au mot brillance, qui correspond à l'éclat d'un objet. [8]

3.6. Le contraste :

C'est l'opposition marqué entre deux régions d'une image, plus précisément entre les régions sombres et les régions claires de cette image. Le contraste est défini en fonction des luminances de deux zones d'image. Si L1 et L2 sont les degrés de luminosité respectivement de deux zones voisines A1 et A2 d'une image, le contraste C est défini par le rapport : [10]

$$C = \frac{L1-L2}{L1+L2}$$ **Équation I-02**



(a)Image mal contrastée (b) Image bien contrastée

Figure I-07: Image avec deux contrastes différents. [11]

3.7. L'histogramme :

L'histogramme des niveaux de gris ou des couleurs d'une image est une fonction qui donne la fréquence d'apparition de chaque niveau de gris (couleur) dans l'image. Pour diminuer l'erreur de quantification, pour comparer deux images obtenues sous des éclairages différents, ou encore pour mesurer certaines propriétés sur une image.

Il permet de donner un grand nombre d'informations sur la distribution des niveaux de gris (couleur) et de voir entre quelles bornes est repartie la majorité des niveaux de gris (couleur) dans les cas d'une image trop claire ou d'une image trop foncée. [2]

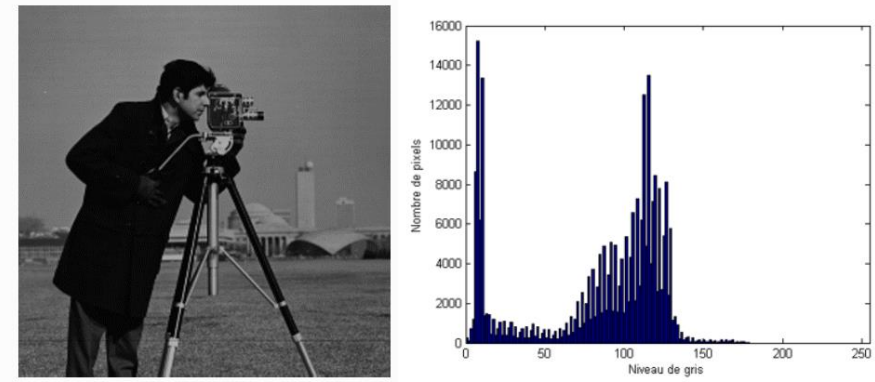
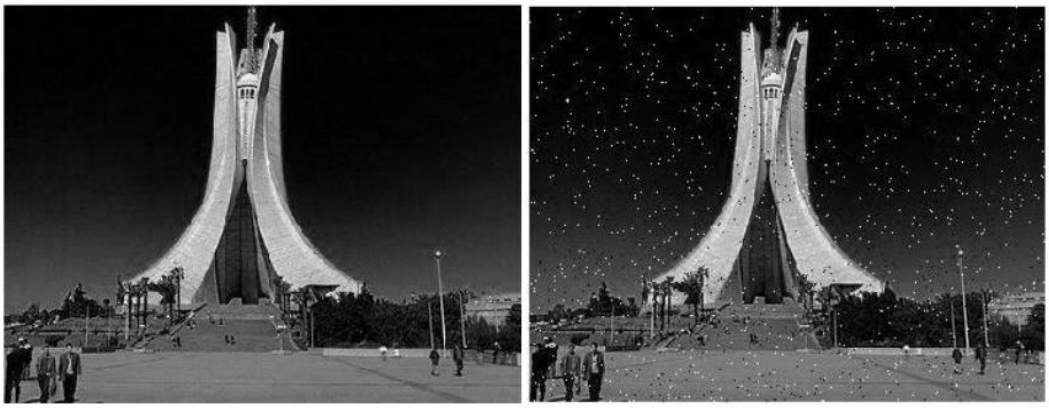


Figure I-08: Histogramme d'image. [12]

3.8. Le bruit :

Un bruit (parasite) dans une image est considéré comme un phénomène de brusque variation de l'intensité d'un pixel par rapport à ses voisins, il provient de l'éclairage des dispositifs optiques et électroniques du capteur. [10]



(a) Image sans bruit (b) Image avec bruit

Figure I-09: Application du bruit sur une image. [2]

3.9. La région :

Une région est un ensemble de pixels connexes et homogènes. Un pixel n'appartient à une région donnée que s'il vérifie les caractéristiques de celle-ci (intensité moyenne, centre de gravité, …). Une région est toujours limitée par un contour. [13]

3.10. Le contour :

Les contours représentent la frontière entre les objets de l'image, ou la limite entre deux pixels dont les niveaux de gris représentent une différence significative. [14]

Figure I-10: Contour d'une image. [2]

4. Introduction à la segmentation d'image :

4.1. Définition de la segmentation d'image :

En vision par ordinateur, la segmentation d'image est le processus de partitionner une image numérique en plusieurs segments ou régions homogènes (ensembles de pixels, également appelés super-pixels). Le but de la segmentation est de simplifier et / ou changer la représentation d'une image en quelque chose de plus significatif et plus facile à analyser. La segmentation est généralement utilisée pour localiser des objets et des limites dans les images. Plus précisément, la segmentation d'image est le processus d'attribution d'une étiquette à chaque pixel d'une image telle que les pixels avec la même étiquette partagent certaines caractéristiques. Segmenter une image signifie donc la diviser en « régions homogènes », selon un ou plusieurs attributs donnés (niveau de gris, texture, couleur,…… etc.). [15]

4.2. Les types de la segmentation d'image :

Il existe plusieurs types de segmentation d'image parmi ces types on trouve :

4.2.1. Segmentation sémantique :

La segmentation sémantique consiste à étiqueter chaque pixel d'une image avec une classe correspondante de ce qui est représenté, Tel que le ciel, l'herbe, la personne ou la voiture……etc. De cette façon, l'image est divisée en régions qui correspondent à des concepts sémantiques. Le résultat de la segmentation sémantique est une image généralement de la même taille que l'image d'entrée dans laquelle chaque pixel est classé dans une classe particulière. Il s'agit donc d'une classification d'image au niveau du pixel. [16]

Figure I-11: Segmentation sémantique d'une image. [17]

4.2.2. Segmentation d'instance :

La segmentation d'instance est une technique qui permet non seulement d'attribuer une étiquette de classe à chaque pixel, mais aussi de distinguer les différentes instances d'une même classe. Par exemple, s'il y a plusieurs personnes dans l'image, la segmentation d'instance identifiera chaque personne en tant qu'objet distinct, plutôt que de les regrouper en une seule région. La segmentation d'instance peut être utile pour compter ou suivre des objets, supprimer ou remplacer des objets, ou créer des masques ou des silhouettes d'objets.

La segmentation des instances est également basée sur des réseaux neuronaux profonds, mais avec des couches ou des modules supplémentaires qui effectuent la détection et la localisation des objets. [18]

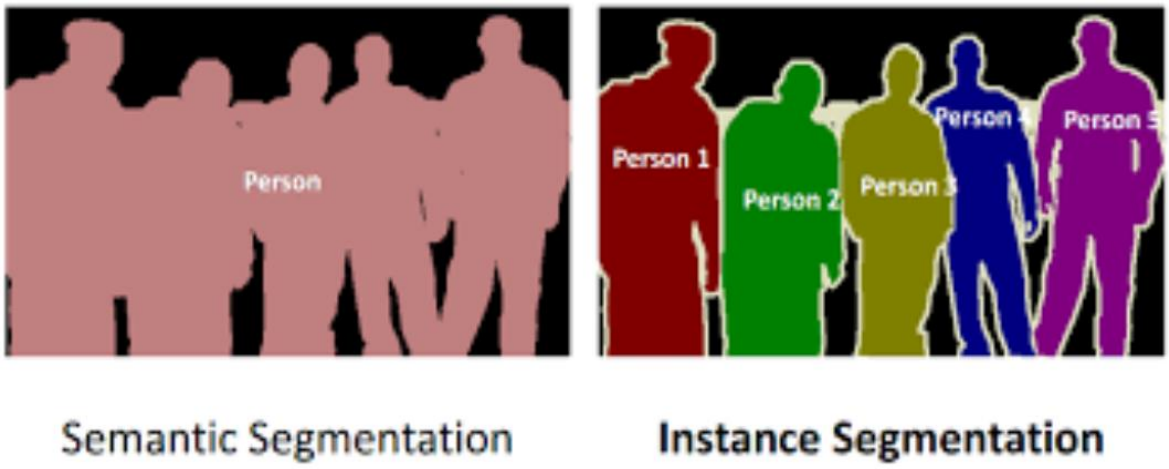


Figure I-12: Segmentation des instances d'une image. [19]

4.2.3. Segmentation panoptique :

La segmentation panoptique est un concept clé dans l'IA et l'apprentissage automatique. Elles déterminent à la fois la classification sémantique de chaque pixel d'une image *et* différencient chaque instance d'objet au sein d'une même image, combinant les avantages de la segmentation sémantique et d'instance.

Dans une tâche de segmentation panoptique, chaque pixel doit se voir attribuer à la fois une étiquette sémantique et un « ID d'instance ». Les pixels partageant le même libellé et le même ID appartiennent au même objet ; pour les pixels déterminés comme ensembles, l'ID d'instance est ignoré. [20]

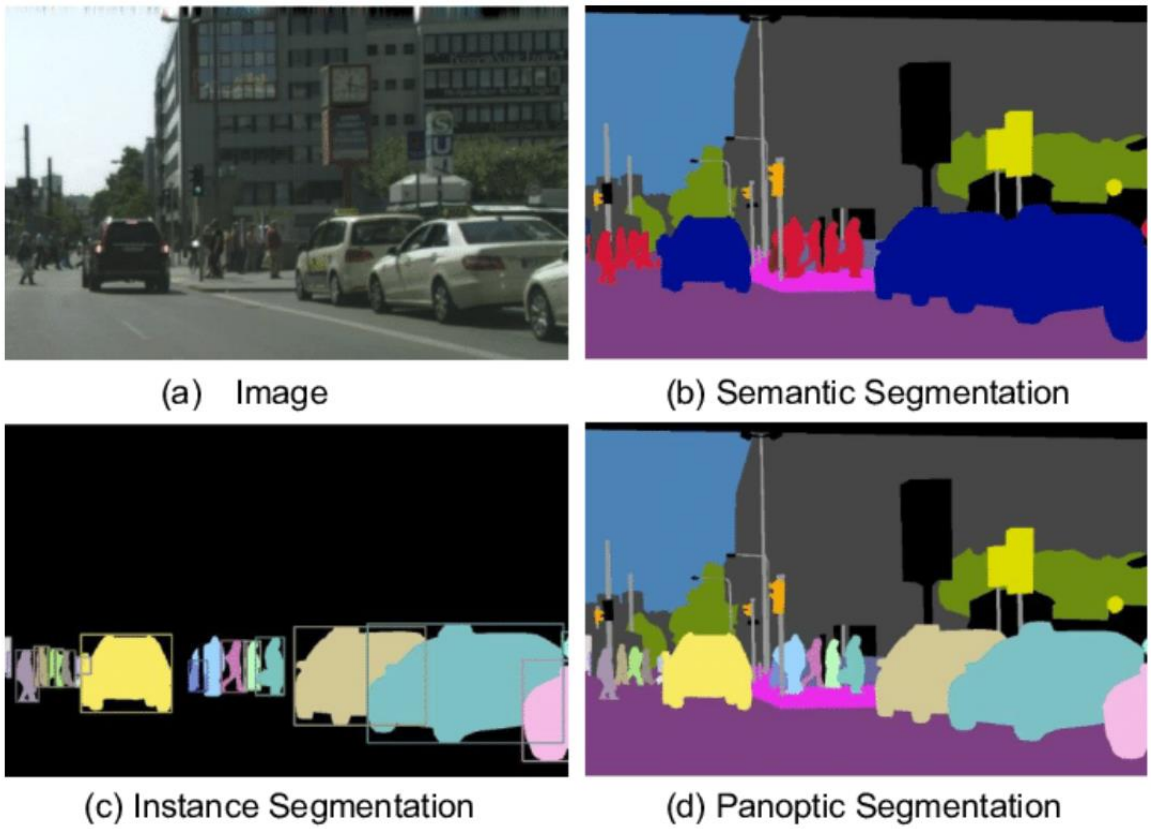


Figure I-13: Les différents types de segmentation de l'image. [21]

4.2.4. Segmentation des instances VS Segmentation sémantique :

Le tableau I-01 ci-dessous présente une comparaison entre la segmentation sémantique et la segmentation d'instance :

Tableau I-01: Une comparaison entre la segmentation sémantique et d'instance. [22]

Caractéristique	Segmentation sémantique	Segmentation d'instance
Classification des pixels	Étiquette chaque pixel avec une étiquette sémantique et une catégorie	Étiquette chaque pixel avec un marqueur spécifique à l'instance
Granularité	Classe d'objet	Instance individuelle d'un objet
Différenciation des objets	Ne différencie pas les objets du même type(les instances)	Distingue les objets du même type(les instances)
Utilisation	Compréhension globale de la scène où l'identité spécifique des objets n'est pas nécessaire	Délimitation, Détection et distinction des objets individuels
Exemples d'Applications	Segmenter le ciel, la route, les bâtiments dans une image de ville	Par exemple Compter le nombre de voitures dans une image, identifier chaque voiture individuellement
Complexité	Moins complexe car elle ne nécessite pas d'identifier des entités uniques (chaque pixel appartient à une classe)	Plus complexe en raison du processus de séparation au niveau de l'instance (chaque pixel doit être assigné à une instance unique)
Niveau de Détail Requis	Moins détaillé	Plus détaillé

5. Les approches et les méthodes de la segmentation:

Généralement, les méthodes de segmentation sont regroupées en trois approches chacune ayant des avantages et ses domaines d'application et elles sont parfois complémentaires, ces approches sont :

1. Segmentation basée sur les contours (en anglais : edge-based segmentation).
2. Segmentation basée sur les régions (en anglais : regions-based segmentation).
3. Segmentation en utilisant la classification. [23]

5.1. Segmentation basée sur les contours (edge-based segmentation):

Le but de la détection de contours est de repérer les points d'une image numérique qui correspondent à un changement brutal de l'intensité lumineuse. La détection des contours d'une image réduit de manière significative la quantité de données et élimine les informations qu'on peut juger moins pertinentes, tout en préservant les propriétés structurelles importantes de l'image [24].

Un élément de contour est un point de l'image appartenant à la frontière de deux ou plusieurs objets ayant des niveaux de gris différents. La figure suivante montre quelques modèles de contours :

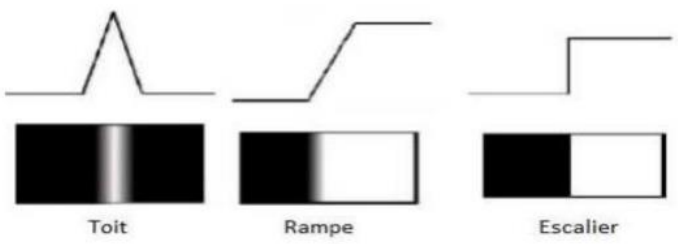


Figure I-14: Quelques modèles de contours. [23]

Un contour peut être défini comme un(e):

- Marche d'escalier : le contour est net (contour idéal).
- Rampe : le contour est plus flou.
- Toit : il s'agit d'une ligne sur un fond uniforme. [23]

Plusieurs méthodes ont été adaptées pour la détection des contours, on distingue principalement **:**

5.1.1. Méthodes dérivatives :

Les méthodes dérivatives sont les plus utilisées pour détecter des transitions d'intensité par différenciation numérique (Première et deuxième dérivé). A chaque position, un opérateur est appliqué afin de détecter les transitions significatives au niveau de l'attribut de discontinuité

choisi. Le résultat est une image binaire constituée de points de contours et de points non-contours. [10]

Les méthodes dérivatives sont très faciles à l'implémentation ainsi que leur temps de calcul relativement court, et leur résultat satisfaisant pour des images non bruitées. Leur inconvénient est qu'elles sont très sensibles au bruit. [4]

De nombreuses techniques d'extraction de contours existent dans la littérature. Elles peuvent être classées comme suit [25]:

• Les algorithmes basés sur le gradient (ou opérateurs du premier ordre) comme les opérateurs de Robert, Prewitt et de Sobel.

Image originale Roberts Perwitt Sobel

Figure I-15: Détection de contour par les différents filtres. [23]

• Les algorithmes basés sur Laplacien (ou opérateurs du second ordre).

Figure I-16: Image originale. [23]

Figure I-17: Contour détecté par le Laplacien.[23]

5.1.2. Méthodes déformables :

Les modèles déformables, introduits par Kass sont aussi connus sous les noms de «snakes » ou « contours actifs ».

L'intérêt principal des contours actifs est de détecter des objets dans une image en utilisant les techniques d'évolution de courbes. L'idée est de partir d'une courbe initiale, généralement un carré ou un cercle, et de la déformer jusqu'à obtenir le contour de l'objet. En effet, celui-ci présente quelques inconvénients tels que la sensibilité à l'initialisation, au bruit, et le réglage difficile de ses différents paramètres. [26]

5.1.3. Méthodes analytiques :

- **Filtre de canny :** Le détecteur de contour de Canny est le plus utilisé. Il est basé sur trois critères : la détection (robustesse au bruit), la localisation (précision de la localisation du point contour), l'unicité (une seule réponse par contour) [27]

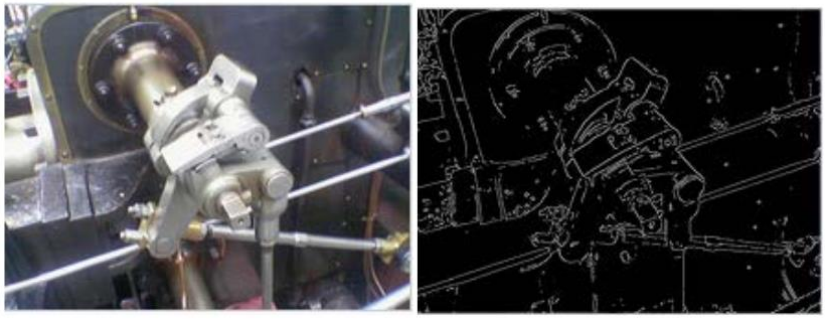


Figure I-18: Exemple d'application du filtre de Canny. [23]

- **Filtre de deriche :** Au filtre de Canny, Deriche a proposé un autre filtre (condition initiale différente) qui permet une simplification de son implémentation .Nous préférons souvent le détecteur de Deriche, qui répond exactement aux mêmes critères de qualité que celui de Canny. [28]

5.2. Segmentation basée sur les régions (Regions-based segmentation) :

La segmentation d'image par l'approche région consiste à découper l'image en régions. Les pixels adjacents sont regroupés en régions distinctes selon un critère d'homogénéité ou de similarité donnée. Ce critère peut être, par exemple, le niveau de gris, couleur, texture…etc.

Un processus de groupement est répété jusqu'à ce que tous les pixels dans l'image soient inclus dans des régions. Cette approche vise, donc, à segmenter l'image en se basant sur des propriétés intrinsèques des régions. [10]

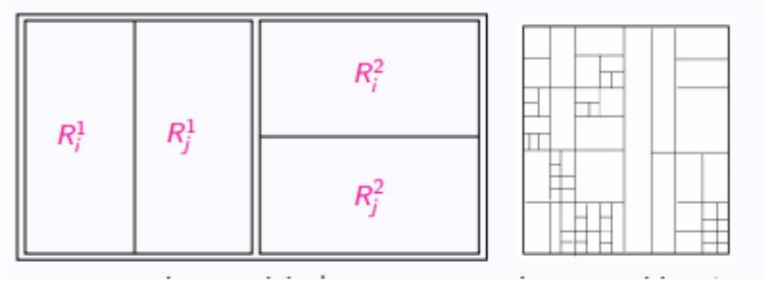


Figure I-19: Segmentation basé sur les régions. [29]

Il existe plusieurs méthodes pour effectuer la segmentation d'image par l'approche région. Parmi ces techniques on a :

5.2.1. Croissance de région (Region growing) :

Cette technique consiste à faire progressivement accroître les régions autour de leur point de départ. L'initialisation de cette méthode consiste à considérer chaque pixel comme une région. On va essayer de les regrouper entre elles avec un double critère de similarité des niveaux de gris et d'adjacence. Le critère de similarité peut par exemple être : la variance des niveaux de gris de la région R est inférieure à un seuil.

Le principe de l'agrégation de pixel est le suivant : on choisit un germe (Le point de départ est le choix d'un ensemble de pixels appelés « germes ») et on fait croitre ce germe tant que des pixels de son voisinage vérifient le test d'homogénéité. Lorsqu'il n'y a plus de pixels candidats dans le voisinage, on choisit un nouveau germe et on itère le processus. [7]

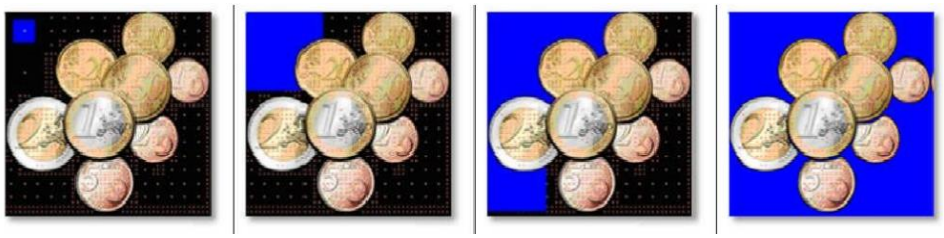


Figure I-20: Croissance progressive des régions. [23]

Parmi les avantages de cette technique, nous pouvons citer :

- La simplicité et la rapidité de la méthode.
- La segmentation d'objet à topologie complexe.
- La préservation de la forme de chaque région de l'image.

Cependant, il existe plusieurs inconvénients comme :

- Une mauvaise sélection des germes ou un choix du critère de similarité mal adapté peuvent entraîner des phénomènes de sous-segmentation1ou de sur-segmentation.
- Il peut y avoir des pixels qui ne peuvent pas être classés. [7]

5.2.2. Segmentation par fusion des régions (Merge) :

Les techniques de réunion (region merging) sont des méthodes ascendantes où tous les pixels sont visités. Pour chaque voisinage de pixel, un prédicat P est testé. S'il est vérifié les pixels correspondants sont regroupés dans une région.

Les inconvénients de cette méthode se situent à deux niveaux :

- Cette méthode dépend du critère de fusion qui peut influer sur le résultat final de la segmentation.
- Elle peut introduire l'effet de sous-segmentation. [7]

5.2.3. Segmentation par division des régions (Split) :

La division consiste à partitionner l'image en régions homogènes selon un critère donné. Le principe de cette technique est de considérer l'image elle-même comme région initiale, qui par la suite est divisée en régions. Le processus de division est réitéré sur chaque nouvelle région (issue de la division) jusqu'à l'obtention de classes homogènes. [25]

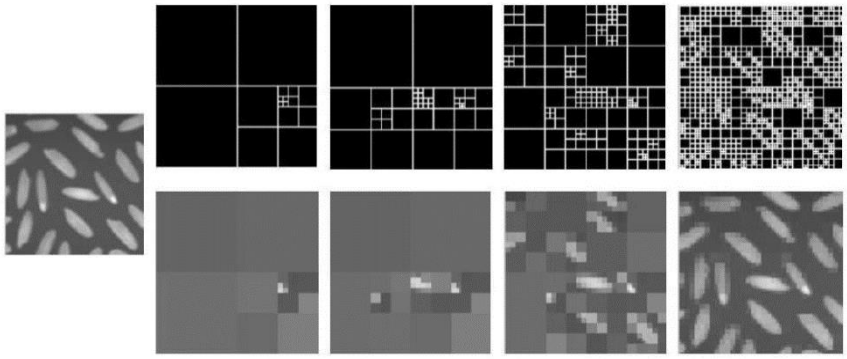


Figure I-21: Décompositions successives des blocs. [23]

Cette méthode présente un inconvénient majeur qui est la sur-segmentation. Toutefois, ce problème peut être résolu en utilisant la méthode de division-fusion que nous présentons dans ce qui suit.

5.2.4. Segmentation par division-fusion (Split and Merge) :

Ces méthodes combinent les deux méthodes décrites précédemment, la division de l'image en de petites régions homogènes, puis la fusion des régions connexes et similaires au sens d'un prédicat de regroupement. On part du principe que chaque pixel représente à lui seul une région. Deux régions seront fusionnées si elles répondent aux critères de similarité des niveaux de gris et d'adjacence de régions .On s'arrête quand le critère de fusion n'est plus vérifié. [10]

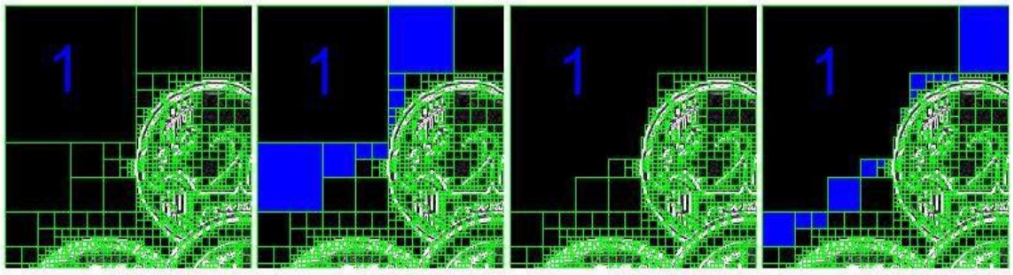


Figure I-22: Agrégation itérative des blocs similaires au bloc numéro 1 de l'image. [23]

Les inconvénients de cette méthode se situent à trois niveaux :

- Les régions obtenues ne correspondent pas, dans tous les cas, aux objets représentés dans l'image.
- Les limites des régions obtenues sont habituellement imprécises et ne coïncident pas exactement aux limites des objets de l'image.
- La difficulté d'identifier les critères pour agréger les pixels ou pour fusionner et diviser les régions. [10]

5.3. Segmentation basée sur la classification des pixels :

La Classification est un processus qui permet de rassembler les pixels d'une image dans des sous-ensembles qui présentent une similitude et une uniformité selon un critère prédéfini, on parle de partitionnement ou de clustering (classe). Cette approche s'appuie sur les concepts de la logique floue [30].

Les méthodes de classification sont issues des méthodes statistiques multidimensionnelles. Il n'existe pas une méthode de classification qui peut s'appliquer à tout type d'image et qui peut fournir un partitionnement optimal. Ce qui explique la grande diversité de méthodes de classification qui existe dans la littérature. Le choix d'une méthode est déterminé par différents facteurs tels que le nombre de classes attendues, la forme des classes extraites ou encore le chevauchement ou non des classes [31].

5.3.1. Classification supervisée des pixels :

La classification de pixels supervisée appelée aussi classification de pixels avec apprentissage consiste à définir une fonction de discrimination effectuant un découpage de l'espace de représentation à partir d'une connaissance a priori de l'image. Ce type de classification nécessite la création d'une base d'apprentissage faisant intervenir une segmentation de référence. [32]

Parmi les algorithmes de classification de pixels supervisée on a :

1. Algorithme des k-plus proches voisins. [33]
2. Algorithme de Bayes. [34]
3. Algorithme des Machines à support de vecteurs. [35]
4. Algorithme des Réseaux de Neurones Multi Couches. [36]

5.3.2. Classification non supervisée de pixels :

La classification de pixels non supervisée appelée aussi classification de pixels sans apprentissage consiste à découper l'espace de représentation en zones homogènes selon un critère de vraisemblance entre les individus. Cette approche est utilisée pour effectuer une classification de pixels en aveugle c'est-à-dire sans connaissance a priori sur l'image et ne nécessite donc pas de phase d'apprentissage. [32]

Parmi les algorithmes de classification de pixels non supervisée on a :

1. L'algorithme des k-moyennes. [37]
2. L'algorithme des C-moyennes floues. [38]
3. L'algorithme de Fisher. [39]

6. Conclusion :

En conclusion, la segmentation d'image est une étape fondamentale dans l'analyse et le traitement des images, jouant un rôle déterminant dans diverses applications de la vision par ordinateur. Les différentes méthodes de segmentation, qu'elles soient basées sur les contours, les régions ou la classification des pixels, offrent des outils puissants pour extraire des informations significatives des images. En comprenant les caractéristiques et les types d'images, ainsi que les diverses approches de segmentation, nous pouvons mieux appréhender les défis et les opportunités dans ce domaine dynamique. La maîtrise de ces techniques ouvre la voie à des avancées significatives dans le développement d'applications intelligentes capables de percevoir et d'interagir avec le monde visuel de manière plus humaine et intuitive.

Le chapitre suivant se concentrera sur les techniques de l'apprentissage profond utilisées dans la segmentation sémantique, une approche qui a démontré des résultats impressionnants dans le domaine de la vision par ordinateur. La maîtrise de ces méthodes avancées permettra de développer des modèles plus précis et robustes pour des applications variées.

Chapitre II:
Deep Learning

1. Introduction :

Le deep learning, une branche de l'intelligence artificielle inspirée du fonctionnement du cerveau humain, a transformé de nombreux domaines grâce à ses capacités avancées de modélisation et de prédiction. Il repose sur des réseaux de neurones artificiels organisés en plusieurs couches, capables d'apprendre des représentations complexes à partir de grandes quantités de données. Cette introduction pose les bases pour explorer en profondeur les concepts, les applications et les développements historiques du deep learning.

Ce chapitre commence par une définition claire du deep learning et examine ses applications dans des domaines tels que la vision par ordinateur, le traitement du langage naturel et la reconnaissance vocale, tout en offrant un aperçu historique depuis ses débuts jusqu'aux développements récents. Il explore ensuite les fondements des réseaux de neurones artificiels, incluant les types comme les réseaux de neurones convolutifs (CNN), et présente leurs architectures utilisées pour la segmentation sémantique, comme U-Net, FCN, VGG et SegNet, il explore les défis du sur-apprentissage et du sous-apprentissage, avec des techniques pour les surmonter. Enfin, une analyse approfondie de l'état de l'art sur la segmentation sémantique par le deep learning mettra en avant les avancées les plus récentes et les approches les plus performantes.

Cette structure offre une exploration approfondie des principes fondamentaux du deep learning ainsi que de ses applications avancées, notamment dans le domaine de la segmentation sémantique.

2. Définition du Deep learning :

Le deep learning, également connu sous le nom d'apprentissage profond, est une branche du machine learning qui vise à développer des systèmes autonomes capables d'apprendre, de prédire et de prendre des décisions. Cette forme d'intelligence artificielle utilise des algorithmes qui imitent le fonctionnement du cerveau humain en utilisant un réseau complexe de neurones artificiels avec plusieurs couches. [40]

3. Les domaines d'application du deep learning :

On trouve le deep learning dans des applications dans divers domaines, notamment

- **Vision par ordinateur :** Reconnaissance d'images, détection d'objets, segmentation sémantique.

- **Le traitement du langage naturel :** par exemple la traduction automatique, l'apprentissage profond permet de reconnaître grâce à la reconnaissance de forme, la langue d'un texte et de le traduire. [41]
- **Médecine :** par exemple la détection de maladies à partir des images médicales.
- Reconnaissance vocale.

4. L'histoire du Deep Learning :

En 1943 apparait le premier modèle de "neurone formel" proposé par Warren McCulloch et Walter Pitts. Il s'agit d'un neurone binaire, la sortie étant équivalente à 0 ou 1. C'est la première représentation informatique de cerveau humain. Puis c'est avec le "perceptron" en 1957 que l'on commence à parler de "réseau de neurones artificiels". [42]

Le perceptron est considéré comme le premier réseau de neurones artificiels, inventé par F. Rosenblatt.

En 1982, Le physicien américain John Hopfield crée le modèle de Hopfield, qui propose un des premiers modèles de réseaux de neurones artificiels récurrents. [43]

En 1986, David Rumelhart et Yann LeCun introduisent le perceptron à couches multiples (MLP, Multilayers Perceptron). Ils présentent une nouvelle approche permettant aux réseaux de neurones de résoudre des problèmes non linéaires avec l'introduction de la rétropropagation du gradient à travers les couches multiples du réseau. [43]

En 1989, Les scientifiques parviennent à créer des algorithmes utilisant des réseaux neuronaux profonds.

À la fin des années 1990, Notamment grâce au travail pionnier de Yann LeCun. LeCun a développé l'architecture LeNet-5.

Dans les années 2000, le deep learning a connu un renouveau grâce à des progrès dans les algorithmes et la puissance de calcul des ordinateurs. De nouvelles architectures de réseaux neuronaux profonds, comme les CNN et les RNN, ont émergé. [40]

Dans les années 2006 jusqu'à 2009, Introduction de deep belief Network et deep Boltzmam machines par Geoffrey Hinton et Salakhutdinov and Hinton respectivement. [44]

En 2012, Introduction de AlexNet qui remporta le challenge ImageNet. [40]

5. Réseau de neurone :

Le concept des réseaux de neurones artificiels, ou Artificiel Neural Networks (ANN), trouve son inspiration dans le fonctionnement des neurones biologiques. À l'instar des neurones dans le cerveau, ces réseaux comportent plusieurs neurones qui travaillent en collaboration. Chaque neurone reçoit des signaux d'entrée, traite ces informations, puis émet un signal de sortie, contribuant ainsi à un traitement global de l'information. [44]

Les réseaux de neurones effectuent des opérations sophistiquées et solutionnent des problèmes complexes tels que la reconnaissance des formes ou le traitement automatique de la langue grâce à l'ajustement de paramètres dans une phase d'entraînement sur les données. [31] [45]

5.1. Neurone Biologique :

Un neurone biologique est une cellule nerveuse fondamentale du système nerveux, qui constitue l'unité de base du traitement de l'information dans le cerveau humain et les organismes vivants. Les neurones reçoivent des signaux (impulsions électriques) par les dendrites et envoient l'information par les axones. Les contacts entre deux neurones (entre axone et dendrite) se font par l'intermédiaire des synapses.

De manière simplifiée, le neurone peut être divisé en trois parties principales :

i. Le corps cellulaire (cell body).
ii. Un groupe de dendrites (environ 7 000 en moyenne).
iii. Un axone. [46]

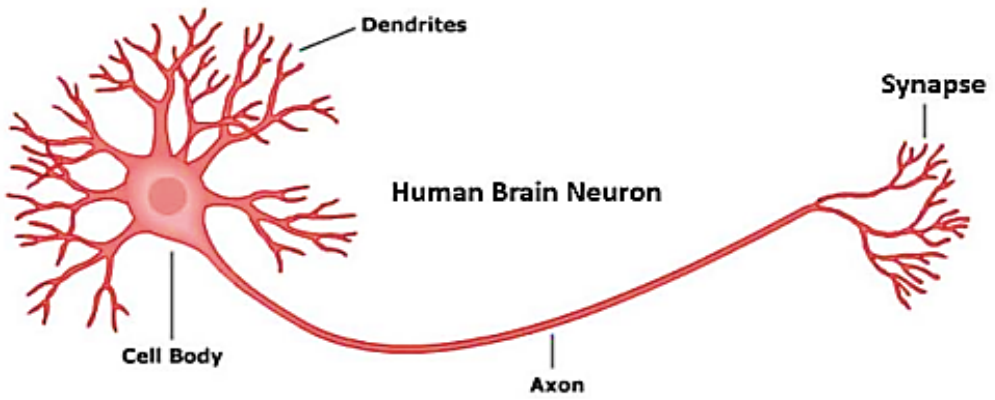


Figure II-01: Neurone biologique. [47]

5.2. Neurone artificielle :

Un neurone artificiel est un type d'apprentissage automatique peut être vu comme une unité de traitement dans un réseau de neurones artificiels. Il reçoit des entrées, les traite à l'aide des fonctions mathématiques, prend une décision et génère une sortie. Cette sortie est ensuite transmise aux neurones suivants dans le réseau.

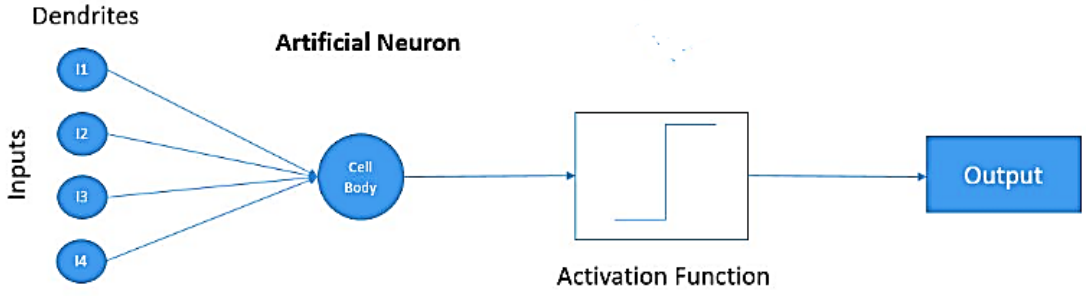


Figure II-02: Neurone Artificiel. [48]

Les neurones biologiques et les neurones artificiels sont des concepts similaires dans le sens où ils sont tous les deux des unités de traitement de l'information. [44]

Neurone biologique	Neurone artificiel
Axones	Signal de sortie
Dendrites	Signal d'entrée
Synapses	Poids de la connexion

Figure II-03 : Neurone biologique et neurone artificiel. [49]

On dit que Les neurones artificiels imitent la structure des neurones biologiques en utilisant des algorithmes de calcul pour traiter et transmettre l'information et peuvent être formés pour effectuer diverses tâches complexes, telles que la reconnaissance d'images, la traduction automatique…etc. [44]

5.3. Perceptron :

Le premier modèle théorique de neurone artificiel a été proposé en 1943 par des chercheurs au MIT. Il a été inventé en 1957 (Frank Rosenblatt) dans les laboratoires de

l'Aérospatiale de l'université de Cornell en utilisant du matériel analogique pour assurer les connexions entre les neurones.

Le perceptron est un algorithme de machine learning, plus spécifiquement un modèle de neurone artificiel, qui est la base des réseaux de neurones. Il est généralement utilisé pour des tâches de classification binaire simples.

Le perceptron prend plusieurs entrées pondérées, les combinant et passant le résultat à travers une fonction d'activation, le perceptron produit une sortie, cette sortie peut alors être utilisée pour classer les données dans l'une des deux catégories possibles. [46]

Un modèle constitué d'un seul neurone est appelé **Perceptron**.

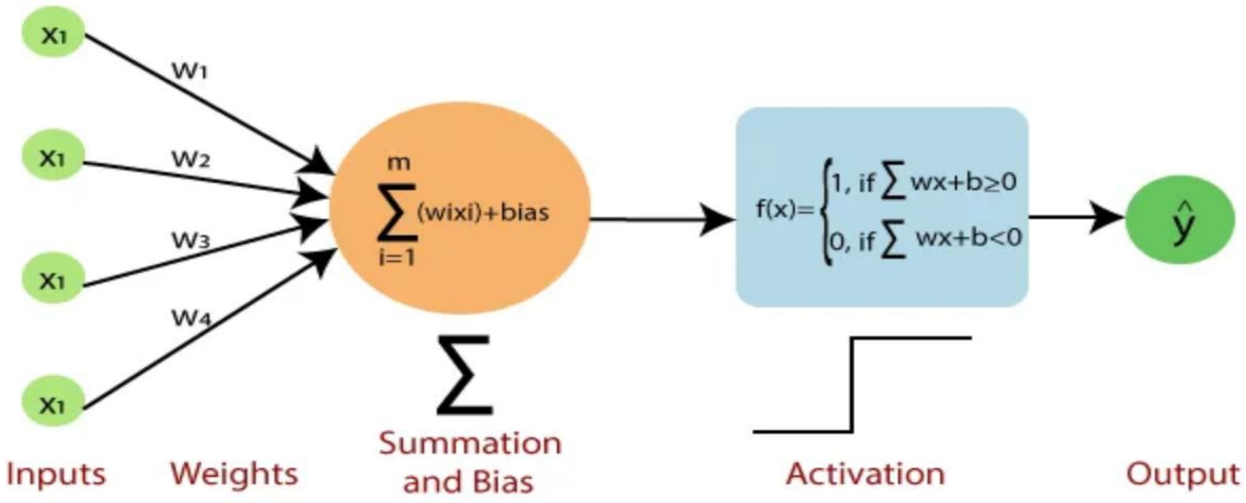


Figure II-04: Un perceptron. [50]

Un perceptron peut être classé comme une unité de calcul simple avec une structure basique. Il est composé des éléments suivants :

- **Entrée et poids.** Chaque perceptron reçoit plusieurs entrées. Ces entrées sont généralement les caractéristiques de la donnée que nous voulons classer. Chaque entrée est associée à un poids, qui indique l'importance relative de cette entrée pour la décision finale.
- **Biais** : Le biais est paramètre supplémentaire ajoutée à la somme pondérée des entrées avant de passer à une fonction d'activation. Il agit comme un ajustement ou une correction pour permettre au modèle de mieux s'ajuster aux données d'entrée.

- **Somme pondérée :** Les entrées pondérées sont sommées pour produire une valeur combinée. Cette somme pondérée est calculée en multipliant chaque entrée par son poids correspondant et en additionnant les résultats en ajoutent un biais.
- **Fonction d'activation** : La somme pondérée des entrées est passée à travers une fonction d'activation. Dans le cas du perceptron, cette fonction est souvent une fonction seuil, qui renvoie 1 si la somme pondérée est supérieure à un certain seuil et 0 sinon.
- **Sortie** : La sortie est le résultat final produit par le perceptron. Cette sortie donne 0 ou 1 dans le cas d'une classification binaire.

Les perceptrons peuvent être combinés pour former des réseaux de neurones plus complexes pour des tâches plus avancées. Et ce qu'on appelle perceptron multicouche. [46]

5.4. Perceptron multicouche (MLP) :

Le perceptron multicouche « ou encore multi-layers perceptron en anglais », est le premier réseau de neurones artificiels composé de plusieurs couches de neurones, où chaque couche est composée de multiples neurones interconnectés, il possède au moins trois couches : une couche d'entrée, au moins une couche cachée, et une couche de sortie. [51]

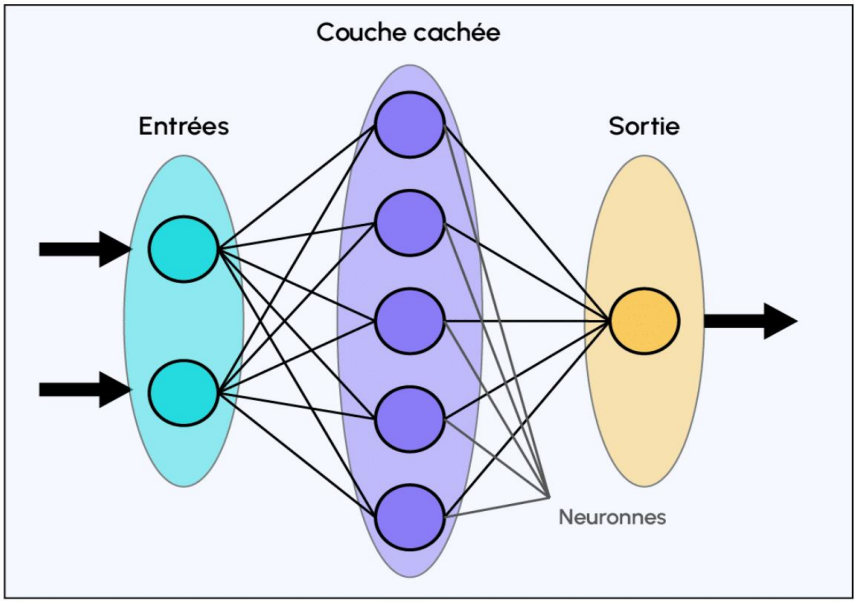


Figure II-05 : Un perceptron multicouche. [52]

La structure typique d'un perceptron multicouche comprend trois types de couches :

- **Couche d'entrée :** La première couche est constituée de neurones "transparents" qui n'effectue aucun calcul mais simplement transmettre leurs entrées à tous les neurones de la couche suivante.
- **Couches cachées :** Il peut y avoir une ou plusieurs couches cachées entre la couche d'entrée et la couche de sortie. Chaque couche cachée est composée de plusieurs neurones, qui reçoivent les entrées de la couche précédente, effectuent des calculs sur ces entrées (somme pondéré, fonction d'activation) et transmettent ensuite leurs sorties à la couche suivante.
- **Couche de sortie :** La sortie finale du réseau de neurones est généralement produite par cette couche, en fonction du type de tâche à exécuter, tel qu'une classification ou une régression. Chaque neurone dans la couche de sortie est responsable de prédire une valeur spécifique ou une classe de sortie.

La sélection de la fonction d'activation dans un perceptron multicouche dépend du type de problème que nous tentons de résoudre.

Généralement, la fonction softmax est couramment utilisée dans la couche de sortie et surtout pour les problèmes de classification multi-classe, tandis que la fonction Relu peut être utilisée dans les couches cachées. [53]

5.5. Les types des réseaux de neurones artificiels :

Il existe plusieurs types de réseaux de neurones en deep learning, chacun ayant ses propres architectures et applications spécifiques. Voici quelques-uns :

5.5.1. Réseaux de neurones récurrents (RNN) :

Un réseau de neurones récurrent (RNN) est un type de réseau de neurones artificiel qui utilise des données séquentielles ou des données de séries temporelles. Ces algorithmes d'apprentissage en profondeur sont couramment utilisés pour des problèmes ordinaux ou temporels, tels que la traduction linguistique, le traitement du langage naturel, la reconnaissance vocale. [54]

5.5.2. Réseaux de neurones autoencodeurs (AE) :

Un auto-encodeur est un type d’architecture de réseau neuronal conçu pour compresser efficacement (encoder) les données d’entrée vers leurs caractéristiques essentielles, puis

reconstruire (décoder) l'entrée d'origine à partir de cette représentation compressée. Cette approche est non supervisée, ce qui signifie qu'elle n'a pas besoin de labels ou de catégories pour l'apprentissage. [55]

5.5.3. Réseaux de neurones convolutifs (CNN) :

Les réseaux de neurones convolutifs (CNN), également connus sous le nom de Convolutional Neural Networks en anglais, sont des modèles informatiques avancés utilisés principalement pour la reconnaissance d'images. [56]

5.6. Introduction aux réseaux de neurones convolutifs (CNN) :

Les réseaux de neurones convolutifs sont des algorithmes d'intelligence artificielle basée sur les réseaux neuronaux multicouche qui apprennent des caractéristiques pertinentes à partir d'image, étant capable d'effectuer plusieurs tâches telles que la classification, la détection et la segmentation d'objets. [57]

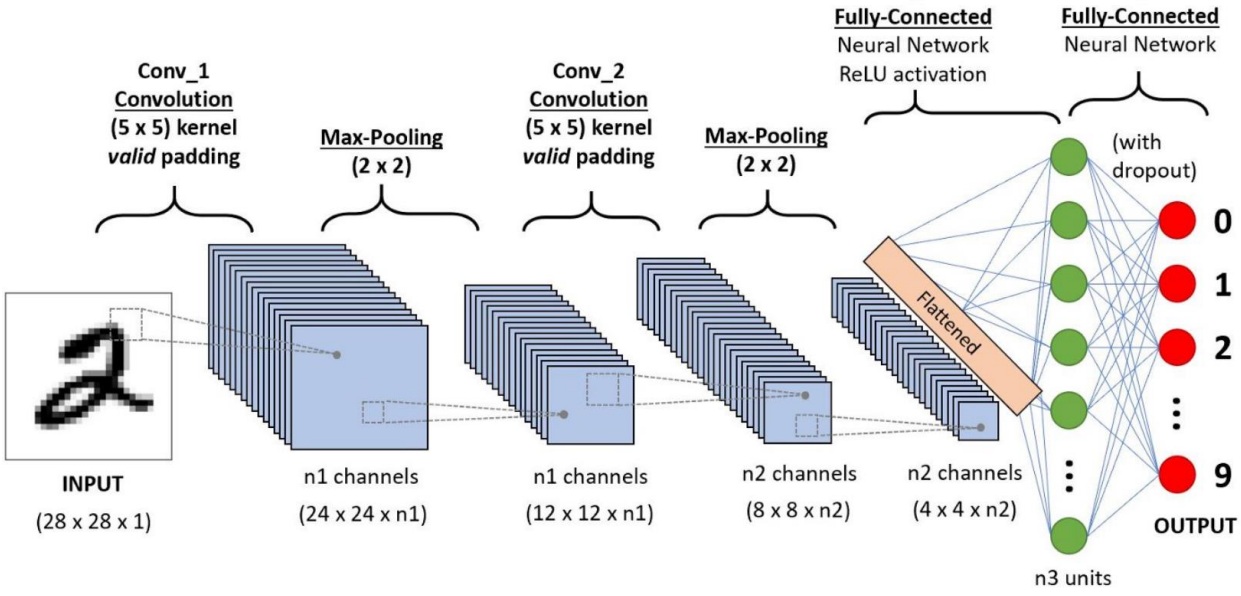


Figure II-06 : Réseau de neurone convolutif. [58]

L'architecture des CNN est inspirée par le fonctionnement du cortex visuel du cerveau humain. La notion de convolution est une opération mathématique qui permet de classer et d'extraire les caractéristiques des données d'entrée.

Les CNN sont composés de plusieurs couches, notamment des couches de convolution, des couches de pooling et des couches entièrement connectées.Voici quelque notion de base de CNN :

- **Couche de convolution :** est l'un des éléments clés d'un réseau de neurones Convolutionnels (CNN). Elle constitue toujours leur première couche. Cette couche permet d'extraire des caractéristiques à partir des données d'entrée en appliquant des filtres de convolution.

  Une couche de convolution est composée de plusieurs filtres de convolution, également appelés noyaux, qui travaillent simultanément sur la même image d'entrée. Chaque filtre est une petite matrice de nombres qui est glissée sur l'image d'entrée pour effectuer une opération de convolution pour produire une carte de caractéristiques en sortie, et ces cartes sont combinées pour former la sortie de la couche de convolution. Tous les filtres dans une couche ont la même largeur et hauteur, et ils partagent des paramètres. Les termes biais et fonction d'activation peuvent être utilisés dans les couches de convolution, comme dans d'autres couches de réseaux neuronaux.

  L'utilisation de plusieurs filtres dans chaque couche de convolution, le réseau de neurones peut apprendre à détecter des caractéristiques visuelles complexes, ce qui lui permet de représenter les données de manière plus significative pour des différentes tâches.

  L'opération de convolution implique de placer le filtre sur chaque petite zone de l'image d'entrée, puis de multiplier les valeurs du filtre par les valeurs correspondantes de l'image dans cette zone. Ces produits sont ensuite additionnés pour obtenir une seule valeur, qui représente l'activation du filtre pour cette zone. Cette étape est répétée pour toutes les zones de l'image, générant ainsi la carte d'activation (carte des caractéristiques).

  Après chaque opération de convolution, un CNN applique une transformation Relu (unité de rectification linéaire) sur la carte d'activation, ce qui permet d'introduire une non-linéarité dans le modèle. [54]

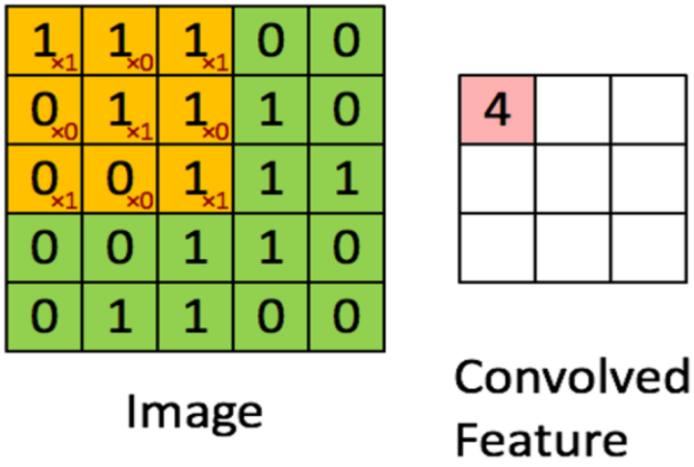


Figure II-07 : La convolution. [59]

- **STRIDE :** indique le nombre de cellules dont il faut déplacer la fenêtre lors de la convolution. [47]
- **PADDING (ou remplissage) :** se réfère à l'ajout de pixels supplémentaires autour de l'image d'entrée avant d'appliquer une opération de convolution. Les pixels ajoutés sont souvent de valeur zéro (padding zéro), mais ils peuvent également prendre d'autres valeurs selon le contexte, afin d'avoir suffisamment de données et s'assurer que tous les pixels sont utilisés dans la convolution, ou si nous voulons que l'image résultante soit de la même taille que notre image d'entrée. [47]
- **Les Hyper paramètres de cette couche :** Le rôle de la couche de convolution (kernels) est de réduire les images en une forme plus facile à traiter sans perdre des caractéristiques qui sont essentielles pour obtenir une bonne prédiction.

  Chaque image en entrée est de dimensions W × H × D où :
  - **W** : largeur en pixel (Width)
  - **H** : hauteur en pixel (height)
  - **D** : le nombre de canaux ou dimension (1 pour image noir/blanc et 3 pour une image couleur RGB).

La couche de convolution possède Quatre hyper paramètres :

- **K :** Le nombre de filtres (kernels)
- **F** : la taille du filtre (kernel dimensions F×F×D pixels).
- **S** : Stride
- **P** : Padding

On peut calculer la dimension de la sortie dans la couche de convolution, Ainsi, pour chaque image Input de taille (W×H×D), la couche de convolution renvoie une matrice de dimensions (Wc x Hc x Dc), où :

- **Wc** = [(W – F + 2P) / S] + 1 **Équation II-01**
- **Hc** = [(H – F + 2P) / S] + 1 **Équation II-02**
- **Dc** = K

Lorsqu'on Choisit $\boldsymbol{P = (F-1) / 2}$ **(Équation II-03)** et $\boldsymbol{S = 1}$ permet ainsi d'obtenir des « Features maps » de même Largeur et Hauteur que celles des Inputs. [47]

- **Couche de pooling :** Ce type de couche est souvent placé entre deux couches de convolution, elle reçoit en entrée plusieurs feature maps (carte de caractéristique), et appliquent à chacune d'entre elles l'opération de pooling.

  Une couche de pooling, agit comme une couche de réduction. Elle divise l'image en blocs et effectue un calcul statistique simple, la technique la plus utilisée est le Max Pooling, est particulièrement utile lorsqu'on travaille avec de grandes images, car elle consiste à prendre la plus grande valeur de chaque bloc et écarter les autres. Cela permet de réduire la dimension de l'image tout en conservant les caractéristiques les plus importantes Ce qui implique moins de calculs et plus de rapidité. On obtient en sortie le même nombre de « Features maps » qu'en entrée, mais celles-ci sont bien plus petites. [47]

  On a aussi Average pooling qui permet de calculer la valeur moyenne dans chaque bloc. [60]

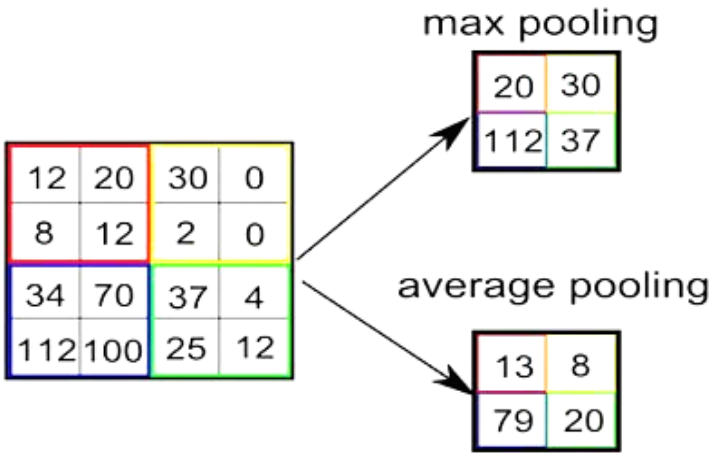


Figure II-08: Max pooling. [61]

- **Couche entièrement connectée (ou Fully Connected) :** est similaire aux couches dans les réseaux de neurones multicouches (MLP).Cette couche s'applique sur une entrée préalablement aplatie (flattened) où chaque entrée est connectée à tous les neurones.

  Elle effectue la classification en fonction des caractéristiques extraites à partir des couches précédentes et de leurs différents filtres. Contrairement aux couches de convolution qui utilise souvent la fonction ReLU, les couches entièrement connectées exploitent généralement une fonction d'activation softmax pour classer les entrées de manière appropriée, produisant une probabilité de 0 à 1. [54] [62]

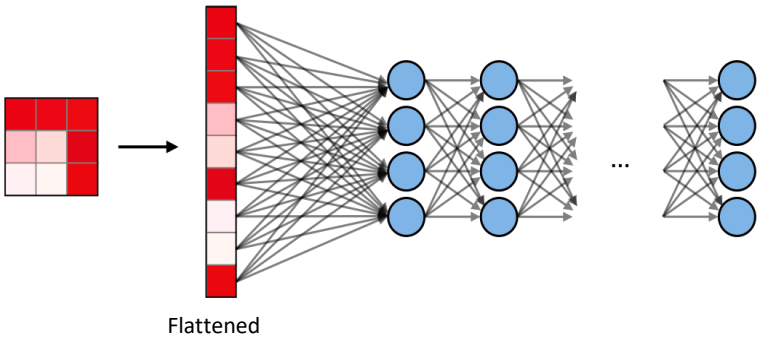


Figure II-09: Fully connected. [63]

- **La couche dropout :** C'est une couche qui éteint certains neurones de manière aléatoire. Ça permet de forcer le réseau à ne pas trop s'appuyer sur un seul neurone mais sur toutes les informations mises à sa disposition. Ça permet aussi de limiter le sur apprentissage. [47]
- **Fonction de coût (fonction de perte ou loss fonction) :** est une fonction qui évalue la différence entre les prédictions faites par le réseau de neurones et les valeurs réelles des données utilisées pendant l'apprentissage. Plus le résultat de cette fonction est faible, plus le réseau de neurones est performant. [47]

  Le choix de la fonction de coût dépend de la nature de la tâche d'apprentissage et du type de modèle utilisé parmi ces fonctions :

  - ✓ ***Erreur quadratique moyenne (Mean Squared Error - MSE) :*** Utilisée pour les tâches de régression, cette fonction mesure la moyenne des carrés des écarts entre les prédictions du modèle et les valeurs réelles.Elle est définie comme suit: [64]

$$\mathbf{MSE} = \frac{1}{n}\sum_i \|\hat{y}_i - y_i\|^2 \qquad \textbf{Équation II-04}$$

$\hat{y}_i$: les prédictions faites par le réseau de neurones
y_i : les valeurs réelles
n : Le nombre total d'échantillons.

- ***Erreur Absolue Moyenne (MAE - Mean Absolute Error) :*** Utilisée principalement pour les problèmes de régression, cette fonction mesure la moyenne absolue des écarts entre les prédictions du modèle et les valeurs réelles. Elle est définie comme suit : [47]

$$\textbf{MAE} = \frac{1}{n}\sum_i |\hat{y}_i - y_i| \qquad \textbf{Équation II-05}$$

- ***Binary cross-entropy*** **:** Détermine la perte lorsque les résultats catégoriques sont binaires, c'est-à-dire deux possibilités : (succès/échec) ou (oui/non). Elle est généralement utilisée pour les modèles de régression logistique [47]
- ***Categorical cross-entropy*** : est essentielle pour les tâches de classification multiclass, elle est couramment utilisée dans les réseaux de neurones avec activation softmax. [65]

$$\textbf{Categorical-cross-entropy} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij}\log(p_{ij}) \qquad \textbf{Équation II-06}$$

N: Le nombre total d'échantillons.
C: Le nombre total de classes.
$\boldsymbol{y_{ij}}$: La véritable étiquette pour l'échantillon i et la classe j.
$\boldsymbol{p_{ij}}$: La probabilité prédite pour l'échantillon i appartenant à la classe j.

- **Epoch :** est un hyper paramètre défini par l'utilisateur qui indique combien de fois l'algorithme d'apprentissage complètera un passage à travers l'ensemble des données d'entraînement. [47]
- **Batches (lots) :** est un hyper paramètre qui détermine le nombre d'exemples d'entraînement à traiter avant que le modèle ne mette à jour ses paramètres internes. [47]
- **Fonction d'activation** : est un composant clé des réseaux de neurones artificiels, qui introduit de la non-linéarité dans le modèle, ce qui permet au réseau de neurones d'apprendre des modèles complexes et de résoudre des problèmes qui ne peuvent pas être séparés de manière linéaire. Il existe plusieurs fonctions d'activation :

- **ReLU :** La fonction Rectified Linear Unit (ReLU) est la fonction d'activation la plus couramment utilisée en Deep Learning. Elle donne x si x est supérieur à 0, 0 sinon. Autrement dit, c'est le maximum entre x et 0 : x si x > 0. [47]

$$\textbf{ReLU(x)=} \begin{cases} x & \text{si } x > 0 \\ 0 & sion \end{cases} \qquad \textbf{Équation II-07}$$

✓ **Sigmoid :** La fonction Sigmoïde est la fonction d'activation utilisée en dernière couche d'un réseau de neurones construit pour effectuer une tâche de classification binaire. Elle donne une valeur entre 0 et 1. [47]

$$\textbf{Sigmoid (x)}= \frac{1}{1+e^{-x}} \qquad \textbf{Équation II-08}$$

✓ **tanh :** La fonction tanh permet d'appliquer une normalisation aux valeurs d'entrée. Elle peut également être utilisée au lieu de la fonction Sigmoïde dans la dernière couche d'un modèle de classification binaire. Elle donne un résultat entre -1 et 1. [47]

$$\textbf{tanh(x)}= \frac{2}{1+e^{-2x}} - 1 \qquad \textbf{Équation II-09}$$

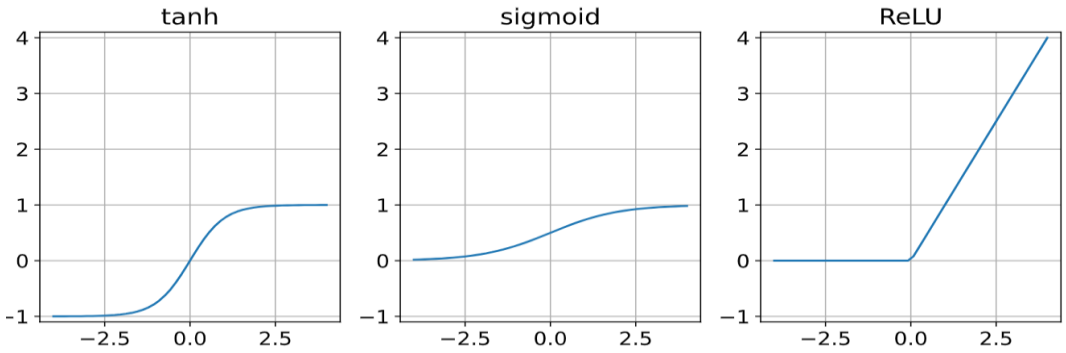


Figure II-10: Les fonctions d'activations. [66]

✓ **Softmax:** la fonction Softmax est la fonction d'activation utilisée en dernière couche d'un réseau de neurones construit pour effectuer une tâche de classification multi-classes.

Pour chaque sortie, la fonction Softmax produit un résultat compris entre 0 et 1. De plus, la somme de toutes ces sorties est égale à 1. [47]

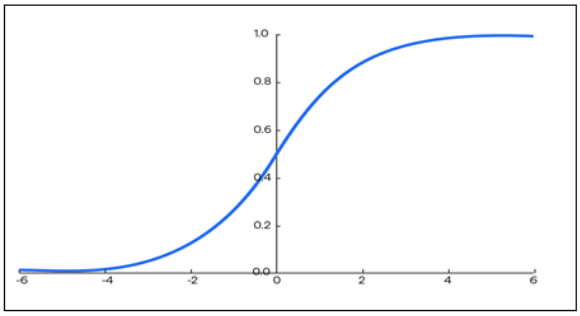


Figure II-11 : La fonction softmax. [67]

- **Les optimiseurs :** sont des algorithmes utilisés pour ajuster les poids des neurones afin de minimiser la fonction de perte. Les optimiseurs sont essentiels pour entraîner les réseaux de neurones de manière efficace et pour assurer la convergence vers un minimum local ou global de la fonction de perte. Parmi les optimiseurs utilisés :
    - ✓ **Stochastic Gradient Descent (SGD) :** utilise un seul exemple de l'ensemble de données pour chaque mise à jour des poids. Cette approche a eu plus de succès et permet un processus d'entraînement plus fluide.
    - ✓ **Adam (Adaptive Moment Estimation) :** L'optimiseur Adam est l'un des algorithmes d'optimisation de descente de gradient les plus populaires et les plus célèbres. C'est une méthode qui calcule les taux d'apprentissage adaptatifs pour chaque paramètre. Efficace pour des problèmes avec des données bruitées ou des gradients rares.
    - ✓ **AdaGard (Adaptive Gradient Algorithm) :** adapte les taux d'apprentissage de manière individuelle pour chaque neurone dans chaque couche cachée du réseau, en les ajustant en fonction des différentes itérations de l'entraînement. [68].
- **Learning rate :** ou le taux d'apprentissage, un hyper paramètre essentiel utilisé pour ajuster les poids d'un modèle pendant l'entraînement. Il contrôle la taille des changements effectués à chaque itération pour minimiser l'erreur de prédiction. [47]

5.7. Le fonctionnement de CNN :

Les réseaux de neurones convolutionnels (CNN) prennent généralement une image en entrée, représentée sous forme d'une matrice de pixels. Par exemple, une image en niveaux de gris pourrait avoir une dimension de (28, 28, 1), où 28 représente la hauteur et la largeur de l'image, et 1 indique le nombre de canaux (pour les images en niveaux de gris, il n'y a qu'un seul canal).

La première couche d'un CNN est souvent une couche convolutive. Cette couche applique une série de filtres à l'image d'entrée, qui sont de petites matrices convoluées avec l'image pour détecter des caractéristiques telles que les bords, les coins et les textures. Après la convolution, une fonction d'activation non linéaire, comme ReLU, est généralement appliquée à la sortie de chaque neurone pour introduire de la non-linéarité dans le modèle.

Ensuite, une opération de pooling est réalisée pour réduire la dimension de l'image tout en préservant les caractéristiques importantes. Le pooling, souvent de type max pooling, consiste à sélectionner la valeur maximale dans une région spécifique de l'image.

Après les couches de convolution et de pooling, les données sont souvent aplaties et introduites dans une couche entièrement connectée. Cette couche effectue des prédictions basées sur les caractéristiques détectées par les couches précédentes.

Enfin, la sortie de la couche entièrement connectée passe par une fonction d'activation finale, telle que la fonction softmax, pour produire les prédictions finales. [54]

5.8. Quelques architectures du CNN :

5.8.1. U-Net :

U-Net est un modèle de réseau de neurones convolutifs, connue pour sa capacité à capturer des informations contextuelles à différentes échelles tout en conservant des détails spatiaux importants. Cette architecture est particulièrement adaptée à la segmentation sémantique des images.

L'architecture d'U-Net se compose de deux chemins principaux : le chemin contractuel (Encodeur) et Chemin d'Expansion (Décodeur), permettant de réduire puis de restaurer la résolution de l'image tout en intégrant les informations contextuelles à chaque étape, ce qui lui confère une architecture en forme de « U ».

• **Le chemin contractuel (encodeur) :** représente la partie gauche de l'architecture U-Net. Son rôle est de capturer les caractéristiques de l'image à différentes échelles en utilisant des couches de :

- ✓ **Convolutions :** Chaque étape du chemin contractant consiste en deux convolutions successives avec des filtres 3x3, suivies d'une fonction d'activation ReLU.
- ✓ **Max-Pooling :** Après les convolutions, une opération de max-pooling 2x2 est appliquée pour réduire la dimension spatiale de l'image tout en augmentant la profondeur des canaux. Cela permet d'extraire des caractéristiques de plus en plus complexes tout en réduisant la résolution spatiale.

Cette séquence de convolutions suivies d'un max-pooling se répète généralement 4 fois.

• **Le chemin (décodeur) :** correspond à la partie droite de l'architecture U-Net. Il est responsable de la localisation précise des caractéristiques grâce :

- ✓ **Transpositions Convolutionnelles (Up-sampling) :** Chaque étape du chemin expansif commence par une opération de transposition convolutionnelle 2x2 pour augmenter la dimension spatiale de l'image.
- ✓ **Convolutions :** Les convolutions 3x3 suivies d'une fonction d'activation ReLU sont appliquées pour affiner les caractéristiques et les détails.

• **Connexions de Pont (Skip Connections, Concatenate) :** Les connexions de pont relient les couches correspondantes du chemin contractant et du chemin expansif. Elles combinent les caractéristiques des deux chemins, permettant de conserver les détails tout en ajoutant un contexte global.

• **Couches de Sortie :** La couche finale est une convolution 1x1 qui réduit le nombre de canaux au nombre de classes de segmentation désiré, généralement suivie d'une fonction d'activation sigmoïde ou softmax selon le problème traité (segmentation binaire ou multi-classes).

Le modèle U-Net présente plusieurs avantages qui en font un choix populaire pour la segmentation d'images. [69] [70]

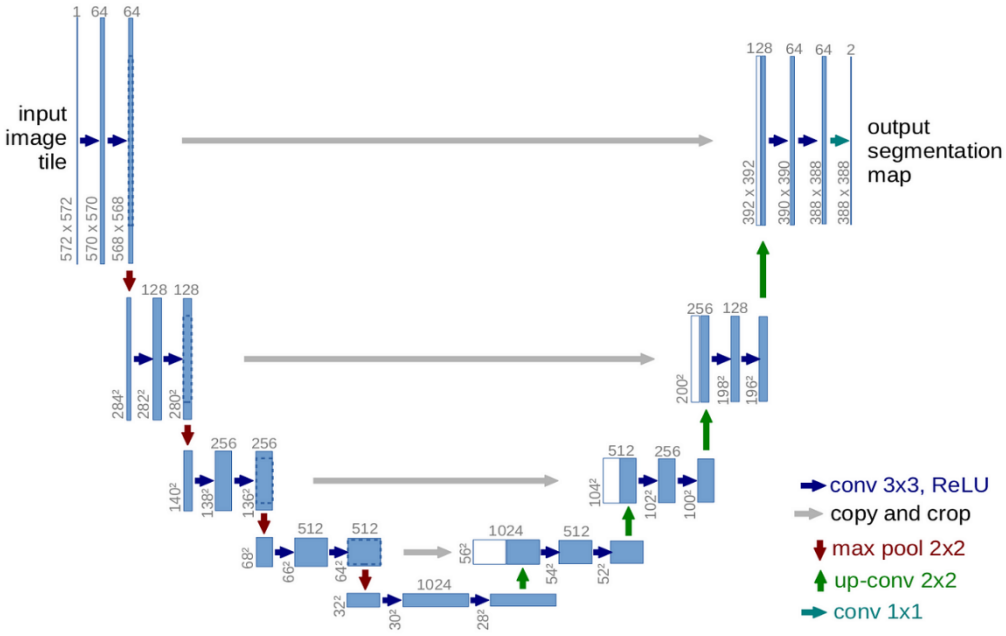


Figure II-12: Architecture U-Net. [71]

5.8.2. Fully Convolutional Networks (FCN) :

Les Fully Convolutional Networks (FCN) sont des réseaux de neurones conçus pour la segmentation sémantique des images. Les FCN utilisent uniquement des couches de convolution, sans couches entièrement connectées, pour produire une carte de segmentation ayant la même dimension spatiale que l'image d'entrée.

L'architecture FCN prend une image en entrée, extrait des caractéristiques via des convolutions, réduit la dimension spatiale via des opérations de max-pooling, et utilise des convolutions transposées pour augmenter la dimension spatiale afin de produire une carte de segmentation avec les mêmes dimensions que l'image d'entrée. La fusion des caractéristiques se fait à chaque étape de la phase de décodage pour combiner les informations de bas et de haut niveau, et la couche de sortie utilise une convolution 1x1 pour prédire les classes pour chaque pixel.[72]

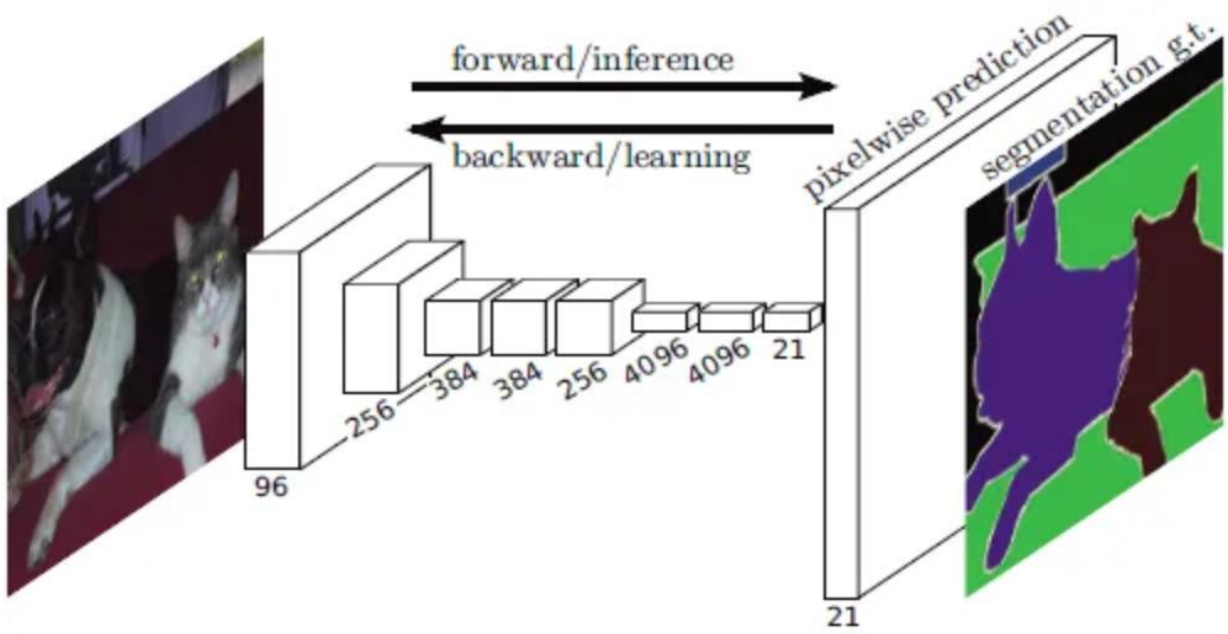


Figure II-13 : Fully Convolutional Networks (FCN). [73]

5.8.3. VGG :

VGG signifie Visual Geometry Group, Il s'agit d'une architecture standard de réseau neuronal convolutif profond (CNN) avec plusieurs couches. Le « profond » fait référence au nombre de couches avec VGG-16 ou VGG-19 composé de 16 et 19 couches convolutives.

L'architecture VGG constitue la base de modèles de reconnaissance d'objets révolutionnaires. Le VGG reste aujourd'hui l'un des architectures de reconnaissance d'images les plus populaires.[74]

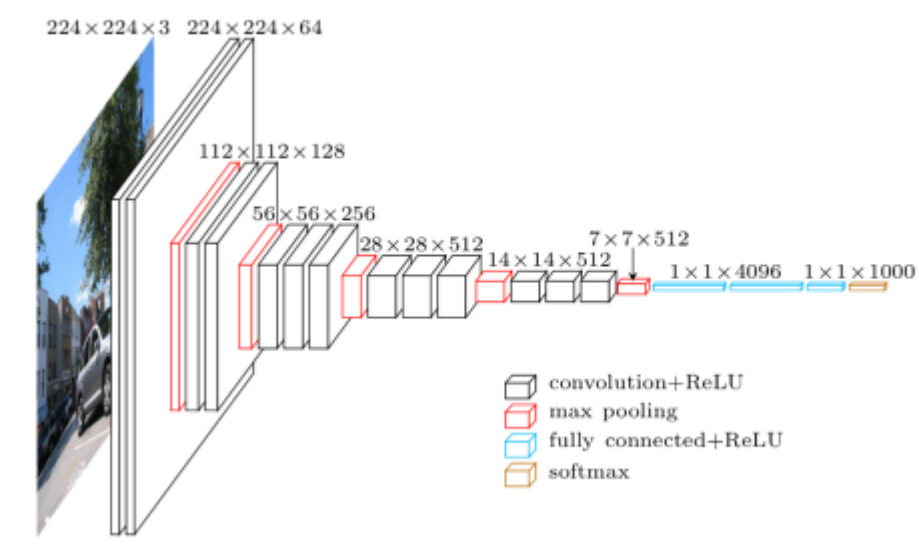


Figure II-14 : Architecture VGG. [75]

5.8.4. SegNet :

SegNet est une architecture de réseau neuronal convolutif (CNN) conçue pour la segmentation sémantique en vision par ordinateur. La segmentation sémantique consiste à classer chaque pixel d'une image dans une catégorie ou une classe spécifique, par exemple en identifiant les objets et leurs limites. SegNet est l'une des nombreuses architectures d'apprentissage en profondeur développées pour résoudre ce problème. Il est basé sur une architecture d'encodeur-décodeur

- **Encodeur :** se compose de couches Convolutionnels et de couches de pooling (sous-échantillonnage) qui extraient les caractéristiques de l'image d'entrée tout en réduisant sa dimension spatiale
- **Décodeur :** utilise des couches d'upsampling (sur-échantillonnage) pour augmenter la dimension spatiale de l'image, et des couches Convolutionnels pour affiner les détails de segmentation. [76]

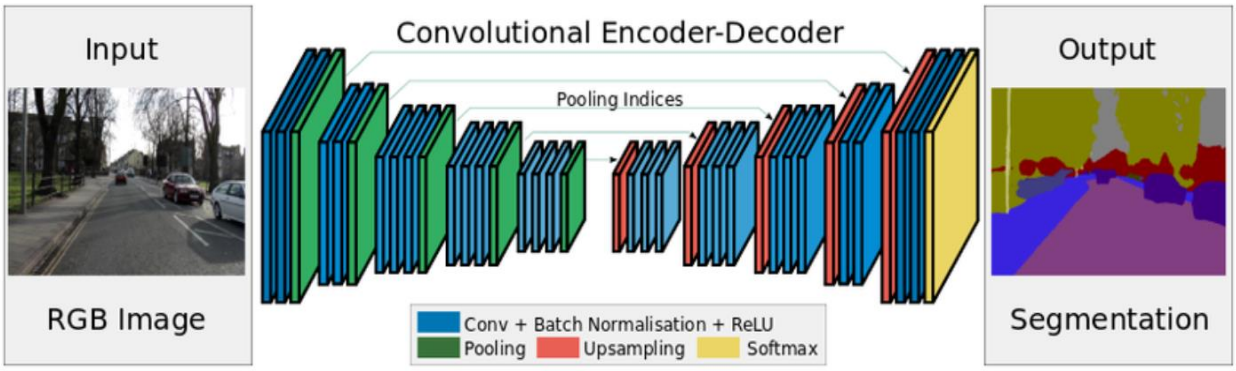


Figure II-15 : Architecture SegNet. [77]

5.9. L'apprentissage par transfert (TL) :

Les réseaux de neurones profonds, largement utilisés dans des domaines tels que la reconnaissance d'images ou le traitement automatique du langage, passent par une phase d'apprentissage gourmande en puissance de calcul. Cela demande des coûts élevés en matière de ressources informatiques et des délais prolongés pour obtenir une intelligence artificielle de haute qualité. Dans ce cas-là, le transfert d'apprentissage permet de réutiliser cette phase d'entraînement d'un modèle à un autre, ce qui permet d'économiser du temps et des ressources.

Le Transfer Learning (TL) ou l'apprentissage par transfert est une stratégie d'apprentissage qui repose sur l'utilisation d'un modèle pré-entraîné sur une tâche spécifique pour résoudre une tâche similaire mais différente. Il se concentre sur le stockage des connaissances apprises lors de la résolution d'un problème afin de les appliquer à un problème différent mais connexe. Par exemple, les connaissances apprises pour reconnaître les chats peuvent s'appliquer à la reconnaissance des tigres. Cette approche (TL) permet d'éviter de concevoir et d'entraîner un modèle à partir de zéro en utilisant plutôt un modèle existant qui effectue une tâche similaire, puis en réutilisant et en adaptant certaines de ses couches pour la nouvelle tâche. [78]

5.10. Le sur-apprentissage et le sous-apprentissage :

L'Overfitting (sur-apprentissage), et l'Underfitting (sous-apprentissage) sont deux problèmes courants rencontrés lors de l'entraînement de modèles.

5.10.1. Sur-apprentissage (Overfitting) :

Le sur-apprentissage se produit lorsque le modèle apprend non seulement les tendances générales présentes dans les données d'entraînement, mais aussi les bruits et les détails spécifiques à ces données. Cela conduit à un modèle qui fonctionne très bien sur les données d'entraînement, mais échoue à bien prédire sur de nouvelles données (validation ou test).

➔Pour éviter ce type de problème :

- Appliquer la technique de dropout pour désactiver aléatoirement certains neurones pendant l'entraînement.
- Augmentation des données (Data augmentation) : Utiliser des techniques d'augmentation des données pour générer plus de données d'entraînement variées.
- Réduire la complexité du modèle (par exemple, réduire le nombre de couches ou de neurones).
- Early Stopping : Arrêter l'entraînement lorsque la performance sur les données de validation cesse de s'améliorer. [46]

5.10.2. Sous-apprentissage (Underfitting) :

Le sous-apprentissage se produit lorsque le modèle est trop simple pour capturer les tendances et les relations présentes dans les données. Cela conduit à une mauvaise performance à la fois sur les données d'entraînement et sur les nouvelles données.

➔Pour résoudre ce problème en doit :

- Ajouter plus de couches ou de neurones au modèle.
- Augmenter le nombre d'époques d'entraînement pour permettre au modèle de mieux apprendre les tendances des données.
- Ajustement du taux d'apprentissage (learning rate). [46]

6. Etat de l’art pour la segmentation sémantique par le deep learning :

La segmentation sémantique est une branche primordiale de la vision par ordinateur qui se concentre sur la classification de chaque pixel d'une image dans une catégorie définie. Les méthodes de deep learning, notamment les réseaux de neurones convolutifs (CNN), ont révolutionné ce domaine en améliorant considérablement ses performances.De nombreux chercheurs ont montré un intérêt marqué pour cette discipline cruciale :

T. S. Arulananth et al [79], ont utilisé le modèle U-Net pour la segmentation sémantique d'images de paysages urbains (dataset Cityscapes). Leur étude a révélé une accuracy de 85%, surpassant les méthodes traditionnelles. La précision obtenue pour les différentes classes d'objets urbains montre que le modèle est capable de gérer des scènes complexes avec un niveau de détail élevé.

La recherche de Yuxiang Sun et al [80], introduisent FuseSeg, un nouveau réseau de fusion de données RVB et thermique, pour améliorer la segmentation sémantique des scènes urbaines. Leur étude se concentre sur ce problème sous des conditions d'éclairage défavorables, proposant une solution par la fusion des données RVB et thermiques. Ils développent un réseau neuronal profond qui prend en charge l'entrée d'une paire d'images (RVB et thermique) et génère des étiquettes sémantiques pixel par pixel. Leur approche a abouti à une précision moyenne de 70,6 % (mAcc).

La segmentation sémantique RGB-T est essentielle pour comprendre les scènes de conduite autonome, et le réseau d'interaction contextuelle (CAINet) a été développé par Ying Lv et al [81] spécifiquement pour cette tâche. Dans cet article, des expériences approfondies sur les ensembles de données MFNet et PST900 démontrent que le CAINet atteint des performances exceptionnelles. Deux métriques, meanAccuracy a été utilisées pour évaluer le modèle :

- Sur l'ensemble de données MFNet, l'utilisation de CAINet a produit un mAcc de 73,2 %.
- Sur l'ensemble de données PST900, l'utilisation de CAINet a produit un mAcc de 94,27%.

Shaimaa Hameed et al [82], présentent un modèle de classification de l'image de réseau VGG16 avec un réseau entièrement convolutif (FCN-8) et transfère la représentation apprise par un réglage fin pour effectuer la segmentation. Skip Architecture est ajouté entre les couches pour combiner des informations d'apparence grossières, sémantiques et locales afin de générer une segmentation précise. Ce modèle est robuste et efficace car il consomme peu de mémoire et un temps d'inférence plus rapide pour les tests et la formation sur l'ensemble de données Camvid. Le système proposé a atteint une précision de 88.04 % sur l'ensemble de données Camvid.

Dans une autre étude, la segmentation des bâtiments urbains dans les images de télédétection à très haute résolution (VHR) est abordée en raison de la complexité des arrière-plans et des apparences variées. Yaning Yi et al [83], ont proposé le modèle DeepResUnet, basé

sur l'architecture U-Net, pour relever ce défi. Ce modèle est capable de segmenter efficacement les bâtiments urbains à partir d'images VHR en produisant des résultats précis à l'échelle des pixels. Les chercheurs ont appliqué DeepResUnet sur des images d'une zone urbaine de Christchurch City, Nouvelle-Zélande, et de Waimakariri, Nouvelle-Zélande, obtenant une précision de 97 % et 96 %, respectivement.

7. Conclusion :

Le deep learning constitue une avancée majeure dans le domaine de l'intelligence artificielle, offrant des capacités de modélisation et de prédiction sans précédent grâce à des algorithmes inspirés du cerveau humain. Ces algorithmes sont organisés en réseaux complexes de neurones artificiels avec plusieurs couches. Plus spécifiquement, les réseaux de neurones convolutionnels (CNN) jouent un rôle crucial en apprenant des caractéristiques pertinentes à partir d'images. Ces réseaux sont capables de réaliser diverses tâches, notamment la segmentation sémantique d'images à l'aide de leurs représentations multicouche.

Dans le prochain chapitre, nous exploiterons le deep learning pour réaliser une segmentation sémantique des images urbaines.

Chapitre III:

Implémentation et Discussion des résultats

1. Introduction :

Dans ce chapitre, nous allons présenter les expérimentations effectuées afin d'obtenir un modèle permettant de segmenter sémantiquement les images de la base de données cityscape. Nous débuterons par une description des environnements et outils de développement utilisés, incluant des bibliothèques comme Python, TensorFlow, Keras.....etc qui ont été essentiels pour notre implémentation. Ensuite, nous aborderons la présentation de la base de données Cityscapes puis on détaille notre approche méthodologique, couvrant le chargement, la préparation et le prétraitement des données, ainsi que la création et l'apprentissage de notre modèle CNN. Nous discuterons ensuite des résultats obtenus, en comparant notre modèle avec d'autres approches et avec un travail de référence de l'état de l'art. Nous explorerons également les défis rencontrés tout au long du projet, suivis de la présentation de l'interface graphique développée pour notre modèle.

2. Environnements et outils de développement :

2.1. Google Colab[1] :

Google Colaboratory, souvent raccourci en "Colab" est un service cloud, offert par Google (gratuit), base sur Jupyter Notebook et destine a la formation à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entrainer des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur. Cet environnement nous permet d'écrire et d'exécuter du code Python dans votre navigateur, avec aucune configuration requise, accès gratuit aux GPU, partage facile.

Figure III-01: logo de google collab. [84]

[1] https://research.google.com/colaboratory/faq.html#resourcelimits

- **Avantages de Colab :**

En plus d'être facile à utiliser, le Colab est assez flexible dans sa configuration et fait beaucoup de travail pour nous :

- Prise en charge de Python 2.7 et Python 3.6
- Accélération GPU gratuite
- Bibliothèques préinstallées : les principales bibliothèques de Python comme TensorFlow, Scikit-learn, entre autres, sont préinstallées et prêtes à être importées.

2.2. Python[2] :

Python est le langage de programmation open source le plus utilisé dans le domaine de l'intelligence artificielle et surtout en Deep Learning vu qu'il contient un nombre important de bibliothèques performantes et utiles pour la vision artificielle et l'utilisation des réseaux de neurones. Il est conçu pour optimiser la productivité des programmeurs en orant des outils de haut niveau et une syntaxe simple à utiliser depuis quelques années. Notre projet est fait avec le langage de Python dans le Cloud et on n'aura pas besoin d'installer un logiciel ou un IDE dans notre ordinateur.

Figure III-02: Logo Python. [85]

2.3. Les bibliothèques utilisés:

2.3.1. Tensorflow[3] :

Est une bibliothèque de logiciels open source pour le calcul numérique haute performance. Son architecture flexible permet un déploiement facile du calcul sur une variété de plates-formes (CPU, GPU, TPU), et des ordinateurs de bureau aux clusters de serveurs en passant par les périphériques mobiles et périphériques. Développé à l' origine par des chercheurs

[2] https://docs.python.org/3/tutorial/
[3] https://docs.python.org/fr/3/tutorial/

et des ingénieurs de l'équipe Google Brain au sein de l'organisation IA de Google, il est livré avec un support solide pour l'apprentissage automatique et l'apprentissage en profondeur et le noyau de calcul numérique flexible est utilisé dans de nombreux autres domaines scientifiques.

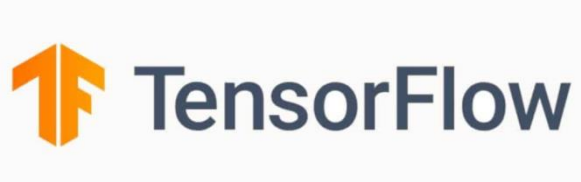


Figure III-03: logo TensorFlow. [86]

2.3.2. Keras :

Est une interface de programmation d'applications (API) basée sur TensorFlow qui facilite la configuration, l'entraînement et l'analyse des réseaux neuronaux. Elle simplifie la construction de réseaux profonds en faisant abstraction de nombreuses fonctionnalités sophistiquées de TensorFlow. Grâce à sa polyvalence et à sa capacité de personnalisation, Keras offre une variété de composants de réseaux neuronaux, notamment des couches épaisses, des couches convolutives, des couches récurrentes et des couches dropout. Pour une utilisation optimale, il gère dynamiquement les ressources CPU et GPU. Afin d'améliorer la convivialité, Keras propose en outre des implémentations de fonctions d'activation, d'optimiseurs, de formules métriques et de techniques de traitement des sessions d'entraînement. [87]

Figure III-04: logo de keras. [88]

2.3.3. Numpy[4] :

Est une bibliothèque permettant d'effectuer des calculs numériques avec Python. Elle introduit une gestion facilitée des tableaux de nombres, des fonctions sophistiquées (diffusion), on peut aussi l'intégrer le code C / C ++ et Fortran.

[4] https://numpy.org/

2.3.4. PIL :

La bibliothèque Python PIL est utilisée pour la manipulation d'images. Elle permet d'ouvrir et de redimensionner les images des bases de données. [89]

2.3.5. Matplotlib :

Est une bibliothèque de traçage pour le langage de programmation Python et son extension mathématique numérique NumPy. Il fournit une API orientée objet permettant d'incorporer des graphiques dans des applications à l'aide de kits d'outils d'interface graphique a usage général tels que Tkinter , wxPython , Qt ou GTK +. [90]

2.3.6. Gradio :

La bibliothèque open-source Python Gradio est utilisée pour créer des composants d'interface utilisateur personnalisés basés sur des algorithmes d'apprentissage automatique. Gradio permet aux utilisateurs de tester des modèles en glissant et déposant des images, en écrivant du texte et en enregistrant des sons pour visualiser les résultats du modèle dans le navigateur. Gradio permet de : créer une démonstration de base basée sur un pipeline d'apprentissage automatique entraîné, obtenir un retour d'information en temps réel sur les performances du modèle, débogage interactif du modèle. [91]

Figure III-05: logo de Gradio. [92]

3. Base de données utilisée :

Nous avons utilisé dans ce travail l'ensemble de données « Cityscapes[5] » qui contient des vidéos étiquetées prises à partir de véhicules conduits en Allemagne. Cette version est un sous-échantillon traité créé dans le cadre de l'article Pix2Pix. L'ensemble de données contient des images fixes des vidéos originales, et les étiquettes de segmentation sémantique sont affichées dans des images à côté de l'image originale.

[5] https://www.kaggle.com/datasets/dansbecker/cityscapes-image-pairs

Cet ensemble de données contient 3475 fichiers d'images. Chaque image a une taille de 512x256 pixels, et chaque fichier est un composite avec la photo originale sur la moitié gauche de l'image, et l'image étiquetée (résultat de la segmentation sémantique) sur la moitié droite. Elle contient 35 classes variées représentant des éléments de l'environnement urbain (voir la figure III-06). [93]

Group	Classes
flat	road · sidewalk · parking+ · rail track+
human	person* · rider*
vehicle	car* · truck* · bus* · on rails* · motorcycle* · bicycle* · caravan*+ · trailer*+
construction	building · wall · fence · guard rail+ · bridge+ · tunnel+
object	pole · pole group+ · traffic sign · traffic light
nature	vegetation · terrain
sky	sky
void	ground+ · dynamic+ · static+

Figure III-06 : Les différentes classes de la base de données cityscape. [94]

4. Notre démarche pour la segmentation sémantique en utilisant le Deep learning :

La schématisation employée par la majorité des projets de vision utilisant l'apprentissage profond est donnée dans le synoptique illustré dans la figure III-07:

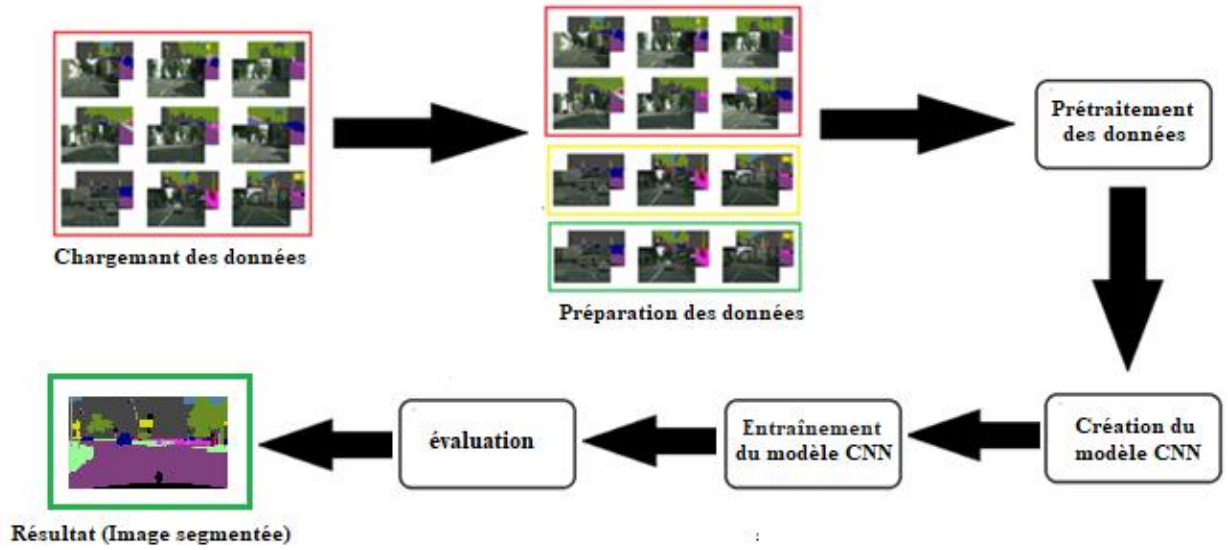


Figure III-07 : Les étapes de la segmentation sémantique en utilisant le Deep learning.

4.1. Chargement des données :

Nous avons téléchargé la base de données « Cityscapes Image Pairs » depuis le site gratuit « Kaggle » et l'avons importée sur notre Google Drive. Pour utiliser cette base de données sur Google Colab, il est nécessaire de monter notre Google Drive dans notre dossier « gdrive ». Pour faire cela, nous exécutons la commande suivante :

Figure III-08 : Comment monter le drive dans notre dossier « gdrive ».

4.2. Préparation des données :

Après le chargement de notre ensemble de données, qui comprend les images et leurs étiquettes de segmentation, plusieurs étapes essentielles sont entreprises pour préparer les données à l'entraînement d'un modèle d’apprentissage profond :

- **Définition des annotations** : Nous avons extrait les labels depuis GitHub[6], utilisés pour annoter les images, en incluant des détails tels que les couleurs et les identifiants uniques, Par exemple, le rouge peut être attribué aux voitures, tandis que le vert pourrait représenter les arbres. Cette étape est cruciale pour assurer que les annotations de vérité terrain sont cohérentes et alignées avec les attentes du modèle, facilitant ainsi une interprétation et une utilisation précises des données.
- **Partitionnement des données :** Les données sont divisées en trois ensembles distincts selon une répartition de 80% pour l'entraînement, 10% pour la validation et 10% pour le teste. Cette séparation permet d'entraîner le modèle sur un ensemble de données, de valider ses performances sur un autre, et d'évaluer son efficacité finale sur un ensemble de teste.
- **Séparation des images :** Les images de ce jeu de données sont combinées avec leurs étiquettes Nous avons développé une fonction spécifique pour les séparer correctement (voir figure III-09).

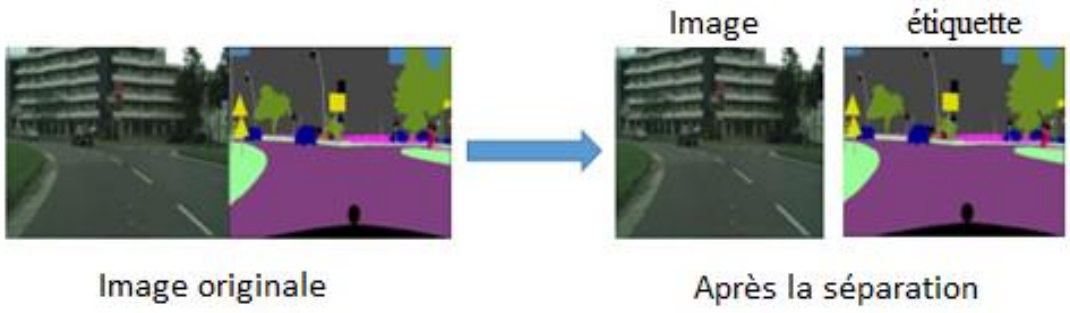


Figure III-09 : Exemple de séparation de nos images

4.3. Prétraitement des données :

Le prétraitement appliqué aux données comprend plusieurs étapes essentielles pour préparer les images et les masques à l'entraînement et à la validation du modèle :

- **Redimensionnement de la taille des images :** Les images et les étiquettes sont uniformément redimensionnés à une taille de 128x128 pixels, afin de correspondre aux dimensions d'entrée requises par notre modèle CNN.

[6] https://github.com/mcordts/cityscapesScripts/blob/master/cityscapesscripts/helpers/labels.py

- **Normalisation des images :** Les valeurs des pixels des images sont normalisées pour améliorer la convergence du modèle lors de l'entraînement. Cette étape consiste à ramener les valeurs des pixels dans une plage de [0…1] en les divisant par 255.
- **Vectorisation :** Les images et les étiquettes sont transformés en un format de vecteurs, facilitant leur gestion et leur traitement par le modèle.
- **Préparation des étiquettes :** Les étiquettes sont également préparées en associant chaque pixel à la classe la plus proche selon une carte de couleurs prédéfinie, en utilisant une mesure de distance euclidienne. Cette carte de couleurs agit comme une table de correspondance entre les couleurs et leurs catégories respectives. Par exemple, si la couleur bleu est associé à la catégorie "voiture", tout pixel dans l'image dont la couleur est proche du bleu sera classé comme appartenant à la catégorie « Voiture». Cette approche permet une conversion efficace des étiquettes en formats utilisables pour l'entraînement.

4.4. Création du notre modèle CNN :

Le modèle CNN prendra en entrée une image et produira en sortie une image segmentée, comme illustré dans la figure III-10,

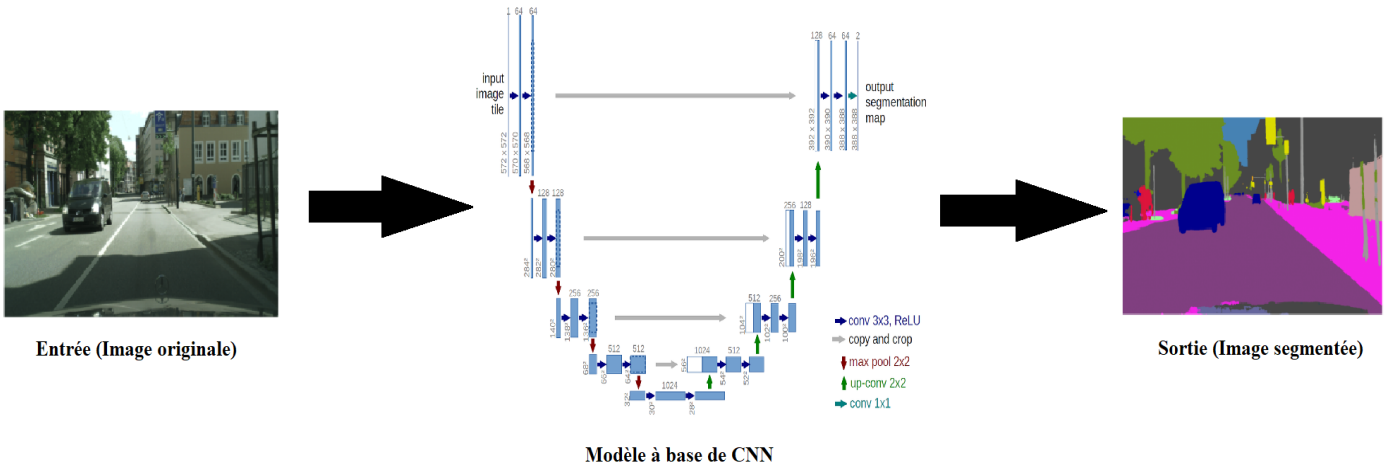


Figure III-10: Structure générale de segmentation sémantique par apprentissage profond

Pour la création de ce modèle CNN, Nous avons choisi d'utiliser l'architecture UNet, décrit en détail dans le chapitre II, et nous avons apporté quelques modifications pour l'adapter à nos besoins spécifiques. Le tableau III-01 décrit notre modèle CNN proposé :

Tableau III-01: Notre modèle CNN qui est basé sur UNet.

Couche	Type	Filtres	Fonction D'activation	Dropout rate
Input	Input Couche d'entrée pour les images			
Downsample 1	Conv2D + **BatchNormalization** + Conv2D + MaxPooling2D 2x2 + **Dropout**	64 filtres 3x3	ReLU	**20%**
Downsample 2	Conv2D + **BatchNormalization** + Conv2D + MaxPooling2D 2x2+ **Dropout**	128 filtres 3x3	ReLU	**20%**
Downsample 3	Conv2D + **BatchNormalization** + Conv2D +MaxPooling2D 2x2 + **Dropout**	256 filtres 3x3	ReLU	**20%**
Downsample 4	Conv2D + **BatchNormalization** + Conv2D +MaxPooling2D 2x2 + **Dropout**	512 filtres 3x3	ReLU	**30%**
Bridge	Conv2D + **BatchNormalization** + Conv2D	1024 filtres 3x3	ReLU	
Couches ajoutées	**Conv2D + BatchNormalization + Dropout**	**1024 filtres 3x3**	**ReLU**	**30%**
Couches ajoutés	**Conv2D + BatchNormalization + Dropout**	**1024 filtres 3x3**	**ReLU**	**30%**
Skip Connection	**Concatenate**	Connexion de saut reliant les sorties des nouvelles convolutions aux sorties précédentes		

Upsample 1	UpSampling2D 2x2 + Concatenate + **Dropout** + Conv2D + **BatchNormalization**+ Conv2D	512 filtres 3x3	ReLU	**20%**
Upsample 2	UpSampling2D 2x2 + Concatenate + **Dropout** + Conv2D + **BatchNormalization**+ Conv2D	256 filtres 3x3	ReLU	**20%**
Upsample 3	UpSampling2D 2x2 + Concatenate + **Dropout** + Conv2D + **BatchNormalization**+ Conv2D	128 filtres 3x3	ReLU	**20%**
Upsample 4	UpSampling2D 2x2 + Concatenate + **Dropout** + Conv2D + **BatchNormalization**+ Conv2D	64 filtres 3x3	ReLU	**20%**
Output	Conv2D	Couche de sortie avec nombre de filtres égal au **nombre de classes**, activation **softmax**		

4.5. Apprentissage par CNN :

Le modèle sera entraîné pour segmenter efficacement les images des scènes urbaines, en distinguant précisément les différents éléments et objets présents dans chaque image.

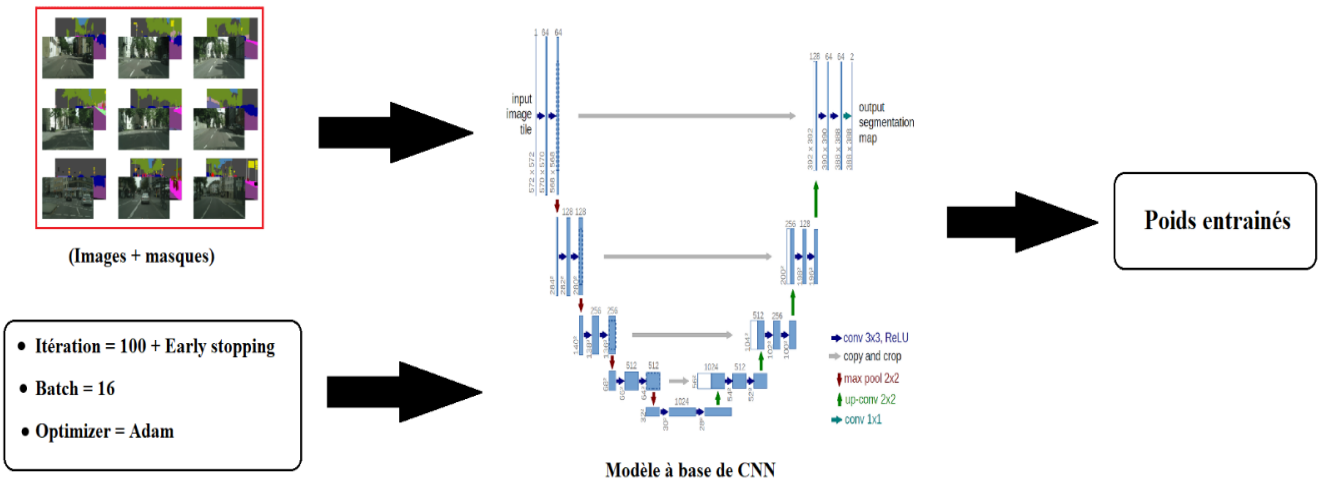


Figure III-11: Phase d'entrainement.

Remarque :

Le nombre d'époques est initialisé à 100, mais il peut varier en fonction de la fonction « early stopping », qui ajuste automatiquement ce paramètre pour prévenir le sur apprentissage.

Avant de lancer l'entraînement, il est essentiel de configurer le type d'exécution pour utiliser le GPU afin d'accélérer le processus. La procédure est illustrée dans la figure III-12.

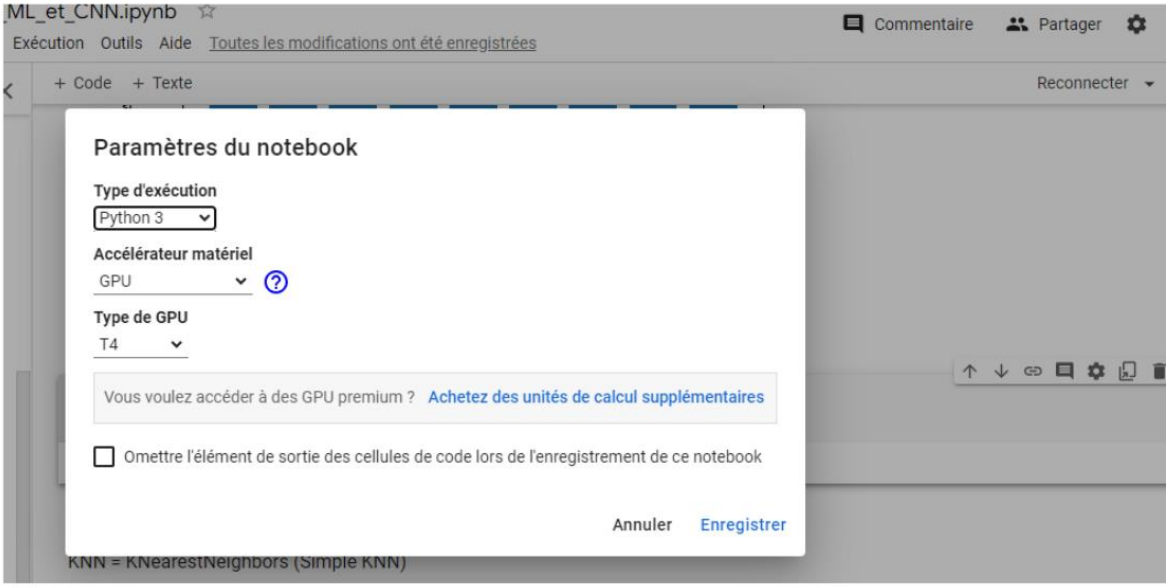


Figure III-12 : Choisir l'accélérateur matériel GPU.

5. Résultat et discussion :

Dans cette section, nous discutons les différents résultats obtenus. Pour évaluer les performances de nos expériences, nous avons utilisé la métrique de précision « Accuracy ».

La précision (Accuracy): La précision des pixels est peut-être le concept le plus facile à comprendre. Il s'agit du pourcentage de pixels de votre image qui sont classés correctement (à la fois vraies positives et vraies négatives) par rapport à les Vrai Positif (VP), Vrai Négatif (VN), Faux Positif (FP) et Faux Négatif (FN). [95]

$$Accuracy = \frac{VP + VN}{VP + FP + FN + VN}$$ **Équation III-01**

Et pour le calcul de la perte, On a utilisé la fonction **« Categorical cross-entropy »** qui est couramment utilisée pour les tâches de classification multi classes (Voir le chapite II).

5.1. Les résultats obtenus de notre approche :

Le graphique illustrant la performance de notre modèle tout au long du processus d'entraînement est présenté dans la figure III-13 suivante :

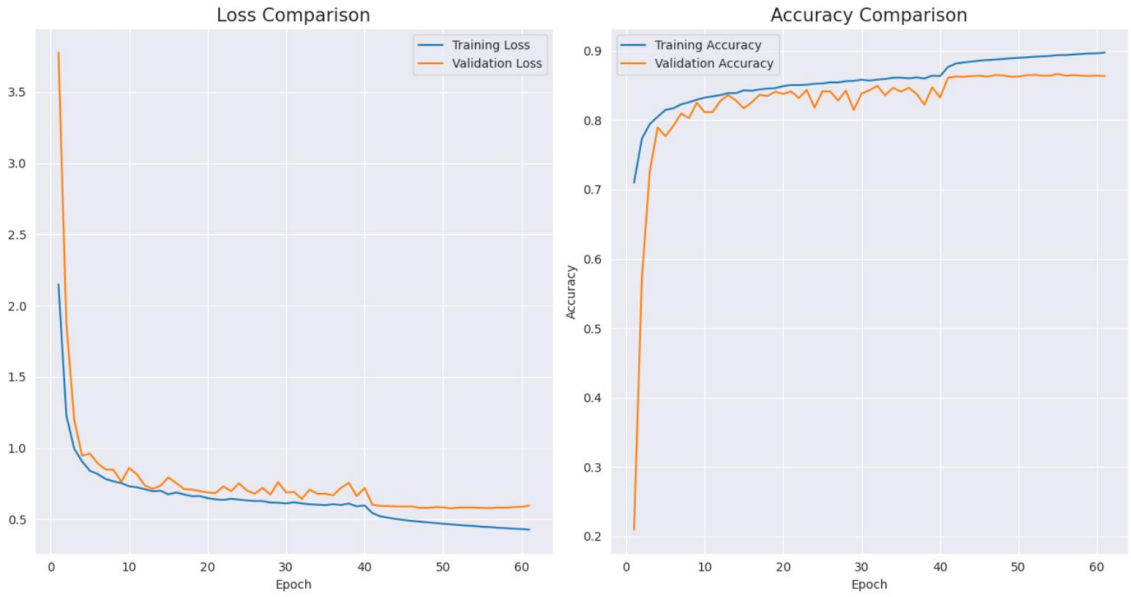


Figure III-13 : Graphique de la perte et la précision en fonction du nombre d'itération.

D'après la figure III-13, On observe qu'à partir d'un certain nombre d'itérations, L'erreur d'entraînement et de validation diminue simultanément, Tandis que la précision d'entraînement et de validation augmente en parallèle.

Le modèle a obtenu une précision de 86,61% sur l'ensemble de validation et de 87,03% sur l'ensemble de test. Cela indique une bonne performance du modèle sur les données de validation et de test, Démontrant son efficacité pour la segmentation sémantique des images de paysages urbains.

5.2. Comparaison avec notre modèle CNN :

A. Comparaison avec Quelque modèles CNN : Nous avons appliqué Trois modèles de Deep learning de type encodeur-décodeur, à savoir SegNet, FCN et UNet (voir Les architectures dans le chapitre II) pour la segmentation sémantique d'images de notre jeu de donnée, cela pour évaluer leur performance dans cette étude.

Tableau III-02 : Les résultats d'évaluation des données de teste sur quelques Modèles CNN.

Modéles	Accuracy de teste	Loss
Modèle SegNet	57.54 %	1.2875
Modèle FCN	80.58 %	0.7623
Modèle UNet	83.83%	0.5969
Notre modèle	**87.03** %	**0.5519**

Les résultats comparatifs des modèles de segmentation montrent des performances variées pour chaque architecture évaluée. Le modèle SegNet obtient une précision de test de 57.54 % avec une perte de 1.2875, soulignant ses limites dans la classification des pixels. Le modèle FCN présente une précision améliorée de 80.58 % et une perte de 0.7623, démontrant des performances nettement supérieures à celles de SegNet. Le modèle UNet va encore plus loin avec une précision de 83.83 % et une perte réduite à 0.5969, illustrant une avancée notable grâce à une architecture optimisée et des hyper paramètres soigneusement ajustés. Toutefois, notre propre modèle surpasse tous les autres avec une précision exceptionnelle de 87.03 % et une perte réduite à 0.5519. Ces résultats mettent en évidence la supériorité de notre modèle, qui non seulement offre la meilleure précision mais aussi une gestion optimale des erreurs, confirmant ainsi son efficacité et sa robustesse dans cette étude comparative.

B. Comparaison avec un travail d'un article de l'état de l'art :

La performance de notre modèle de la segmentation sémantique est évaluée en comparant par un autre travaille et ce dernier et effectuer par TS Arulananth et ses collègues [79], Ils ont proposé le modèle U-Net pour la segmentation sémantique des scènes urbaines voici le résultat obtenu dans le tableau III-03 :

Tableau III-03: Comparaison avec un travail d'article**.**

	Dataset	Méthode	Epoch	Optimiser	Batch size	Learning rate	Validation accuracy	Nombres de classes
TS Arulananth [79]	Cityscapes	Basé sur U-net	35	Adam	32	0.0001	85%	8
Notre approche	Cityscapes	Basé sur U-net	62	Adam	16	0.001	86,61%	31

Notre modèle, spécialement conçu pour la segmentation d'images sur la base de données Cityscapes, se distingue par sa robustesse et sa précision, préservant les détails spatiaux tout en capturant les informations contextuelles à différentes échelles. Arulananth [79] a utilisé une architecture U-Net de base avec l'ajout de couches de normalisation par lots après chaque couche de convolution, ainsi que des couches de dropout avec des ajustements sur les hyper paramètres. Cependant, ces modifications n'ont pas permis d'obtenir des résultats aussi satisfaisants. Le tableau III-03 montre la configuration des deux approches : nous avons entraîné notre modèle sur 62 epochs, une période d'apprentissage plus longue par rapport aux 35 epochs utilisées par le modèle d'Arulananth [79]. Cette différence suggère que notre modèle a eu davantage d'opportunités pour ajuster ses paramètres et apprendre des données. Notre modèle a atteint une précision de validation 86.61%, comparativement à 85% pour le modèle décrit dans l'article. De plus, notre modèle a segmenté 31 classes contre seulement 8 pour celui d'Arulananth, ce qui démontre une capacité accrue à traiter des images complexes et diversifiées. Cette performance supérieure peut être attribuée à une meilleure adaptation aux spécificités du domaine, à une optimisation plus efficace des hyper paramètres et à l'ajout de couches de convolution supplémentaires, cruciales pour l'extraction précise des caractéristiques.

Voici la figure III-14, qui présente quelques résultats obtenus et illustre l’efficacité de notre modèle.

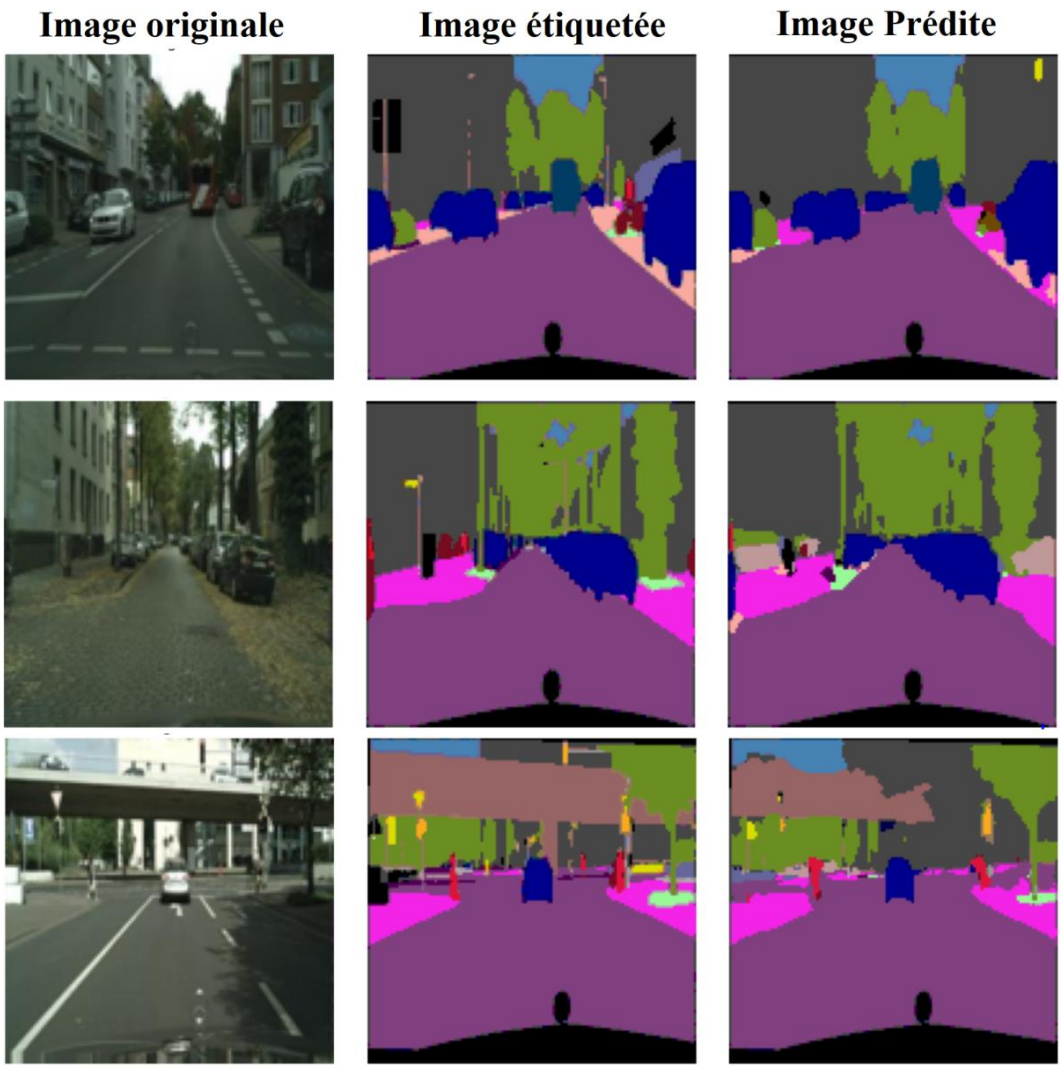


Figure III-14 : Les résultats de notre segmentation sémantique sur les données de validation.

6. Les défis auxquels nous avons été confrontés :

Atteindre des résultats satisfaisants avec notre modèle de segmentation a représenté un défi complexe, en raison de plusieurs obstacles rencontrés. Parmi ceux-ci :

a. Transfer learning:

Malgré l'application de transfert d'apprentissage en utilisant différents modèles tels que VGG16, VGG19, et autres, ainsi que, plusieurs ajustements d'hyper paramètres, les performances obtenues sont demeurées relativement constantes, variant entre 83 % et 85 %. Ce

constat met en lumière une certaine stagnation dans les performances, même après des modifications significatives des hyper paramètres.

Voici le tableau III-04 qui montre certains résultats obtenus en utilisant le transfer learning sur notre notre propre approche basé sur UNet :

Tableau III-04: Les résultats d'évaluation des données de teste sur notre propre model CNN en utilisant le transfer learning

Modéles	Accuracy de teste	Loss
VGG16+notre modèle CNN basé sur UNet	83.16	0.7046
VGG19+notre modèle CNN basé sur UNet	83.68	0.6669

b. Accès aux ressources sur Google Colab:

La limitation d'accès aux ressources, en particulier sur Google Colab, a restreint notre capacité à explorer des architectures plus complexes ou à augmenter la taille des modèles pour améliorer les performances. Cette contrainte a compliqué l'implémentation de configurations plus sophistiquées, telles que l'ajustement fin des hyper paramètres, l'utilisation de réseaux de neurones plus profonds, ou l'intégration de techniques d'apprentissage avancées, qui pourraient potentiellement résoudre les problèmes rencontrés. Malgré ces défis, notre modèle a néanmoins réussi à fournir de bons résultats.

7. Interface graphique :

Nous avons finalisé ce travail en développant une interface graphique, programmée en Python à l'aide de la bibliothèque « Gradio », spécialement conçue pour créer des interfaces utilisateur interactives (voir les figures III-15, III-16, III-17).

Premièrement on lance l'interface :

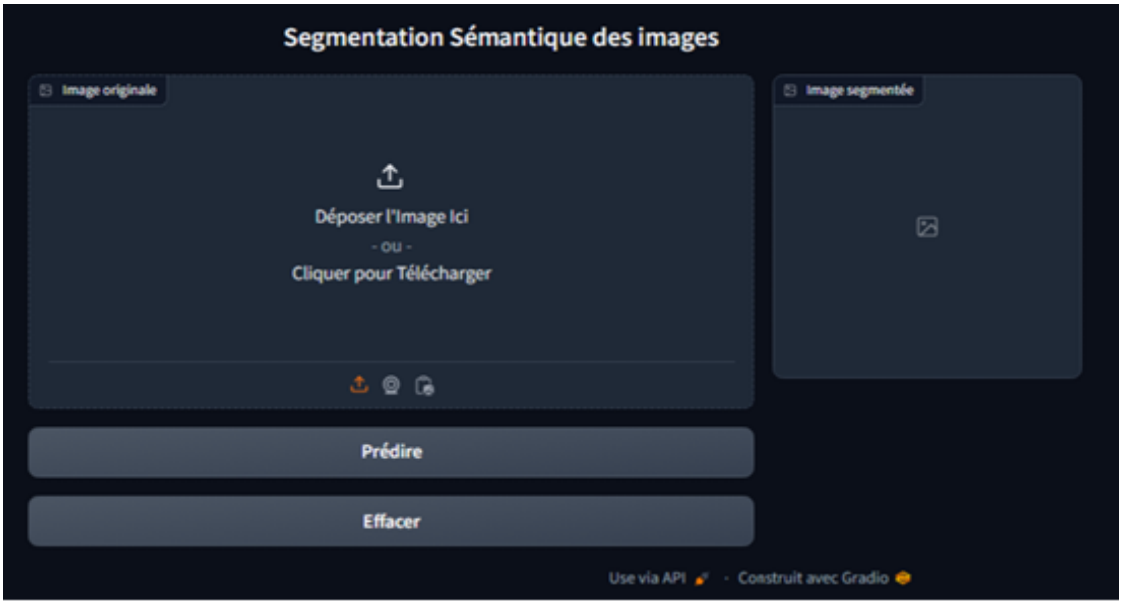


Figure III-15 : Lancement de l'interface.

Deuxièmement faire une clique sur « Déposer l'image ici » à gauche et on choisit l'image pour la segmenter :

Figure III-16 : Importer l'image à segmenter.

Troisièmement faire une clique sur le bouton « prédire » à gauche :

Figure III-17 : Prédire l'image segmentée.

8. Conclusion :

Ce chapitre a détaillé le processus complet de développement d'un modèle de segmentation sémantique des images urbaines en utilisant le Deep learning, En se concentrant sur l'architecture U-Net. Nous avons commencé par décrire les outils et environnements de développement essentiels, Suivi d'une explication approfondie de la préparation des données et du prétraitement nécessaire. La construction et l'entraînement du modèle ont été discutés, Avec une analyse des résultats obtenus et une comparaison avec d'autres approches et un modèle de référence.

Les résultats montrent que notre modèle, avec des ajustements spécifiques pour les scènes urbaines, Offre des performances supérieures en termes de précision et de gestion des erreurs. En surmontant les défis posés par la complexité des scènes urbaines, Notre approche démontre la robustesse et l'efficacité de notre modèle. Cette étude met en évidence l'importance de l'innovation continue et de l'optimisation des architectures de deep learning pour atteindre des résultats de segmentation sémantique précis et fiables.

CONCLUSION GENERALE

Conclusion générale

Ce mémoire a exploré les approches de Deep learning pour la segmentation sémantique d'images, un domaine crucial de la vision par ordinateur. La segmentation sémantique consiste à attribuer une étiquette à chaque pixel d'une image, offrant ainsi une compréhension détaillée du contenu visuel. Les récents progrès en Deep learning, notamment les réseaux de neurones convolutionnels (CNN), ont considérablement amélioré la précision et l'efficacité des algorithmes de segmentation sémantique.

L'étude s'est concentrée sur le modèle U-Net, réputé pour ses performances exceptionnelles en segmentation d'images, et l'avons adapté pour la segmentation de scènes urbaines en utilisant l'ensemble de données Cityscapes. Des ajustements ont été apportés pour optimiser la performance du modèle dans ce contexte spécifique, urbaines.

En plus de l'implémentation technique, ce mémoire inclut une revue de la littérature, analysant les fondements théoriques et les développements récents dans le domaine de la segmentation sémantique avec une revue générale sur le deep learning.

Cependant, la segmentation sémantique, malgré les progrès réalisés, elle continue de faire face à des défis majeurs. L'un des principaux obstacles est la diversité et la complexité des scènes urbaines avec des conditions d'éclairage variables, où une multitude d'objets, de structures et de contextes interagissent de manière dynamique. De plus, la disponibilité de données annotées de haute qualité est essentielle pour l'entraînement efficace des modèles de segmentation, mais la création de telles bases de données reste souvent coûteuse et fastidieuse.

Les avancées en segmentation sémantique sont étroitement liées aux progrès des réseaux de neurones profonds. Pour améliorer encore la précision des modèles de segmentation, il est crucial de relever les défis clés identifiés dans cette étude. Parmi les perspectives futures pour ce projet, plusieurs axes de recherche méritent d'être explorés :

1. **Augmentation des Données** : Implémenter des techniques d'augmentation des données, telles que les transformations géométriques, les ajustements de couleur, et les perturbations

de bruit, pour accroître la diversité des données d'entraînement et rendre le modèle plus robuste aux variations des scènes urbaines.

2. **Utilisation de Colab Pro pour un Accès Amélioré au GPU** : L'utilisation de Colab Pro offre un accès à des GPU plus puissants, permettant d'exécuter des architectures plus complexes et profondes. Cette approche pourrait accélérer les temps d'entraînement et permettre l'expérimentation avec des modèles plus avancés.
3. **Génération d'Images Photoréalistes via Transfert de Style** : Utiliser des réseaux antagonistes génératifs (GAN) pour traduire des images urbaines en images photoréalistes représente une autre direction potentielle. En combinant ces images traduites avec leurs annotations correspondantes, il devient possible de former des réseaux de segmentation sémantique de manière entièrement supervisée, améliorant ainsi la qualité et la précision des segmentations.

En conclusion, Ce mémoire souligne l'importance de l'innovation et de l'adaptation des modèles de deep learning pour améliorer la segmentation sémantique. Les travaux futurs devront continuer à optimiser les architectures de réseau pour mieux s'adapter aux variations et conditions environnementales, ouvrant la voie à de nouvelles applications et améliorations dans ce domaine dynamique.

BIBLIOGRAPHIE

Bibliographie

[1] M. Andre , « Introduction aux techniques de traitement d'images » , Edition Eyrolles, 1987.

[2]M. Sandeli,« Traitement d'images par des approches bio-inspirées application à la segmentation d'images », Magister en informatique ,Université Constantine 2. 2014.

[3]MEDJAOUI Amina, FARES Fadia, « Segmentation des Images par Contours Actifs : Application sur les Images Satellitaires à Haute Résolutions », Mémoire de master, L'université Abou Bakr Belkaid – Tlemcen Faculté des sciences : Département d'Informatique, 2012.

[4]LAMICHE Hamza, « Segmentation des images aux niveaux de gris par les lignes de partage des eaux (LPE) ». Mémoire de master. L'université Mohamed Boudiaf – M'Sila Tlemcen Faculté de Technologie, Département d'électronique, 2019.

[5] Site web https://www.mit-university.net/index.php?view=entry&id=2%3Ala-notion-de-pixel&option=com_lyftenbloggie&Itemid=123 Consulté le 06/06/2024

[6]BENKAMOUCH Wiam, « Détection d'objets et deep Learning dans un trafic routier », Mémoire de master, Université 8 May 1945 – Guelma, Faculté de Mathématiques et d'informatique : Département : d'informatique, juin 2023.

[7]M.Hadallah, « Codage des images fixes par une méthode hybride basée sur la QV et les approximations fractales», Mémoire de magistère, Université USTHB – Alger, 1997.

[8]BENALI Moustafa, « Reconnaissance Automatique des Chiffres Manuscrits ». Thèse de Doctorat. Université Abou Bakr Belkaid – Tlemcen, 2017.

[9]ABDI Zohra, MAHIA Daima, « Développement d'un modèle deep learning pour la segmentation et la classification d'images pulmonaires », Projet de certification d'une startup dans le cadre de la décision ministérielle 1275, Mémoire de master, Université Ibn Khaldoun – Tiaret, 2023.

[10]C.Houassine, « Segmentation d'images par une approche biomimétique hybride », Mémoire de magistère, Université M'Hamed Bougara – Boumerdes, 2012.

[11]REFICE Ismail, « Critères d'évaluation pour les méthodes de segmentation d'images », Mémoire de master, Université M'Sila, Faculté de technologie, Département : d'électronique, juin 2012.

[12]HAMMOUDI Mahmoud, KECHRA Radhouane Abdelkhaliq, « Systèmes d'Identification Automatique des Véhicules », Mémoire de master, Université Ibn Khaldoun – Tiaret, Faculté de Mathématiques et d'informatique : Département : d'informatique, juin 2022.

[13]CHADLI Rachid, LAKAF Houria, « Recherche d'Images dans un Contexte Big Data». Mémoire de master II, Université d'Ibn Khaldoun – Tiaret Faculté des Mathématiques et de l'Informatique : Département Informatique, 2016.

[14]N. MERABET, M. MAHLIA, « Recherche d'images par le contenu », Mémoire de master, Faculté des Sciences : Département d'Informatique, Université Abou Bakrbelkaid – Tlemcen, 2011.

[15] SAADI Khaled Iben El Walid, BENDAHI Khaled. « Segmentation D'Image Médicale Via Non superviser Réseau de neurones convolutif ». Mémoire de master. L'université Mohamed Boudiaf – M'Sila, Faculté de technologie : Département d'électronique, 2022.

[16]BERRAHOU Zoulikha. « Segmentation des polypes colorectaux par les réseaux de neurones entièrement convolutifs ». Mémoire de master. Université Abou Bakr Belkaïd – Tlemcen, Faculté de technologie : Département de génie biomédical, 2022.

[17]Site web https://www.ictjournal.ch/news/2016-10-04/bright-box-devoile-un-systeme-de-conduite-autonome-entraine-a-laide-de-jeux-video Consulté le 22/06/2024.

[18]Site web https://fr.linkedin.com/advice/1/what-advanced-image-segmentation-techniques-used-detailed-cdqsc?lang=fr Consulté le 06/06/2024

[19]Site web https://www.analyticsvidhya.com/blog/2019/02/tutorial-semantic-segmentation-google-deeplab/ Consulté le 22/06/2024.

[20]Site web https://www.ibm.com/fr-fr/topics/image-segmentation#:~:text=La%20segmentation%20s%C3%A9mantique%20est%20le,information%20(comme%20les%20objets) Consulté le 06/06/2024.

[21] Site web https://www.labellerr.com/blog/semantic-vs-instance-vs-panoptic-which-image-segmentation-technique-to-choose/ Consulté le 22/06/2024.

[22]Site web https://www.innovatiana.com/post/understand-panoptic-segmentation Consulté le 06/06/2024

[23]LARBI Nacerdine , « Segmentation d'images avec le Deep Learning», Mémoire de master , Université Mouloud Mammeri De Tizi-Ouzou, Faculté de Génie Electrique Et D'informatique: Département d'automatique, 2018.

[24]K. Larbi, « Segmentation d'image basée sur la modélisation statistique d'histogramme », Mémoire de Magister, Université Mouloud Mammeri de Tizi-Ouzou, 2012

[25]M. Melliani, « Segmentation d'image par cooperation regions-contours », Mémoire de magistère, école national supérieur d'informatique, 2012.

[26]M. Kass, A. Witkin et D. Terzopoulos , « snakes : active contour models », Journal: Cinternational journal of computer vision , p321–331, 1987.

[27]J. Canny, « Computational approach to edge detection », IEEE trans. on pattern analysis and machine intelligence, vol. 8, n°6, pp. 679-698, novembre 1986.

[28]R. Deriche, « Using Canny's criteria to derive a recursively implemented optimal edge detector », international journal of computer vision, pp. 167-187, 1987.

[29]SAHLI Aoulia, « Segmentation des images médicales par apprentissage profond», Mémoire de master, Université Larbi Tebessi , Faculté des sciences exactes et sciences de la nature et de la vie: Département : Mathématiques et Informatiquen, Domaine : Informatique,2021.

[30]A.N. Benaichouche, « Conception de méta heuristiques 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.

[31]Lotfi A. Zadeh . « Fuzzy sets ». Information and control, 8(3) : 338–353 , 1965.

[32]BOUBAYA Semail , BERBIT Djamel , « Deep learning pour la segmantation d'images », Mémoire de master , Université Mohamed Boudiaf – M'Sila Faculté de technologie : Département d'électronique, 2021.

[33]Site web https://www.ibm.com/fr-fr/topics/knn Consulté le 10/06/2024.

[34]Site web https://www.ibm.com/docs/fr/db2/11.5?topic=building-naive-bayes Consulté le 10/06/2024.

[35]Site web https://www.geeksforgeeks.org/support-vector-machine-algorithm/ Consulté le 10/06/2024.

[36]Site web https://www.ibm.com/docs/fr/spss-statistics/saas?topic=perceptron-architecture-multilayer Consulté le 10/06/2024.

[37]Site web https://blent.ai/blog/a/k-means-comment-ca-marche Consulté le 10/06/2024.

[38]M.-H. Masson, T. Denœux, « ECM : Algorithme évidentiel des C-moyennes », Université de Picardie Jules Verne et l'Université de Technologie de Compiègne, Laboratoire Heudiasyc, UMR CNRS 6599 BP 20529 60205 Compiègne.

[39]Site web https://www.ibm.com/docs/fr/spss-statistics/saas?topic=equations-generalized-estimating-estimation Consulté le 10/06/2024.

[40]Site web « blog.hubspot », https://blog.hubspot.fr/marketing/deep-learning Consulté le 06/06/2024

[41]Site web « retengr », https://www.retengr.com/le-blog/deep-learning-definitions-applications-avantages-inconvenients Consulté le 06/06/2024

[42]Site web « sicara », https://www.sicara.fr/fr/parlons-data/deep-learning Consulté le 06/06/2024.

[43]Site web « praedictia » SOFAD, https://praedictia.com/page/reseaux-de-neurones/lhistoire-des-reseaux-de-neurones.html Consulté le 06/06/2024

[44]N. OUM-HANI, « Etude Comparative des CNNs et de L'algorithme K-NN en mammographie », Faculté: Mathématiques et Informatiquen, Département : Informatique, Mémoire de master, Université Ibn Khaldoun – Tiaret, 2023.

[45]P. Drouin, Site web «datafranca,», https://datafranca.org/balados/quest-ce-quun-reseau-de-neurones/ Consulté le 06/06/2024.

[46]Site web «free-work», https://www.free-work.com/fr/tech-it/blog/actualites-informatiques/quest-ce-quun-perceptron-et-a-quoi-sert-il Consulté le 06/06/2024.

[47]Dr. Benomar Mohammed lamine, « Deep learning », Cours du module Deep Learning, Université Belhadj Bouchaib – Ain Témouchent, 2024.

[48]Site web http://www.mplsvpn.info/2017/11/what-is-perceptron-in-deep-learning.html Consulté le 18/06/2024.

[49]Site web https://iatpe2015.wordpress.com/le-fonctionnement/le-reseau-de-neurones-artificiel/comparaison-entre-reseau-de-neurones-biologique-et-artificiel/?fbclid=IwZXh0bgNhZW0CMTAAAR26S8lei5JjeQ5HUjenbMznu2kuHzLQOYbzc_55WzhPJgzsOYsTnfRM-qc_aem_P1n9sRaoFdfF9G9Dpivcjg Consulté le 18/06/2024.

[50]Site web https://medium.com/@sasirekharameshkumar/understanding-deep-learning-basics-part-2-466a7422d24b Consulté le 22/06/2024.

[51]Site web «kongakura», https://kongakura.fr/article/Le-perceptron-multicouches Consulté le 06/06/2024.

[52]Site web https://datascientest.com/convolutional-neural-network Consulté le 18/06/2024.

[53]Philippe Thomas et André Thomas, « Sélection de la structure d'un perceptron multicouches pour la réduction d'un modèle de simulation d'une scierie », Source: arXiv, Décembre 2008.

[54]Site web «ibm»,https://www.ibm.com/fr-fr/topics/recurrent-neural-networks Consulté le 06/06/2024.

[55]C.S. Dave Bergmann, Site web «ibm», https://www.ibm.com/fr-fr/topics/autoencoder Consulté le 06/06/2024.

[56]J. Robert, Site web « datascientest », https://datascientest.com/convolutional-neural-network Consulté le 06/06/2024.

[57]ADDAHOUM Mohamed, « Réseau de neurones Convolutfs – CNN », Cours en ligne.

[58]Site web https://www.agrotic.org/uncategorized/les-one-stage-detector-en-elevage-yolo/ Consulté le 18/06/2024

[59]Site web https://forum.huawei.com/enterprise/fr/Qu-est-ce-qu-un-r%C3%A9seau-de-neurones-convolutifs/thread/667491061695660032-667481001632346112 Consulté le 22/06/2024.

[60]Site web « blent », https://blent.ai/blog/a/cnn-comment-ca-marche Consulté le 18/06/2024.

[61]Site web https://www.ia-insights.fr/comprendre-les-reseaux-de-neurones-convolutifs-cnn-le-guide-ultime-pour-une-croissance-rapide-dans-le-deep-learning/ Consulté le 18/06/2024.

[62]A. A. a. S. Amidi, Site web « stanford.edu », https://stanford.edu/~shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels Consulté le 18/06/2024.

[63]Université Standford, « CS230 - Apprentissage profond », Site web https://stanford.edu/~shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels Consulté le 22/06/2024.

[64]AKHENAK Khalissa et AMAOUCHE Ilham, « Apprentissage profond pour la segmentation sémantique des tumeurs cervicales », Mémoire de master, Université Ferhat Abbas – Sétif 1, Faculté des Sciences, Département : de Physique, juin 2022.

[65]Site web «365datascience», https://365datascience.com/tutorials/machine-learning-tutorials/cross-entropy-loss/ Consulté le 22/06/2024.

[66]Site web https://rtavenar.github.io/deep_book/fr/content/fr/mlp.html Consulté le 20/06/2024.

[67] Site web https://botpenguin.com/glossary/softmax-function Consulté le 20/06/2024.

[68]BELLAHMER Hacene, « Implémentation et évaluation d'un modèle d'apprentissage automatique pour l'estimation de la valeur marchande de propriétés immobilières », Mémoire de master, Université Mouloud Mammeri – Tizi-Ouzou, Faculté de Génie électrique et Informatique, Département : d'Informatique, 2020.

[69]Site web « blent.a », https://blent.ai/blog/a/unet-computer-vision Consulté le 20/06/2024.

[70]Site web « geeksforgeeks.org », https://www.geeksforgeeks.org/u-net-architecture-explained/ Consulté le 22/06/2024.

[71]Site web https://geoafrica.fr/detection-de-constructions-illegales/ Consulté le 20/06/2024.

[72]Sheng-Yao Huang, Wen-Lin Hsu, Ren-Jun Hsu et Dai-Wei Liu, « Fully Convolutional Network for the Semantic Segmentation of Medical Images: A Survey », Novembre 2022.

[73]Site web https://towardsdatascience.com/review-fcn-semantic-segmentation-eb8c9b50d2d1 Consulté le 20/06/2024.

[74]G. Boesch, Site web « viso.a », https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/ Consulté le 20/06/2024.

[75]Site web https://medium.com/@qucitfr/apprentissage-automatique-pour-la-d%C3%A9tection-danomalies-dans-les-donn%C3%A9es-ouvertes-application-%C3%A0-23268284a992 Consulté le 20/06/2024

[76]S. Hesaraki, Site web «medium,», https://medium.com/@saba99/segnet-a139ce77b570 Consulté le 22/06/2024

[77]Site web https://www.researchgate.net/figure/Illustration-du-reseau-de-segmentation-semantique-SegNetBADRINARAYANAN-et-al-2015_fig67_322498355 Consulté le 22/06/2024

[78]A. Crochet-Damais, Site web « journaldunet.fr », https://www.journaldunet.fr/intelligence-artificielle/guide-de-l-intelligence-artificielle/1501859-transfer-learning/ Consulté le 22/06/2024

[79]T. S. Arulananth, P. G. Kuppusamy,Ramesh Kumar Ayyasamy ,Saadat M. Alhashmi ,M. Mahalakshmi,K. Vasanth,P. Chinnasamy, « Semantic segmentation of urban environments: Leveraging U-Net deep learning model for cityscape image analysis », PLOSE ONE, 2024.

[80]Yuxiang Sun, Weixun Zuo, Peng Yun, Hengli Wang, Ming Liu, « FuseSeg : Semantic Segmentation of Urban Scenes Based on RGB and Thermal Data Fusion », ieeexplore, vol. 18, n° %13, pp. 1000 - 1011, 2021.

[81]Ying Lv, Zhi Liu, Gongyang Li, « Context-Aware Interaction Network for RGB-T Semantic Segmentation, ieeexplore », vol. 26, pp. 6348 - 6360, 2024.

[82]Amani Noori, Shaimaa Hameed, Raghad Abdulaali Azeez, « Semantic Segmentation of Urban Street Scenes Using Deep Learning » Webology , vol. 19, n° %11, pp. 2294-2306, 2022.

[83]Yaning Yi, Zhijie Zhang, Wanchang Zhang, Chuanrong Zhang ,Weidong Li and Tian Zhao, « Semantic Segmentation of Urban Buildings from VHR Remote Sensing Imagery Using a Deep Convolutional Neural Network » , mdpi, vol. 11, n° %115, 2019.

[84]Site web https://www.hwlibre.com/fr/colaboratoire-google/ Consulté le 22/06/2024.

[85]Site web https://resotel.net.ma/recherche/afficher/langage-de-programmation-python Consulté le 22/06/2024

[86]Site web https://datascientest.com/formation-tensorflow Consulté le 22/06/2024.

[87]Chicho, B. T., & Sallow, A. B. (2021), « A Comprehensive Survey of Deep Learning Models Based on Keras Framework », Journal of Soft Computing and Data Mining, 2(2), Article 2.

[88]Site web https://www.actuia.com/actualite/publication-de-la-documentation-de-keras-en-francais/ Consulté le 22/06/2024.

[89]Kumar, V. (2020), « Age Prediction using Image Dataset using Machine Learning. International Journal of Innovative Technology and Exploring Engineering », 8, 107-113.

[90]DIALLO Nene Adama Dian, « La reconnaissance des expressions faciales », Mémoire de master, Université 8 Mai 1945 – Guelma, Faculté des Mathématiques, d'Informatique et des Sciences de la matière, Département : d'Informatique, juillet 2019.

[91]Pycaret. (s. d.). « Supercharge Your ML with PyCaret and Gradio ». Site web https://pycaret.gitbook.io/docs/learn-pycaret/official-blog/supercharge-your-mlwith-pycaret-and-gradio Consulté le 22/06/2024.

[92]Site web https://medium.com/@shreshthbansal2505/crafting-conversations-build-your-chatbot-with-gradio-and-openai-6294bd064b56 Consulté le 22/06/2024

[93]Site web https://www.kaggle.com/datasets/dansbecker/cityscapes-image-pairs Consulté le 10/06/2024.

[94]Site web https://www.cityscapes-dataset.com/dataset-overview/ Consulté le 10/06/2024.

[95]BERRAHOU Zoulikha, « Segmentation des polypes colorectaux par les réseaux de neurones entièrement convolutifs », mémoire de master en genie biomedical, Université Abou Bakr Belkaïd – Tlemcen, 2022.