

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'Aïn Témouchent - BELHADJ Bouchaib  
كلية العلوم والتكنولوجيا  
Faculté des Sciences et de la Technologie



قسم الرياضيات والإعلام الآلي  
Département des Mathématiques et Informatique

## Rapport de projet de fin d'études

Pour l'obtention du Diplôme de Master en Informatique Spécialité :  
Réseaux et Ingénierie des Données (RID)

Réalisé par :

Melle. MERKACH Fatima

Mme. BENGOU DIFA Safaa

---

# *Utilisation de l'apprentissage profond pour la détection d'objets*

---

Devant le jury composé de :

---

Dr. BENOMER Mohamed Lamine	U.B.B.A.T. (Aïn Témouchent)	Président
Dr. BOUHALOUAN Djamila	U.B.B.A.T. (Aïn Témouchent)	Examineur
Mme SAIDI Samira	U.B.B.A.T. (Aïn Témouchent)	Encadreur

---

## REMERCIEMENTS

Nous souhaitons tout d'abord exprimer notre gratitude envers DIEU le Tout-Puissant, le Miséricordieux, pour nous avoir accordé le courage, la force, la santé et la persévérance nécessaires pour mener à bien ce travail dans les meilleures conditions.

Nous tenons à remercier notre superviseur, Mme SAIDI, pour nous avoir guidés tout au long de ces mois. Il n'a jamais semblé se lasser de répondre à nos questions.

Cela nous a aidés à identifier nos lacunes et elle nous a enseigné une méthode de recherche scientifique appropriée au fur et à mesure de nos avancées dans notre investigation.

Nos remerciements vont ensuite aux membres du jury, qui ont accepté, sans réserve, de lire et d'évaluer ce mémoire à sa juste valeur, et de nous faire part de leurs remarques sûrement pertinentes qui, avec du recul, contribueront, sans aucun doute, à l'amélioration de notre travail.

Nous tenons à remercier nos parents pour leur soutien inébranlable, ainsi que nos sœurs, nos frères et nos amis qui nous ont soutenus de différentes manières tout au long de l'élaboration de ce mémoire.

Pour la même occasion, nous adressons nos remerciements à tous nos enseignants pour leurs efforts qui ont guidé nos pas et enrichi nos travaux tout au long de nos études universitaires.

## *Dédicaces*

*Je dédie ce mémoire*

*À mon père 'Bengoudifa rabah' et ma mère 'Maarouf kheira', je souhaite exprimer ma profonde gratitude pour leur amour inestimable, leurs sacrifices, leur confiance et leur soutien indéfectible. Leurs valeurs inculquées ont joué un rôle crucial dans ma formation et ont contribué à façonner la personne que je suis aujourd'hui.*

*Je leur suis infiniment reconnaissante pour tout ce qu'ils ont fait et continuent de faire pour moi.*

*À mes sœurs Chahinez, Nawal et mon frère Sofiane, je tiens à exprimer ma profonde gratitude pour leur tendresse, et leurs encouragements et leur soutien constant ont été une source d'inspiration pour moi.*

*À mes amies proches Azzi malek et Ben abdelmoula bouchra et amir aichouche et ghitri houda, je tiens à vous remercier pour votre amitié sincère, votre soutien inconditionnel et les moments de joie que nous avons partagés.*

*À mon binôme Merkach fatima, nous avons formé une équipe solide et je vous suis reconnaissante pour votre collaboration tout au long de ce projet. Et j'apprécie grandement notre partenariat.*

*Enfin, à mes professeurs et à mon encadreuse Madame saidi, je vous suis reconnaissante pour votre expertise, vos conseils éclairés et votre disponibilité. Votre enseignement de qualité et vos orientations précieuses ont été essentiels dans la réalisation de ma mémoire de Master 2.*

***Bengoudifa Safaa***

## *Dédicaces*

*La vie n'est qu'un éclair, Et un jour de réussite est un jour très cher.*

*Je dédie ce modeste mémoire à tous ceux qui me sont chers :*

*A mon merveilleux père décédé l'année passée et qui là où il est, continue de veiller sur moi « رحمك الله يا أبي و اداامك سندا لي »*

*Ma mère qui m'a entouré d'amour, d'affection et qui fait tout pour ma réussite, que dieu la garde*

*A mes frères, ma sœur et mes nièces et tous les membres de ma famille, qui m'ont encouragé et aidé durant toutes mes études.*

*À mon cher collègue et binôme un grand merci pour cette incroyable aventure que nous avons vécue ensemble. Tu as été un partenaire fiable, compétent et motivé, et je suis convaincu que notre projet n'aurait pas été aussi réussi sans toi.*

*A toutes personnes que j'ai senties redoutable de leur dédier ce modeste en termes d'amour et de profonde gratitude.*

***Merkach Fatima***

## Résumé

Notre projet vise à utiliser l'apprentissage en profondeur pour la détection d'objets, en se concentrant spécifiquement sur la détection de personnes dans une image. Pour entraîner notre modèle, nous avons utilisé deux bases de données provenant de Roboflow. Ensuite, nous avons mis en place une méthode de détection basée sur le deep learning en utilisant l'architecture YOLO (You Only Look Once). Cette architecture nous permettra de détecter les personnes dans une image en utilisant des boîtes englobantes pour les localiser précisément. Nous avons fait des entraînements de notre modèle à travers aux différentes étapes. Notre objectif est de trouver un entraînement pour la détection d'objet avec des mesures de performance plus améliorées.

### Mots clés :

Détection d'objets, Réseaux de neurones convolutifs (CNN), Apprentissage en profondeur (Deep Learning), Yolo...

### ملخص

يهدف مشروعنا إلى استخدام التعلم العميق لاكتشاف الأشياء، مع التركيز بشكل خاص على اكتشاف الأشخاص في الصورة. لتدريب نموذجنا، سنستخدم قاعدتي بيانات من Roboflow. بعد ذلك، سنقوم بتنفيذ طريقة الكشف بناءً على التعلم العميق باستخدام بنية YOLO (أنت تنظر مرة واحدة فقط). ستسمح لنا هذه البنية باكتشاف الأشخاص في صورة ما باستخدام المربعات المحيطة لتحديد موقعهم بدقة. قمنا بتدريب بياناتنا حتى وجدنا معدل دقة كشف أعلى. هدفنا هو السعي للحصول على تدريب أفضل على البيانات لاكتشاف الأشياء بمزيد من مقاييس الأداء المحسنة.

### الكلمات المفتاحية:

اكتشاف الكائنات، الشبكات العصبية التلافيفية (CNN)، التعلم العميق (التعلم العميق)، YOLO...

## Abstract

Our project aims to use deep learning for object detection, focusing specifically on people detection in image. To train our model, we will use two databases from Roboflow. Then, we will implement a detection method based on deep learning using the YOLO (You Only Look Once) architecture. This architecture will allow us to detect people in an image using bounding boxes to locate them precisely. We trained our data until we found a higher detection accuracy rate. Our objective is to seek better data training for object detection with more improved performance measures.

### Key words:

Object detection, Convolutional Neural Networks (CNN), deep learning, YOLO ...

# Table des matières

Résumé.....	5
ملخص .....	5
Abstract .....	5
<b>TABLE DE FIGURE</b>	<b>8</b>
<b>INTRODUCTION GENERALE</b>	<b>9</b>
<b>I. CHAPITRE 01 : IMAGERIE NUMERIQUE ET LE DEEP LEARNING</b>	<b>10</b>
I.1. Introduction .....	10
I.2. Généralité sur intelligence artificielle .....	10
I.3. Image numérique .....	10
I.3.1. Type d'image .....	10
I.3.2. Traitement d'image numérique .....	11
I.4. Apprentissage automatique .....	12
I.4.1. Les méthodes d'apprentissage automatique les plus populaires : .....	12
I.5. Apprentissage profond.....	13
I.5.1. Fonctionnement de l'apprentissage profond .....	13
I.5.2. Applications du l'apprentissage profond.....	13
I.5.3. Pour quoi l'Apprentissage profond.....	14
I.6. Les réseaux de neurones artificiels.....	14
I.6.1. Les composants de réseaux de neurone artificiel .....	14
I.7. Conclusion.....	16
<b>II. CHAPITRE 02 : DETECTION D'OBJET PAR CNN</b>	<b>17</b>
II.1. Introduction .....	17
II.2. Les Réseaux de neurones convolutif.....	17
II.2.1. Architecture d'un réseau de neurone convolutif .....	18
II.3. Détection d'objets.....	19
II.3.1. Qu'est-ce que la détection d'objets .....	19
II.3.2. Types de modèles de détection d'objets .....	20
II.4. Détection d'objet par CNN .....	21
II.4.1. Les algorithmes de détection d'objet basées CNN.....	21
II.4.2. Etat de l'art.....	Erreur ! Signet non défini.

II.5.	Conclusion.....	30
<b>III.</b>	<b>CHAPITRE 3 : CONCEPTION ET IMPLEMENTATION</b>	<b>31</b>
III.1.	Introduction .....	31
III.2.	Mesures de performances.....	31
III.2.1.	Cas de détection.....	31
III.2.2.	Précision moyenne (PA) .....	32
III.2.3.	Mean Average Precision (MAP).....	33
III.2.4.	Rappel .....	33
III.2.5.	Précision .....	33
III.2.6.	Matrice de confusion .....	33
III.2.7.	Intersection over Union (IoU) .....	33
III.3.	L'approche adoptée .....	34
III.4.	L'architecture de notre application .....	34
III.5.	L'algorithme de détection d'objet .....	35
•	Backbone.....	35
•	Model Neck.....	35
•	Model Head .....	35
III.5.1.	YOLOv5 .....	35
III.5.2.	YOLOv7 .....	36
III.6.	Implémentation .....	37
III.6.1.	Environnement logiciel.....	37
III.6.2.	L'environnement matériel.....	37
III.6.2.2.	Préparation des données .....	38
III.6.2.3.	Training .....	39
III.7.	Résultats .....	42
	<b>CONCLUSION</b>	<b>44</b>
	<b>CONCLUSION GENERALE</b>	<b>45</b>
	<b>REFERENCES</b>	<b>46</b>

## Table de figure

Figure 1: la différence entre image matricielles et image vectorielles	11
Figure 2: filtrage d'image	12
Figure 3: comment fonctionne le deep learning	13
Figure 4: Les réseaux de neurones artificiels	14
Figure 5: Les neurons	15
Figure 6: Les poids et biais	15
Figure 7: les fonctions d'activation	16
Figure 8: l'architecture de réseaux de neurones convolutifs.	17
Figure 9: la couche de convolution	18
Figure 10: la différence entre Max pooling et Average pooling	19
Figure 11: La couche entièrement connectée	19
Figure 12: Un exemple de détection d'objet dans une image	20
Figure 13: La Différence entre les détecteurs	21
Figure 14: Architecture de R-CNN	22
Figure 15: Architecture de Fast R-CNN	22
Figure 16: Architecture Faster R-CNN	23
Figure 17: architecture de Mask R-CNN	23
Figure 18: Architecture de SSD	24
Figure 19: Architecture RetinaNet	24
Figure 20: détection d'objet utilisant YOLO	25
Figure 21: Intersection sur Union pour la détection d'Object.	34
Figure 22: Les boîtes d'ancrage	26
Figure 23: exemple de Non-maximal suppression	27
Figure 24 : Exemple de Vrai Positif	31
Figure 25: Exemple de Faux positif	32
Figure 26: Exemple de Vrai négatif	32
Figure 27: Exemple de Faux négatif	32
Figure 28: Architecture de notre système.	35
Figure 29: architecture de YOLOv5	36
Figure 30: Architecture de YOLOv7	36
Figure 31: l'environnement matériel	38
Figure 32: quelque image de la base de donnée 1	38
Figure 33 : Le premier fichier ".yaml"	39
Figure 34 Le deuxième fichier ".yaml"	40
Figure 35: graphe de pert loss	42
Figure 36: matrice de confusion	43
Figure 37: testimg1	43
Figure 38: testimg2	44
Figure 39: testimg3	44



## Introduction générale

La détection d'objets est une technique essentielle de vision par ordinateur qui permet d'identifier et de localiser des objets dans des images ou des vidéos. Elle trouve de nombreuses applications, notamment dans les domaines des voitures autonomes, de la surveillance et de la recherche d'images. En comprenant le contenu des images et des vidéos, les ordinateurs peuvent accomplir une multitude de tâches, de la reconnaissance des panneaux de signalisation et des piétons par les voitures autonomes à la détection des intrusions par les systèmes de sécurité. Dans le domaine médical, la détection d'objets permet le diagnostic de maladies, tandis que dans le commerce de détail, elle facilite la gestion des stocks et l'analyse du comportement des clients.

Le deep learning, une branche de l'apprentissage automatique, offre des opportunités prometteuses pour la détection d'objets. Dans notre projet, nous avons choisi d'utiliser l'algorithme YOLOv5, qui repose sur des réseaux de neurones convolutionnels (CNN) pour détecter les objets dans les images et les vidéos. YOLOv5 se distingue par sa précision et son efficacité, en plus d'être facile à utiliser et de pouvoir être entraîné sur divers ensembles de données. En adoptant YOLOv5, nous avons pu obtenir des résultats précis pour la détection des objets, tout en réduisant le temps et les ressources nécessaires à l'entraînement et à l'inférence.

L'objectif de notre projet de recherche est de mettre en œuvre l'algorithme YOLOv5 pour détecter les personnes dans les images. Nous avons effectué plusieurs entraînements en utilisant différents ensembles de données et évalué les performances du modèle en fonction de mesures telles que la précision, la sensibilité et la spécificité. Notre démarche vise à développer un modèle de détection d'objets extrêmement précis et efficace, pouvant être appliqué à diverses applications.

Notre projet de recherche est divisé en trois chapitres. Le premier chapitre aborde l'imagerie numérique et l'apprentissage approfondi, en fournissant une introduction à l'intelligence artificielle, une explication des images numériques et du traitement d'images, ainsi. Le deuxième chapitre se concentre sur les réseaux neuronaux convolutifs (CNN), leur architecture, les différents types de modèles de détection d'objets, et l'algorithme YOLO. Enfin, le troisième chapitre présente la conception de notre application et l'implémentation de notre programme de détection des personnes avec YOLOv5, en détaillant l'environnement logiciel et matériel, l'architecture des détecteurs à étape unique, les choix d'algorithme, le processus de préparation des données, l'entraînement du modèle et les résultats obtenus. Une conclusion est proposée à la fin de chaque chapitre pour récapituler les points essentiels.

# I. Chapitre 01 : Imagerie numérique et le deep learning

## I.1. Introduction

La détection d'objet utilisant l'apprentissage approfondi est une technique d'IA qui permet d'identifier et de localiser des objets dans une image. Cette méthode utilise des réseaux de neurones profonds pour extraire des caractéristiques de l'image et entraîner un modèle à détecter les objets. Cette technologie a de nombreuses applications dans des domaines tels que la surveillance, la reconnaissance de forme, la conduite autonome.

## I.2. Généralité sur l'intelligence artificielle

L'IA est le terme général désignant la science de l'intelligence artificielle. Il utilise des ordinateurs pour simuler des comportements intelligents humains et il entraîne des ordinateurs à apprendre des comportements humains tels que l'apprentissage, le jugement et la prise de décision. L'IA est un projet de connaissance qui prend la connaissance comme objet, acquiert des connaissances, analyse et étudie les méthodes d'expression des connaissances et utilise ces approches pour obtenir l'effet de simulation des activités intellectuelles humaines.

L'IA est une compilation d'informatique, de logique, de biologie, de psychologie, de philosophie et de nombreuses autres disciplines, et elle a obtenu des résultats remarquables dans des applications telles que la reconnaissance vocale, le traitement d'images, le traitement du langage naturel, la démonstration de théorèmes automatiques et les robots intelligents. [1]

## I.3. Image numérique

Une image numérique est une image acquise, traitée et stockée en bits.

Une image numérique est un tableau de pixel : chaque pixel est codé par un nombre binaire pour un niveau de gris, ou par trois nombres binaires qui correspondent à une nuance de rouge, de vert et de bleu (codage RVB). [2]

Les images numériques peuvent être enregistrées dans différents formats de fichiers, tels que JPEG, PNG, GIF, etc.

### I.3.1. Type d'image

Les formats de l'image numérique sont très divers. On les différencie en deux grandes catégories (matricielle ou vectorielle) qui sont des approches totalement différentes du traitement de l'image. [3]

#### *I.3.1.1. Image vectorielle*

Image vectorielle Est une image numérique composée de plusieurs objets géométriques individuels (droites, polygones, arcs de cercle). L'image vectorielle est créée à partir d'équations

mathématiques. Chaque forme dépend de plusieurs paramètres (hauteur, largeur, rayon) donnés à des vecteurs. [4]

### *1.3.1.2. Une image matricielle (bitmap)*

Est un « objet » numérique constitué de pixels, c'est-à-dire de tous petits carrés de couleur composés d'un mélange de lumière rouge, verte et bleue (également connus sous le nom de sous-pixels). Ces pixels forment une grille statique. [5]

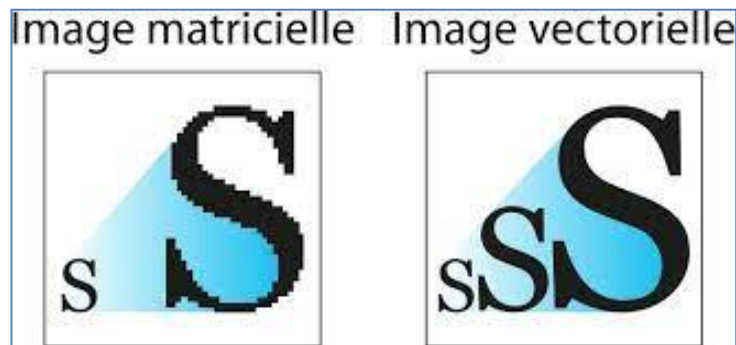


Figure 1: la différence entre image matricielles et image vectorielles

## **1.3.2. Traitement d'image numérique**

Théoriquement, le traitement d'image numérique est un ensemble d'approches, de méthodes, de techniques et d'outils dont l'ambition est de résoudre la majorité des problèmes qui peuvent se présenter lorsqu'il est nécessaire d'extraire et d'analyser de façon automatique les informations présentes dans une image. [6]

### *1.3.2.1. Le prétraitement*

Le prétraitement dans l'image numérique est l'ensemble des opérations ou ensemble d'opérations appliquées à une image avant sa manipulation consistant à préparer l'image en éliminant les bruits, afin de faciliter son traitement ultérieur et d'améliorer la qualité des résultats obtenus. De ce fait, la donnée de départ est l'image initiale et le résultat est également une image.

### *1.3.2.2. Le filtrage*

Le filtrage est une technique de modification ou d'amélioration d'une image. Par exemple, vous pouvez filtrer une image pour mettre en valeur certaines fonctionnalités ou supprimer d'autres fonctionnalités. Les opérations de traitement d'image mises en œuvre avec le filtrage comprennent le lissage, la netteté et l'amélioration des contours.

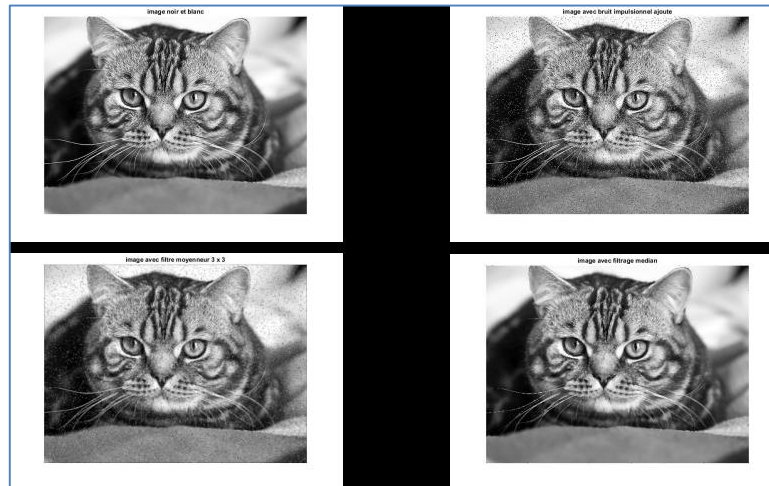


Figure 2: filtrage d'image

### I.4. Apprentissage automatique

L'apprentissage machine est une application d'intelligence artificielle (IA) qui permet aux systèmes d'apprendre et de s'améliorer automatiquement à partir de l'expérience elle-même sans être explicitement programmée. L'apprentissage machine se concentre sur le développement de programmes informatiques qui peuvent accéder à des données et les utiliser pour apprendre par eux-mêmes.

Ce processus d'apprentissage commence par des observations ou des données, comme des exemples, une expérience de première main ou des instructions, dans le but de rechercher des modèles dans les données et de prendre de meilleures décisions à l'avenir sur la base des exemples que nous fournissons. Votre objectif premier est de permettre aux ordinateurs d'apprendre automatiquement sans intervention ou assistance humaine et d'adapter les actions en conséquence.

#### I.4.1. Les méthodes d'apprentissage automatique les plus populaires :

Il existe deux méthodes d'apprentissage automatique largement répandues : l'apprentissage supervisé et l'apprentissage non supervisé.

##### I.4.1.1. Apprentissage supervisé

Le but de ce type d'apprentissage est d'apprendre une fonction qui, compte tenu d'un échantillon de données et de résultats souhaités, se rapproche le mieux de la relation entre les entrées et les sorties observables dans les données. Dans ce cas, on a une connaissance préalable de ce que les valeurs de sortie pour nos échantillons devraient être.

##### I.4.1.2. Apprentissage non supervisé

Les algorithmes sont utilisés lorsque l'information utilisée pour entraîner le modèle n'est ni classifiée ni étiquetée. Le modèle en question étudie ses données d'entraînement dans le but de déduire une fonction pour décrire une structure cachée à partir ces données.

## I.5. Apprentissage profond

Par l'apprentissage approfondie en anglais 'deep learning', on entend un mode d'apprentissage automatique géré par un réseau de neurones artificiels. Le réseau est composé de plusieurs couches dotées de fonctions précises. Le deep learning tend donc vers la simulation du cerveau humain en activant ces couches neuronales qui interagissent et fournissent un apprentissage progressif à partir de grands volumes de données. Comme les neurones du cerveau, les neurones artificiels arrivent donc à communiquer pour favoriser l'apprentissage et l'assimilation de divers éléments d'information.

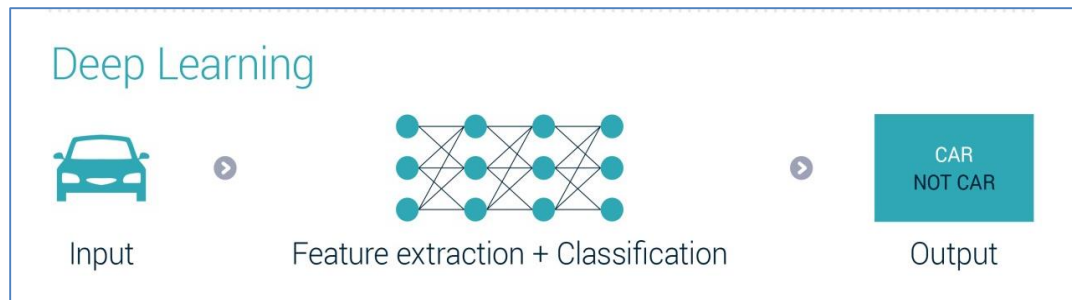


Figure 3: comment fonctionne le deep learning

### I.5.1. Fonctionnement de l'apprentissage profond

Le deep Learning s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain. Ce réseau est composé de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente. Le système apprendra par exemple à reconnaître les lettres avant de s'attaquer aux mots dans un texte, ou détermine s'il y a un visage sur une photo avant de découvrir de quelle personne il s'agit.

À chaque étape, les « mauvaises » réponses sont éliminées et renvoyées vers les niveaux en amont pour ajuster le modèle mathématique. Au fur et à mesure, le programme réorganise les informations en blocs plus complexes. Lorsque ce modèle est par la suite appliqué à d'autres cas, il est normalement capable de reconnaître un chat sans que personne ne lui ait jamais indiqué qu'il n'a jamais appris le concept de chat. Les données de départ sont essentielles : plus le système accumule d'expériences différentes, plus il sera performant.

### I.5.2. Applications du l'apprentissage profond

Les IA à deep learning sont très efficaces pour les analyses d'images. Elles sont, par exemple, employées dans l'imagerie médicale pour détecter des maladies ou dans le secteur automobile dans le cas des voitures autonomes. Mais aussi pour les reconnaissances faciales comme sur les smartphones ou sur Facebook. D'autre part, il est utilisé en sécurité informatique pour identifier les dangers documentés et les risques inconnus. Il est, en effet, capable de détecter des anomalies et de renforcer les mesures de sécurité. [7]

Le Traitement du langage naturel en anglais "Natural Language Processing" est une autre application du Deep Learning. Son but étant d'extraire le sens des mots, voire des phrases pour faire de l'analyse de sentiments.

### I.5.3. Pour quoi l'Apprentissage profond

Les modèles de Deep learning ont tendance à bien fonctionner avec une grande quantité de données alors que les modèles d'apprentissage automatique plus classique cessent de s'améliorer après un point de saturation. Au fil des années, avec l'émergence du big data et de composants informatiques de plus en plus puissant, les algorithmes de Deep Learning gourmands en puissance et en données ont dépassé la plupart des autres méthodes.

## I.6. Les réseaux de neurones artificiels

Les réseaux de neurones artificiels (ANN) sont des algorithmes d'apprentissage en profondeur inspirés du cerveau humain. Ils sont constitués de couches nodales et de neurones artificiels connectés les uns aux autres avec des poids et des seuils. Les données sont transmises à la couche suivante si la sortie du neurone dépasse le seuil spécifié. Les réseaux de neurones sont entraînés avec des données pour améliorer leur précision et sont utilisés pour classer et regrouper des données, comme dans la reconnaissance vocale ou d'images. Un exemple célèbre de réseau de neurones est l'algorithme de recherche de Google. [8]

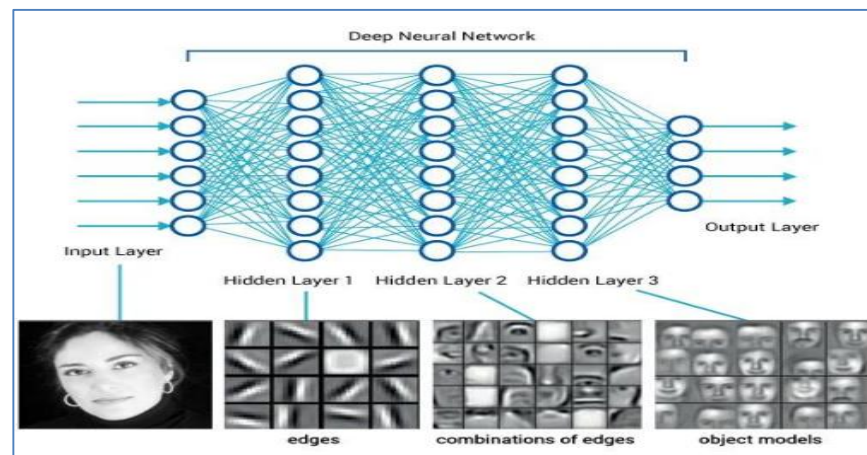


Figure 4: Les réseaux de neurones artificiels

### I.6.1. Les composants de réseaux de neurone artificiel

Un réseau de neurones est composé de trois principaux éléments, détaillés ci-dessous :

#### I.6.1.1. Les neurones

Dans un réseau, le neurone est une unité qui reçoit l'information, procède à des calculs simples, et la transmet à une autre unité. On distingue trois types de neurones dans un réseau artificiel :

- Les neurones d'entrées qui reçoivent les données du monde extérieur ;
- Les neurones de traitement ;
- Les neurones de sorties.

Ils composent respectivement la couche d'entrée (input layer), la couche cachée (hidden layer) et la couche de sortie (output layer). Dans la pratique, un réseau de neurones peut avoir une multitude de couches à chaque niveau : entrée, traitement, sortie... On parlera alors de Deep Learning.

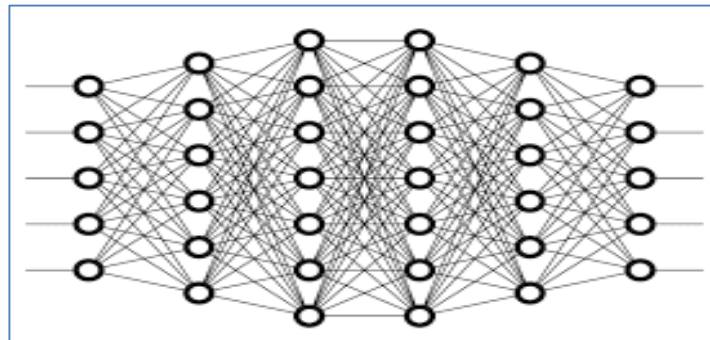


Figure 5: Les neurones

### 1.6.1.2. Les poids et biais

Les connexions entre neurones (biologiquement appelés synapses) dans un réseau artificiel ne sont pas équivalentes. Chaque connexion a en effet un poids spécifique (weight en anglais) qui influence la transmission de l'information d'une unité à une autre. Les informations provenant d'une synapse avec un poids important seront ainsi dominantes à l'entrée du neurone suivant, comparativement à celles provenant de connexions à poids faible. Ces dernières ne sont cependant pas ignorées lors du traitement de l'information. Les poids sont mis à jour de façon continue par les réseaux de neurones artificiels pour améliorer la précision des outputs. Ce sont en réalité des valeurs d'auto-apprentissage.

Dans le même ordre d'idées, les biais sont des paramètres additionnels utilisés dans les réseaux de neurones pour ajuster les valeurs d'entrées auxquelles les poids ont été appliqués, avant l'obtention de valeurs de sorties définitives. Elles influencent le comportement de la fonction d'activation.

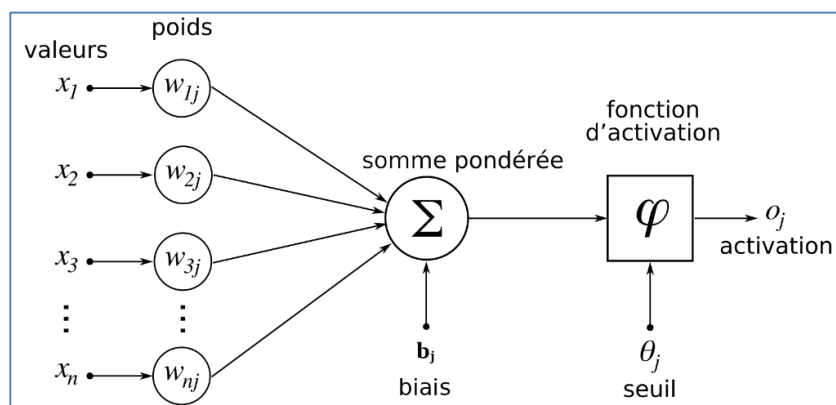


Figure 6: Les poids et biais

1.6.1.3. La fonction d'activation

La fonction d'activation est une fonction mathématique qui permet de déterminer la sortie d'un neurone dans un réseau de neurones. Elle définit la valeur de sortie en fonction de l'entrée et des poids associés aux entrées. Les fonctions d'activation les plus couramment utilisées sont la sigmoïde, la tangente hyperbolique, la fonction ReLU et la softmax. Chacune de ces fonctions d'activation a ses propres avantages et inconvénients en termes de performance et de capacité à traiter les différents types de données. [9] [10]

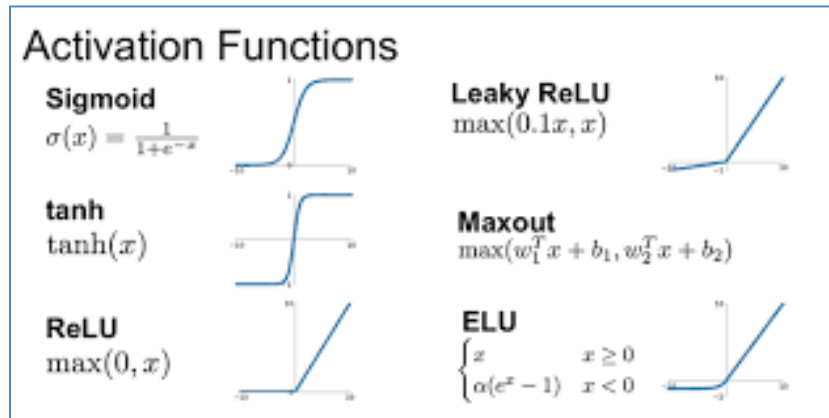


Figure 7: les fonctions d'activation

1.7. Conclusion

En conclusion, la détection d'objet dans les images utilisant l'apprentissage profond est une technologie puissante qui permet de détecter automatiquement des objets dans une image avec une grande précision. Les réseaux de neurones profonds peuvent extraire des caractéristiques de l'image qui ne sont pas visibles à l'œil nu, permettant ainsi de détecter des objets même dans des conditions difficiles. Cette technologie continue d'évoluer rapidement et a des applications potentiellement infinies dans de nombreux domaines.



## II. Chapitre 02 : DETECTION D'OBJET PAR CNN

### II.1. Introduction

La détection d'objets est une tâche importante dans le domaine de l'intelligence artificielle et du traitement de l'image. Les algorithmes de détection d'objets peuvent être utilisés dans de nombreuses applications, telles que la sécurité, la surveillance, la reconnaissance de visages, la reconnaissance de caractères, la détection de personnes et d'animaux, et bien plus encore.

L'objectif de ce chapitre est de fournir une vue d'ensemble des réseaux de neurones convolutionnels (CNN) et de l'algorithme YOLO pour la détection d'objets dans les images. Nous allons examiner les concepts de base des CNN et de l'algorithme YOLO.

### II.2. Les Réseaux de neurones convolutif

Les Réseaux Neuronaux Convolutifs (RNC ou CNN) sont une classe de réseaux de neurones artificiels spécialisés dans le traitement de données en grille, comme les images. Ils utilisent des filtres convolutifs pour extraire des caractéristiques importantes de ces données, ce qui leur permet de reconnaître des objets, des visages ou d'autres motifs. Les RNC ont révolutionné la reconnaissance d'images et ont été appliqués à d'autres domaines tels que la reconnaissance de la parole et la traduction automatique. Ils sont formés à partir de données d'entraînement et leur architecture varie en fonction du problème à résoudre, mais tous les RNC ont une architecture de base de couches de convolution, de couches de pooling et de couches entièrement connectées. [11]

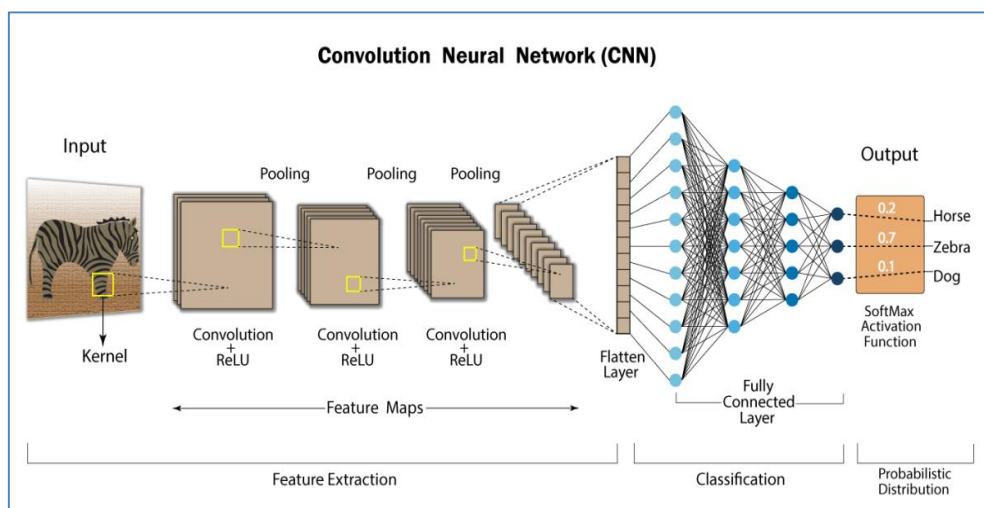


Figure 8: l'architecture de réseaux de neurones convolutifs.

## II.2.1. Architecture d'un réseau de neurone convolutif

### II.2.1.1. La couche de convolution

La couche de convolution est le bloc de construction de base d'un CNN, elle traite les données d'un champ récepteur. Trois paramètres permettent de dimensionner le volume de la couche de convolution la profondeur, le pas et la marge. [12]

- Profondeur : nombre de noyaux de convolution (ou nombre de neurones associés à un même champ récepteur).
- Le pas : contrôle le chevauchement des champs récepteurs. Plus le pas est petit, plus les champs récepteurs se chevauchent et plus le volume de sortie sera grand
- La marge (à 0) : mettre des zéros à la frontière du volume d'entrée. Cette marge permet de contrôler la dimension

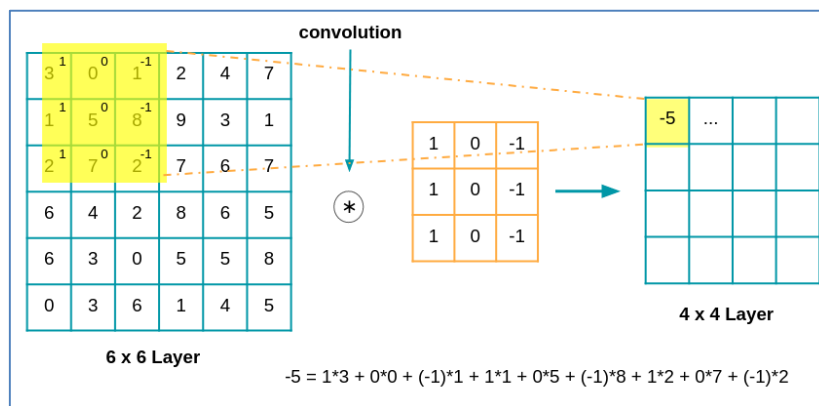


Figure 9: la couche de convolution

### II.2.1.2. Couche de mise en commun (Pooling Layer)

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling. L'opération de pooling consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes. Pour cela, on découpe l'image en cellules régulières, puis on garde au sein de chaque cellule la valeur maximale. [13]

Il existe deux types de pooling :

- Le Maximum Pooling : Chaque opération de pooling sélectionne la valeur maximale de la surface.
- L'Average Pooling : Chaque opération de pooling sélectionne la valeur moyenne de la surface.

Le maximum pooling est le plus utilisé surtout pour la reconnaissance des objets de grandes tailles et complexes. [14]

Type	Max pooling	Average pooling
But	Chaque opération de pooling sélectionne la valeur maximale de la surface	Chaque opération de pooling sélectionne la valeur moyenne de la surface
Illustration		
Commentaires	<ul style="list-style-type: none"> <li>• Garde les caractéristiques détectées</li> <li>• Plus communément utilisé</li> </ul>	<ul style="list-style-type: none"> <li>• Sous-échantillonne la <i>feature map</i></li> <li>• Utilisé dans LeNet</li> </ul>

Figure 10: la différence entre Max pooling et Average pooling

### II.2.1.3. La couche entièrement connectée (Fully-Connected)

La couche entièrement connectée ressemble à un réseau neuronal régulier connectant tous les neurones et forme les dernières couches du réseau. La sortie de la couche d'aplatissement est envoyée à cette couche entièrement connectée.

Le vecteur de caractéristiques de la couche entièrement connectée est en outre utilisé pour classer les images entre différentes catégories après l'entraînement. Toutes les entrées de cette couche sont connectées à chaque unité d'activation de la couche suivante. Étant donné que tous les paramètres sont occupés dans une couche entièrement connectée, cela provoque un sur ajustement. Le décrochage fait partie des techniques qui réduisent le sur apprentissage. [15]

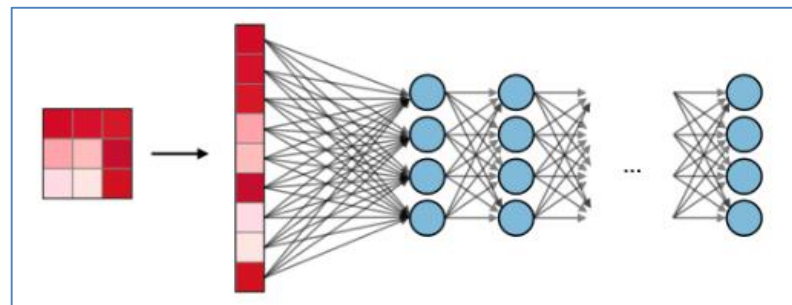


Figure 11: La couche entièrement connectée

## II.3. Détection d'objets

### II.3.1. Qu'est-ce que la détection d'objets

On entend par détection d'objets en informatique et plus spécifiquement en géomatique, l'ensemble des techniques et moyens pour détecter, identifier et classier des objets dans une image. La détection d'objets s'appuie sur deux étapes intermédiaires sont la localisation d'objets et la classification d'image.

La première consiste à trouver la position d'un ou plusieurs objets dans une image et de dessiner leurs contours c'est comme si on déterminait les coordonnées de chaque pixel de l'objet sur l'image.

Quant à la seconde, elle consiste à prédire et affecter à chaque objet une classe. Lors des travaux en télédétection, il est possible que vous ayez déjà eu à réaliser de la détection d'objet pendant les

classifications d'images. Avec le deep learning la détection d'objet est devenue très populaire dans des sphères autres que la géomatique, avec plusieurs familles de modèles (R-CNN, YOLO, etc.). [16]



Figure 12: Un exemple de détection d'objet dans une image

### II.3.2. Types de modèles de détection d'objets

#### II.3.2.1. Détecteur à un étage

Un détecteur à une étage (one-stage detector) également appelé détecteur à étage unique, est un type de modèle de détection d'objets en vision par ordinateur qui prédit directement les coordonnées des boîtes englobantes (bounding boxes) et les étiquettes de classe en une seule étape. Les détecteurs à un étage utilisent un seul réseau neuronal pour traiter une image d'entrée et générer les coordonnées des boîtes englobantes, ainsi que les probabilités de classe, pour plusieurs objets présents dans l'image. Des exemples de détecteurs à un étage populaire incluent You Only Look Once (YOLO) et Single Shot MultiBox Detector (SSD). Les détecteurs à une étape sont connus pour leur vitesse d'inférence en temps réel, ce qui les rend adaptés aux applications où la latence de traitement est cruciale

#### II.3.2.2. Détecteur à deux étages

Un détecteur à deux étages (two-stage detector), également appelé pipeline à deux étages ou approche à deux étages, est un type de modèle de détection d'objets qui implique deux étapes principales : la génération de propositions de régions (region proposals) et la classification des objets. Dans la première étape, le modèle génère des propositions de régions, qui sont des propositions de boîtes englobantes potentielles pour les objets dans l'image. Dans le deuxième étage, les propositions de régions sont affinées et classifiées en différentes classes d'objets. Des exemples de détecteurs à deux étapes populaires incluent Faster R-CNN (Region-based Convolutional Neural Network) et Mask R-CNN. Les détecteurs à deux étages sont connus pour leur haute précision et leur robustesse, car ils génèrent généralement un grand nombre de propositions de régions et les affinent soigneusement, ce qui permet d'obtenir une localisation précise des objets et une classification précise.

### II.3.2.3. La Différence entre les détecteurs

La principale différence entre les détecteurs à un étage et à deux étages réside dans leur approche pour générer des propositions d'objets et prédire les boîtes englobantes. Les détecteurs à un étage prédisent directement les boîtes englobantes et les étiquettes de classe en une seule étape, tandis que les détecteurs à deux étages utilisent un processus en deux étapes de génération de propositions de régions et d'affinement pour la classification des objets.

Les détecteurs à un étage ont tendance à avoir des vitesses d'inférence plus rapides, mais peuvent sacrifier une certaine précision par rapport aux détecteurs à deux étages, qui ont généralement une meilleure précision mais des vitesses d'inférence plus lentes. Le choix entre les détecteurs à un étage et à deux étages dépend des exigences spécifiques de l'application, telles que la nécessité d'un traitement en temps réel ou d'une haute précision, ainsi que des ressources informatiques disponibles. [17]

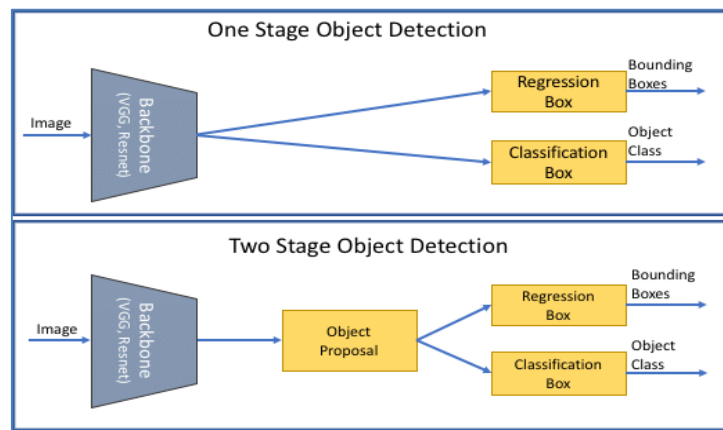


Figure 13: La Différence entre les détecteurs

## II.4. Détection d'objet par CNN

### II.4.1. Les algorithmes de détection d'objet basés CNN

Il existe plusieurs algorithmes de détection d'objets basés sur les réseaux de neurones convolutifs (CNN) qui ont été développés et utilisés avec succès. Voici quelques-uns des algorithmes les plus populaires.

#### II.4.1.1. R-CNN (Region-based Convolutional Neural Networks)

C'est l'un des premiers algorithmes à avoir introduit la détection d'objets basée sur les CNN. Il propose de générer des régions d'intérêt (ROIs) à l'aide d'une méthode de proposition de région (comme Selective Search) et d'appliquer un CNN à chaque ROI pour extraire les caractéristiques. Ensuite, un classificateur est utilisé pour prédire les classes des objets et une régression est effectuée pour ajuster les boîtes englobantes. [18]

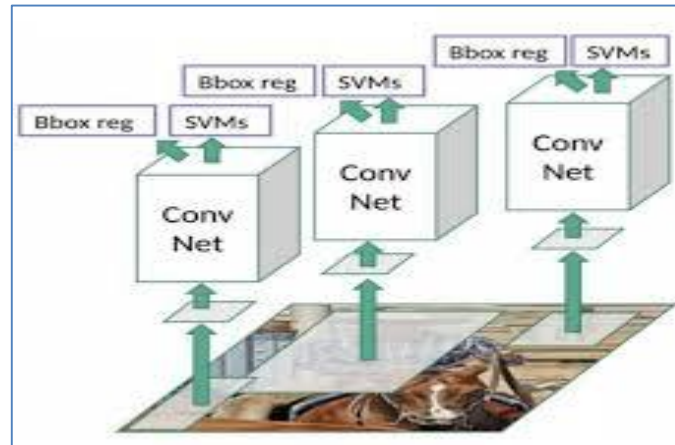


Figure 14: Architecture de R-CNN

#### II.4.1.2. Fast R-CNN

Fast-R-CNN est le premier successeur de R-CNN, conservant la plupart des notions fondamentales de R-CNN et introduisant peu de raffinements. Dans Fast R-CNN, le CNN principal avec plusieurs couches convolutives est appliqué à l'image pour la détection des caractéristiques avant de proposer des régions. Cela signifie qu'il ne nécessite pas plusieurs CNN sur plusieurs régions qui se chevauchent. Le SVM est également remplacé par une couche Softmax, étendant ainsi le réseau neuronal pour les prédictions au lieu de créer un nouveau modèle. Un autre raffinement a été la couche de mise en commun des régions d'intérêt (RoI). [19]

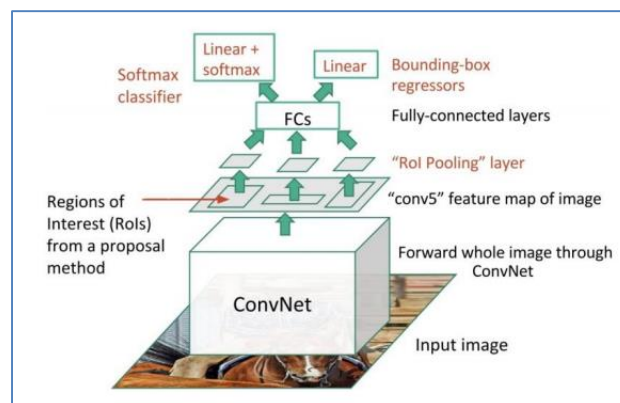


Figure 15: Architecture de Fast R-CNN

#### II.4.1.3. Faster R-CNN

Faster R-CNN développé par Shaoqing Ren et al. en 2016, est le successeur de Fast RCNN. Il est le plus couramment utilisé pour la détection d'objets tridimensionnels, segmentation, etc. Théoriquement, le Faster RCNN se compose de 3 réseaux neuronaux tels que le réseau de fonctionnalités (Feature Network), le réseau de proposition régionale (Regional Proposal Network) et le réseau de détection (Detection Network). La figure 16 représente l'architecture de l'algorithme Faster RCNN. [20]

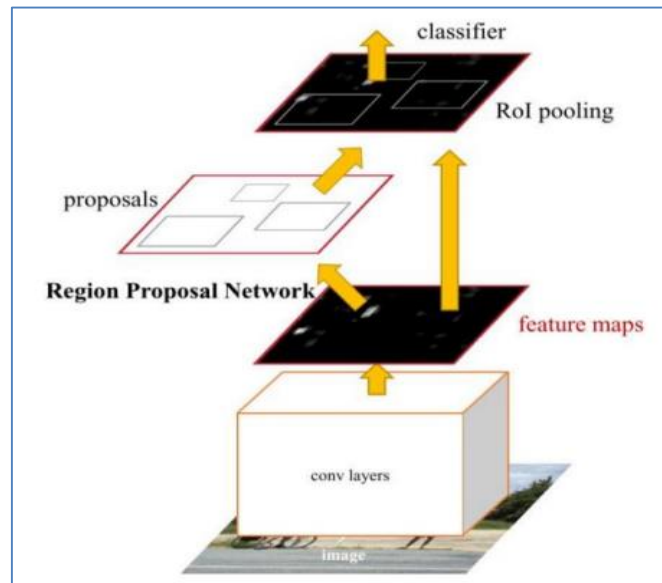


Figure 16: Architecture Faster R-CNN

#### II.4.1.4. Mask R-CNN

Le Mask R-CNN peut être considéré comme une extension du Faster R-CNN. Il ajoute une petite surcharge au Faster R-CNN en ajoutant une branche supplémentaire pour prédire un masque d'objet en parallèle avec les branches existantes pour la classification et la régression de la boîte englobante.

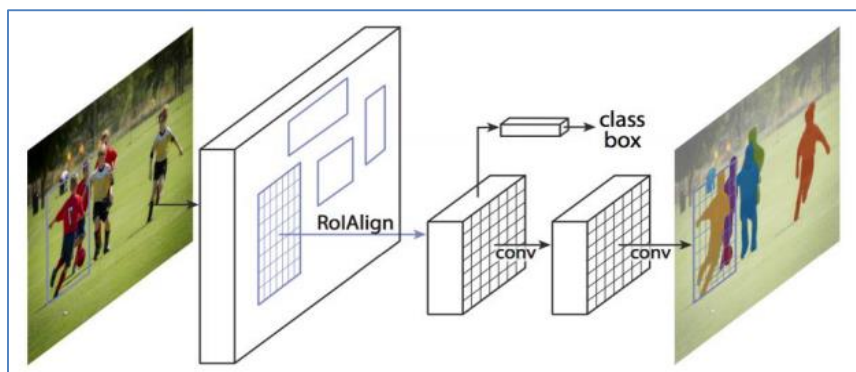


Figure 17: architecture de Mask R-CNN

#### II.4.1.5. SSD

Le SSD en anglais Single Shot MultiBox Detector présenté par Liu et al, qui est capable de fournir des performances en temps réel avec une grande précision. Ce réseau n'utilise pas la méthode de proposition régionale. Dans ce réseau, la localisation et la classification des objets sont effectuées en un seul passage allé du réseau tout en utilisant une technique connue sous le nom de «MultiBox» pour effectuer la régression de la boîte englobant (bounding box). Le SSD est donc capable d'effectuer des calculs end-to-end. [21]



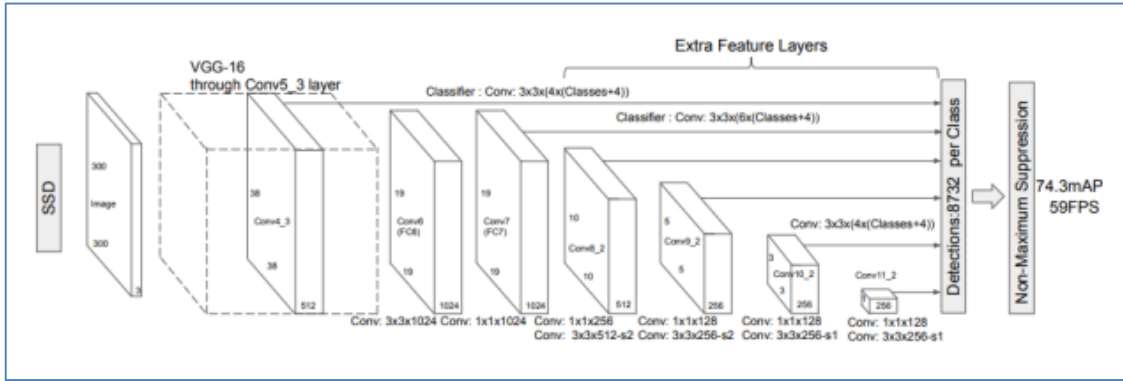


Figure 18: Architecture de SSD

II.4.1.6. RetinaNet

RetinaNet est un détecteur d'objet à un seul stage avec un backbone et deux sous-réseaux, un pour la classification et un pour la régression. Le backbone génère les feature map convolutives sur une image d'entrée entière similaire à Faster R-CNN. RetinaNet utilise un Feature Pyramid Network (FPN) construit sur ResNet comme backbone. Le subnet de classification est chargé de prédire la probabilité de présence d'objet à chaque position spatiale pour chacune des ancrés A et K classes d'objets. Il prend une carte d'entités d'entrée avec des canaux C à partir d'un niveau de pyramide et applique quatre couches de convolution 3x3, chacune avec des filtres C et chacune suivie par des activations ReLU. Enfin, les activations sigmoïdes sont attachées aux sorties. [22]

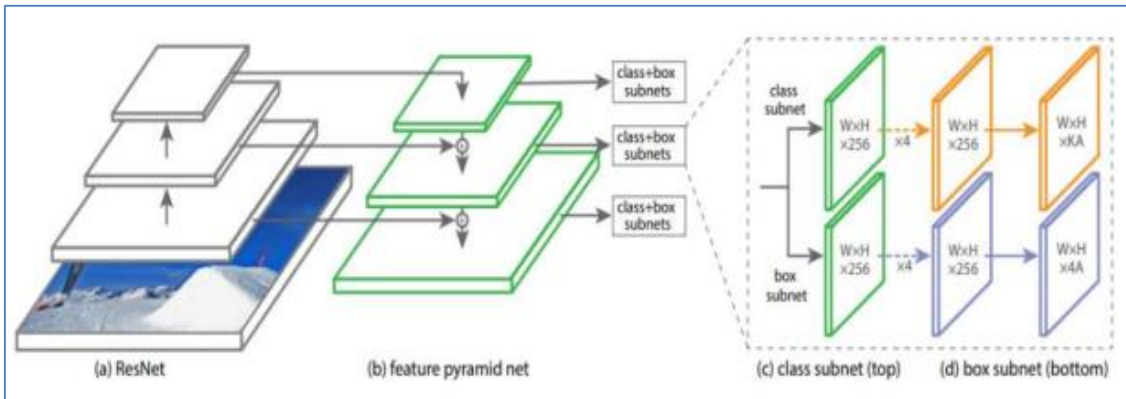


Figure 19: Architecture RetinaNet

II.4.1.7. YOLO

YOLO signifie "You Only Look Once" et est un système de détection d'objets en temps réel. Il s'agit d'un algorithme d'apprentissage profond qui peut détecter des objets dans des images et des vidéos, et les étiqueter avec leur classe respective (comme "personne", "voiture", "chien", etc.) et leur emplacement dans l'image ou la vidéo.

YOLO diffère des algorithmes de détection d'objets traditionnels en ce qu'il utilise un seul réseau neuronal pour effectuer des prédictions directement à partir des entrées de pixels brutes, plutôt que d'utiliser des modèles distincts pour la proposition et la classification d'objets. Cela le rend plus rapide



et plus efficace que les autres méthodes de détection d'objets, ce qui le rend bien adapté aux applications en temps réel telles que la conduite autonome, la surveillance vidéo et la robotique.

### II.4.1.7..1. Comment fonctionne l'algorithme YOLO

Notre système divise l'image d'entrée en une grille  $S \times S$ .

Si le centre d'un objet tombe dans une cellule de la grille, cette cellule de la grille est responsable de la détection de cet objet.

Chaque cellule de la grille prédit  $B$  boîtes englobantes ( $B$  bounding boxes) et les scores de confiance (confidence scores) pour ces boîtes. Ces scores de confiance reflètent la confiance du modèle dans le fait que la boîte contient un objet et également la précision avec laquelle il pense que la boîte prédit. Formellement, nous définissons la confiance comme  $\text{Pr}(\text{Object}) * \text{IOU}_{\text{truth pred}}$ .

Si aucun objet n'existe dans cette cellule, les scores de confiance doivent être nuls. Sinon, nous voulons que le score de confiance soit égal à l'intersection sur l'union (IOU) entre la boîte prédite et la vérité terrain.

Chaque boîte englobant se compose de 5 prédictions :  $x$ ,  $y$ ,  $w$ ,  $h$  et confiance. Les coordonnées  $(x, y)$  représentent le centre de la boîte par rapport aux limites de la cellule de la grille. La largeur et la hauteur sont prédites par rapport à l'image entière. Enfin, la prédiction de confiance représente l'intersection sur union (IOU) entre la boîte prédite et toute boîte de vérité terrain.

Ces prédictions sont encodées sous la forme :  $S \times S \times (B * 5 + C)$  tensor

Tel que  $5$  c'est  $x$  et  $y$ ,  $w$ ,  $h$  et confiance et  $C$  c'est le nombre des classes. [23]

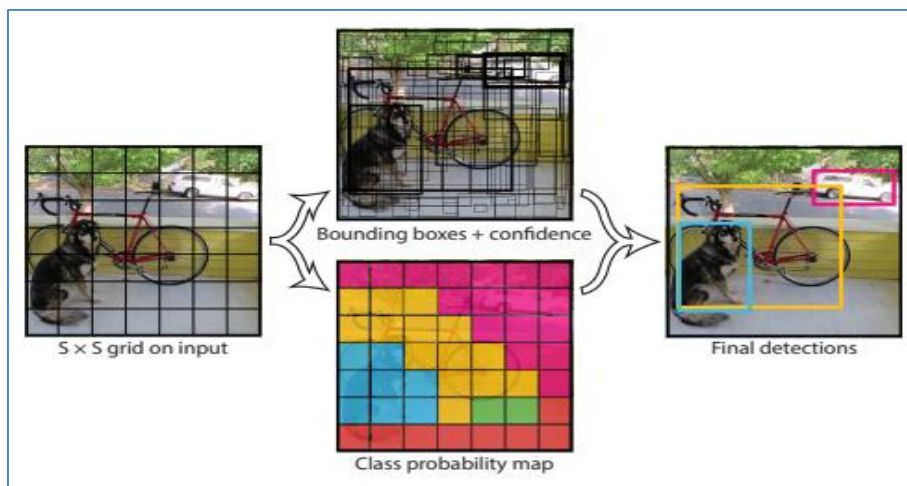


Figure 20: détection d'objet utilisant YOLO

### II.4.1.7..2. Les boîtes d'ancrage (Anchor boxes)

YOLO peut bien fonctionner pour la détection de plusieurs objets, où chaque objet est associé à une cellule de grille. Cependant, dans le cas de chevauchements, où une cellule de grille contient en réalité les points centraux de deux objets différents, on peut utiliser quelque chose appelé "boîtes d'ancrage" pour permettre à une cellule de grille de détecter plusieurs objets. [24]

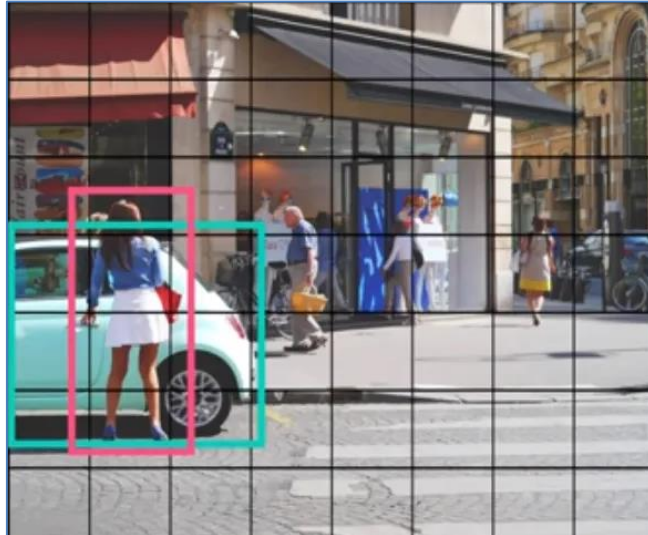


Figure 21: Les boîtes d'ancrage

Les boîtes d'ancrage sont un ensemble de boîtes englobantes prédéfinies d'une certaine hauteur et largeur.

Ces boîtes sont conçues pour capturer l'échelle et le rapport d'aspect des classes d'objets spécifiques que vous souhaitez détecter, et sont généralement choisies en fonction de la taille des objets dans vos jeux de données d'entraînement.

Dans l'image ci-dessous, nous voyons que nous avons une personne et une voiture qui se chevauchent dans l'image. Puisque nous détectons une voiture large et une personne debout, nous définirons une boîte d'ancrage qui a à peu près la forme d'une voiture, cette boîte sera plus large que haute. Et nous définirons une autre boîte d'ancrage pouvant contenir une personne debout à l'intérieur, qui sera plus haute que large. Puisque nous utilisons deux boîtes d'ancrage, nous obtiendrons deux boîtes d'ancrage prédites pour chaque cellule de la grille. Certaines, en fait la plupart des boîtes d'ancrage prédites auront une valeur PC (probabilité qu'un objet y soit présent) très faible.

Après avoir produit ces vecteurs de sortie, nous utilisons une non-maximal suppression pour nous débarrasser des boîtes englobantes improbables. Pour chaque classe, la non-maximal suppression élimine les boîtes englobantes qui ont une valeur PC inférieure à un seuil donné. [25]

### II.4.1.7..3. Non-maximal suppression

YOLO utilise la non-maximal suppression (NMS) pour ne conserver que la meilleure boîte englobantes.

- La première étape dans NMS consiste à supprimer toutes les boîtes englobantes prédites qui ont une probabilité de détection inférieure à un seuil NMS (NMS threshold) donné. Par exemple, nous définissons le seuil NMS sur 0,6. Cela signifie que toutes les boîtes englobantes prédites qui ont une probabilité de détection inférieure à 0,6 seront supprimées.

- la deuxième étape consiste à sélectionner les boîtes englobantes avec la probabilité de détection la plus élevée et à éliminer toutes les boîtes englobantes dont la valeur d'intersection sur l'union (IOU) est supérieure à un seuil IOU donné car il y a une similarité. Par exemple, nous fixons ce seuil IOU à 0,4. Cela signifie que toutes les boîtes englobantes prévues qui ont une valeur IOU supérieure à 0,4 par rapport aux meilleures boîtes englobantes seront supprimées.

Il répétera cela jusqu'à ce que toutes les boîtes englobantes non-maximales aient été supprimées pour chaque classe. [26]

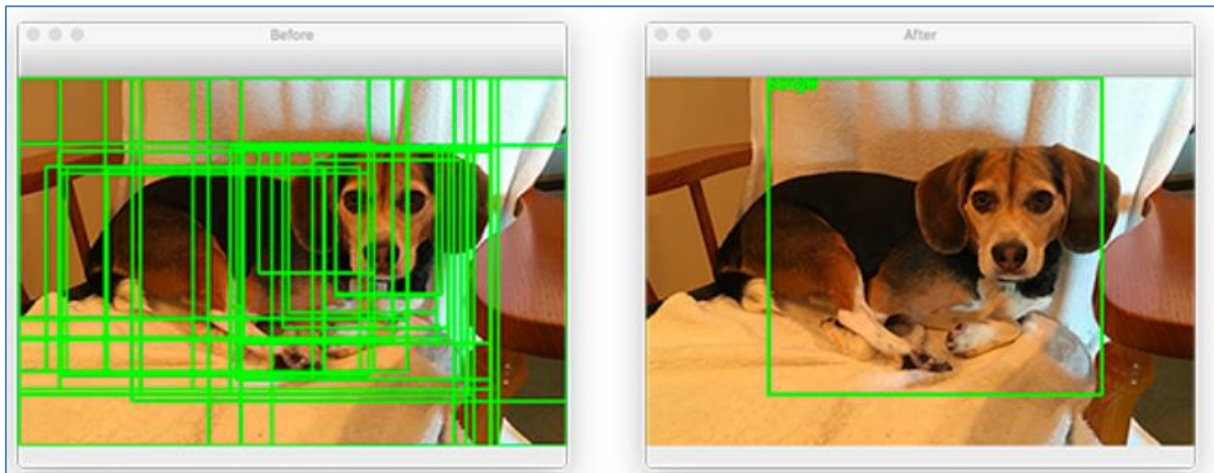


Figure 22: exemple de Non-maximal suppression

### II.4.2. Etat de l'art

La détection d'objets dans les images à l'aide de réseaux de neurones convolutifs (CNN) est un domaine de recherche en pleine expansion. De nombreux chercheurs se sont penchés sur cette question en proposant des solutions pour améliorer la précision de la détection d'objets. Plusieurs articles récents ont été publiés sur ce sujet, se concentrant notamment sur l'utilisation de CNN en combinaison avec des techniques telles que les réseaux pré-entraînés et l'augmentation de données. Dans cette partie, nous allons présenter quelques travaux récents sur la détection d'objet par CNN :

#### **L'article(1) : Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks [27]**

Dans ce travail, les auteurs introduisent un réseau de proposition de région (RPN) qui partage des caractéristiques convolutives d'image complète avec le réseau de détection, permettant ainsi des propositions de région presque gratuites. Un RPN est un réseau entièrement convolutif qui prédit simultanément les limites d'objet et les scores d'objectivité à chaque position. Les RPN sont formés de bout en bout pour générer des propositions de région de haute qualité, qui sont utilisées par Fast R-CNN pour la détection. ce système de détection a une fréquence d'images de 5 ips (y compris toutes les étapes) sur un GPU, tout en atteignant une précision de détection d'objets de pointe sur PASCAL VOC 2007 (73,2 % mAP) et 2012 (70,4 % mAP) en utilisant 300 propositions par image.

### **L'article(2) : Deep Residual Learning for Image Recognition [28]**

Les auteurs présentent un cadre d'apprentissage résiduel pour faciliter la formation de réseaux qui sont sensiblement plus profonds que ceux utilisés précédemment. Ils reformulent explicitement les couches comme des fonctions résiduelles d'apprentissage en référence aux entrées de la couche, au lieu d'apprendre des fonctions non référencées. Ils fournissent des preuves empiriques complètes montrant que ces réseaux résiduels sont plus faciles à optimiser et peuvent gagner en précision à partir d'une profondeur considérablement accrue. Sur l'ensemble de données ImageNet. Les auteurs évaluent les filets résiduels avec une profondeur allant jusqu'à 152 couches - 8 × plus profondes que les filets VGG mais ayant toujours une complexité moindre. Un ensemble de ces filets résiduels atteint une erreur de 3,57 % sur l'ensemble de test ImageNet. Ce résultat a remporté la 1ère place sur la tâche de classification ,ils obtiennent une amélioration relative de 28% sur l'ensemble de données de détection d'objets COCO.

### **L'article (3) : Détection des anomalies par segmentation des images médicales cérébrales [29]**

A travers ce projet, les auteurs s'intéressent à la segmentation des IRM cérébrales pour détecter les anomalies (pathologies/maladies) existantes au sein de ces images. Le but du projet est d'implémenter un algorithme qui nous permet de détecter les anomalies dans une image IRM cérébrale. Ils ont utilisé les techniques de l'apprentissage approfondi pour la détection des tumeurs cérébrales dans les images IRM. ils ont implémenté un modèle appelé VGG16 en utilisant la technique de transfert d'apprentissage dans un environnement cloud. Des résultats d'apprentissage et de test satisfaisants ont été obtenus.

### **L'article (4) : Détection de tumeur cérébrale avec l'apprentissage profond [30]**

Ce travail décrit la détection des tumeurs cérébrales à l'aide des réseaux de neurones convolutifs « CNN ». La détection des tumeurs cérébrales est l'objectif de ce travail pour créer ce système afin de sauver la vie des patients. Donc, ce système se base sur trois étapes nécessaires à savoir : Prétraitement d'images, extraction de caractéristiques avec le modèle CNN. Et enfin, Les couches entièrement connectées de CNN pour classifier l'IRM en normale ou tumorale.

Nous allons comparer ces articles en termes d'objectifs, de méthodes, de résultats et de limitations comme illustré dans le tableau 1

Les articles	Descriptions	Les Avantages	Les Inconvénients
<b>Article 1</b>	<ul style="list-style-type: none"> <li>Cet article présente une nouvelle méthode de détection d'objets en temps réel, qui utilise des RPN pour générer des propositions de régions d'intérêt et des réseaux de neurones convolutifs pour classer et localiser les objets.</li> </ul>	<ul style="list-style-type: none"> <li>Cette méthode est plus rapide et plus précise que les méthodes Précédentes.</li> </ul>	<ul style="list-style-type: none"> <li>Temps de calcul élevé</li> <li>Difficulté à détecter de petits objets</li> </ul>
<b>Article 2</b>	<ul style="list-style-type: none"> <li>Cet article présent les réseaux ResNet atteignent des performances supérieures à celles des autres architectures de réseaux de neurones profonds pour la reconnaissance d'images sur plusieurs ensembles de données standard</li> </ul>	Cette méthode permet d'entraîner des réseaux de neurones profonds (avec plusieurs couches) plus facilement qu'auparavant. permettant de sauter des couches entières et de propager l'erreur plus facilement.	<ul style="list-style-type: none"> <li>Temps de calcul élevé</li> <li>Besoin d'un grand ensemble de données d'entraînement</li> </ul>
<b>Article 3</b>	<ul style="list-style-type: none"> <li>Cet article présent Le modèle VGG16 des réseaux de neurones convolutionnels (CNN) pour la classification d'images, a été implémenté pour classifier des images IRM cérébrales en distinguant celles qui sont saines de celles qui sont malades</li> </ul>	Sa capacité à extraire des caractéristiques précises, son pré-entraînement sur de nombreuses images, et sa large utilisation dans la communauté de la vision par ordinateur.	<ul style="list-style-type: none"> <li>La complexité du modèle, le risque de sur apprentissage, et la difficulté d'interpréter les caractéristiques extraites.</li> </ul>
<b>Article 4</b>	Cet article présent les méthodes actuelles de détection de tumeurs cérébrales et les techniques de traitement d'images médicales associées. Par des méthodes basées sur les seuils, les méthodes de segmentation, les méthodes de classification et les méthodes de détection automatique de tumeurs. concepts clés de l'apprentissage profond et de l'utilisation de réseaux de neurones convolutifs (CNN) pour la classification d'images médicales et ces différentes couches notamment les couches de convolution, de sous-échantillonnage et de sortie, ainsi que la fonction d'activation et la rétro propagation.	<ul style="list-style-type: none"> <li>-Obtenir des résultats très précis.</li> <li>-cette méthode peut traiter des volumes de données plus importants en un temps plus court que les méthodes traditionnelles.</li> <li>-Elle peut être adapté à différents types d'imagerie cérébrale</li> </ul>	<ul style="list-style-type: none"> <li>-Il faut généralement un grand ensemble de données d'imagerie cérébrale.</li> <li>-Les algorithmes de détection de tumeurs cérébrales basés sur l'apprentissage profond peuvent être complexes et nécessiter une expertise en matière de programmation et de modélisation de données.</li> <li>L'algorithme est difficiles à interpréter.</li> </ul>

Tableau 1: Tableau comparatif pour les quatre articles

## **II.5. Conclusion**

En conclusion, les réseaux de neurones convolutifs ont transformé la vision par ordinateur en permettant aux machines de comprendre le contenu des images et des vidéos de manière plus précise que jamais. YOLO est un exemple d'algorithme de détection d'objets qui utilise les avantages des CNN pour une détection rapide et précise d'objets dans des images en temps réel. Ces avancées sont importantes car elles ouvrent la voie à des applications pratiques telles que la détection d'objets dans les véhicules autonomes, la surveillance vidéo, la reconnaissance faciale et de nombreuses autres applications.

## III. Chapitre 3 : Conception et implémentation

### III.1. Introduction

Dans ce chapitre, nous présenterons les étapes de mise en œuvre de notre modèle d'expérimentation et nous présenterons les résultats obtenus. Pour cela, nous allons en premier lieu présenter un bref aperçu des mesures de performances et de l'environnement de développement utilisés dans notre travail, puis les étapes suivies pour la conception et l'implémentation ainsi que les résultats obtenus. De plus, afin de sélectionner le meilleur résultat possible, nous avons effectué plusieurs entraînements du modèle en variant certains paramètres clés. Cela nous a permis d'évaluer les performances du modèle dans différentes configurations et de choisir la meilleure solution. Nous discuterons en détail de ces différentes étapes et des critères utilisés pour sélectionner le résultat optimal.

### III.2. Mesures de performances

#### III.2.1. Cas de détection

##### III.2.1.1. Vrai positif (TP)

Nombre de cas que le test déclare positifs et qui le sont réellement.  
Détection l'objet alors qu'il se trouve dans le cadre photo.



Figure 23 : Exemple de Vrai Positif

##### III.2.1.2. Faux positif (FP)

Nombre de cas que le test déclare positifs et qui sont en réalité.



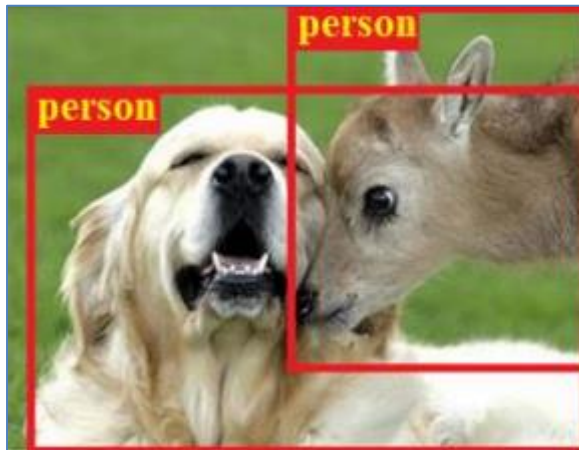


Figure 24: Exemple de Faux positif

### III.2.1.3. Vrai négatif (VN)

Nombre de cas que le test déclare négatifs et qui sont en réalité négatifs.



Figure 25: Exemple de Vrai négatif

### III.2.1.4. Faux négatif (FN)

Nombre de cas que le test déclare négatifs et qui sont en réalité positifs. [31]



Figure 26: Exemple de Faux négatif

## III.2.2. Précision moyenne (PA)

Il est calculé à l'aide de l'aire sous la courbe (AUC) de la courbe Précision x Rappel. En faisant la moyenne de la précision de toutes les valeurs de rappel entre 0 et 1.



### III.2.3. Mean Average Precision (MAP)

Le compare la boîte réelle à la boîte détectée et renvoie un score. Plus le score est élevé, plus le modèle est précis dans ses détections.

### III.2.4. Rappel

Mesure à quel point vous trouvez tous les points positifs. Par exemple, nous pouvons trouver 80% des cas positifs possibles dans nos prédictions top K.

$$\text{Rappel} = TP / (TP + FN)$$

### III.2.5. Précision

Le pourcentage de vos prédictions est correct.

$$\text{Précision} = TP / (TP + FP)$$

### III.2.6. Matrice de confusion

La matrice de confusion est un outil essentiel pour évaluer les performances d'un modèle de classification en comparant les prédictions du modèle avec les valeurs réelles [32] [33]. Elle met en évidence les prédictions correctes et incorrectes et permet de mesurer la performance d'un algorithme de Machine Learning [33]. Elle se présente sous la forme d'un tableau qui compare les prévisions et les résultats des tests avec les valeurs réelles.

### III.2.7. Intersection over Union (IoU)

L'intersection sur l'union est une métrique populaire pour mesurer la précision de la localisation et calculer les erreurs de localisation dans les modèles de détection d'objets.

Pour calculer l'IoU entre les boîtes englobantes prédites et celles de la réalité terrain, nous prenons d'abord la zone d'intersection entre les deux boîtes englobantes correspondantes pour le même objet. Ensuite, nous calculons la surface totale couverte par les deux boîtes englobantes, également appelées "Union" et la zone de chevauchement entre elles appelée "Intersection". Il est calculé comme suit :

$$\text{IoU} = \text{zone de chevauchement} / \text{zone d'union}$$

IoU varie entre 0 et 1, où 0 n'indique aucun chevauchement et 1 signifie un chevauchement parfait entre gt et pd. [34]

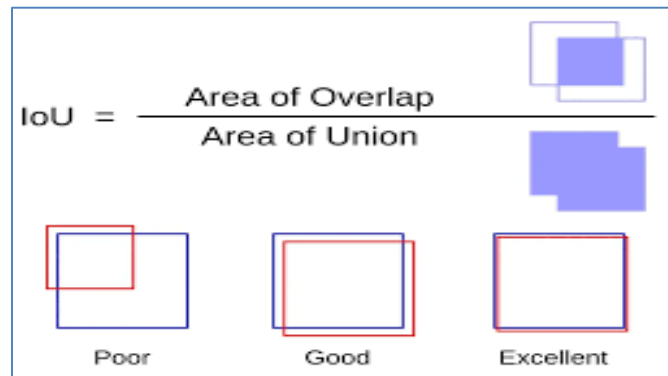


Figure 27: Intersection sur Union pour la détection d'Object.

On utilise l'IoU :

- comme mesure de la qualité des boîtes englobantes prévues
- dans le calcul des pertes (YOLO Version 2,3)
- dans l'affectation de la boîte d'ancrage à la boîte englobant de l'objet
- dans la sélection de la boîte BEST Predicted (dans l'algorithme de suppression non max.)

### III.3. L'approche adoptée

Dans ce travail nous avons appliqué un algorithme de détection d'objet sur une image d'entrée, la classe de l'objet choisi est "personne". L'objectif de notre approche est d'améliorer le taux de détection. Pour ce faire, nous avons apporté quelques modifications à la base de données (augmenter la taille de la base de données), ainsi que des modifications aux hyperparamètres du modèle d'apprentissage choisi dans ce travail. Nous avons utilisé le métrique map50 et d'autres paramètres pour mesurer les performances de détection.

### III.4. L'architecture de notre application

L'architecture de notre système de détection de personnes dans une image est résumée ci-dessous (figure 28), nous avons disposé d'un ensemble de données d'images contenant des personnes. Ces images doivent être annotées avec des boîtes englobantes indiquant l'emplacement des personnes dans chaque image. Une fois que nous avons un jeu de données annoté, nous pouvons appliquer des techniques d'augmentation d'images, pour augmenter la diversité des exemples d'entraînement. Ensuite, prétraitement des images en les redimensionnant, en normalisant les valeurs des pixels, etc. Puis nous sommes entraînés notre modèle YOLO en utilisant cet ensemble de données prétraitées et fournies les performances du modèle en utilisant des mesures telles que la précision, le rappel, etc. L'objectif est d'atteindre un modèle entraîné capable de détecter efficacement les personnes dans les images fournies.

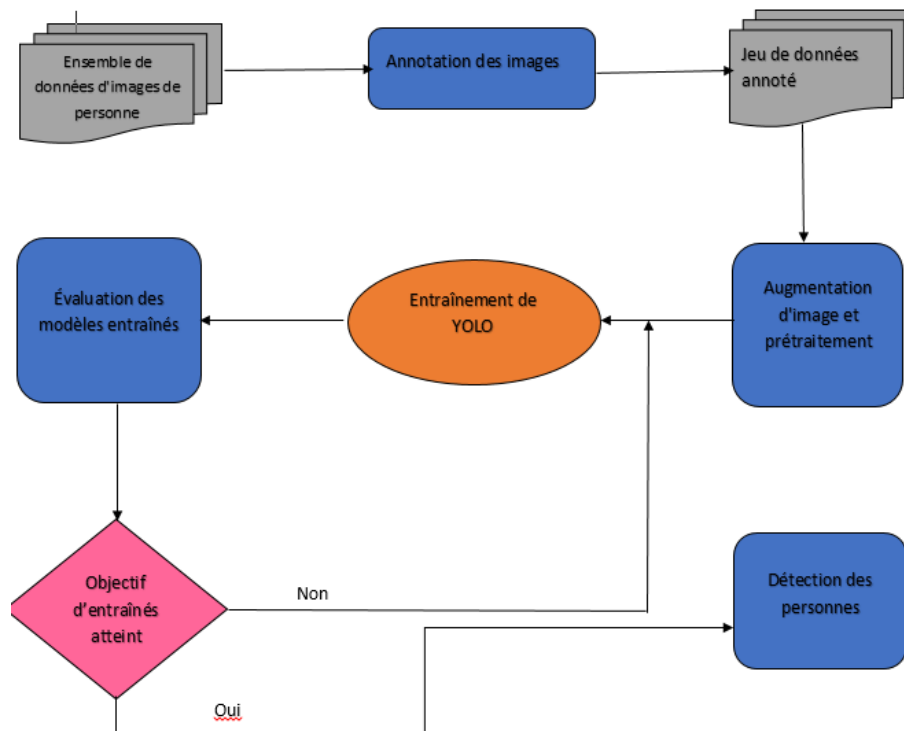


Figure 28: Architecture de notre système.

### III.5. L'algorithme de détection d'objet

Les détecteurs d'objets à un étage unique (comme YOLO) sont composés de trois éléments : une épine dorsale (backbone), un cou (neck) et une tête (Head) pour effectuer des prédictions denses. [35]

- **Backbone**

Le modèle de base (backbone) est un réseau pré-entraîné utilisé pour extraire des représentations de caractéristiques riches pour les images. Cela aide à réduire la résolution spatiale de l'image et à augmenter sa résolution de fonctionnalités (canaux).

- **Model Neck**

Le cou (neck) du modèle est utilisé pour extraire des pyramides de caractéristiques. Cela permet au modèle de généraliser efficacement aux objets de différentes tailles et échelles.

- **Model Head**

La tête (Head) du modèle est utilisée pour effectuer les opérations de la dernière étape. Elle applique des boîtes d'ancrage sur les cartes de caractéristiques et rend la sortie finale : les classes, les scores d'objectivité et les boîtes englobantes.

#### III.5.1. YOLOv5

YOLOv5 est un modèle de la famille des modèles de vision par ordinateur "You Only Look Once" (YOLO) largement utilisé pour la détection d'objets. YOLOv5 est disponible en quatre versions principales : petite (s), moyenne (m), grande (l) et extra grande (x), offrant chacune des taux de précision progressivement plus élevés. Chaque variante prend également un temps différent pour l'entraînement. [36]

Il existe plusieurs versions de YOLO proposées à ce jour, chacune ayant pour objectif d'améliorer les performances du modèle en termes de rapidité et de précision de classification. YOLOv5 est une version récente qui ressemble beaucoup à YOLOv4, mais qui présente quelques différences significatives :

- Le Framework Darknet a été utilisé pour publier YOLOv4, tandis que YOLOv5 est basé sur le Framework PyTorch.
- YOLOv4 utilise le fichier ".cfg" pour la configuration, tandis que YOLOv5 utilise le fichier .yaml. Bien que tous
- les modèles de base de YOLO aient été entraînés sur la base de données COCO, certains modèles ont été entraînés sur d'autres bases de données, telles que l'ensemble de données Image Net de Google
- Vitesse de détection d'objets : YOLOv5 est plus rapide que YOLOv4, avec des temps d'inférence jusqu'à 90% plus rapides que YOLOv4. [37]

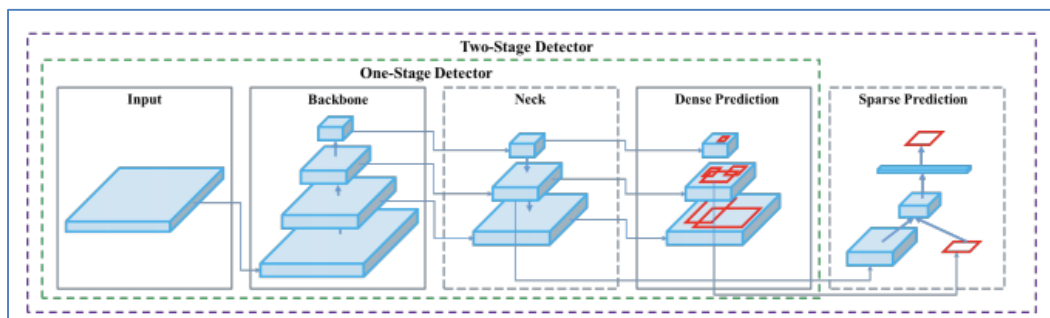


Figure 29: architecture de YOLOv5

### III.5.2. YOLOv7

YOLOv7 est un détecteur d'objets en temps réel à un étage. Il a été présenté à la famille YOLO en juillet 22. Selon l'article YOLOv7, il s'agit du détecteur d'objets en temps réel le plus rapide et le plus précis à ce jour. [38]

Le code source a été publié en open source sous la licence GPL-3.0, une licence libre de droits d'auteur, et peut être trouvé dans le référentiel officiel YOLOv7 GitHub qui a reçu plus de 4,3 000 étoiles le premier mois après sa sortie. [39]

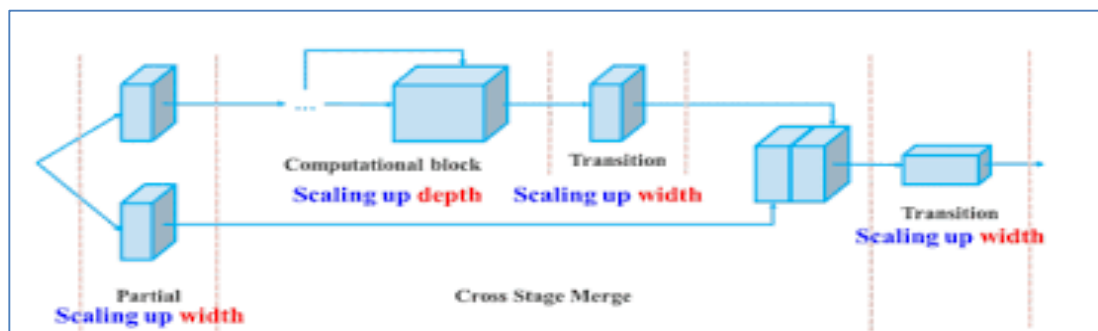


Figure 30: Architecture de YOLOv7

### III.6. Implémentation

#### III.6.1. Environnement logiciel

##### III.6.1.1. PYTHON

Python a été choisi comme langage de programmation car il est un langage de programmation libre et gratuit [40], et Il offre une syntaxe simple et lisible, une grande variété de bibliothèques et d'outils puissants, ce qui en fait un choix privilégié pour les développeurs travaillant dans ce domaine l'apprentissage en profondeur. Il est utilisé pour créer des applications Web, des jeux vidéo, des applications de bureau, des scripts, des outils de calcul scientifique, des systèmes d'exploitation, etc. [41]

##### III.6.1.2. Les bibliothèques utilisées

- PyTorch : YOLOv5 est basé sur PyTorch, une bibliothèque d'apprentissage en profondeur très populaire.
- NumPy : NumPy est une bibliothèque Python essentielle pour le calcul scientifique, souvent utilisée dans le prétraitement des données.
- OpenCV : OpenCV est une bibliothèque d'analyse d'images et de vision par ordinateur. Elle est souvent utilisée pour le traitement des images avant et après l'inférence des modèles.
- Matplotlib : Matplotlib est une bibliothèque de visualisation de données en Python. Elle est utile pour afficher des images, des graphiques et des résultats d'évaluation de YOLOv5.
- PyYAML : PyYAML est une bibliothèque permettant de lire et d'écrire des fichiers YAML, qui sont utilisés pour configurer YOLOv5.

#### III.6.2. L'environnement matériel

##### III.6.2.1. Google Colab

Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner 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. Cool, n'est-ce pas ? Avant de présenter ce magnifique service, nous rappellerons ce qu'est un Jupyter Notebook.

Google Colab offre différentes options pour spécifier l'environnement matériel, notamment :

- CPU : L'environnement par défaut est configuré pour utiliser un processeur central (CPU) pour l'exécution des tâches.
- GPU : Colab permet également d'accéder à un processeur graphique (GPU) pour accélérer les calculs intensifs, notamment pour le deep learning.

- TPU : Les notebooks de Colab peuvent également utiliser les unités de traitement tensoriel (TPU) de Google, qui offrent une accélération encore plus importante pour certains types de tâches liées à l'apprentissage automatique.

### Paramètres du notebook

Accélérateur matériel  
GPU ?

Type de GPU  
T4

Vous voulez accéder à des GPU premium ? [Achetez des unités de calcul supplémentaires](#)

Omettre l'élément de sortie des cellules de code lors de l'enregistrement de ce notebook

Annuler [Enregistrer](#)

Figure 31: l'environnement matériel de notre application

### III.6.2.2. Préparation des données

Nous avons utilisé dans ce projet deux bases de données qui contiennent des images étiquetées avec des instances de la classe "personne", ce qui la rend particulièrement adaptée pour entraîner un modèle de détection de personnes. :

- **Base de donnée 1(BD1)** :« Person\_Roboflow » cette base a été générée en 2022, elle contient 1500 images pour l'entraînement et 300 pour la validation



Figure 32: quelque image de la base de donnée 1

- **Base de donnée 2(BD2)** : "Person\_Object\_Detection" a été générée le 21 avril 2022. Elle contient 10000 images pour l'entraînement et 208 images pour la validation.

Les données ont été téléchargées au format PyTorch pour être compatibles avec le modèle YOLOv5, simplifiant ainsi le processus d'entraînement.



Figure 8 : quelque image de la base de donnée 2

### III.6.2.3. Training

Cette procédure se compose de trois étapes :

- **Etape1** : Nous avons fait deux entraînements en utilisant BD1 et BD2. La taille de l'ensemble de données utilisé a été augmentée dans le deuxième entraînement du projet pour améliorer la précision du modèle de détection d'objets.

Le premier entraînement a été effectué avec YOLOv5s en utilisant BD1 Le modèle a été entraîné pendant 50 époques avec un lot de 16 images. Le fichier ".yaml" (figure33) utilisé pour ce premier entraînement spécifie les chemins vers les répertoires contenant les images d'entraînement, de validation et de test, ainsi que le nom de classe.

```
File Edit Selection View Go Run Terminal Help
! pythorch (4).yaml X ! pythorch10.yaml • ! YAMLYaml •
C: > Users > HP > Downloads > ! pythorch (4).yaml
1 |train: /content/yolov5/myproject/train/images
2 |val: /content/yolov5/myproject1/valid/images
3 |test: /content/yolov5/myproject1/test/images
4
5 |nc: 1
6 |names: ['person']
7
```

Figure 33 : Le premier fichier ".yaml"

**Le deuxième entraînement** a utilisé un ensemble de données BD2. Le modèle YOLOv5x a été utilisé, et l'entraînement a été effectué pendant 5 époques pour éviter le risque de surajustement (overfitting) avec un lot de 16 images. Le fichier ".yaml"(figure34) utilisé pour ce deuxième entraînement comprend des

paramètres supplémentaires spécifiques à YOLOv5, tels que le type de modèle, l'utilisation de poids pré-entraînés, les dimensions des boîtes d'ancrage, etc. Il définit également les configurations pour les parties "backbone", "neck" et "Head" de l'architecture YOLOv5.

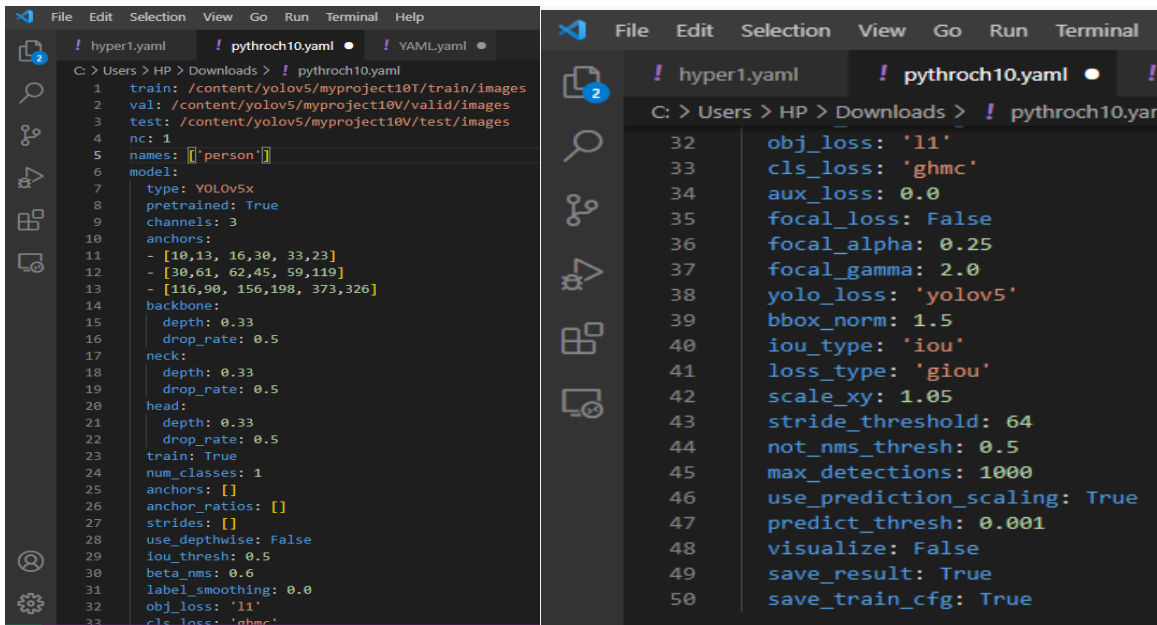


Figure 34 Le deuxième fichier ".yaml"

Les résultats de l'entraînement 1 et 2 sont affichés dans le tableau 2

base de données	maP-50	F1	Precision
BD1	0.552	0.59	0.65
BD2	0.884	0.84	0.880

Tableau 2: Comparaison entre deux entraînements de l'étape 1

- **Etape 2** : dans cette étape nous avons entraîné le jeu de donnée BD2 avec le model yolov5 trois fois, avec 5, 10 et 20 époques respectivement

**Exemple : Epoques=10** le modèle va effectuer 10 itérations complètes de l'ensemble de données d'entraînement pour améliorer les prédictions.



Les résultats des trois entraînements de l'étape2 sont présentés dans le tableau 3

Les époques	maP-50	F1	Precision
5	0.884	0.84	0.880
10	0.898	0.85	0.911
20	0.881	0.84	0.925

Tableau 3: Comparaison entre les époques

**Etape 3** : dans cette étape un entraînement a été effectué avec la base de données BD2 en utilisant deux modèles de l'apprentissage yolov5 et yolov7. Nous comparons les mesures de performance de ces modèles comme indiqué dans le tableau 4

	maP-50	F1	Precision
YOLO v5	0.898	0.85	0.911
YOLO v7	0.616	0.60	0.692

Tableau 4 : Comparaison entre les algorithmes

### III.6.2.3..1. Discussion

- **Première tableau**

Bien que les résultats du premier entraînement ne soient pas mauvais, nous avons réalisé qu'il y avait encore une marge d'amélioration. Pour y parvenir, nous avons augmenté la quantité de données d'entraînement, car l'entraînement d'un modèle d'apprentissage en profondeur avec davantage de données peut souvent conduire à de meilleures performances.

En parallèle, nous avons ajusté le fichier ".yaml" en utilisant des paramètres qui offrent une plus grande flexibilité et un meilleur contrôle sur le modèle et le processus d'entraînement.

- **Deuxième tableau**

Ces résultats suggèrent que 10 époques étaient suffisantes pour atteindre un bon niveau de généralisation. En augmentant le nombre d'époques à 20, nous avons observé un surapprentissage, où le modèle s'est spécialisé de manière excessive dans les données d'entraînement spécifiques.

Il est important de trouver un équilibre dans le nombre d'époques afin d'obtenir les meilleures performances du modèle sans surajustement. Dans notre cas, nous avons constaté que 10 époques étaient plus appropriées pour notre ensemble de données spécifique. Cela indique que le modèle a réussi à généraliser efficacement et à bien se comporter sur de nouvelles données.

- **Troisième tableau**

Dans le tableau, il semble que les résultats de YOLOv5 soient meilleurs que ceux de YOLOv7.

Certaines sources suggèrent que YOLOv5 est meilleur que YOLOv7 en termes de précision, mAP@0.5. [42]

Cependant, d'autres sources suggèrent que YOLOv7 est meilleur que YOLOv5 en termes de précision, avec un score MAP plus élevé on the COCO dataset. [43] [44]

Et les performances de YOLOv5 et YOLOv7 peuvent varier en fonction du jeu de données, du matériel et des critères d'évaluation. [45]

### III.7. Résultats

Voici les résultats de training et les images de test de notre modèle avec l'algorithme yolov5 :

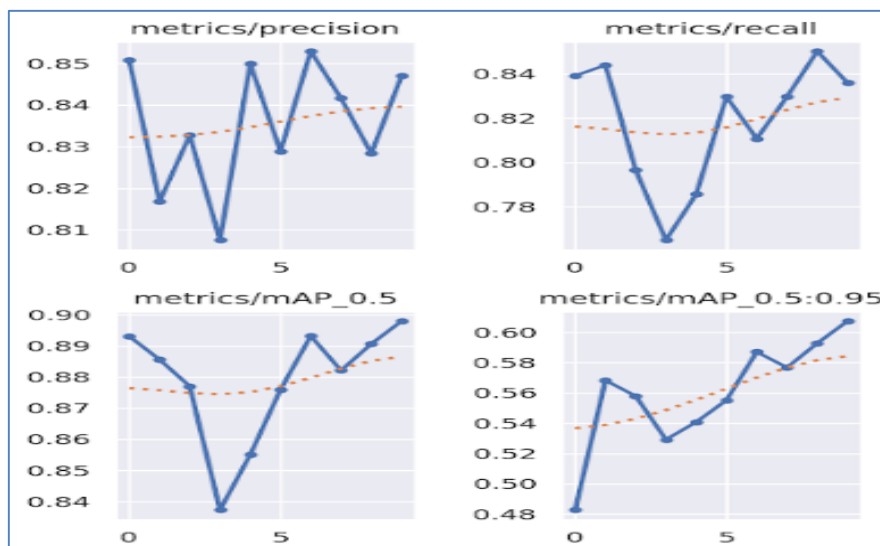


Figure 35: graphe de pert loss

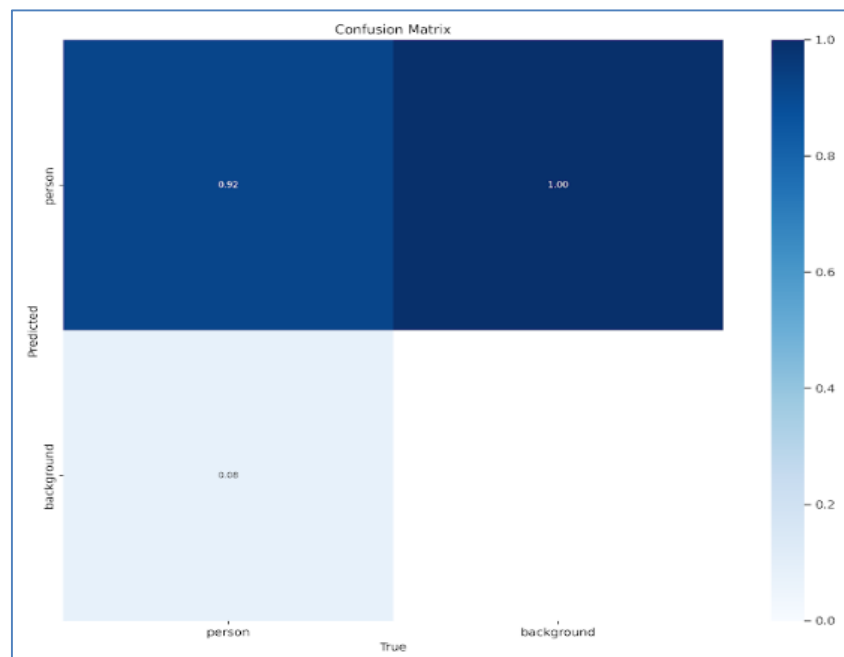


Figure 36: matrice de confusion



Figure 37: test mg1

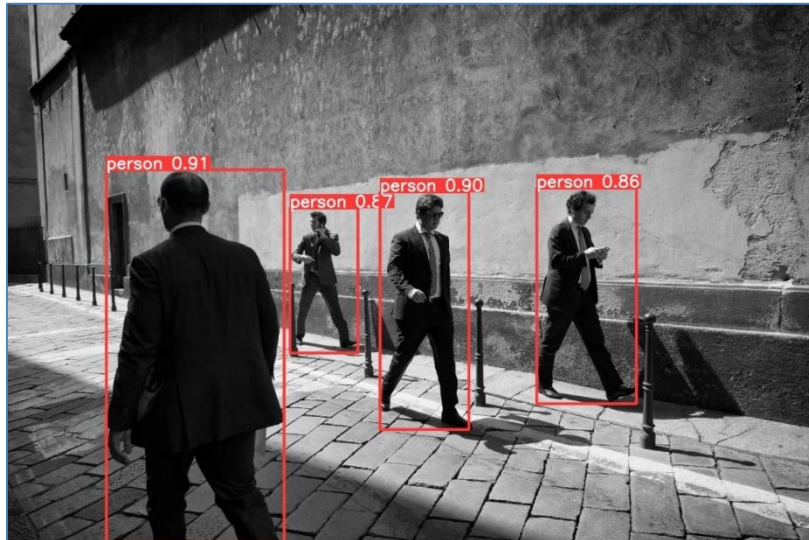


Figure 38: test img2



Figure 39: test img3

## Conclusion

En conclusion, notre chapitre a mis en lumière les différentes étapes de mise en œuvre de notre modèle d'expérimentation ainsi que les résultats obtenus. Nous avons présenté un aperçu des mesures de performances et de l'environnement de développement utilisés, Les résultats obtenus ont confirmé l'efficacité de notre modèle, ouvrant ainsi la voie à de potentielles applications et améliorations futures.

# Conclusion Générale

En conclusion, notre projet de détection de personnes dans les images en utilisant l'algorithme YOLOv5 a été réalisé avec succès. Nous avons utilisé une approche itérative en entraînant le modèle avec différentes configurations et en évaluant les performances obtenues.

En expérimentant avec différents ensembles de données, nous avons pu constater l'importance d'un ensemble de données de taille adéquate pour améliorer la capacité de détection du modèle. Les résultats ont démontré que l'utilisation d'un ensemble de données plus volumineux a conduit à de meilleures performances en termes de précision et de sensibilité.

De plus, nous avons ajusté le nombre d'époques lors de l'entraînement du modèle pour trouver la configuration optimale. Les résultats ont montré que l'utilisation de 10 époques a permis d'obtenir les meilleurs résultats en termes de précision et de temps d'entraînement.

Ces résultats confirment l'importance de l'ensemble de données et des hyperparamètres pour obtenir des performances optimales dans la détection de personnes avec YOLOv5. Ils soulignent également l'importance de la méthodologie itérative dans le processus d'optimisation du modèle.

Nos travaux offrent des perspectives intéressantes pour des recherches futures dans le domaine de la détection d'objets, en particulier la détection de personnes. Les pistes à explorer comprennent l'évaluation d'autres architectures de réseaux neuronaux, tels que EfficientDet ou Mask R-CNN, l'extension de la détection de personnes aux vidéos en utilisant des techniques de suivi d'objets, l'application de techniques de transfert d'apprentissage pour améliorer les performances avec des ensembles de données plus restreints, et l'exploration de la détection de personnes dans des contextes spécifiques tels que la surveillance en temps réel ou la détection d'anomalies comportementales. Ces perspectives permettront de continuer à développer et à améliorer les techniques de détection d'objets, avec des applications pratiques dans des domaines variés.

En conclusion, notre projet a permis de mettre en évidence l'importance de l'ensemble de données, des hyperparamètres et de la méthodologie itérative dans la détection de personnes avec YOLOv5. Il contribue à l'amélioration des performances de détection d'objets et offre des perspectives pour des recherches futures visant à optimiser davantage les résultats obtenus.

## Références

- [1] C. Zhang et Yang Lu , «Study on artificial intelligence: The state of the art and future prospects,» September 2021.
- [2] «image-numerique,» [En ligne]. Available: <https://www.schoolmouv.fr/definitions/image-numerique/definition>. [Accès le 04 2023].
- [3] M. Klein, «formats-image-numerique,» 1 mars 2020.
- [4] B. MACAP, «image-vectorielle,» 19 OCTOBRE 2020.
- [5] J. Jaehnig, «what-is-a-bitmap-image,» 21 JANVIER 2021.
- [6] «TRAITEMENT D'IMAGE A VOISINAGES ADAPTATIFS GENERAUX,» [En ligne]. Available: <https://theses.hal.science/tel-00749859v1/document>. [Accès le 04 2023].
- [7] H. Welter, «hubspot,» 1 mars 2023.
- [8] «ibm,» [En ligne]. Available: <https://www.ibm.com/fr-fr/topics/neural-networks>. [Accès le 04 2023].
- [9] «cnil,» [En ligne]. Available: <https://www.cnil.fr/fr/definition/fonction-dactivation>. [Accès le 04 2023].
- [10] B. Maurice, «deeplylearning,» 26 septembre 2018.
- [11] «mathworks,» [En ligne]. Available: <https://fr.mathworks.com/discovery/convolutional-neural-network-matlab.html>. [Accès le 05 2023].
- [12] C. Crouspeyre, «medium,» 1017.
- [13] N. Belaidi, «blent,» 21 juin 2022.
- [14] A. Amidi et Shervine Amidi, «stanford».
- [15] B. L, «lebigdata,» 05 avril 2019.
- [16] TODOME, «veillecarto2-0,» 22 janvier 2022.
- [17] X. Lu, Quanquan Li, Buyu Li et Junjie Yan, «MimicDet: Bridging the Gap Between».
- [18] R. Girshick, Jeff Donahue, Trevor Darrell et Jitendra Malik, «feature hierarchies for accurate object detection and semantic segmentation».
- [19] E. I. Conference, «Proceedings of the IEEE international conference on computer vision,» 1995.
- [20] S. Ren, Kaiming He, Ross Girshick et Jian Sun, «Faster R-CNN: Towards real-time object detection with region proposal networks,» p. 14, 2016.
- [21] W. Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szege, Scott Reed, Cheng-Yang Fu et Alexander C. Berg, «SSD: Single Shot MultiBox Detector,» p. 17, 2016.
- [22] T.-Y. Lin, Priya Goyal, Ross Girshick, Kaiming He et Piotr Doll, «Focal Loss for Dense Object Detection,» p. 10, 2018.

- [23] J. Redmon, Santosh Divvala , Ross Girshick et Ali Farhadi, «You Only Look Once:», p. 10, 9 May 2016.
- [24] G. Nishad, «medium,» 1 Mar 2019.
- [25] O. Chernytska, «towardsdatascience,» 18 Auguste 2022.
- [26] G. Nishad, «medium,» 1 Mar 2019.
- [27] S. Ren, Kaiming He, Ross Girshick et Jian Sun, «Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,» p. 14, 06 Janvier 2016.
- [28] K. He, Xiangyu Zhang, Shaoqing Ren et Jian Sun, «Deep Residual Learning for Image Recognition,» p. 12, 10 Decembre 2015.
- [29] Y. CHERGUI et BOUSSAHA, Chaimaa, «Détection des anomalies par segmentation des images médicales cérébrales,» 2022.
- [30] [En ligne]. Available: <https://archives.univ-biskra.dz/bitstream/123456789>. [Accès le 05 2023].
- [31] «xlstat,» [En ligne]. Available: <https://www.xlstat.com/fr/solutions/fonctionnalites/analyse-detaillee-de-sensibilite-et-specificite>. [Accès le 05 2023].
- [32] M. Emilion, «jedha».
- [33] «datascientest,» [En ligne]. Available: <https://datascientest.com/matrice-de-confusion>. [Accès le 05 2023].
- [34] R. Kundu, «v7labs».
- [35] [En ligne]. Available: <https://iq.opengenus.org/yolov5/>.
- [36] J. Solawetz, «roboflow,» 29 juin 2020.
- [37] «github,» [En ligne]. Available: <https://github.com/ultralytics/yolov5>.
- [38] S. Rath et Kukil, «learnopencv,» 2 août 2022.
- [39] G. Boesch, «viso,» 2023.
- [40] «01net,» [En ligne]. Available: <https://www.01net.com/telecharger/programmation/creation/python.html>. [Accès le 2023].
- [41] «machinelearning.fr,» [En ligne]. Available: <https://machinelearning.fr/product/lapprentissage-profond-avec-python-francois-chollet/>.