République **A**lgérienne **D**émocratique et **P**opulaire

Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche Scientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DU GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire de Fin d'Etudes
De MASTER ACADEMIQUE

Domaine : **Sciences et Technologies**

Filière : **Electronique**

Spécialité : **Electronique des Systèmes Embarqués**

Présenté par

RAHMANI Tinhinane

SADI Fatma

Thème

Système embarqué pour la détection d'objets basé sur YOLOv5

Mémoire soutenu publiquement le 26/06/2024 devant le jury composé de :

Mr TAHANOUT Mohammed
President, **M.C.A**, UMMTO
Mr Lahdir Mourad
Encadrant, Professeur, UMMTO
Mr ALOUACHE Djamal
Examinateur, **M.C.A**, UMMTO

Année universitaire 2023-2024

Remerciements

Tout d'abord, nous adressons nos remerciements à Dieu le Très-Haut, pour nous avoir donné la force et la détermination nécessaires à la réalisation de ce travail.

Nous tenons à exprimer notre gratitude à toutes les personnes qui ont contribué, de près ou de loin, à la réalisation de ce mémoire.

Nous remercions vivement M. Lahdir Mourad, notre encadrant de mémoire, pour ses conseils avisés et son encadrement tout au long de ce travail.

Nos remerciements s'adressent également aux membres du jury, qui ont accepté d'évaluer notre travail et nous ont apporté des suggestions précieuses.

Enfin, nous exprimons notre reconnaissance à tous les responsables et enseignants du département d'Electronique université Mouloud Mammeri Tizi Ouzou, pour leur soutien et leur contribution à notre formation.

TINHINANE RAHMANI

FATMA SADI

Dédicaces

Après de nombreuses années de travail et d'engagement, je dédie ce mémoire à :

Tout d'abord, je remercie Dieu pour toute la force qu'Il m'a donnée.

À mes parents,

Pour leur soutien constant, leur amour inconditionnel et leurs encouragements tout au long de ce parcours. Merci de m'avoir toujours cru en moi, pour leur patience, leur compréhension et leur soutien inébranlable tout au long de mes études. Je vous dédie ce travail en signe de ma profonde gratitude.

À ma chère mère,

Pour tes conseils avisés, tes immenses sacrifices, ton soutien constant et tes encouragements précieux. Merci pour tout.

À mon père,

Ma gratitude ne suffira jamais à exprimer tout ce que je te dois pour tes sacrifices depuis ma naissance, pendant mon enfance, et même à l'âge adulte. Merci infiniment, papa.

À mon frère et mes sœurs,

Pour votre soutien et votre présence.

À ma binôme adorée Fatma Sadi,

Pour tous les moments qu'on a passés ensemble.

À tous mes enseignants,

Pour leur dévouement et leur patience infinie.

À toute la famille Rahmani,

Pour votre soutien et vos encouragements tout au long de ce parcours. Ce travail est dédié à notre famille et à tous les moments précieux partagés ensemble.

TINHINANE RAHMANI

Dédicace

Je suis ravi de dédier ce travail :

Tout d'abord, je voudrais remercier Dieu pour tout.

À mon très cher père, Lah irahmou, qui peut être fier de voir ici le résultat de longues années de sacrifices et de privations qui m'ont aidé à avancer dans la vie.

A ma très chère mère, A la femme qui a souffert sans me laisser souffrir, qui n'a jamais dit non â mes exigences et qui n'a épargné aucun effort pour me rendre heureuse *qui me donne toujours l'espoir de vivre et qui a œuvré pour ma réussite qui m'ont permis de me sentir aimé et soutenu dans mes choix.*

À ma chère petite sœur Romaissa, qui m'a toujours soutenu et a été à mes côtés.

À mon cher fiancé Qui n'a pas cessé de me conseiller, encourager et soutenir tout au long de mes études. Que Dieu la protège et lui offre la chance et le bonheur

À toutes mes chères cousines précisément Ouldkaci Sabrina et mes chères tantes pour leur soutien

Mes copines qui ont toujours été présentes et m'ont apporté leur soutien.

À la famille Sadi

À ma binôme Tinhinane et sa famille, pour leur précieuse collaboration et leur soutien constant tout au long de ce travail.

À tous ceux qui m'ont aidé, encouragé, et conseillé, que ce soit de près ou de loin, merci pour votre confiance en moi et pour l'encouragement que vous m'avez apporté.

Fatma Sadi

Liste Des Figures

Liste des Figures

Chapitre 1 : Vision Artificielle

Chapitre2 : Méthodes de la détection d'objets

Chapitre3 : Système embarqués de détection d'objets basé sur deep learning

Chapitre4: Application et Résultats

Liste des Figures

Liste des tableaux

Liste des Tableaux

Liste des abréviations

Liste des abréviations

YOLO (You Only Look Once) - On ne regarde qu'une fois

IA (Intelligence Artificielle) - Artificial Intelligence

ML (Machine Learning) - Apprentissage automatique

ASI (Apprentissage Supervisé) - Supervised Learning

GPU (Graphics Processing Unit) - Unité de traitement graphique

DL (Deep Learning) - Apprentissage profond

RNN (Recurrent Neural Network) - Réseau neuronal récurrent

GAN (Generative Adversarial Network) - Réseau antagoniste génératif

CNN (Convolutional Neural Network) - Réseau neuronal convolutif

MLP (Multilayer Perceptron) - Perceptron multicouche

RCNN (Region-based Convolutional Neural Network) - Réseau neuronal convolutif basé sur la région

SPP-NET (Spatial Pyramid Pooling Network) - Réseau de regroupement de pyramides spatiales

FRCNN (Fast Region-based Convolutional Neural Network) - Réseau neuronal convolutif basé sur la région rapide

G-CNN (Gated Convolutional Neural Network) - Réseau neuronal convolutif à portes

SSD (Single Shot MultiBox Detector) - Détecteur de boîtes unique

SVM (Support Vector Machine) - Machine à vecteurs de support

NMS (Non-Maximum Suppression) - Suppression des non-maximum

ROI (Region of Interest) - Région d'intérêt

FC (Fully Connected) - Entièrement connecté

RPN (Region Proposal Network) - Réseau de proposition de région

VGG-16 (Visual Geometry Group-16) - Groupe de géométrie visuelle-16

FPN (Feature Pyramid Network) - Réseau de pyramide de caractéristiques

WRC (Weighted Residual Connections) - Connexions résiduelles pondérées

CSP (Cross Stage Partial connections) - Connexions partielles en travers de l'étage

Liste des abréviations

CMBN (Context Module with Batch Normalization) - Module de contexte avec normalisation par lots

SAT (Spatial Attention) - Attention spatiale

SPP (Spatial Pyramid Pooling) - Regroupement de pyramides spatiales

PAN (Path Aggregation Network) - Réseau d'agrégation de chemin

CSP (Cross Stage Partial connections) - Connexions partielles en travers de l'étage

BBOXES (Bounding Boxes) - Boîtes englobantes

ReLU (Rectified Linear Unit) - Unité linéaire rectifiée

SGD (Stochastic Gradient Descent) - Descente de gradient stochastique

ADM (Adam) - Adam algorithme d'optimisation

GNU (GNU's Not Unix)

ARM (Advanced RISC Machine)

RAM (Random Access Memory) - Mémoire vive

HDMI (High-Definition Multimedia Interface) - Interface multimédia haute définition

USB (Universal Serial Bus) - Bus série universel

RCA (Radio Corporation of America) - Radio Corporation of American

SD (Secure Digital) - Digital sécurisé

GRIO (General-Purpose Input/Output) - Entrée/Sortie à usage général

CPU (Central Processing Unit) - Unité centrale de traitement

OS (Operating System) - Système d'exploitation

MAP : Mean Average Precision

Sommaire

Sommaire

Chapitre3 : Système embarqués de détection d'objets basé sur deep learning

Chapitre4: Application et Résultats

Sommaire

Introduction

Introduction

L'intelligence artificielle (IA) a transformé notre manière d'interagir avec le monde qui nous entoure en permettant aux machines d'apprendre à partir de données et de prendre des décisions autonomes. Parmi les nombreuses applications de l'IA, la détection d'objets en temps réel représente une avancée significative, offrant des possibilités d'automatisation et de sécurité dans divers domaines, de la surveillance à la robotique.

Au cœur de cette innovation se trouve la vision par ordinateur, une discipline de l'IA qui permet aux machines de percevoir et d'interpréter le monde visuel qui les entoure. Avec l'avènement de solutions de cristallisation avancées telles que YOLO (You Only Look Once), il est désormais possible d'effectuer des tâches de détection d'objets en temps réel avec une efficacité et une précision impressionnante.

Dans ce contexte, notre travail se concentre sur l'utilisation d'un système embarqué Raspberry Pi 4 comme plateforme pour déployer un système de détection d'objets en temps réel. En particulier, nous explorons l'utilisation du modèle YOLOv5, qui représente l'une des dernières itérations de la série YOLO, caractérisée par sa rapidité et sa précision dans la détection d'objets. YOLOv5 a été conçu avec une architecture plus légère tout en maintenant des performances de détection hautement efficaces, ce qui le rend particulièrement adapté pour les applications nécessitant un traitement en temps réel sur des ressources limitées telles que le Raspberry Pi 4.

En résumé, notre travail vise à démontrer l'efficacité de l'utilisation de YOLOv5 sur Raspberry Pi 4 pour des applications de détection d'objets en temps réel, ouvrant ainsi la voie à de nouvelles opportunités dans le domaine de la vision par ordinateur embarquée.

À travers notre travail, nous visons à fournir une compréhension approfondie de la détection d'objets en temps réel avec YOLOv5 sur Raspberry Pi 4, tout en mettant en lumière les défis et les opportunités dans ce domaine en constante évolution.

Pour réaliser notre travail, nous avons structuré notre mémoire en quatre chapitres distincts, chacun se concentrant sur un aspect spécifique de notre sujet :

Le premier chapitre établit les principes fondamentaux de l'intelligence artificielle (IA), de la vision par ordinateur, la détection d'objets et Apprentissage automatique (Machine Learning).

Dans le deuxième chapitre, nous plongeons dans les détails des modèles de détection d'objets, en mettant particulièrement l'accent tout en présentant son principe de fonctionnement, ses avantages et ses inconvénients.

Le troisième chapitre explore la mise en œuvre pratique de notre système de détection d'objets en temps réel sur un système embarqué Raspberry Pi 4 avec Yolov5.

Le chapitre quatre contient la description des étapes de configuration du système de détection et la discussion des résultats obtenus.

Enfin, nous terminons notre mémoire par une conclusion permettant de résumer le travail fait ainsi que par des perspectives pouvant améliorer notre système.

Chapitre 1 : Vision Artificielle

1. Préambule :

Ce chapitre se concentrera sur les fondements de l'intelligence artificielle et de la vision par ordinateur, une discipline qui convertit une image en données objet et identifie les objets par l'extraction et l'analyse de caractéristiques abstraites. La détection d'objets, l'un des domaines les plus explorés de la vision par ordinateur, sera particulièrement étudiée en raison de ses nombreuses applications, notamment en vidéosurveillance et robotique, ainsi que de son rôle dans l'automatisation de processus tels que la reconnaissance faciale.

Le concept vision par ordinateur est défini comme une branche de l'intelligence artificielle intégrant diverses méthodes pour créer automatiquement des modèles à partir de données visuelles. Dans ce chapitre, nous aborderons également le concept d'apprentissage automatique et l'apprentissage profond. Ensuite, nous examinerons en détail l'algorithme le plus répandu les réseaux de neurones.

2. L'intelligence artificielle :

L'intelligence artificielle (IA) est un domaine vaste et interdisciplinaire qui cherche à développer des algorithmes et des systèmes informatiques capables de reproduire le comportement humain dans ses activités de raisonnement. Depuis ses débuts dans les années 1950 [1], l'IA a évolué et est maintenant largement utilisée dans de nombreux secteurs tels que la finance, la médecine, la robotique, la surveillance et les jeux vidéo. Pour cette raison, l'IA s'est divisée en de nombreux sous-domaines, chacun essayant de traiter une partie du problème. Comme indiqué dans la figure 1, ces principes sont fondamentaux pour parvenir à la vision par ordinateur, et ils impliquent l'utilisation de l'apprentissage automatique et de l'apprentissage profond (Deep Learning) [1].

La figure 1 présente une vue d'ensemble des sous-domaines de l'intelligence artificielle (IA) [68] :

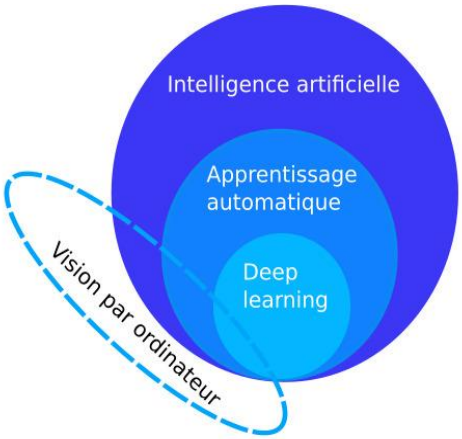


Figure 1: L'intelligence artificielle et ses sous-domaines [68].

a. Domaine d'application :

- Résolution de problèmes généraux : L'objectif est de créer des algorithmes généraux pour résoudre des problèmes concrets.
- Traitement du langage naturel : Ce sous-domaine vise à la compréhension, la traduction, ou la production du langage (écrit ou parle).
- Vision artificielle : est de permettre aux ordinateurs de comprendre les images et la vidéo (par exemple, de reconnaitre des visages ou des chiffres).
- Robotique : Cette discipline vise à réaliser des agents physiques qui peuvent agir dans le monde.
- Apprentissage automatique : Dans cette branche de l'IA, on essaie de concevoir des programmes qui peuvent s'auto-modifier en fonction de leur expérience [2].

b. Type d'intelligence artificielle :

- **L'intelligence artificielle générale (AGI)** :

L'AGI est dans la classe d'IA forte, car elle travaille à un niveau supérieur, qui correspond à l'intelligence humaine.

- **La super intelligence artificielle (ASI)** :

Bien que ce type d'IA ne soit actuellement pas développé, l'ASI signifie est une machine est plus intelligente qu'un humain [3].

c. Fonctionnement :

L'IA fonctionne en mémorisant des comportements à partir de bases de données et en utilisant des algorithmes pour résoudre des problèmes et agir en conséquence. Elle est étroitement liée au Big Data, car elle nécessite d'importantes quantités de données pour effectuer des analyses et des apprentissages.

- **Les unités de traitement graphique (GPU)** :

C'est un processeur spécialisé conçu pour effectuer des calculs parallèles intensifs, initialement pour les graphismes, mais maintenant aussi utilisé dans l'intelligence artificielle et le machine learning.

- **Le Big Data** :

Les algorithmes utilisés par l'IA sont façonnés par la grande quantité d'informations trouvées dans les méga données. Cela permet de traiter l'information à un rythme élevé et de rendre les données plus accessibles et utilisables.

- **Algorithmes :**

Les algorithmes évoluent constamment en devenant plus complexes grâce à l'utilisation de couches avec des variables cachées, ce qui permet de trier et d'optimiser les résultats. Ils permettent d'accomplir des tâches que l'on croyait réservées uniquement à l'intelligence humaine [4].

3. Vision par ordinateur :

La vision par ordinateur est un domaine de l'IA qui permet aux ordinateurs de traiter et d'analyser des images et des vidéos. Grâce à l'IA, les ordinateurs deviennent suffisamment puissants pour percevoir le monde de la même manière que les humains. Inspiré par le fonctionnement des neurones du cerveau, ce domaine est largement utilisé dans une variété d'applications qui exploitent l'intelligence artificielle et l'apprentissage automatique (IA/ML) pour traiter ces données avec précision. Ces applications incluent l'identification d'objets, la reconnaissance faciale, la classification, la recommandation, la surveillance et la détection [5].

Les réseaux de neurones, une composante essentielle de la vision par ordinateur, sont capables de fonctionner et de traiter de vastes bases de données en parallèle. Ils peuvent être entraînés à partir de données en utilisant des techniques d'apprentissage automatique, ce qui leur permet de reconnaître des motifs dans les données et d'accomplir des tâches complexes en vision par ordinateur [6].

3.1. Les quatre grandes catégories d'algorithmes :

Cette figure illustre divers exemples de détection et de segmentation d'objets dans des images [69].

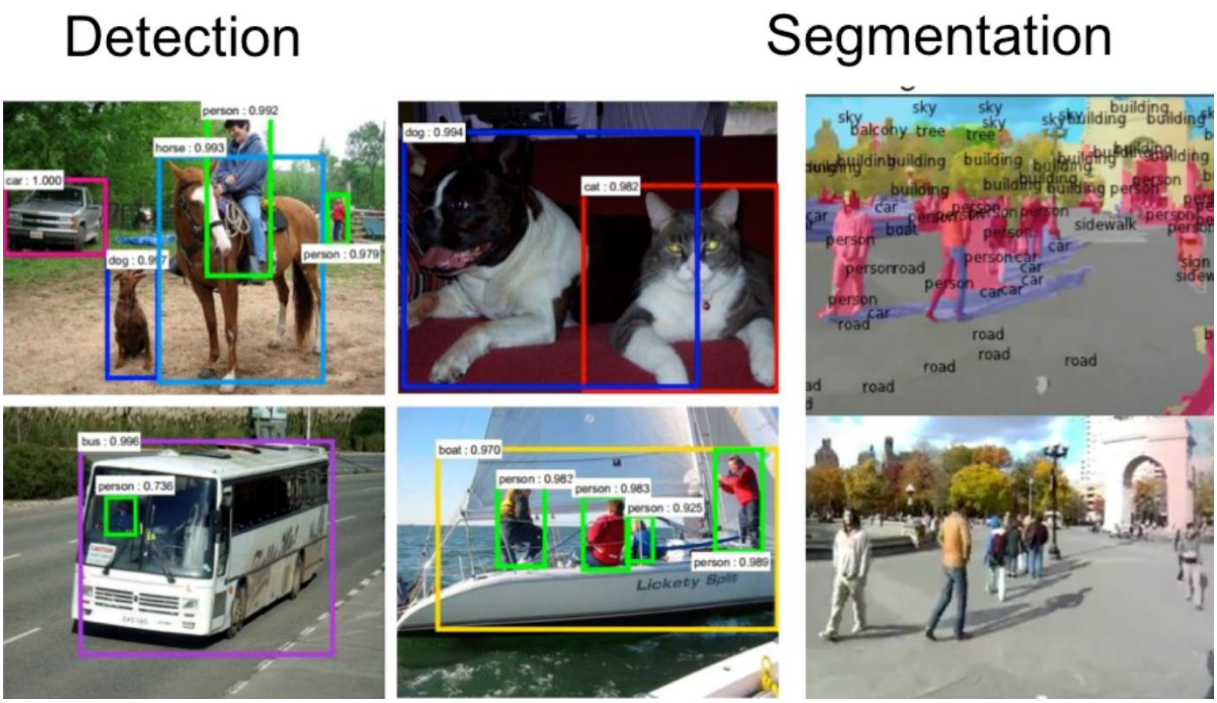


Figure 2: Exemples de détections et de segmentations d'objets à l'intérieur d'images [69].

3.1.1.La détection d’objets :

La détection d'objets, partie intégrante de la vision par ordinateur, permet aux systèmes d'intelligence artificielle de repérer et de situer des catégories spécifiques d'objets dans des flux d'images ou via des caméras, Elle se divise généralement en deux étapes principales : la localisation et la classification.

La première étape vise à repérer la position d'un ou plusieurs objets dans une image et à fournir les coordonnées de la boîte englobante correspondant à chaque objet détecté. La seconde phase consiste à prédire la classe à laquelle chaque objet appartient.

3.1.2.La segmentation d’images :

La segmentation d'images est une technique en vision par ordinateur qui automatise la découpe d'une image en régions de pixels appartenant à des classes d'objets distinctes.

Elle se divise en deux principaux types : la segmentation sémantique et la segmentation par instance. Dans la segmentation sémantique, l'objectif est de classifier les pixels de l'image en fonction de leur appartenance à une catégorie spécifique. En revanche, la segmentation par instance permet une analyse plus détaillée de l'image en identifiant et en différenciant les différentes occurrences d'un même objet [7].

3.1.3.La classification d’images :

La classification d'images donne aux ordinateurs la capacité de regarder une image et de déterminer avec précision à quelle classe elle appartient. Cela implique que la vision par ordinateur peut reconnaître et étiqueter différents éléments comme des personnes, des animaux, des paysages, etc. Par exemple, un logiciel de reconnaissance d'images peut identifier les chiens dans une photo et les distinguer des autres objets [8].

3.1.4. La reconnaissance de formes, d’images ou de mouvements :

La reconnaissance de formes consiste à identifier des objets spécifiques dans une image, comme des visages, des voitures ou des bâtiments Cela implique d'extraire des caractéristiques visuelles comme les bords, les angles et les textures, puis de les comparer à des modèles appris pour classifier l’objet [9].

La reconnaissance d'images permet d'identifier le contenu global d'une image, comme le type de scène (intérieur, extérieur, urbain, naturel, etc.) ou les objets présents Des techniques d'apprentissage profond comme les réseaux de neurones convolutifs sont souvent utilisés pour cette tâche [10].

La reconnaissance de mouvement analyse une séquence d'images ou une vidéo pour détecter et suivre le mouvement d'objets Cela peut servir à détecter des événements, suivre des cibles mobiles ou estimer la trajectoire d'objets en mouvement Des méthodes courantes incluent la soustraction de fond et le flot optique [11].

La figure 3 présente trois exemples de reconnaissance de forme [70] :

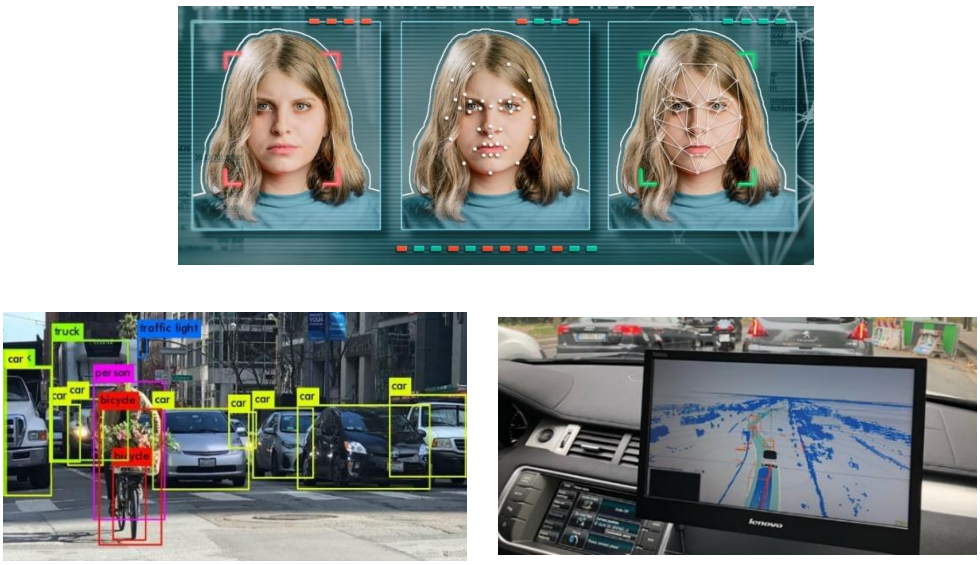


Figure 3: Exemples de reconnaissance de forme : a) Reconnaissance faciale, b) détections d'objets, c) voiture autonome [70].

4. Détections d'objets :

La détection d'objets est l'un des domaines de recherche les plus actifs dans le domaine de la vision par ordinateur ainsi elle implique à la fois la classification de chaque objet dans l'image et visent à crée des systèmes qui se rapproches des compétences de l'être humain dans la perception et le suivi et la reconnaissance d'objet. L'importance de la détection vienne du fait que le bon résultat dans cette phase donne un bon résultat de la reconnaissance et aussi cette phase n'est pas facile à traiter à cause de plusieurs problèmes tels que la taille d'objet détecter, la lumière, la forme, la grande variété des objets, la vitesse de réponse en temps réel, la complexité des backgrounds… etc.

La figure ci-dessous illustre la détection et la classifications d'objets [74] :

Figure 4: Détections et classifications d'objets [74].

La détection d'objet elle est parmi les applications pratiques les plus intéressantes dans la vie courante.

Par exemple elle est utilisée dans :

- La surveillance du trafic ou bien pour un véhicule avec une assistance de conduite automatique ou partiellement autonome.
- La détection des personnes qui ne portent pas des masques pendent la situation du « COVID - 19 ».
- Dans les entreprises pour détecter les produits bien fait et le mal fait (entreprise qui produit des billets ou pièces donc le système va vérifier la qualité).
- Robotique.
- L'analyse d'images médicales.
- Militaire.
- Et la détection de la somnolence des conducteurs sur l'autoroute afin d'éviter les accidents peut être obtenue par la détection d'objets aussi [12].

5. Apprentissage automatique (Machine Learning) :

C'est un champ d'étude d'IA qui se fonde sur des algorithmes (approche mathématiquement et statistique) pour donner aux machines (systèmes informatique) la capacité « d'apprendre » à partir de données, améliorer la performance à résoudre des taches sans être explicitement programmés, Les paramètres du modèle sont ajustés progressivement lors d'une phase d'apprentissage, utilisant des ensembles de données d'entraînement pour découvrir des relations et les classifier. Les concepteurs choisissent parmi diverses méthodes d'apprentissage automatique en fonction des caractéristiques des tâches à accomplir, telles que le regroupement ou la construction d'arbres de décision. Ces méthodes sont Généralement regroupées en trois catégories :

- L'apprentissage supervisé.
- L'apprentissage non supervisé.
- L'apprentissage par renforcement.

Chacune de ces catégories comprend différentes approches, telles que les réseaux de neurones et l'apprentissage profond. Comme cité dans la figure 5 certaine catégorie de l'apprentissage automatique [71].

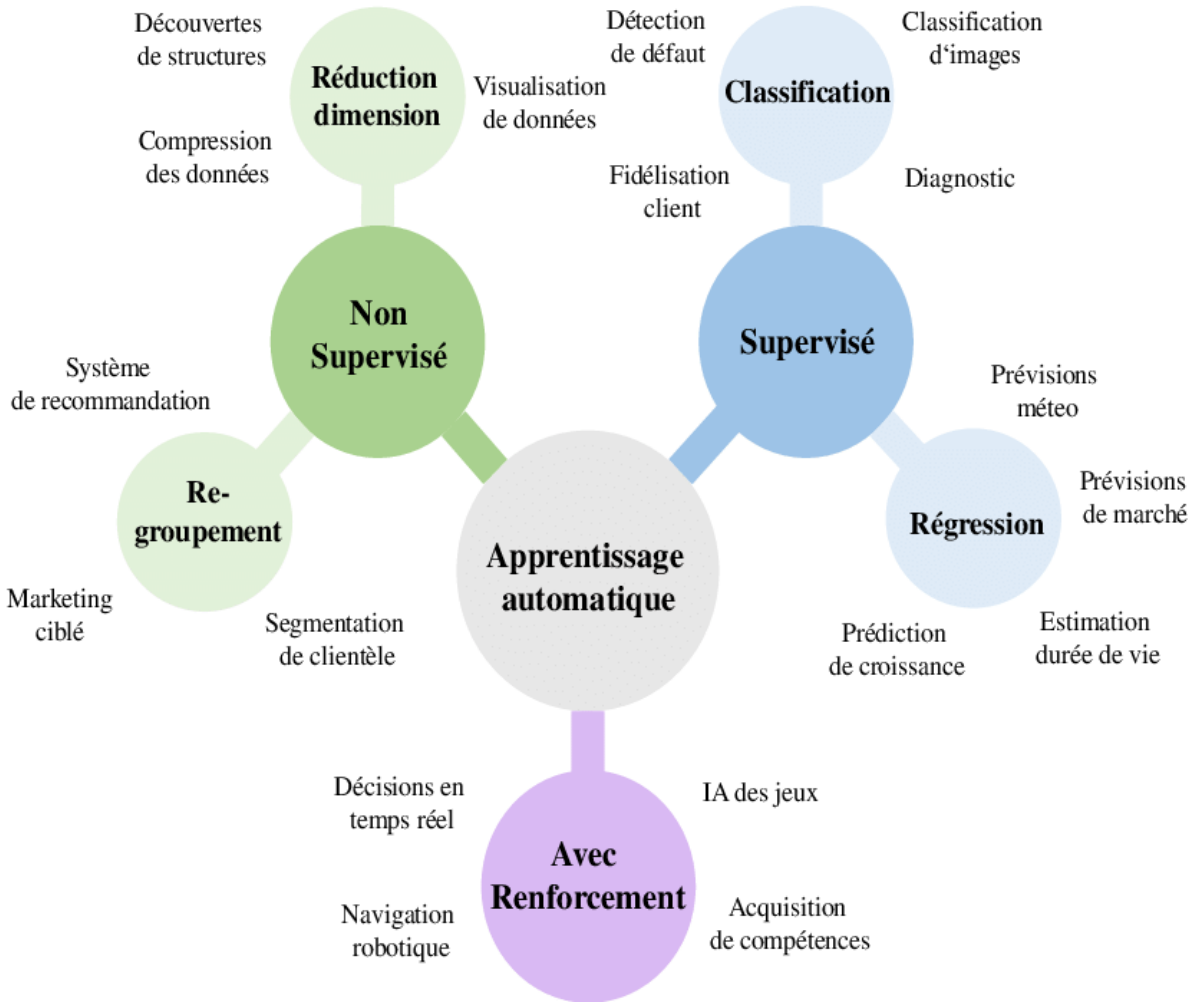


Figure 5: Catégories d'apprentissage automatique [71].

L'apprentissage supervisé :

Utilise, des données étiquetées pour créer un modèle permettant de faire des prédictions sur de nouvelles données. Il comporte deux types principaux de problèmes : la classification, où les données de sortie sont des étiquettes prédéfinies, et la régression, où la sortie est une valeur continue.
La classification peut être binaire (deux classes) ou multi classe (plusieurs classes). Exemple: Gmail qui est en mesure de prédire si un mail donné est un spam ou mail normal.
La régression vise à prédire une valeur aussi proche que possible de la valeur réelle, Plus l'erreur est petite, plus la précision de notre modèle de régression est grande [14].

- Nous distinguons plusieurs types d'apprentissage supervisés :

L'apprentissage non supervisé : qui utilise des données d'entrée sans étiquettes pour tirer des inférences. Le clustering est une technique de cet apprentissage, où les données sont regroupées en clusters en fonction de leurs similitudes, sans que ces clusters ne soient prédéterminés [15].

L'apprentissage semi-supervisé : Combine les deux types précédents en utilisant à La fois des données étiquetées et non étiquetées, ce qui peut réduire les coûts et la complexité associés à l'étiquetage des données [16].

L'apprentissage par renforcement : Implique un agent interagissant avec son environnement, prenant des décisions et recevant des récompenses en fonction de ses actions, afin d'apprendre et de s'améliorer dans cet environnement [17].

6. Apprentissage Profond (Deep Learning) :

Pourrait être défini comme un type de machine Learning, mais un peu plus complexe. Il s'agit d'un ensemble d'algorithmes qui utilisent des réseaux de neurones, inspirés du fonctionnement du cerveau humain. Dans cette technologie, la machine apprend par elle-même, en suivant des étapes organisées en couches. La profondeur du réseau dépend du nombre de ces couches dans le modèle. Les avancées du Deep Learning sont attribuées à l'augmentation exponentielle de la puissance de calcul et de stockage des machines, ainsi qu'à la disponibilité de vastes ensembles de données (big data).

L'apprentissage profond repose sur des réseaux de neurones composés de milliers d'unités (appelées « neurones »), chacune effectuant de petites opérations simples. Les résultats de la première couche de neurones servent d'entrée pour les calculs de la deuxième couche, et ainsi de suite. Par exemple, dans le domaine de la reconnaissance visuelle, les premières couches d'unités identifient des lignes, des courbes et des angles, tandis que les couches supérieures reconnaissent des combinaisons de formes, des objets et des contextes. Les progrès du Deep Learning ont été rendus possibles notamment grâce à l'augmentation de la puissance des ordinateurs et au développement de vastes bases de données (« big data ») [18].

6.1. Les réseaux de neurones :

6.1.1. Les réseaux de neurones biologiques :

Sont les connexions entre les neurones du cerveau et du système nerveux. Ils facilitent la transmission d'informations et la régulation des fonctions corporelles. Ces réseaux, complexes, permettent le traitement des signaux électriques et chimiques, soutenant ainsi les processus cognitifs, sensoriels, moteurs et émotionnels chez les êtres vivants. Ils jouent un rôle essentiel dans le fonctionnement du cerveau et la coordination des activités physiologiques [19].

6.1.2.Les réseaux de neurones artificiels :

Sont des systèmes inspirés du fonctionnement des neurones biologiques du cerveau. Composés de neurones artificiels interconnectés, ils résolvent divers problèmes grâce à des calculs mathématiques. Ces réseaux sont largement utilisés en intelligence artificielle pour des applications telles que la reconnaissance d'images, le diagnostic médical, le marketing ciblé, les prévisions financières, etc.

Ils permettent aux ordinateurs d'apprendre à partir de données et de prendre des décisions intelligentes en modélisant les relations complexes entre les entrées et les sorties [20].

6.1.3.Du réseau de neurone a réseau de neurone profond :

Les réseaux de neurones profonds, aussi appelés Deep Neural Networks, sont des structures complexes inspirées du fonctionnement du cerveau humain. Composés de plusieurs couches de neurones interconnectés, ils traitent des données complexes et effectuent des tâches sophistiquées en intelligence artificielle. Utilisés dans divers domaines tels que le traitement des données non structurées, la reconnaissance d'images, la détection de fraudes, la santé, l'aviation, les recommandations personnalisées, etc., ces réseaux apprennent à partir de données et améliorent leurs prédictions au fil du temps. Leurs capacités avancées d'automatisation et de prise de décision contribuent à transformer divers secteurs d'activité [21].

6.2. Les architectures d'apprentissage :

Les architectures d'apprentissage mentionnées sont des concepts clés en intelligence artificielle :

6.2.1.Réseaux neuronaux récurrents(RNN) :

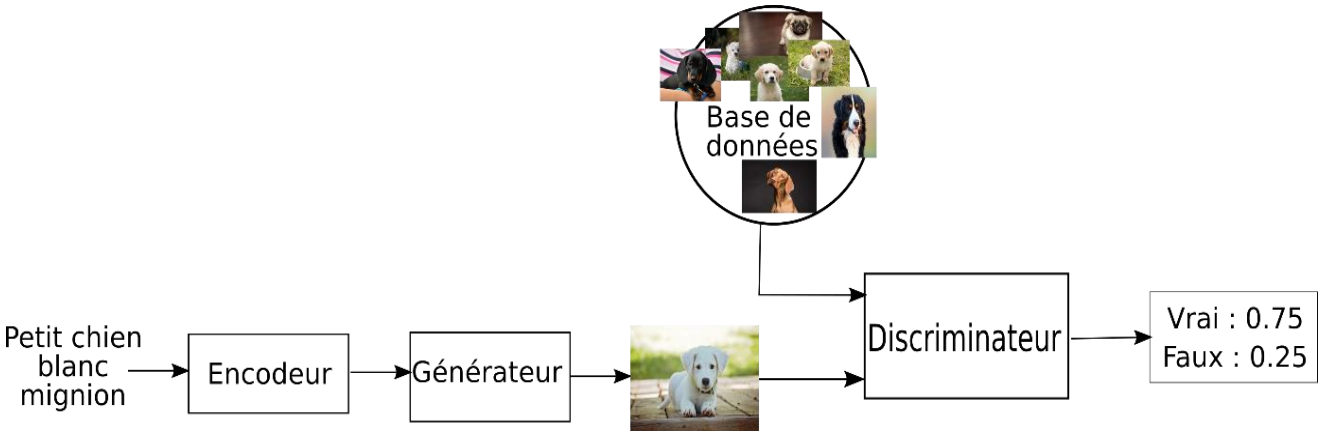
Ces réseaux sont conçus pour traiter des données séquentielles en conservant une mémoire des informations précédentes, ce qui les rend efficaces pour des tâches telles que la traduction automatique et la génération de texte [22].

Cette figure montre une architecture de réseau de neurones où une couche de neurones récurrents (RNN) précède une couche de neurones traditionnels (feedforward). Les neurones récurrents sont interconnectés, permettant le traitement séquentiel des données. La couche de neurones traditionnels reçoit les sorties de la couche RNN pour des transformations finales. Cette configuration est idéale pour des tâches comme le traitement du langage naturel et la prévision de séries temporelles. Comme on peut le voir dans la figure [72]:

Figure 6: Couche de neurones récurrents devant une couche de neurones traditionnel [72].

6.2.2.Réseaux adversariaux génératifs(GAN) :

Les GAN sont utilisés pour générer des données réalistes à partir de données brutes, ce qui est



particulièrement utile en vision par ordinateur et en création artistique [23], Cette figure illustre l'architecture d'un réseau adversarial génératif (GAN) [73].

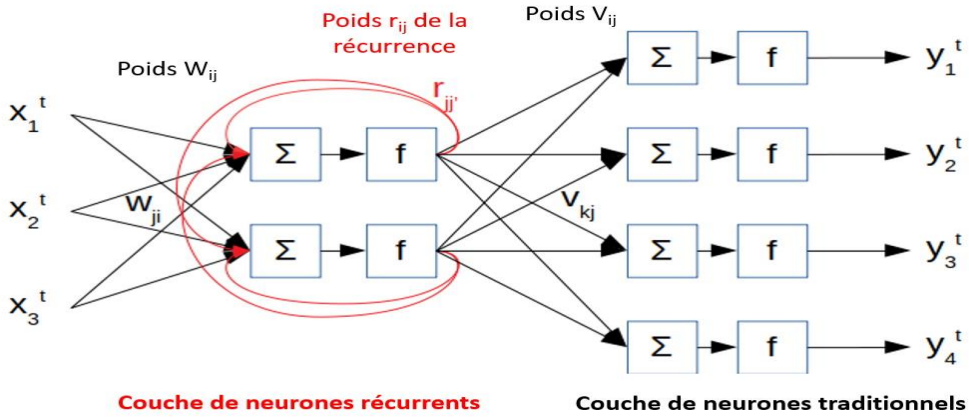


Figure 7: Architecture réseaux adversariaux génératifs (GAN) [73].

6.2.3.Réseaux neuronaux convolutifs(CNN) :

Les Réseaux neuronaux convolutifs (CNN)sont spécialement conçus pour le traitement d'images en extrayant des caractéristiques importantes à partir de celles-ci est dans la figure9 [9], les rendant efficaces pour la reconnaissance d'images et la vision par ordinateur [24].

Ils utilisent trois principaux types de couches pour accomplir leurs tâches [25] :

- **Couche de convolution** : Identifie les caractéristiques à l'intérieur des pixels.
- **Couche de mise en commun**: Résume les caractéristiques en vue d'un traitement ultérieur.
- **Couche entièrement connectée (FC)**: Utilise les caractéristiques acquises pour la prédiction.

La figure 8 illustre l'architecture d'un réseau de neurones convolutif (CNN) [74] :

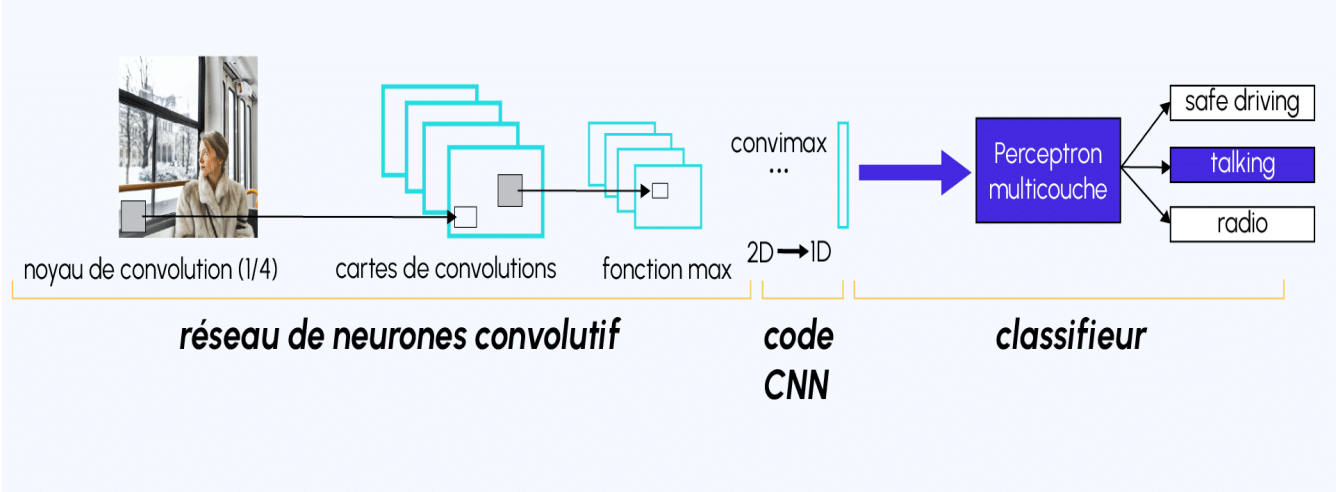


Figure 8: Réseaux convolutifs (CNN) [74].

6.2.4. Multicouche (MLP) :

Un Perceptron multicouche, ou MLP pour faire court, est un réseau neuronal artificiel avec plus d'une seule couche. Il possède une couche d'entrée qui se connecte aux variables d'entrée, une ou plusieurs couches masquées et une couche de sortie qui produit les variables de sortie [26].

Cette figure présente la structure d'un réseau multicouche (MLP), également connu sous le nom de réseau de neurones à propagation [75] :

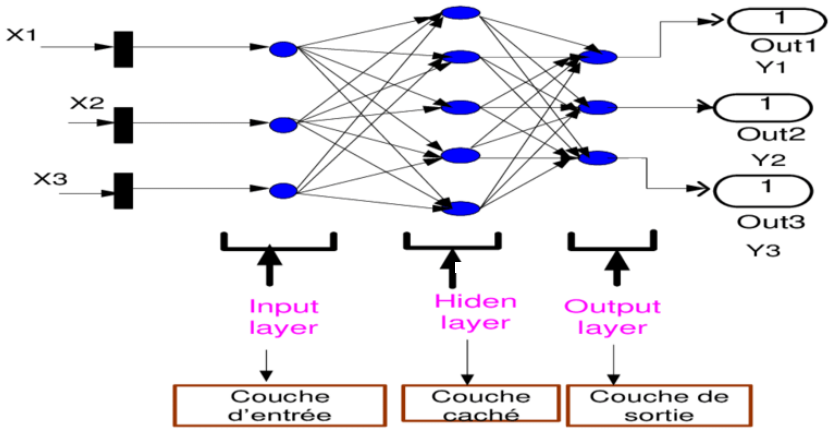


Figure 9: Topologie d'un réseau multicouche (MLP)[75].

7. Différence entre « Machine Learning » et « Deep Learning » :

- L'apprentissage automatique et l'apprentissage profond sont deux méthodes qui visent à entraîner des modèles à partir de données pour effectuer des prédictions sans intervention humaine.
- L'apprentissage profond est largement utilisé pour détecter des objets spécifiques dans des images, offrant des performances de reconnaissance élevées et une utilisation en temps réel.

- Il trouve une variété d'applications telles que la détection d'objets, l'estimation de distances et la prédiction de comportements, avec une intégration simple et à moindre coût nécessitant peu d'équipements.
- Contrairement à l'apprentissage automatique, où les caractéristiques doivent être définies manuellement, l'apprentissage profond identifie automatiquement les caractéristiques à apprendre.
- Les algorithmes de machine learning reposent principalement sur des données structurées, tandis que l'apprentissage profond utilise des réseaux de neurones artificiels.
- Dans le deep learning, seuls les réseaux de neurones sont utilisés, tandis que dans le machine learning, ils sont une des approches parmi d'autres.
- Le deep learning représente une évolution du machine learning, permettant aux machines de prendre des décisions précises sans intervention humaine [27].

7. Discussion :

Dans ce chapitre, nous avons exploré la vision artificielle qui est l'un des domaines de l'intelligence artificielle le plus utilisé. Pour approfondir notre compréhension de ce domaine, nous avons abordé la détection d'objets, qui figure parmi les applications pratiques les plus intéressantes dans la vie quotidienne. Dans le domaine de l'IA, nous rencontrons deux types d'apprentissage plus connus : l'apprentissage automatique et l'apprentissage profond, chacun avec ses propres méthodes distinctes. Ainsi, nous avons introduit le concept de réseaux de neurones pour une meilleure compréhension du principe d'apprentissage profond. Dans le prochain chapitre, nous allons nous intéresser à l'architectures de l'apprentissage profond (Deep Learning) qui sera la partie principale de notre système.

Chapitre 2 :

Méthodes de détection d’objets

1. Préambule :

Dans le domaine de la vision par ordinateur, plusieurs techniques sont utilisées, parmi lesquelles se trouve la détection d'objets. Ce domaine de recherche est particulièrement actif en raison de sa combinaison de la classification et de la localisation, et par extension, de l'identification, en utilisant différents algorithmes liés aux modèles de détection d'objets.

Le principe de la détection d'objets, ou des termes alternatifs tels que la reconnaissance d'objets, la classification d'images, la localisation d'objets et la segmentation d'images, repose sur la capacité à identifier et à localiser des objets spécifiques dans des images ou des vidéos. Pour obtenir une méthode efficace de détection d'objets, il est essentiel de disposer à la fois d'un algorithme robuste de détection des régions et d'un algorithme précis de classification d'images.

Il existe différents algorithmes pour la détection d'objets, certains sont des techniques classiques tandis que d'autres sont des techniques avancées développées récemment. Dans ce chapitre, nous présentons les divers algorithmes classiques et modernes de détections d'objets.

2. Les modelés de la détection d'objets :

Les modèles de détection d'objets, qu'ils soient classiques ou modernes, sont des algorithmes utilisés en vision par ordinateur pour localiser et identifier des objets spécifiques dans une image ou une vidéo. Voici une définition pour chacun :

2.1. Modèles de détection d'objets classiques :

Les diverses méthodes de détection du mouvement sont soumises à deux contraintes majeures, à savoir l'éclairage de la scène et le mouvement du capteur, comme celui d'une caméra. En effet, la dynamique de la situation varie en fonction de la stabilité du capteur, qu'il reste immobile, qu'il se déplace lentement ou qu'il soit mobile avec un mouvement compensé. Ainsi, parmi les différentes approches, nous avons sélectionné celles qui répondent le mieux à notre objectif de suivi en temps réel d'objets, ce qui implique de passer par la détection d'objets [28].

2.1.1. Détection basée sur la différence inter image :

La détection basée sur la différence inter-images compare deux images successives pour trouver les différences entre elles, comme montré dans la figure 10. Ces différences peuvent être causées par le mouvement d'un objet ou par des changements dans l'environnement [10]. Elles apparaissent sous différentes formes et tailles. Cependant, cette méthode peut être sensible au bruit et aux changements de luminosité [29].

Cette figure illustre les étapes clés du processus de détection d'inter-image [76]:

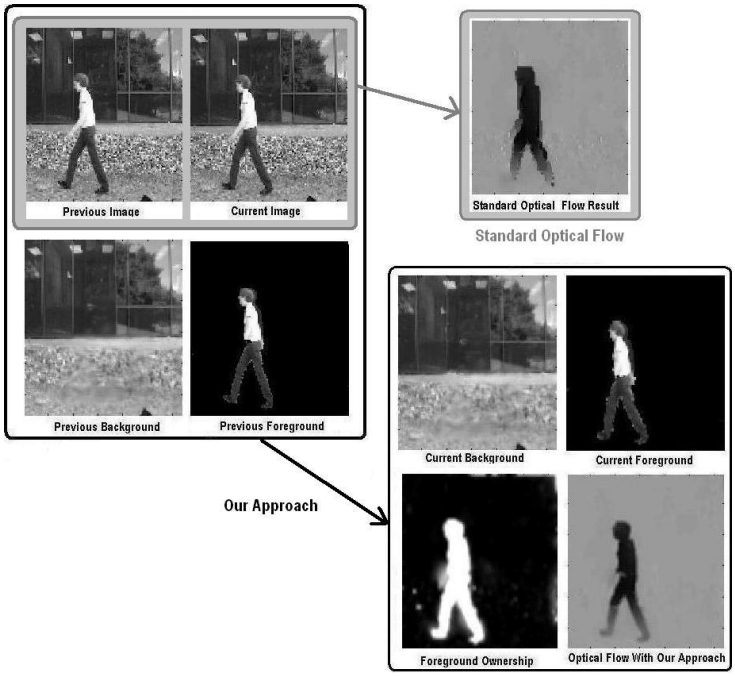


Figure 10: différentes étapes de la détection d'inter-image [76].

2.1.2. Détection basée sur la modélisation du fond :

La détection basée sur la modélisation du fond consiste à modéliser l'arrière-plan d'une scène. Comme indiqué dans la figure 11, la soustraction du fond permet de détecter l'objet en mouvement comme une anomalie par rapport au modèle de l'arrière-plan [77]. Cette technique est généralement plus robuste au bruit et aux changements de luminosité que les techniques basées sur la différence inter-images. Cependant, elle peut être plus complexe à mettre en œuvre [30]. Cette figure présente le processus de segmentation d'images par soustraction de fond :

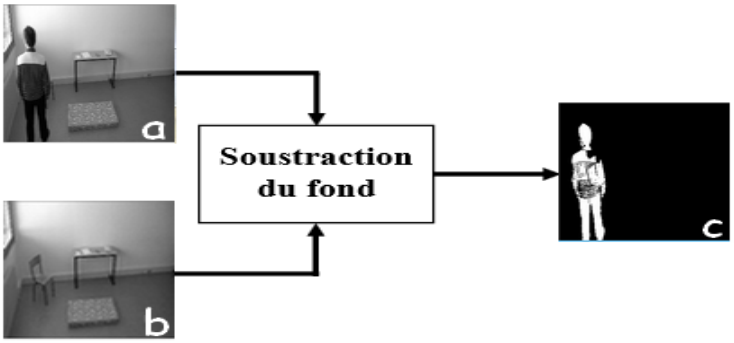


Figure 11: Segmentation d'images par soustraction de fond.

(a) Image courante. (b) Image du fond. (c)

2.1.3.Les détections utilisant la notion de cohérence :

Un mouvement cohérent peut être caractérisé comme un déplacement ayant de fortes probabilités d'être attribué à une "cible" classique, comme une personne ou un véhicule. Cette définition a été exploitée dans le contexte de la segmentation du mouvement en introduisant l'hypothèse suivante : Un objet en mouvement cohérent se déplace dans une direction approximativement constante sur une courte période, généralement quelques images en pratique. La distinction et l'avantage ici résident dans la détection exclusive des objets cohérents, permettant ainsi l'application directe d'autres procédés tels que l'identification ou le suivi [31].

2.2. Les modèles de détection moderne par Réseaux neuronaux convolutifs (CNN) :

En apprentissage profond, un réseau de neurones convolutifs (CNN) est une classe de réseaux de neurones profonds, principalement utilisés pour analyser les images visuelles. Contrairement aux réseaux de neurones classiques, les CNN utilisent une technique spéciale appelée convolution, qui est une opération mathématique sur deux fonctions produisant une troisième fonction. Cela permet d'exprimer comment la forme de l'une est modifiée par l'autre [32].

Dans la détection en deux étapes, qui inclut des modèles bien connus tels que RCNN, SPP-NET, FRCNN et Faster RCNN, le processus se divise en deux phases distinctes : la proposition initiale de régions et la classification suivie du raffinement de la localisation des objets. Cette approche permet un ajustement plus précis des positions des objets détectés, ce qui peut améliorer la précision, mais elle peut aussi être plus lente en raison de ces deux étapes séparées.

En revanche, la détection en une étape, qui comprend des modèles comme MultiBox, AttentionNet, G-CNN, YOLO, SSD, etc., combine la localisation et la classification des objets en une seule étape. Cela rend cette méthode généralement plus rapide, car il n'y a pas de phase distincte de proposition de régions. Cependant, cette approche peut parfois être moins précise, car elle peut manquer de détails fins dans la localisation des objets ou dans la distinction entre des objets très proches les uns des autres [33].

La figure 12 illustre les différents algorithmes de détection d'objets basés sur CNN [78].

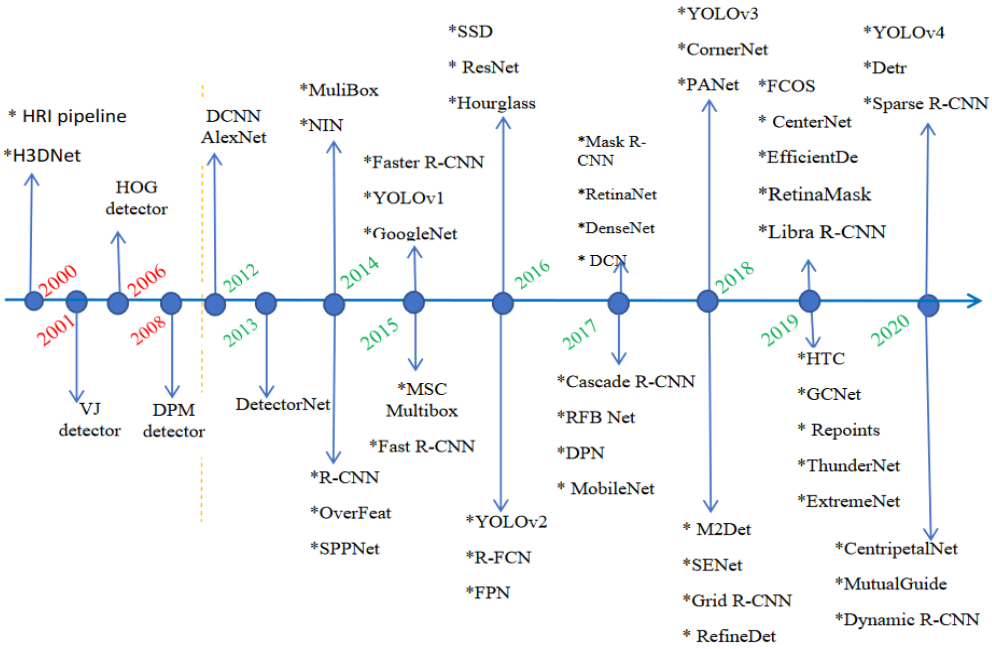


Figure 12: Algorithmes de détection d'objets par CNN [78].

2.2.1.Region-based Convolutional Neural Network (R –CNN) :

Région-CNN (R-CNN) est une approche de détection d'objets utilisant des réseaux de neurones convolutifs (CNN), proposée par Girshick et al en 2014. Elle combine des propositions de régions avec des CNN pour détecter des objets dans une image via des boîtes de délimitation. R-CNN fonctionne en trois étapes principales :

1) Il utilise la recherche sélective pour générer environ 2000 propositions de régions par image
2) Il utilise un CNN pour classifier chaque région proposée
3) Il affine les boîtes de délimitation à l'aide de la régression.

Les régions proposées sont redimensionnées pour être compatibles avec le CNN, et les caractéristiques extraites sont classées par des machines à vecteurs de support (SVM).

Il présente un inconvénient majeur de lenteur, principalement en raison du traitement individuel de chaque région et de la complexité de son pipeline. Pour résoudre ce problème, Fast R-CNN a été introduit pour accélérer le processus en propageant l'image entière à travers le CNN une seule fois pour obtenir une carte de caractéristiques, puis en extrayant des vecteurs de caractéristiques pour chaque région proposée [34].

Cette figure présente l'architecture de R-CNN (Region-based Convolutional Neural Network) [79] :

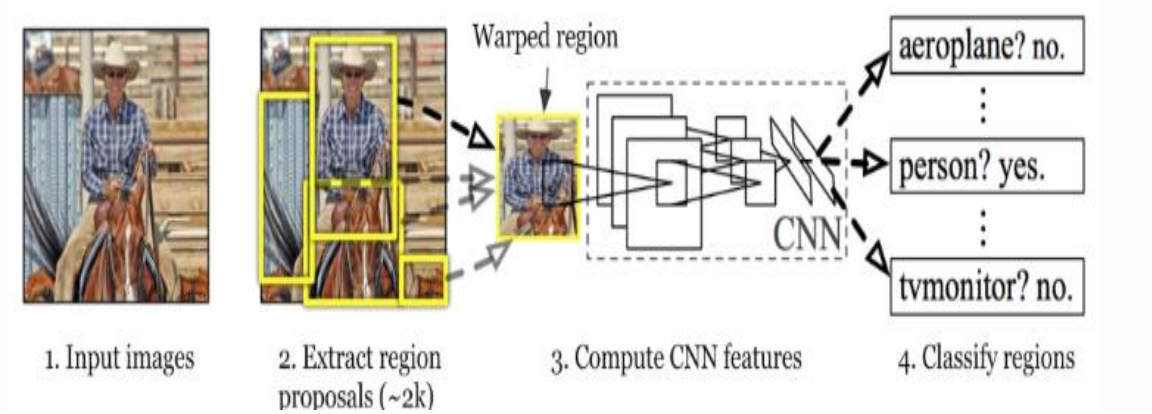


Figure 13: Architecture R-CNN [79].

2.2.2.Fast Region-based Convolutional Neural Network (Fast R-CNN) :

En 2015, Ross Girshick a introduit Fast R-CNN pour accélérer et améliorer la précision de R-CNN. Au lieu de transmettre chaque proposition de région au CNN séparément, Fast R-CNN propage l'image entière pour générer une carte de caractéristiques convolutives. À partir de cette carte, les propositions d'objets sont identifiées et redimensionnées en une taille fixe à l'aide d'une couche de regroupement des régions d'intérêt (RoI pooling layer). Cette couche utilise un algorithme de max pooling pour transformer chaque région d'intérêt (RoI) en une carte de caractéristiques de taille fixe. Ensuite, ces caractéristiques sont passées à une couche entièrement connectée. Finalement, une couche softmax prédit la classe de chaque région proposée et ajuste les boîtes englobantes. En procédant ainsi, Fast R-CNN effectue l'opération de convolution une seule fois par image, ce qui le rend beaucoup plus rapide que R-CNN, tout en augmentant la précision des détections d'objets [35]. Cette figure illustre le pipeline de Fast R-CNN, une méthode améliorée pour la détection d'objets dans les images, basée sur l'architecture R-CNN [80] :

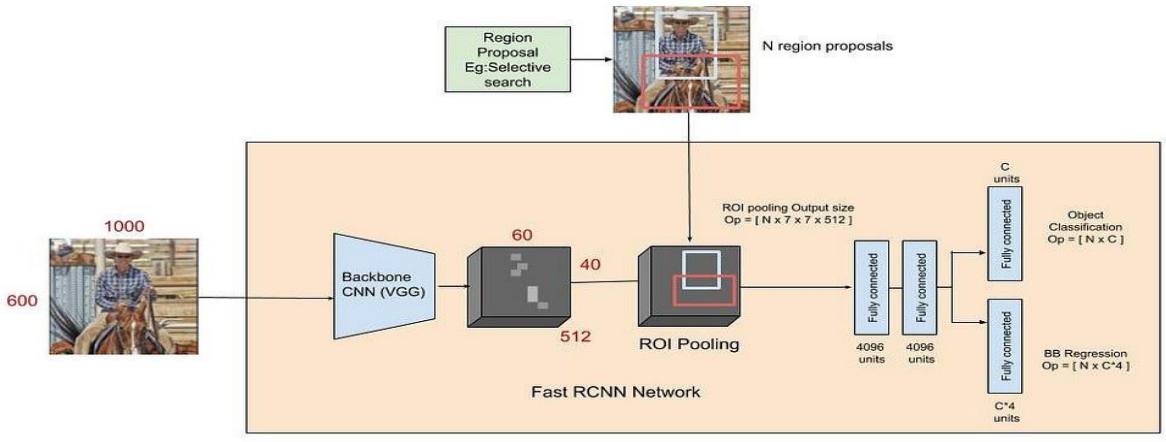


Figure 14: Pipeline Fast R-CNN [80].

2.2.3.Faster Region-based Convolutional Neural Network (Faster R-CNN):

Faster R-CNN, introduit en 2015 par l'équipe Microsoft Research, améliore la vitesse et la précision de la détection d'objets par rapport à Fast R-CNN. Semblable à Fast R-CNN, Faster R-CNN commence par fournir l'image en entrée à un réseau convolutif pour générer une carte de caractéristiques convolutives. Cependant, au lieu d'utiliser un algorithme de recherche sélective pour identifier les propositions de région, il utilise un Réseau de Propositions Régionales (RPN) pour prédire ces propositions. Les propositions de région sont ensuite remodelées avec une couche de regroupement des régions d'intérêt (RoI pooling layer), puis classées et ajustées pour les boîtes englobantes [36].

Le RPN fonctionne en glissant une fenêtre spatiale sur la carte de caractéristiques entière, créant des boîtes d'ancrage de différentes échelles et rapports d'aspect. Cette approche permet d'alterner l'apprentissage entre la proposition de région et la détection d'objets, tout en partageant les caractéristiques convolutives, ce qui accélère la convergence de l'apprentissage [37].

Faster R-CNN surpasse ses prédécesseurs en termes de vitesse et de précision, atteignant un mAP de 73,2% sur Pascal VOC 2007 et de 70,4% sur Pascal VOC 2012. Ces améliorations résultent de l'intégration du RPN, qui remplace la recherche sélective plus lente, permettant ainsi une détection d'objets plus efficace et précise [38].

Cette figure présente l'architecture de Faster R-CNN[81] :

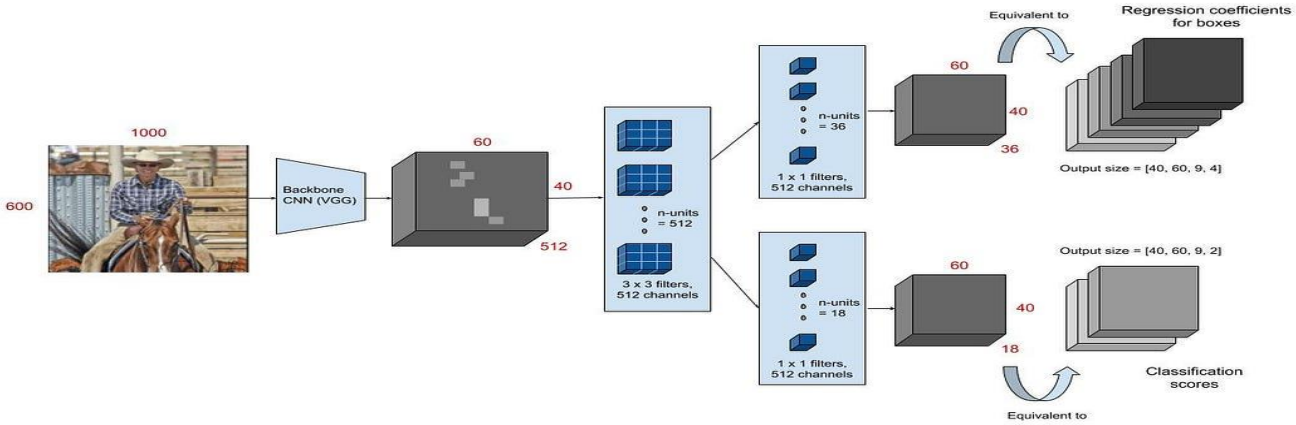


Figure 15: Faster R-CNN for object detection [81].

2.2.4.Mask Region-based Convolutional Neural Network (Mask R-CNN):

Le Mask R-CNN, introduit par Facebook AI Research en 2017, représente une avancée majeure par rapport au Faster R-CNN en introduisant l'instance segmentation. Conçu pour répondre aux besoins de ceux qui recherchent à la fois la détection d'objets précise et la segmentation fine des objets, ce modèle étend l'architecture du Faster R-CNN [39].

Il commence par extraire des caractéristiques des images pour former une carte de caractéristiques spatiales. Un Réseau de Propositions Régionales (RPN) génère ensuite des boîtes englobantes autour des objets potentiels, avec des scores de confiance pour chaque classe. Ce qui distingue le Mask R-CNN est l'ajout d'une branche parallèle dédiée à la segmentation : elle prédit des masques binaires pixel par pixel pour chaque objet détecté, permettant une segmentation fine. Pendant l'entraînement, le modèle est optimisé pour la classification des objets, l'ajustement précis des boîtes englobantes et la génération précise des masques de segmentation, réduisant ainsi une fonction de perte combinée [40].

L'avantage clé du Mask R-CNN réside dans sa capacité à résoudre simultanément le problème des boîtes englobantes et celui de la segmentation sémantique. Cette intégration offre une solution robuste pour la segmentation précise des objets, ce qui en fait un choix attractif pour divers domaines de recherche et d'application nécessitant une analyse fine des objets dans les images et les vidéos.

Cette figure illustre l'architecture de Mask R-CNN, une extension de Faster R-CNN qui ajoute la capacité de segmentation sémantique précise :

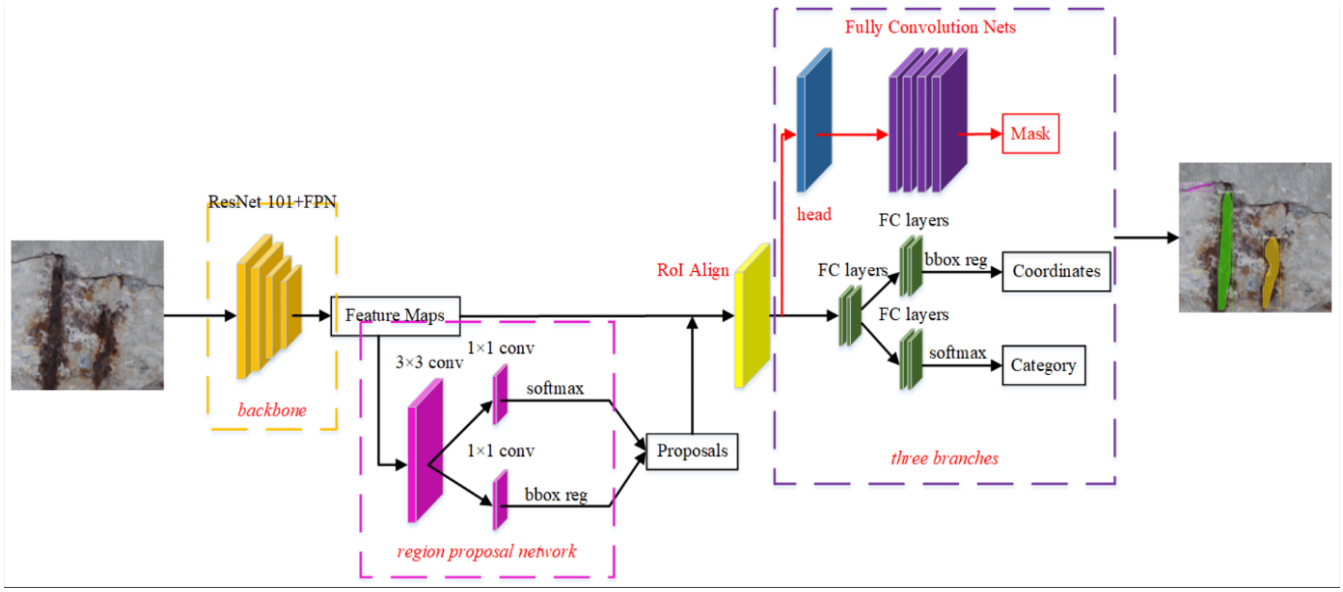


Figure 16: Mask R-CNN par exemple la segmentation [82].

2.2.5. Single Shot MultiBox Detector (SSD):

L'approche de la SSD est basée sur un réseau convolutif feed-forward qui produit une collection de boîtes englobantes de taille fixe et des scores pour la présence d'instances de classes d'objets dans ces boîtes, suivi d'une étape de suppression non maximale pour produire les détections finales Les premières couches du réseau sont basées sur une architecture standard utilisée pour la classification d'images de haute qualité (VGG-16) (tronquée avant toute couche de classification), qui s'appelle réseau de base [41].

En résumé, la figure 18 montre que le modèle SSD intègre de manière efficace des couches convolutionnelle multi-échelles pour une détection d'objets rapide et précise [18].

Ensuite une structure auxiliaire au réseau pour produire des détections avec les caractéristiques clés suivantes :

✓ Cartes de caractéristiques multi-échelles pour la détection.

✓ Prédicteurs convolutifs pour la détection.

✓ Boîtes et aspect par défaut.

Le résultat obtenu du modele SSD512 entrainer sur COCO trainval35k puis affiner en pascal voc 2007+2012 c'est le meilleur résultat : 81,6% mAP, et 68.0% mAP avec le model SSD300 apprend sur voc 2007, et le résultat obtenu des modelés SSD300 et SSD512 entrainer sur la base de données COCO trainval35k est 41.2% mAP et 46.5% Map [47].

La figure 18 présente l'architecture du SSD (Single Shot MultiBox Detector)[83] :

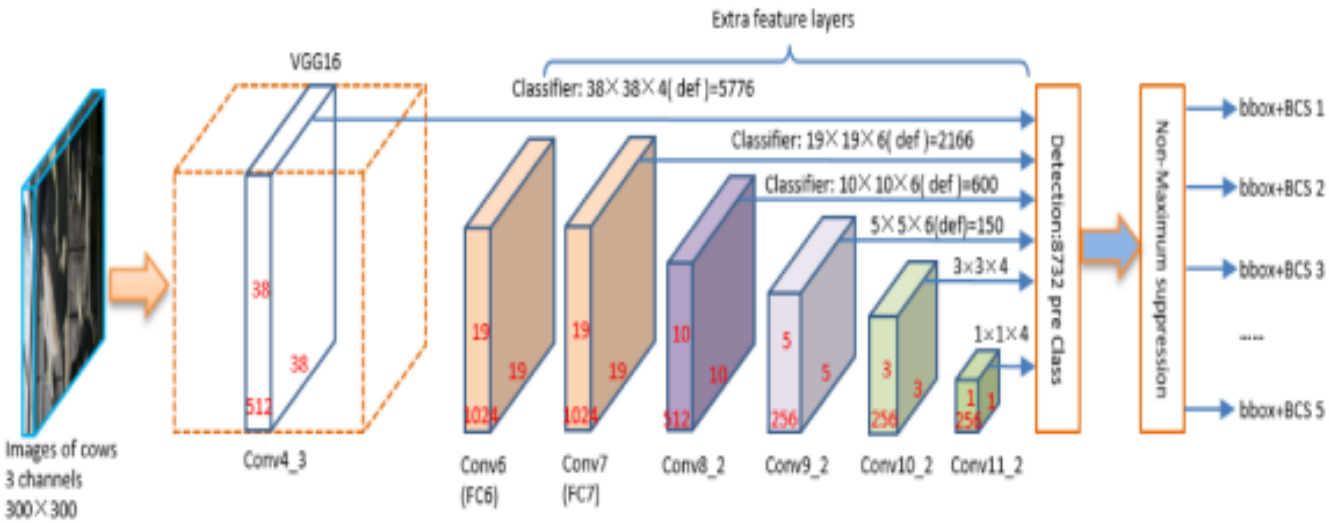


Figure 17: architecture de model SSD [83]

2.2.6.You Only Look Once (YOLO):

YOLO (You Only Look Once) est un algorithme de détection d'objets conçu et amélioré par ses créateurs Joseph Redmon et Ali Farhadi ,Il utilise un réseau neuronal convolutif appelé Darknet, basé sur GoogLeNet, mais modifié pour intégrer directement les phases de détection et de classification sans nécessiter de réseau secondaire. Contrairement aux approches traditionnelles, YOLO entraîne son modèle sur l'image complète, assurant ainsi que les informations spatiales pertinentes sont conservées à chaque couche du réseau pour éviter les faux négatifs. L'image d'entrée est divisée en cellules de 32x32 pixels, où YOLO traite jusqu'à 5 cadres d'ancrage par cellule, déterminés par un algorithme de clustering k-means avec une mesure d'intersection sur union (IOU). Chaque cadre d'ancrage est utilisé pour estimer la présence d'un objet et calculer la précision de localisation par rapport à la taille et à la classe prédite. En réduisant les caractéristiques dans ses couches de convolution finales, YOLO produit une grille d'ancrages où chaque cellule contient des cadres avec les plus hauts niveaux de confiance pour la présence d'un objet, ainsi que les confidences associées à chaque classe connue par le réseau.

Cette approche permet à YOLO d'effectuer la localisation et la classification d'objets de manière intégrée, tout en évitant les limitations de vitesse rencontrées par d'autres méthodes de détection d'objets [42].

La figure 18 le système divise l'image d'entrée en une grille $S \times S$ [84]. Si le centre d'un objet tombe dans une cellule de grille, cette cellule de grille est chargée de détecter cet objet. Chaque cellule de la grille prédit, B cadres de délimitation et scores de confiance pour ces cadres. Ces scores de confiance reflètent à quel point le modèle est sûr que la boîte contient un objet et également à quel point il pense que la boîte est précise.

Figure 18: Application de la grille [84].

Le score de confiance de la boîte englobante peut être obtenu comme suit :

$$score = Pr(objet) * IOU_{pred}^{truth}$$

Où $Pr(objet)$ est la probabilité qu'un objet se trouve à l'intérieur de la boîte englobante.

Figure 19 Est illustrer L'IOU_{pred}^{truth} signifie Intersection Over Union est la zone de l'intersection des boîtes de vérité prédite et terrain divisée par la zone de l'union [85] des mêmes boîtes de vérité prédite et terrain [43].

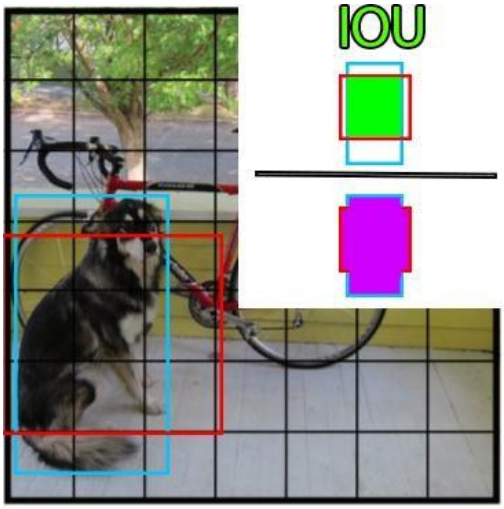


Figure 19: Exemple d'un IOU [85].

Dans la figure 20 Chaque cellule de la grille prédit également les probabilités de classe conditionnelles $C = Pr(class_i|objet)$. Ces probabilités sont conditionnées à la maille contenant un objet. YOLO prédit qu'un seul ensemble de probabilités de classe par cellule de grille, quel que soit le nombre de cases B [6] et s'il y a plusieurs objets de différentes classes dans une cellule de la grille, l'algorithme ne parviendra pas à les classer correctement [86]. Ainsi, chaque prédiction d'une cellule de grille sera de forme $C + B * 5$, où C et la probabilité de classes et B est le nombre de boîtes englobantes prédites. B est ici multiplié par 5 car il inclut $(x, y, w, h, confiance)$ pour chaque case. Puisqu'il y a des cellules de grille $S \times S$ dans chaque image, la prédiction globale du modèle est un tenseur de forme $S \times S \times (C + B * 5)$ [44].

Au moment du test, YOLO multiplie les probabilités de classe conditionnelles et les prédictions de confiance des boîtes individuelles,

$$Pr(class_i|objet) * Pr(objet) * IOU_{pred}^{truth} = Pr(class_i|IOU_{pred}^{truth})$$

Ce qui nous donne des scores de confiance spécifiques à la classe pour chaque boîte. Ces scores codent à la fois la probabilité de cette classe apparaissant dans la boîte et dans quelle mesure la boîte prédite correspond à l'objet [43].

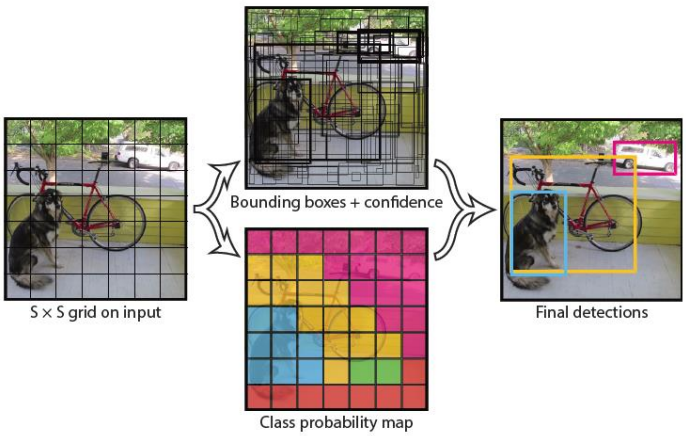


Figure 20: Exemple de détection avec YOLO [86].

3. Discussion :

Dans ce chapitre, nous nous concentrons sur l'utilisation de l'apprentissage profond pour détecter des objets dans des images complexes. Cette approche utilise des réseaux neuronaux comme les CNN, qui sont capables d'apprendre à reconnaître automatiquement des caractéristiques visuelles importantes, comme les formes et les textures des objets. Par exemple, dans des photos de circulation, les CNN peuvent être formés pour repérer précisément des voitures en se basant sur ces caractéristiques.

L'apprentissage profond utilise différents types de réseaux neuronaux adaptés à différentes tâches. Les CNN sont particulièrement efficaces pour extraire des informations visuelles, tandis que d'autres types comme les RNN sont utilisés pour des données séquentielles comme le langage, et les GAN pour générer des données réalistes.

Comparé aux méthodes traditionnelles d'apprentissage machine, qui nécessitent souvent des instructions précises sur les caractéristiques à rechercher, l'apprentissage profond est plus autonome et capable d'apprendre directement à partir des données. Cela le rend particulièrement efficace pour la détection d'objets dans des environnements complexes et variés.

Dans le prochain chapitre, nous explorerons différents modèles YOLO (You Only Look Once), notamment YOLOv5, choisi pour sa rapidité et sa précision dans la détection d'objets. Ce modèle sera intégré dans un système embarqué basé sur Raspberry Pi 4, offrant une solution robuste et autonome pour la détection d'objets dans divers contextes d'application.

Chapitre3 : Système embarqué de détection d’objets basé sur Deep Learning

1. Préambule :

Dans les chapitres précédents, nous avons étudié divers modèles de Deep Learning, notamment les réseaux de neurones convolutifs (CNN), pour la détection d'objets. Nous avons identifié YOLO (You Only Look Once) comme le modèle le plus adapté en raison de son efficacité et de sa rapidité.

Dans ce chapitre, nous examinerons les différentes versions de YOLO et comparerons leurs performances en termes de précision moyenne. Cette comparaison mettra en évidence les améliorations apportées à chaque version. Pour faciliter la détection d'objets sans connexion Internet, nous avons choisi une solution embarquée. Après avoir évalué plusieurs options, nous avons opté pour le Raspberry Pi en raison de ses caractéristiques techniques et de sa polyvalence. Nous présenterons les différentes versions du Raspberry Pi et les composants que nous avons utilisés pour réaliser notre système.

2. Les différentes versions du You Only Look Once (YOLO) :

2.1. Le modèle You Only Look Once version 1 (YOLO v1):

YOLOv1 est l'une des premières versions de YOLO, connue pour sa capacité à traiter les images en temps réel à une vitesse de 45 images par seconde. Ce modèle repose sur un "backbone", une partie du réseau de neurones composée de couches de convolution, de max pooling et de couches entièrement connectées. Le backbone est généralement pré-entraîné sur des tâches de classification d'images telles que VGG, ResNet ou Darknet pour YOLOv1 [44].

L'architecture de YOLOv1 s'inspire de GoogLeNet, avec 24 couches de convolution qui réduisent la taille de l'image d'entrée. Cette architecture inclut une couche entièrement connectée qui génère une grille de 7x7 cellules. Chaque cellule prédit deux boîtes englobantes avec leur confiance respective, ainsi que 20 prédictions de classe par cellule grâce à une régression logistique. La sortie finale est une grille de 7x7x30 valeurs.

Comme illustré dans la figure 21, YOLO (You Only Look Once) est une architecture avancée pour la détection d'objets. Elle divise l'image en une grille et effectue des prédictions simultanées pour chaque région, permettant une détection rapide et précise en une seule étape [87].

YOLOv1 a atteint une précision moyenne (mAP) de 63,4 % sur le jeu de données PASCAL VOC 07+12 [45].

Cette figure présente l'architecture de YOLO (You Only Look Once), une méthode de détection d'objets en temps réel dans les images [87].

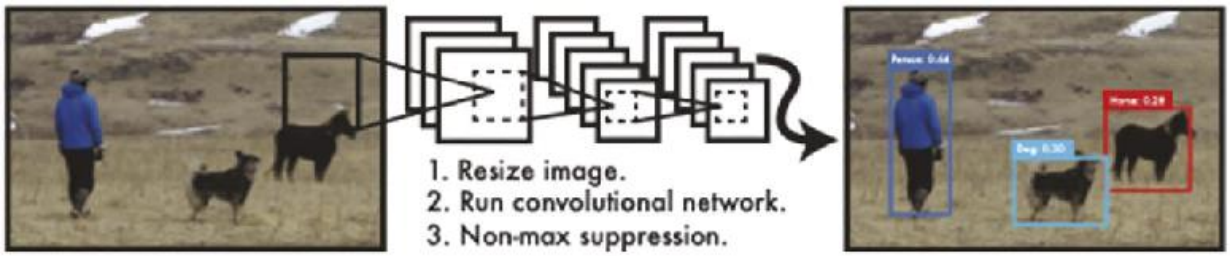


Figure 21: Système de détection YOLO [87].

Défauts de YOLOv1 :

Quand deux objets ont leurs centres dans la même cellule de la grille, YOLOv1 ne peut pas prédire deux objets et leurs catégories en même temps. Il ne peut prédire que la catégorie avec la probabilité la plus élevée, ce qui signifie qu'il ne reconnaîtra qu'un seul objet par cellule de la grille, même si plusieurs sont présents.

2.2. Le modèle You Only Look Once version 2(YOLO v2):

YOLOv2 améliore YOLOv1 en introduisant plusieurs innovations pour une meilleure détection et localisation des objets. Contrairement à YOLOv1, YOLOv2 prédit la valeur de la classe séparément pour chaque boîte de prédiction, permettant ainsi à une cellule de grille de produire plusieurs types de boîtes.

Ce modèle utilise DarkNet-19 au lieu de GoogLeNet, une architecture plus légère et efficace. DarkNet-19 est composé de 19 couches de convolution et 5 de max-pooling, intégrant des boîtes d'ancrage pour une détection améliorée. YOLOv2 augmente également la taille d'entrée des images de 224x224 à 416x416 pixels, comme illustré dans la figure 22 qui montre l'architecture de YOLOv2, particulièrement adaptée pour détecter les petits objets. De plus, le "neck" élargi de YOLOv2 permet de détecter des objets de différentes tailles [46].

Sur le jeu de données PASCAL VOC 07+12, YOLOv2 a atteint une précision moyenne (mAP) de 78,6%, représentant une amélioration significative par rapport à YOLOv1 qui avait une mAP de 63,4% sur le même jeu de données [46].

Cette figure illustre l'architecture de YOLOv2 (You Only Look Once version 2) [88]:

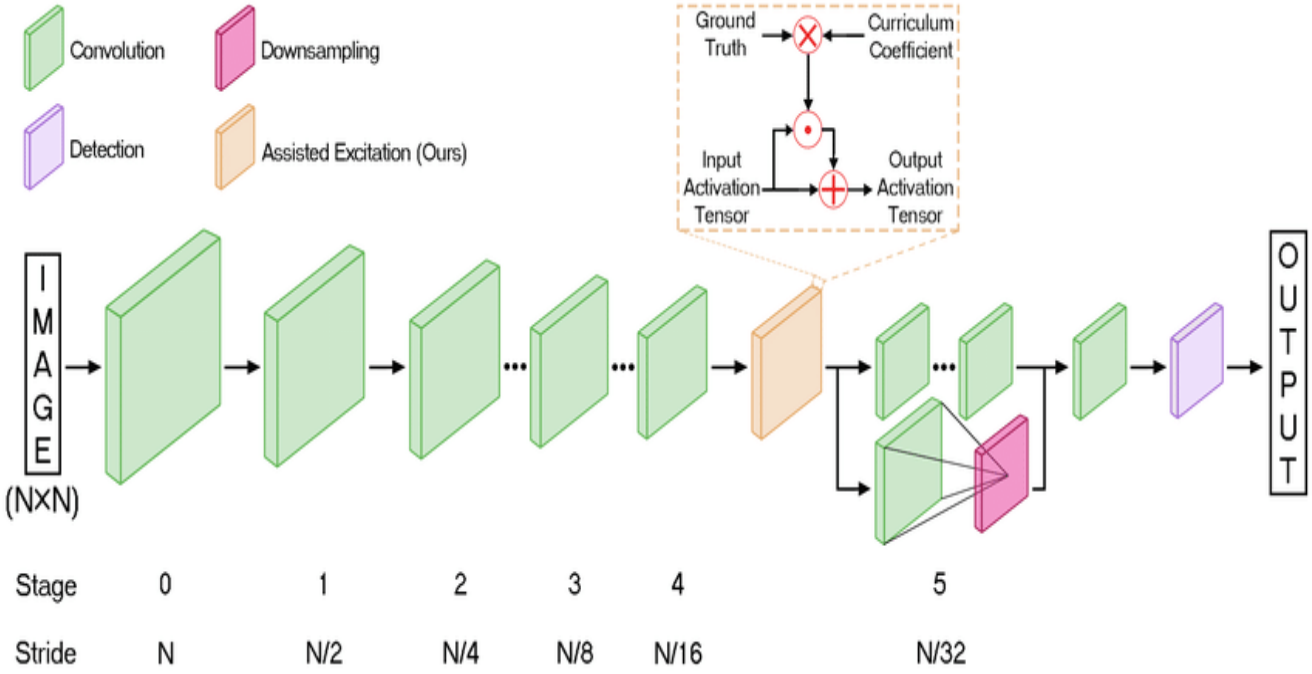


Figure 22: Architecture YOLOv2 [88].

2.3. Le modèle You Only Look Once version 3 (YOLOv3):

YOLOv3, la troisième itération de l'algorithme de détection d'objets, a progressivement amélioré la version précédente. Cette version utilise Darknet-53 comme backbone pour l'extraction des caractéristiques, comme illustré dans la figure 23, et calcule un score d'objectivité pour chaque boîte de délimitation à l'aide d'une régression logistique [89]. Les prédictions de classe sont effectuées en utilisant la perte d'entropie croisée binaire.

Pour surmonter les difficultés rencontrées dans YOLOv2 pour détecter de petits objets, YOLOv3 adopte une approche à trois échelles différentes. Cela améliore la représentation des boîtes en permettant au modèle de mieux capturer les détails des objets de différentes tailles [47].

Concernant l'architecture de YOLOv3 [48] :

L'architecture de YOLOv3 s'appuie sur un réseau appelé Darknet-53, qui comporte 53 couches de convolution, inspiré par ResNet. Chaque couche est suivie d'une normalisation par lot et utilise l'activation Leaky ReLU. YOLOv3 détecte les objets à trois niveaux différents (couches 82, 94 et 106), ce qui permet de reconnaître des objets de différentes tailles.

Pour y parvenir, il utilise une technique appelée Feature Pyramid Network (FPN). Cette méthode combine des caractéristiques de différentes échelles avec celles des niveaux précédents pour améliorer la précision de la détection. L'image est divisée en une grille, et chaque cellule de cette grille prédit

plusieurs boîtes englobantes avec des probabilités de classe, en se basant sur des boîtes d'ancrage prédéfinies de différentes tailles et ratios.

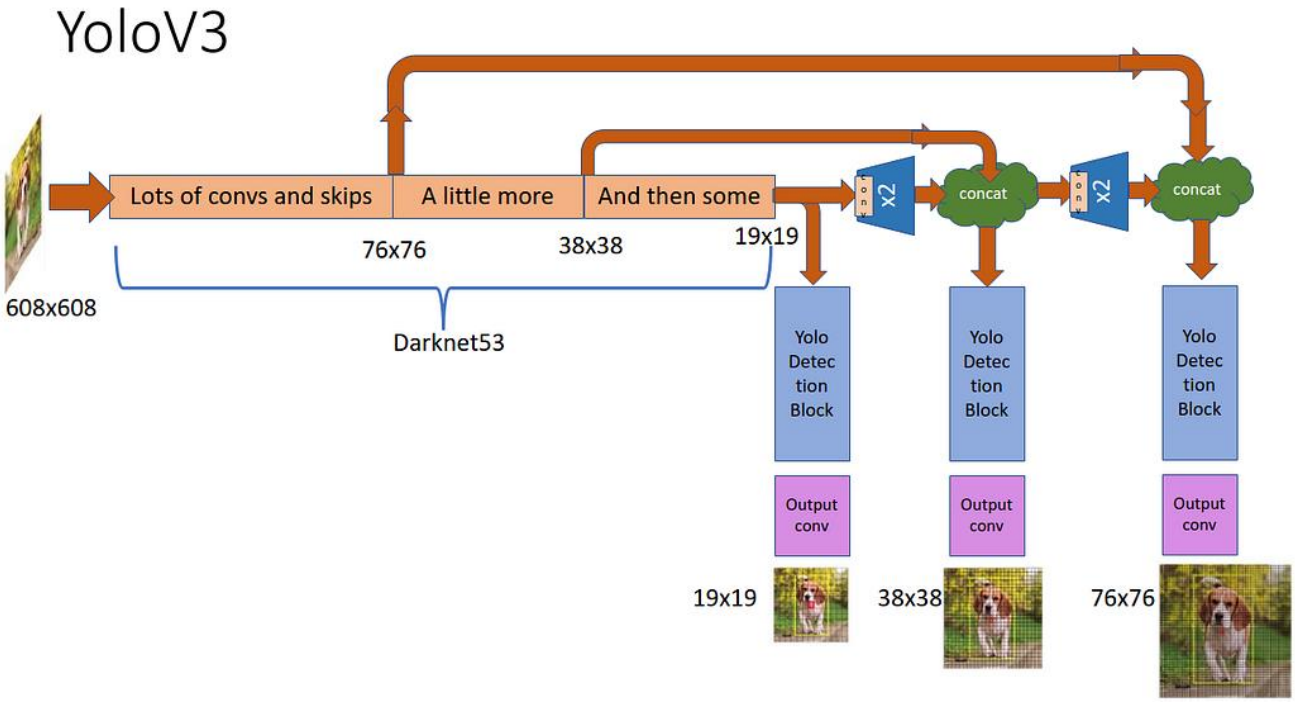


Figure 23:Architecture YOLO-V3[89].

2.4. Le modèle You Only Look Once version 4 (YOLOv4):

YOLOv4 utilise deux techniques principales : « Spatial Pyramid Pooling » (SPP) pour extraire des caractéristiques de l'image à différentes échelles et résolutions, ce qui permet de détecter des objets de différentes tailles, et « Cross Stage Partial » (CSP) pour améliorer la précision du modèle. De plus, il combine les prédictions de plusieurs modèles avec des architectures et des échelles différentes pour obtenir une meilleure précision.

Concernent l'architecture de YOLOv4 [49] :

- YOLOv4 utilise un backbone appelé CSP-Darknet53, une version améliorée de Darknet53 utilisée dans YOLOv3, avec l'ajout de la technique Cross Stage Partial Network (CSP) pour améliorer l'efficacité.
- Le cou de YOLOv4 comprend deux éléments clés :
  - **Spatial Pyramid Pooling (SPP)** : Capturer des caractéristiques à différentes échelles.
  - **Path Aggregation Network (PAN)** : Fusionner les caractéristiques de différentes couches.

- La tête de détection de YOLOv4 est similaire à celle de YOLOv3, réalisant des prédictions à trois échelles différentes en utilisant les caractéristiques extraites par le cou. Chaque prédiction d'échelle produit un tenseur de sortie avec les dimensions suivantes :
  (batch_size, 3 * (5 + num_classes), grid_size, grid_size)

  - **batch_size** est la taille du lot.
  - **3** correspond aux 3 boîtes d'ancrage par cellule.
  - **5** inclut les coordonnées x, y, la largeur, la hauteur et le score de confiance.
  - **num_classes** est le nombre de classes d'objets.
  - **grid_size** est la taille de la grille de prédiction.

- YOLOv4 introduit plusieurs innovations pour améliorer les performances :
  - **Weighted Residual Connections (WRC) :** Connexions résiduelles pondérées
  - **Cross Stage Partial Network (CSP) :** Réseau partiel inter-étages
  - **Cross mini-Batch Normalization (CmBN) :** Normalisation croisée par mini-lots
  - **Self-Adversarial Training (SAT) :** Entraînement auto-adversarial

Dans la figure 24 Schéma de l'architecture de YOLOv4. Présentation de la conception complexe du réseau de YOLOv4, y compris les composants de l'épine dorsale, du cou et de la tête, ainsi que leurs couches interconnectées pour une détection optimale des objets en temps réel [90].

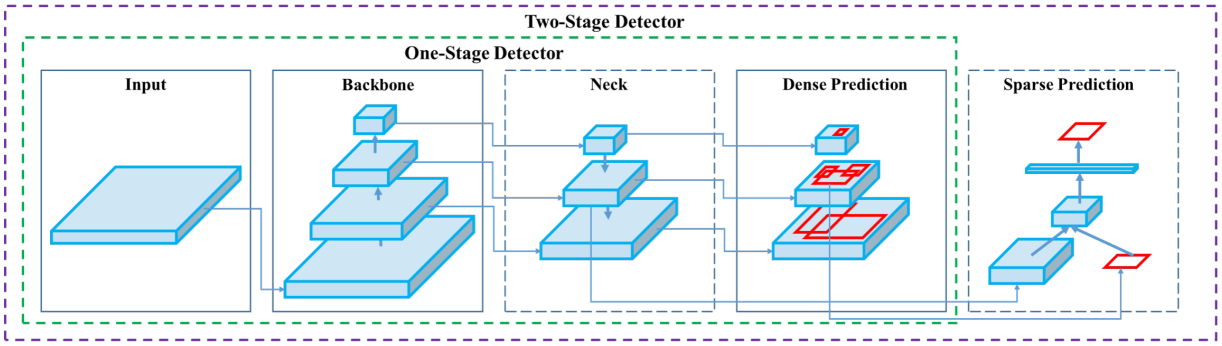


Figure 24: Architecture yolov4 [90].

2.5. Le modèle You Only Look Once version 5 (YOLOv5):

YOLOv5, la cinquième itération de l'approche YOLO, a été présentée par Glenn Jocher le 9 juin 2020. Cette version se distingue comme une référence majeure et une source d'inspiration pour les modèles de détection d'objets en un seul passage. Comparativement aux modèles à deux étages, YOLOv5 se distingue par sa rapidité d'exécution tout en maintenant une précision élevée.

Cette itération représente une extension de YOLOv3, qui était déjà prisée par les développeurs pour transposer les poids de Darknet YOLOv3 vers PyTorch. YOLOv4 est sortie juste avant YOLOv5, et celle-ci a été renommée pour éviter toute confusion entre les versions [50].

YOLOv5 est proposé en quatre variantes principales : small (S), medium (M), large (L) et extra-large (X). Chacune de ces versions offre un niveau de précision graduellement plus élevé, tout en nécessitant des temps d'entraînement différents pour être pleinement opérationnelles.

Dans la figure 25, chaque version de YOLO est comparée en termes de précision et d'amélioration des performances des procédures d'entraînement de PyTorch, avec le temps d'inférence sur l'axe X. YOLOv5 peut traiter des images rapidement et avec une précision presque identique à celle du modèle EfficientDet-D4.

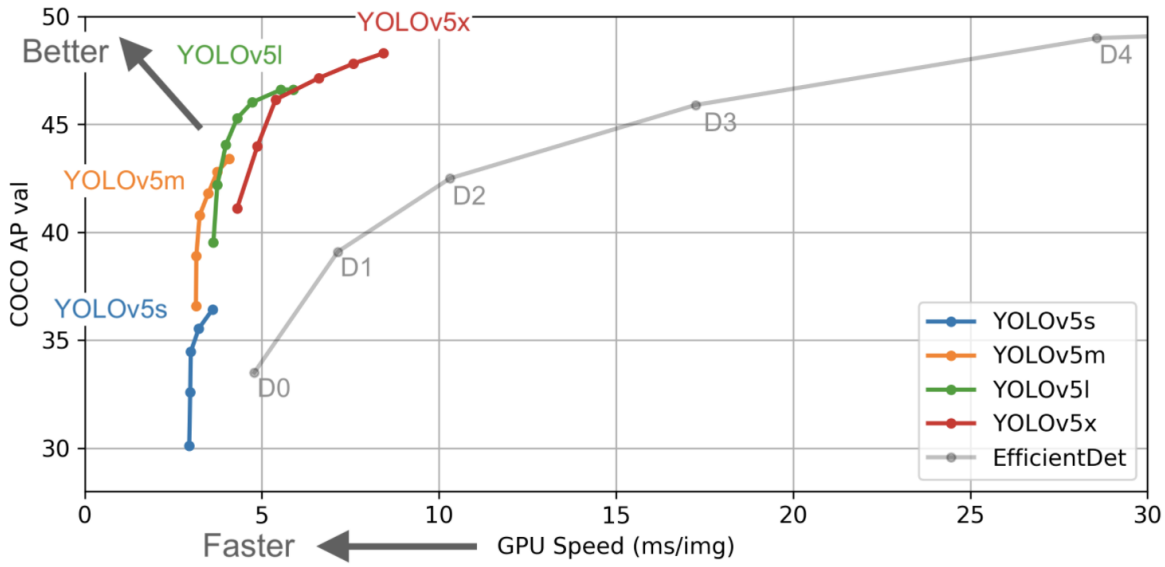


Figure 25: La version initiale de YOLOv5 montre la promesse d'une détection d'objets de pointe (citer le dépôt YOLOv5) [91].

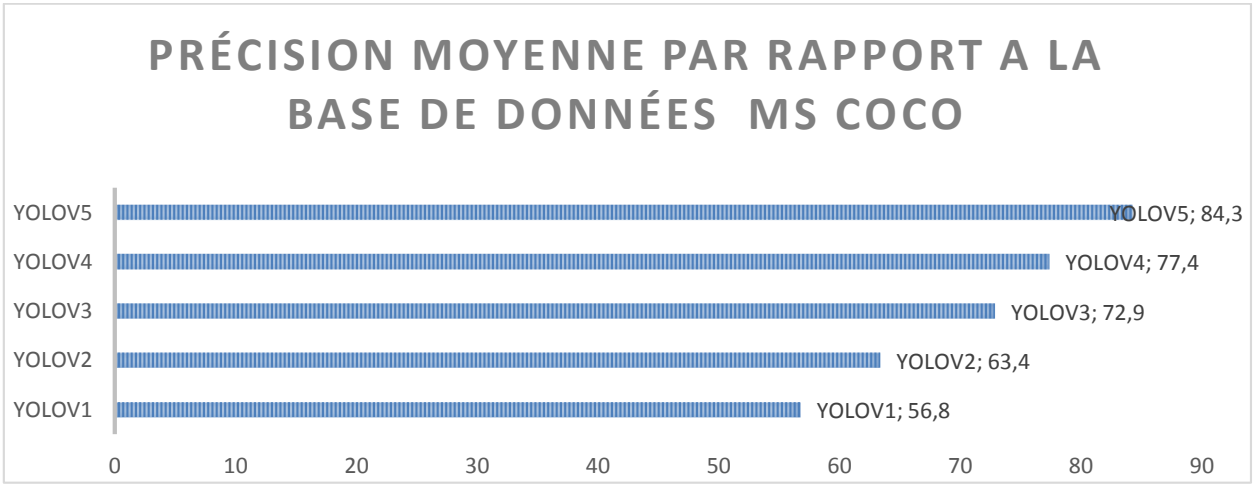


Figure 26:précision moyenne par rapport à la base de données MS COCO .

En analysant la figure 26, on peut constater une amélioration progressive des taux de précision moyenne (mAP) avec chaque nouvelle version de YOLO. YOLOv1, la version initiale, montre un taux de précision inférieur par rapport aux versions ultérieures. YOLOv2 et YOLOv3 apportent des

améliorations significatives, tandis que YOLOv4 et YOLOv5 montrent les meilleures performances, avec des mAP plus élevés.

2.5.1.Architecture YOLOV5

- **Architecture globale [51] :**

Dans YOLOv5, l'image est d'abord traitée par une couche d'entrée, puis envoyée au "backbone" pour extraire les caractéristiques de l'image. Le backbone produit des cartes de caractéristiques de différentes tailles, qui sont ensuite fusionnées via un réseau de fusion de caractéristiques (le "neck"). Cela génère trois cartes de caractéristiques, P3, P4 et P5, avec des dimensions respectives de 80×80, 40×40 et 20×20 dans YOLOv5. Chaque carte est utilisée pour détecter les petits, moyens et grands objets dans l'image.

Dans la figure 27, on voit qu'EfficientDet utilise un backbone EfficientNet et une structure de réseau de fusion des caractéristiques, appelée BiFPN (Bi-directional Feature Pyramid Network), pour améliorer la performance [92].

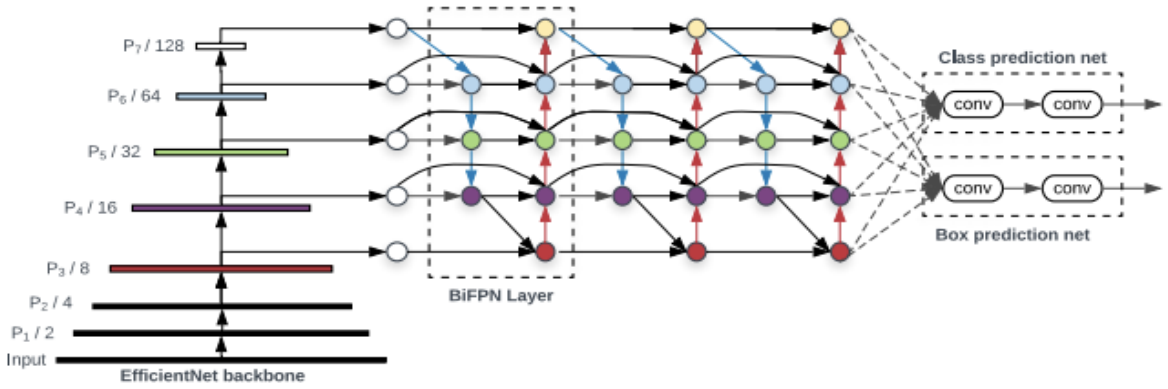


Figure 27: EfficientDet architecture [92].

Une fois les trois cartes de caractéristiques envoyées à la "tête de prédiction", le modèle calcule la confiance et effectue la régression des boîtes englobantes pour chaque pixel de la carte de caractéristiques.

Cela se fait en utilisant des ancres prédéfinies pour obtenir un tableau multidimensionnel de boîtes englobantes (BBoxes), comprenant des informations telles que la classe de l'objet, la confiance de la classe, les coordonnées de la boîte, ainsi que des informations sur la largeur et la hauteur.

En définissant des seuils correspondants (comme le seuil de confiance et le seuil d'objet) pour filtrer les informations non pertinentes dans le tableau, et en appliquant ensuite un processus de suppression non maximale (NMS), le modèle génère les informations finales de détection des objets.

- **YOLOv5 est compose de trois parties : la colonne vertébrale (backbone), le cou (neck), et la tête (head)**

a) Backbone :

Backbone examine l'image et identifie les détails de base à différentes échelles, comme un filtre photo qui capture les contours et les formes. Ensuite le backbone basé sur CSP Darknet53, intègre plusieurs modules CBS et C3 comme cité dans la figure 28, ainsi qu'un module SPPF pour améliorer l'extraction et l'expression des caractéristiques [93]. Le module CBS assiste le module C3 dans cette tâche, tandis que le SPPF optimise l'agrégation des fonctionnalités spatiales, améliorant ainsi la vitesse de traitement [51].

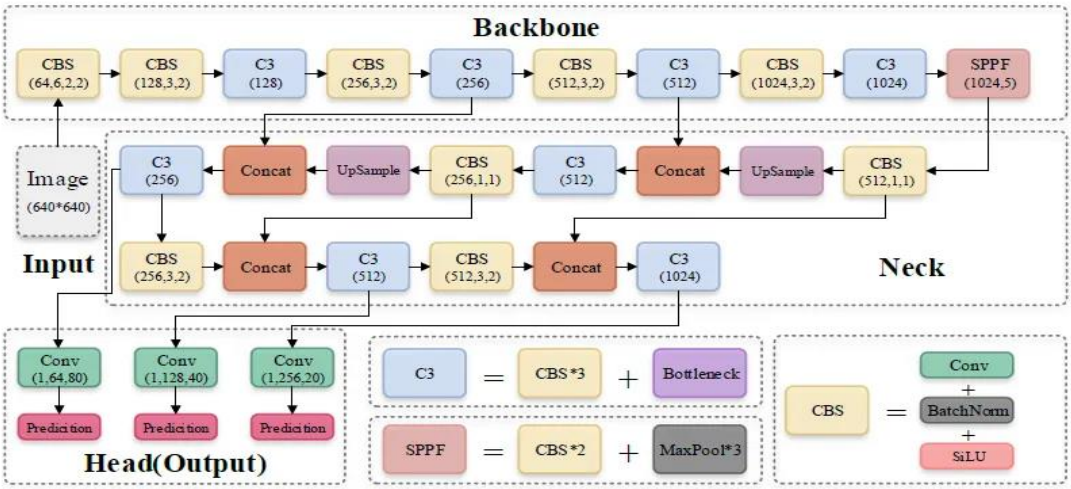


Figure 28: YOLOv5 Architecture du modèle [93].

b) Neck :

Le cou prend les caractéristiques détectées par le backbone et les mélange pour obtenir une vue d'ensemble de l'image, prête à être analysée pour la détection des objets.

Dans la figure 29 Neck est utilisé les méthodes FPN et PAN pour détecter des cibles à différentes échelles. FPN sur-échantillonne les cartes de caractéristiques de sortie (C3, C4 et C5) pour générer de nouvelles cartes de caractéristiques (P3, P4 et P5) [94], permettant ainsi une détection multi-échelle [52].

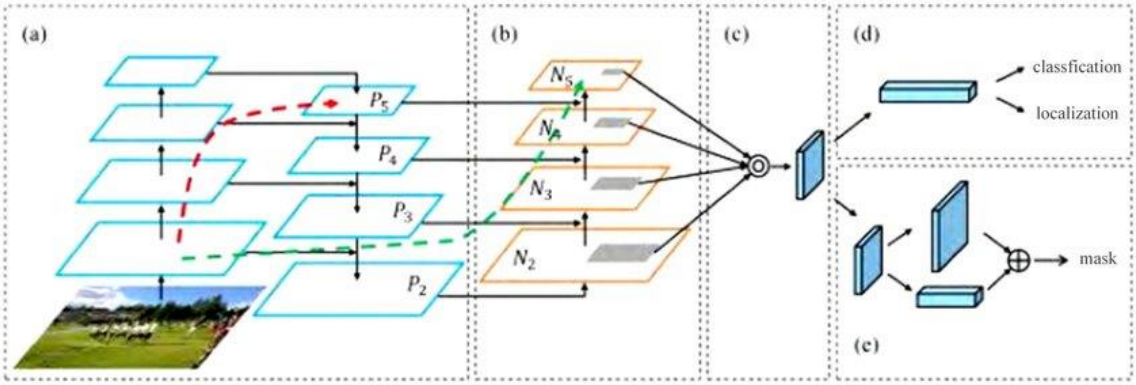
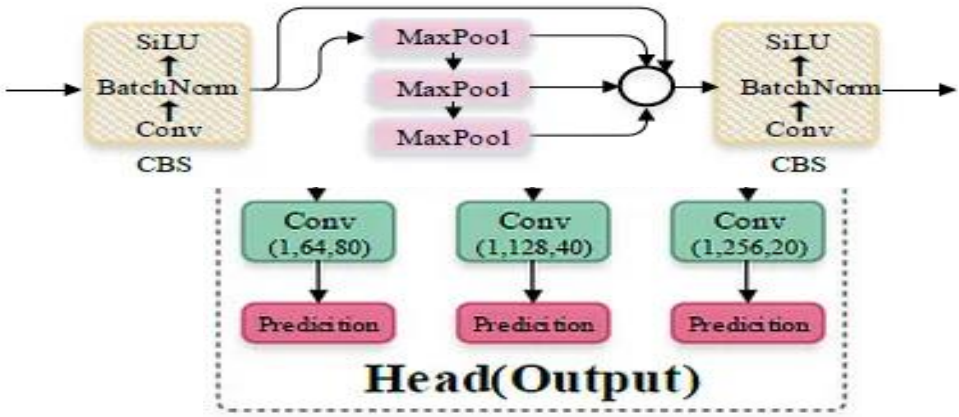


Figure 29: PANet. (a) FPN. (b) voie ascendante. (c) mise en commun adaptative des fonctionnalités. (d) branche de détection. (e) couche de fusion [94].

c) Tête :

La tête prend cette vue d'ensemble créée par le cou et l'utilise pour identifier ce qu'il y a dans l'image - qu'il s'agisse d'un chat, d'une voiture ou autre chose. Ensuite, elle prend des décisions sur les boîtes qui entourent ces objets et les classes auxquelles ils appartiennent [53].



$$g_x = 2\sigma(s_x) - 0.5 + r_x$$
$$g_y = 2\sigma(s_y) - 0.5 + r_y$$
$$g_h = p_h(2\sigma(s_h))^2$$
$$g_w = p_w(2\sigma(s_w))^2$$

Figure 30: Structure du SPPF / Régression de la boîte englobante [95].

La figure 30 représente un schéma de la structure du SPPF et de la régression de la boîte englobante. Le SPPF utilise des niveaux de pooling spatial pour extraire des caractéristiques à différentes échelles, tandis que la régression de la boîte englobante affine les prédictions de localisation des objets [95].

- Les coordonnées du coin supérieur gauche de la carte des caractéristiques sont fixées à (0, 0).
- -rx et ry représentent les coordonnées non ajustées du point central prédit.
- gx, gy, gw et gh contiennent les informations de la boîte de prédiction ajustée.

- pw et ph correspondent aux informations de l'ancre précédente.
- sx et sy représentent les décalages calculés par le modèle.
- Le processus vise à ajuster la coordonnée centrale et la taille de l'ancre préalablement définie pour les aligner avec la coordonnée centrale et la taille de la boîte de prédiction finale.

- **Activation et optimisation :** YOLOv5 utilise l'activation sigmoïde et ReLU fuyante pour normaliser les sorties et éviter les problèmes de gradient, ainsi que SGD et ADAM comme options d'optimisation et permettant une mise à jour flexible des poids du modèle [54].
- **Fonction de perte :** Il utilise l'entropie croisée binaire avec une perte, adaptée à la classification binaire des objets détectés [55].

4. Présentation Raspberry pi :

Le Raspberry Pi est un micro-ordinateur conçu par des professeurs du Département d'Informatique de l'Université de Cambridge, grâce à la fondation Raspberry Pi. Son objectif est de faciliter l'apprentissage de la programmation et de permettre aux utilisateurs de découvrir des systèmes d'exploitation open source, tout en pouvant être intégré dans divers systèmes embarqués [56]. Le Raspberry Pi peut être connecté à divers périphériques tels que des souris et des claviers. Cette carte est conçue pour aider à l'étude des ordinateurs et servir de support à l'apprentissage de la programmation, notamment en utilisant des langages comme Python et Scratch. Elle est également capable de lire des vidéos en haute définition et même d'installer des jeux vidéo. L'intérêt du Raspberry Pi réside dans sa capacité à interagir avec le monde extérieur et à exécuter différentes versions de systèmes d'exploitation libres comme GNU/Linux et Raspbian Debian, ainsi que d'autres logiciels compatibles. Malgré sa très petite taille, il offre une puissance suffisante pour de nombreuses possibilités de développement et de réalisation de projets en électronique, informatique et télécommunication [57].

La figure 31 représente la carte Raspberry Pi 4[96].

Figure 31: Carte raspberry pi4 , avec son boitier[96].

4.1. Les composants standards du Raspberry Pi :

a) **Processeur ARM** : Les architectures ARM sont des architectures de processeurs, à faible consommation, introduites à partir de 1983 par « Acorn Computers » et développées depuis 1990 par « ARM Ltd » [58].

b) **Mémoire vive RAM** : C'est la mémoire dans laquelle le Raspberry place les données lors de son traitement [67].

c) **Une connectique variée [59] :**

- **HDMI** : « High Définition MultiMedia Interface » permet de relier le Raspberry Pi à un dispositif compatible : écran LCD ou un vidéoprojecteur…
- **Port USB 2.0** : Le port « Universal Serial Bus » est un port série qui sert à connecter le Raspberry aux autres périphériques.
- **Port Ethernet** : C'est un port qui correspond au protocole international ETHERNET de réseau local à commutation de paquets.
- **Prise RCA** : « Radio Corporation of America » est un connecteur électrique utilisé dans le domaine audio/vidéo.
- **Un slot pour les cartes SD** : Le Raspberry a besoin d'une mémoire externe supplémentaire pour fonctionner. Ce slot permit de connecter la mémoire externe.
- **Une prise jack** : C'est une connectique audio-vidéo.
- **GPIO** : « General Purpose Input/Output » sont des ports d'Entrée/Sortie.

4.2. Domaine applications:

- **Serveurs Web et Cloud :**

Le Raspberry Pi peut être configuré comme un serveur web, un serveur de fichiers ou même un serveur de cloud personnel. Il est souvent utilisé pour héberger des sites web, des bases de données ou des applications simples [68].

- **Systèmes Embarqués :**

Le Raspberry Pi est utilisé dans les systèmes embarqués pour divers dispositifs, notamment dans l'automobile, l'aérospatial, et les appareils médicaux. Il permet de développer des solutions compactes et puissantes à faible coût [60].

- **Domotique**

Le Raspberry Pi peut être utilisé pour automatiser des tâches domestiques, telles que le contrôle des lumières, des systèmes de chauffage, des caméras de sécurité et autres appareils ménagers intelligents. Il est souvent intégré à des systèmes de gestion de maison intelligente comme Home Assistant [60].

- **Intelligence artificielle (détections d'objets**) [60] :

Systèmes de Sécurité : Surveillance vidéo pour détecter des intrusions ou des mouvements suspects. **Automatisation Domestique** : Reconnaissance d'objets pour contrôler des appareils domestiques Intelligents.

Robots Autonomes : Robots capables de naviguer et d'interagir avec leur environnement en reconnaissant et en évitant les obstacles.

4.3. Comparaison entre les différentes cartes Raspberry Pi :

Les différents modèles du Raspberry Pi	**Date de sortie**	**Caractéristique**	**Dimension (H x L x P)**	**Poids**	**Fréquence du processeur**	**Mémoire vive (Ram)**	**Consommation maximale Mesurée**
Modèle Zéro	2015	WIFI / NON BLUETOOTH/ CAMERA	**(65 × 31× 5) mm**	9 g	1 GHz	512 Mo	140 Ma
Modèle Zéro W	2018	WIFI / BLUETOOTH/ CAMERA	**(65 × 31 × 5) mm**	9 g	1 GHz	512 Mo	230 Ma
Modèle Zéro WH	2018	WIFI / BLUETOOTH/NON CAMERA	**(65 × 31 × 13) mm**	12 g	1 GHz	512 Mo	180 mA
Modèle Zéro 2W	2017	WIFI / BLUETOOTH/ CAMERA	**(65 × 31 × 5) mm**	9gNo	1GHz	512Mo	Non précisé
Modèle 1 A	2013	NON WIFI / NON BLUETOOTH/NON CAMERA	**(85,60 × 53,98 × 17) mm**	45 g	700 MHz	256 Mo(GPU)	320 Ma
Modèle 1 A+	2014	NON RESEAU (ETHERNET 10/100 Mbit/s) / NON BLUETOOTH/NON CAMERA	**(65 × 53,98 × 17) mm**	23 g	700 MHz	256 Mo(GPU)	230 mA
Modèle 1 B	2012	RESEAU (ETHERNET 10/100 Mbit/s) / NON BLUETOOTH/N	**(85,60 × 53,98 × 17) mm**	45 g	700 MHz	512 Mo	480 mA

		ON CAMERA					
Modèle 1 B+	2014	RESEAU (ETHERNET 10/100 Mbit/s) / NON BLUETOOTH/NON CAMERA	**(85,60 × 53,98 × 17) mm**	45 g	700 MHz	512 Mo	330 mA
Modèle 2 B	2015	NON WIFI /RESEAU (ETHERNET 10/100 Mbit/s) / NON BLUETOOTH/ CAMERA	**(85,60 × 53,98 × 17) mm**	45 g	900 MHz	1 Go	250 mA
Modèle 3 B	2016	WIFI/ RESEAU (ETHERNET 10/100 Mbit/s) / BLUETOOTH/ CAMERA	**(85,60 × 53,98 × 17) mm**	45 g	1.2 GHz	1 Go	720 mA
Modèle 3 A+	2018	WIFI /RESEAU (ETHERNET 10/100 Mbit/s) / BLUETOOTH/ CAMERA	**(65 × 56 × 12,50) mm**	29 g	1.4 GHz	512 Go	Non précisé
Modèle 3 B+	2018	WIFI /NON RESEAU (ETHERNET 10/100 Mbit/s) / BLUETOOTH/ CAMERA	**(85,60 × 53,98 × 17) mm**	45 g	1.4 GHz	1 Go	Non précisé
Modèle 4 B	2019	WIFI /RESEAU (Gigabit Ethernet) / BLUETOOTH/ CAMERA	**(85 × 56 × 16) mm**	46 g	1.5 GHz	(1-2-4-8) Go (LPDDR4 SDRAM)	885 mA
Modèle 5 B	2023	WIFI /RESEAU (Gigabit Ethernet) / BLUETOOTH/ CAMERA	**(85,6x56,5x11)mm**	46g	2.4 GHz	(4 8)GB (LPDDR4)	Non precisé

Tableau 1 : Comparaison entre les différents modèles Raspberry Pi[69].

5. Choix de modèle (Raspberry Pi 4) [61]:

Pour notre projet, nous avons choisi le Raspberry Pi 4, pour sa disponibilité faible consommation en énergie par rapport à d'autres types de Raspberry.

Le Raspberry Pi 4 est une nouvelle génération d'ordinateurs Raspberry Pi, offrant des performances améliorées en termes de processeur, graphique et entrées/sorties, tout en prenant en charge plus de RAM. Malgré ces améliorations, il conserve le même format, la même consommation d'énergie et le même prix que le modèle précédent. Le Pi4B est disponible avec 1, 2 ou 4 ou 8 giga octets de LPDDR4 SDRAM.

En terme d'utilisation finale, il est important de noter que les performances du nouveau Raspberry Pi 4 modèle B sont équivalentes à celles d'un ordinateur x86 d'entrée de gamme. Parmi les principales caractéristiques de ce dernier ordinateur Raspberry Pi, on peut noter :

5.1. Caractéristiques du Raspberry Pi 4 modèle B 4Go [62] :

- SoC Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bits à 1.5GHz.
- DRAM 4 Go LPDDR4-2400.
- LAN sans fil 2.4 GHz et 5.0 GHz IEEE 802.11b/g/n/ac, Bluetooth 5.0, BLE.
- Ethernet True Gigabit.
- 2 Ports USB 3.0, 2 Ports USB 2.0
- Connecteur GPIO 40 broches entièrement rétrocompatible
- 2 ports micro HDMI supportant une résolution vidéo allant jusqu'à 4K 60Hz.
- Ports MIPI DSI/CSI à 2 voies pour caméra et écran.
- Sortie audio stéréo et port vidéo composite, 4 pôles.
- Emplacement pour carte Micro SD, pour le système d'exploitation et le stockage des données.
- Nécessite une alimentation 5.1V, 3A via USB-C ou GPIO.
- PoE (Power over Ethernet) activé (nécessite PoE HAT).

La figure 32 montre les Caractéristiques techniques de la Raspberry Pi 4 [97].

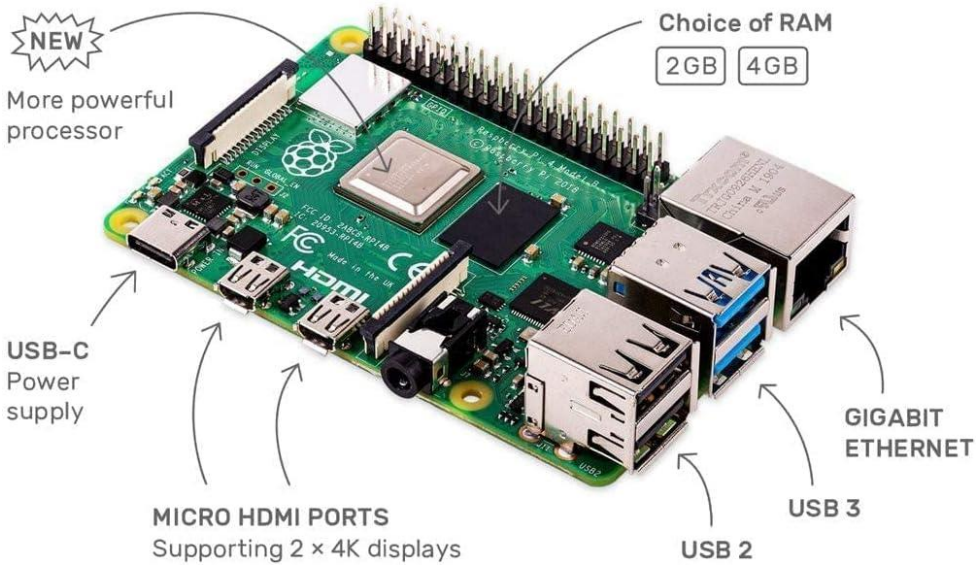


Figure 32: Caractéristique du raspberry pi4 [97].

6. Discussion :

Dans ce chapitre, nous avons discuté de notre choix du modèle YOLOv5 pour notre projet. YOLOv5 est un algorithme de détection d'objets en temps réel, connu pour son équilibre entre précision et vitesse.

Bien que non officiel, YOLOv5 est largement utilisé pour sa simplicité d'utilisation et son efficacité. Écrit en PyTorch, il est facilement intégrable dans divers pipelines d'apprentissage automatique.

Nous avons opté pour YOLOv5 après avoir comparé plusieurs versions précédentes. YOLOv1 a introduit le concept de détection d'objets en un seul passage, réduisant considérablement le temps de calcul. YOLOv2, aussi appelé YOLO9000, a amélioré la précision avec de meilleures techniques de normalisation et la capacité de détecter plus de 9000 classes d'objets. YOLOv3 a ajouté une architecture plus profonde et des techniques de détection multi-échelle, tandis que YOLOv4 a apporté des améliorations comme CSPDarknet53 et des méthodes d'augmentation des données avancées. YOLOv5, avec sa simplicité et son efficacité, a finalement retenu notre attention pour son équilibre parfait entre performance et facilité d'utilisation.

Nous avons choisi d'implémenter notre modèle sur un Raspberry Pi 4, un système embarqué abordable et flexible. Le Raspberry Pi 4 peut supporter divers systèmes d'exploitation et bibliothèques nécessaires pour faire tourner YOLOv5. Avec ses multiples ports USB, GPIO, HDMI et son support pour les réseaux sans fil, il offre de nombreuses options pour intégrer d'autres composants et capteurs. Sa large communauté de développeurs assure également un bon support et des ressources abondantes.

Cependant, le Raspberry Pi 4 a des limitations. Comparé à des systèmes plus puissants comme les GPU dédiés, il est moins performant en termes de capacité de traitement. De plus, sa mémoire RAM est limitée à 8 Go maximum, ce qui peut être un obstacle pour des modèles très gourmands en ressources. Malgré ces limitations, nous l'avons choisi pour son équilibre entre coût, flexibilité et connectivité.

Dans le prochain chapitre, nous allons tester et évaluer notre modèle YOLOv5 entraîné avec une base de données comportant plusieurs classes d'objets. Nous détaillerons le processus de préparation des données, l'entraînement du modèle et les résultats obtenus. De plus, nous explorerons des stratégies pour améliorer la précision moyenne (mAP), comme l'augmentation de la base de données et l'utilisation de techniques d'augmentation des données avancée.

Chapitre 4 : Application et Résultats

1. Préambule :

Dans ce chapitre, nous visons à implémenter la détection automatique d'objets dans une vidéo provenant d'une caméra fixe (webcam)au temps réel, à l'aide de la carte Raspberry Pi.

Après avoir exploré en profondeur YOLOv5 dans le chapitre précédent, nous allons maintenant découvrir la carte Raspberry Pi et ses différentes configurations. Nous donnerons également des informations détaillées sur la façon d'installer les différentes bibliothèques nécessaires à notre projet. Nous verrons ensuite comment le modèle YOLOv5 peut être utilisé pour la détection d'objets sur le Raspberry Pi. Ensuite, nous expliquerons les étapes nécessaires pour entraîner ce modèle et l'intégrer dans le Raspberry Pi, ce qui en fait un puissant système embarqué pour la détection d'objets dans diverses applications réelles.

Enfin, nous testerons le modèle et réaliserons une évaluation de sa performance dans des situations réelles, évaluant ainsi sa performance et son efficacité dans des applications pratiques.

La figure 32 représente le Schéma qui résume toutes les étapes de la détection d'objets en temps réel avec YOLOv5 :

Pour expliquer les étapes principales de la détection d'objets avec YOLOv5 déployé sur une carte Raspberry Pi 4, le processus commence par le pré-traitement des données. Cela inclut le redimensionnement, la normalisation et l'augmentation des images prélevées dans la base de données MS COCO, avec un accent particulier sur cinq classes spécifiques telles que "person" et "car".

YOLOv5 utilise un backbone comme CSPDarknet pour extraire efficacement les caractéristiques des images. Ensuite, un "neck" combine ces caractéristiques à différentes échelles pour améliorer la détection des objets de diverses tailles.

L'entraînement du modèle ajuste les poids à travers des itérations d'entraînement et de validation, en utilisant des algorithmes d'optimisation pour optimiser les performances. La tête de YOLOv5 prédit les classes et les boîtes englobantes des objets détectés, générant des scores de confiance pour filtrer les détections moins probables.

Ce processus global garantit une reconnaissance d'objets précise et rapide, adaptée aux capacités de traitement du Raspberry Pi 4.

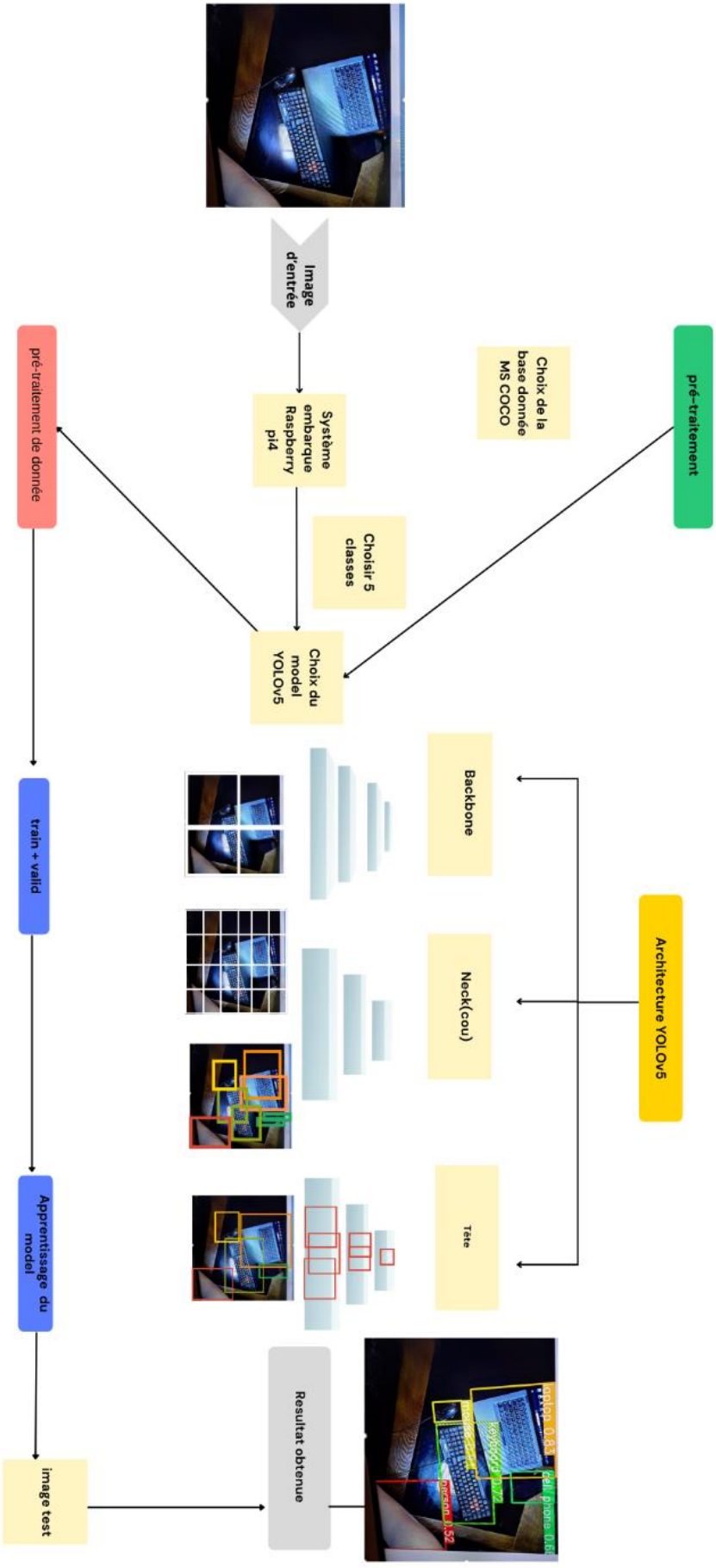


Figure 33: Conception d'apprentissage yolov5

2. Logiciels de développement :

2.1. Les outils :

On a utilisé un ordinateur ayant les caractéristiques suivant:

- Processeur: Intel(R) Core (TM) i7-8665U CPU
- Une RAM de 16Go.
- Type Processus : Système d'exploitation 64 bits, processeur x64
- Un disque dur 500 Go.
- Résolution de l'écran : 1920 X 1080
- Système exploitation : Windows 10 Professionnelle

Pour commencer, nous devons assembler et configurer le matériel nécessaire :

- Une carte Raspberry Pi 4.
- Une alimentation électrique compatible.
- Une carte microSD (au moins 32 Go) avec le système d'exploitation Raspbian.
- Un écran, un clavier et une souris pour la configuration initiale.

2.2. L'environnement informatique :

a) Langage Python :

Python est un langage interprété, libre et gratuit, multiplateforme, mature et de plus en plus Populaire, y compris dans le domaine scientifique grâce à Numpy, Scipyé et Matplotlib. Il est Multi-paradigmes : orienté, impératif, fonctionnel et objet. Il contient nativement de nombreuses bibliothèques. Il est fourni avec la plupart des distributions Linux. Sous Windows Il est possible d'installer python, puis d'installer les bibliothèques une à une [63].

b) Opencv :

OpenCV (Open Source Computer Vision) est une bibliothèque libre de vision par ordinateur. Cette bibliothèque est écrite en C et C++ et peut être utilisée sous Linux, Windows et Mac OS. Des interfaces ont été développées pour Python, Ruby, Matlab et autre langage. Open CV Est orienté vers des applications en temps réel. Un des buts d'OpenCV est d'aider les gens à Construire rapidement des applications sophistiquées de vision à l'aide d'infrastructure simple De vision par ordinateur. La bibliothèque d'OpenCV contient plus de 500 fonctions [64].

c) Numpy :

Numpy fournit des classes de matrices et tableaux qui sont des standards et qui permette de Dialoguer facilement entre les différentes libraires. Numpy fournit aussi un certain nombre D'opérations mathématiques utilisant ces tableaux [65].

d) Google colab :

Est un environnement de développement intégré basé sur le cloud, fourni par Google. Il permet d'écrire et d'exécuter du code Python directement dans votre navigateur. Colab est particulièrement populaire pour les projets d'apprentissage automatique et de science des données en raison de son accès gratuit à des GPU, ce qui accélère considérablement les calculs nécessaires pour l'entraînement de modèles complexes [66]. Dans la figure 34 on voit logo du Google colab [98].

Figure 34: la revue dur Google colab [98].

e) Pytorch :

Dans la figure 34, on voit la revue de PyTorch, [99] qui est une bibliothèque open source d'apprentissage automatique en Python, développée par Facebook. Elle permet d'effectuer les calculs tensoriels nécessaires pour l'apprentissage profond, optimisés pour être réalisés par le processeur central (CPU) ou le processeur graphique (GPU), si disponible. Le CPU (Central Processing Unit)**,** ou processeur central, est le composant principal de l'ordinateur responsable de l'exécution des instructions des programmes. Il réalise les calculs de base et gère les tâches générales de traitement des données. En revanche, le GPU (Graphics Processing Unit)**,** ou processeur graphique, est spécialement conçu pour gérer des calculs complexes liés à l'affichage graphique. Les GPU sont également utilisés pour des tâches de calcul intensif comme l'apprentissage profond, car ils peuvent traiter un grand nombre d'opérations en parallèle, ce qui les rend beaucoup plus efficaces que les CPU pour ce type de travail [67].

Figure 35: la revue de Pytorch [99].

3. Configuration SD et installation d'un Raspberry Pi :

Le Raspberry Pi est un petit ordinateur mono carte, peu coûteux mais puissant, qui a révolutionné le monde de la technologie en permettant à quiconque, du débutant au professionnel, d'explorer et de créer une variété infinie de projets informatiques. Que vous souhaitiez créer un serveur domestique, un centre multimédia, un système de surveillance, ou même un robot, le Raspberry Pi offre une plateforme flexible et abordable pour concrétiser vos idées.

Avant de plonger dans les vastes possibilités offertes par le Raspberry Pi, la première étape consiste à préparer son installation. Cela inclut la configuration d'une carte SD, qui servira de support de stockage pour le système d'exploitation et les données de votre Raspberry Pi.

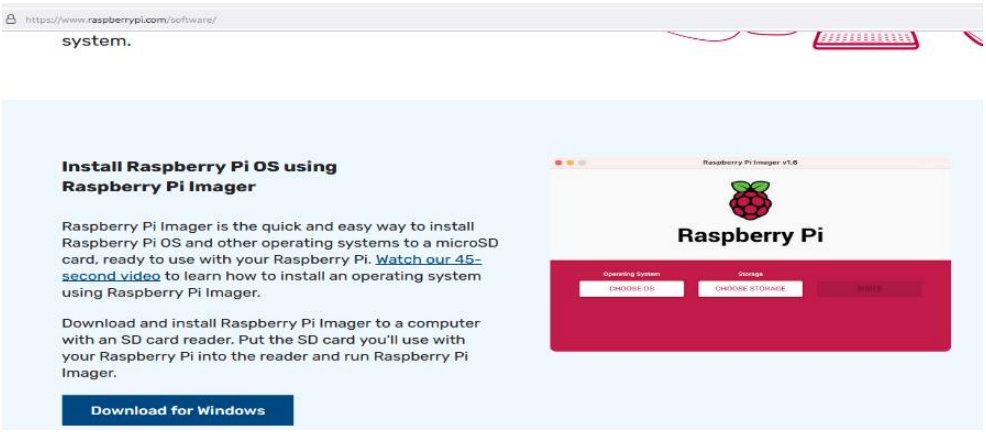


Figure 36 :Logiciel Raspberry pi image.

3.1. Installation de Système exploitation :

On télécharge sur le site Raspberry Pi Imager le logiciel Raspberry Pi Imager comme ci-dessus dans la figure 36.

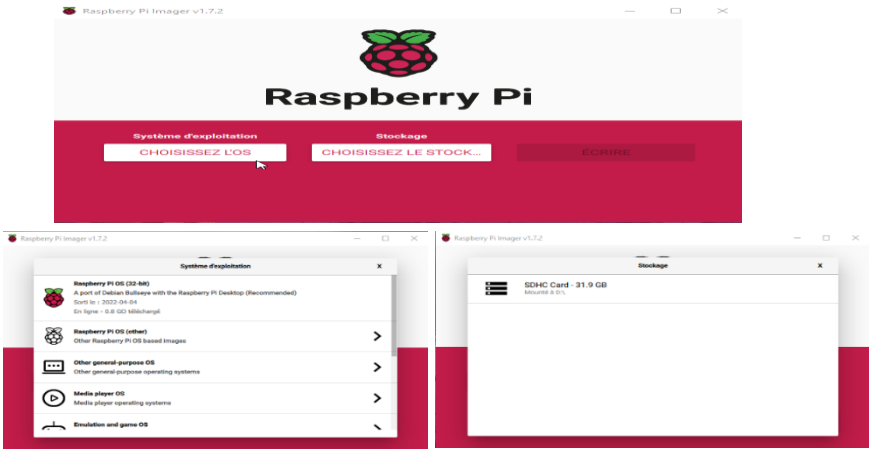


Figure 37: la revue du logicielle raspberry pi image.

3.2. Création de l'image :

Pour installer Raspberry OS sur la carte, on crée une image du système d'exploitation sur la carte SD du nano-ordinateur via un PC personnel est mentionner dans la figure 37 sur la revue du logicielle raspberry pi imager. Pour créer cette image, on insère la carte SD dans le PC et on commence par télécharger Raspberry Pi Imager : https://www.raspberrypi.com/software/ On choisit Raspberry OS 64bits et la carte SD.

- On autorise un lien SSH en cliquant Enable SSH :

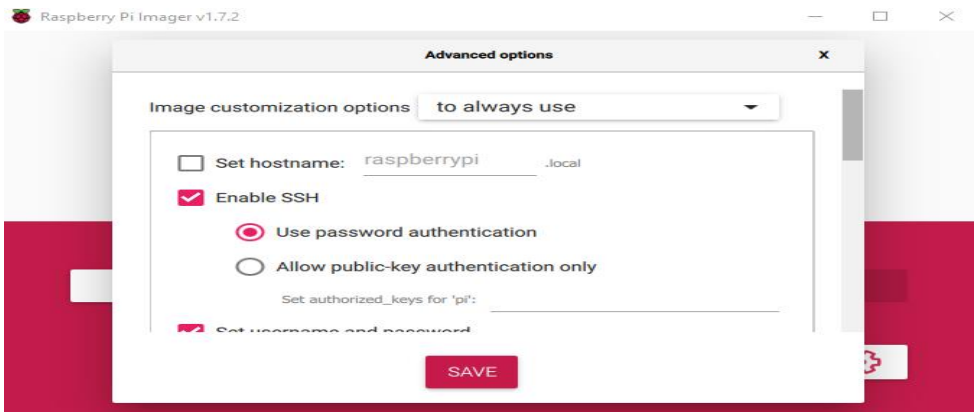


Figure 38:autorisation SSH.

- On crée un utilisateur avec son mot de passe (ici le user est pi et le mdp est raspberry).
- Dans Set Locals, on choisit le Layout FR pour le clavier.

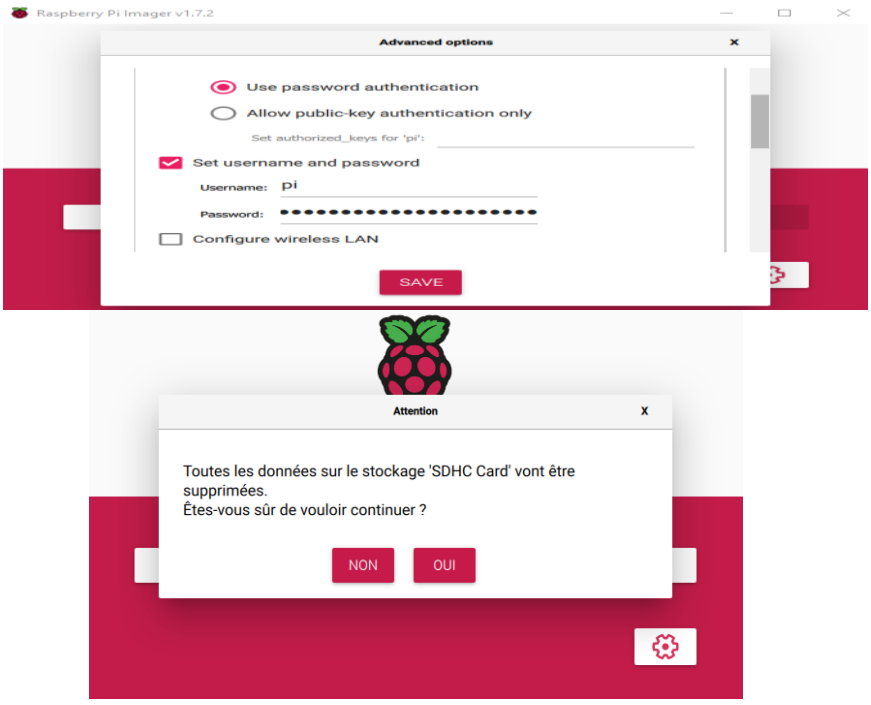


Figure 39: set user nom et mots de passe pour système du la raspberry pi.

- On vérifie si on est bien sur la carte SD puis on clique sur oui :

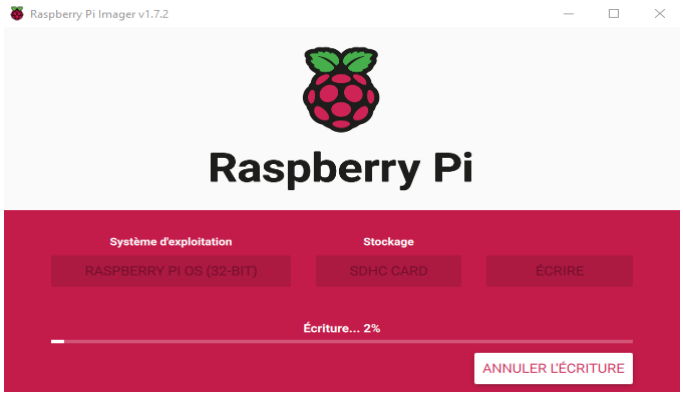


Figure 40:formater le stockage de la carte SD/ début d'écriture.

3.3. Recherche de l'IP du nano-ordinateur (Raspberry pi4) :

Une fois ces configurations effectuées, le nano-ordinateur démarre. Pour s'y connecter à distance, il est nécessaire d'avoir son adresse IP, adresse notée sous la forme de 4 nombres 192.168.1.4 par exemple (en IPv4).

Le PC doit être bien évidemment connecté sur le même réseau local.

- Une solution simple est de faire un ping avec une recherche DNS locale, depuis une console (nommée invite de commande sous windows) avec le hostname défini dans la partie précédente : raspberrypi.local

Le nano-ordinateur répond alors avec son adresse IP.

- Si le nano-ordinateur est connecté sur un réseau Ethernet filaire ou wifi avec un serveur DHCP (c'est-à-dire qu'une machine, le routeur ou le box habituellement sur les petits réseaux, distribue les adresses IP), on peut y trouver l'adresse IP du nano ordinateur. Pour cela, on se rend sur la page web du box ou du routeur (la page web de la box internet pour une utilisation à domicile, souvent accessible à l'adresse 192.168.1.1). On présente ici les pages d'une LiveBox. Les routeurs grand public et les boxes ont la même apparence avec une vue globale (overview) du réseau où on trouve les IP des équipements connectés et des pages de configuration avancée (parfois nommées LAN network), notamment pour la réservation des adresses IP.

Il est alors possible de réserver l'adresse IP dans le serveur DHCP, de sorte que le nano-ordinateur obtienne toujours la même adresse lors de ces nouvelles connexions.

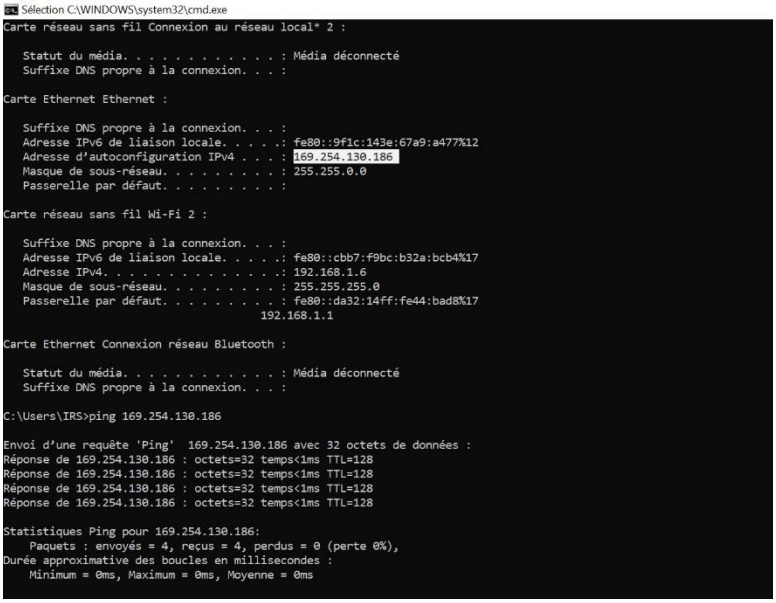


Figure 41: Adresse IP du la Raspberry pi 4.

3.4. Connexion SSH :

Une fois l'IP découverte, il est aisé de se connecter en SSH (Secure Shell). SSH est un protocole de communication sécurisée permettant de dialoguer via une console du serveur SSH ouverte sur le client SSH.

Le nano-ordinateur est serveur SSH et le PC utilisateur est client SSH.Très utile pour les accès à distance, le serveur SSH est installé par défaut sur les Raspberry Pi (et il a été activé dans les options lors de la création de la carte SD.

Le client SSH est installé sur les PC Linux (et sans doute Mac). Sur les PC Windows, on peut installer Putty par exemple. Depuis une console (terminal Linux ou Putty sous Windows), on tape :

ssh login@<adresse IP de la raspberry pi >

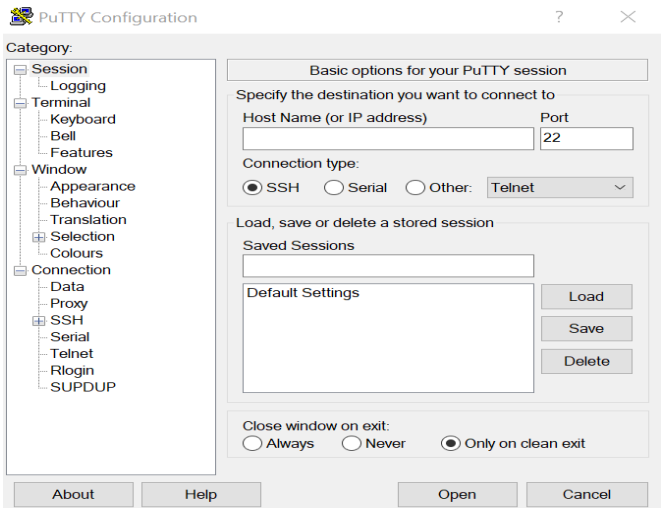


Figure 42: Putty configuration.

L’authenticité de ce serveur SSH n’étant pas attesté, une demande de confirmation attend la réponse yes, lors de la première connexion, puis le mot de passe de l’utilisateur.

Lorsque la connexion est réussie, le prompt indique : le_nom_de_l_utilisateur@raspberrypi, ici pi@raspberrypi.

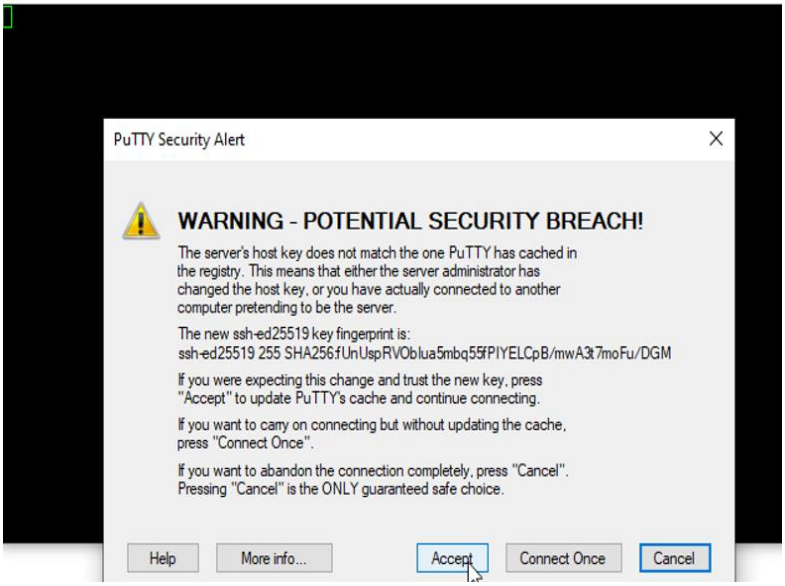


Figure 43: Connexion SSH à la Raspberry Pi depuis un PC sur le même réseau local.

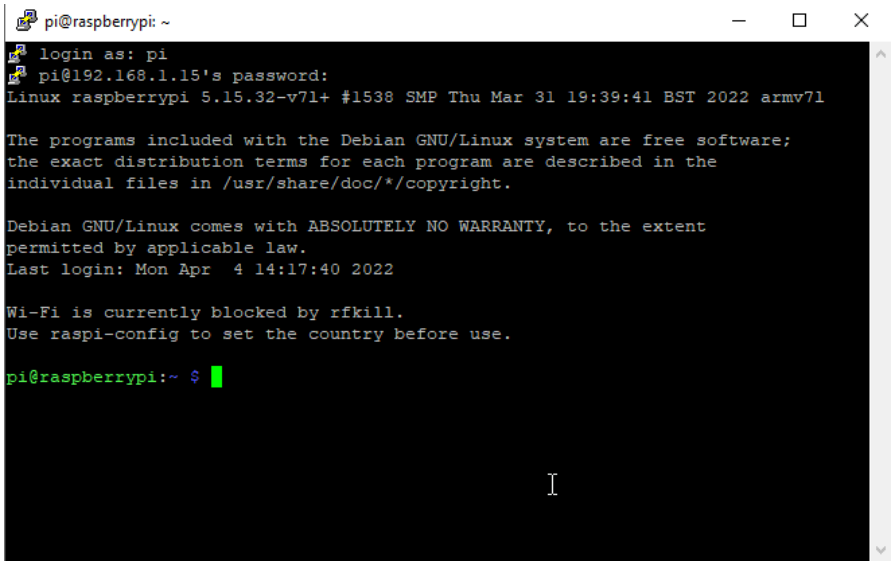


Figure 44: intégré dans raspberrypi.

3.5. Bureau à distance

Pour faciliter l'utilisation du raspberry, les outils graphiques sont plus accessibles que la ligne de commande.

Mise en œuvre d'un bureau à distance VNC :

VNC (Virtual Network Computing) est un protocole de bureau à distance. Le serveur VNC est la Raspberry Pi et le client VNC est le PC utilisateur.

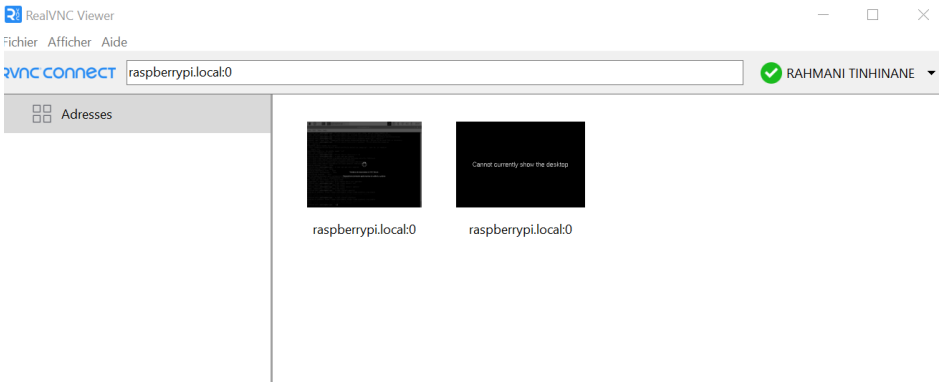


Figure 45: Commande pour accéder aux paramètres de la Raspberry Pi 4.

Figure 46: Client VNC Viewer lors de la 1ère connexion à la Raspberry Pi.

- **Installation du client sur le PC :**

Raspberry propose la solution gratuite mais propriétaire de RealVNC. Il est possible de télécharger le client vnc (nommé VNC Viewer) : sur le site (https://www.realvnc.com/en/), onglet Download, choisir la version adaptée de VNC Viewer à l'OS de l'ordinateur et l'installer.

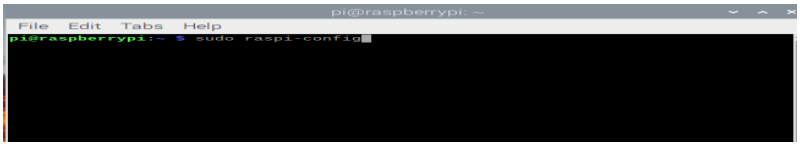


Figure 47: Extrait de la fenêtre raspi-config.

- **Activation du serveur VNC sur le nano-ordinateur Raspberry Pi :**

Realvnc-vnc-serveur est installé avec l'image de Raspberry OS. Pour l'activer, utiliser la fenêtre de configuration raspi-config, depuis l'accès SSH :

Ensuite, choisir Interfacing Options > VNC › Yes.

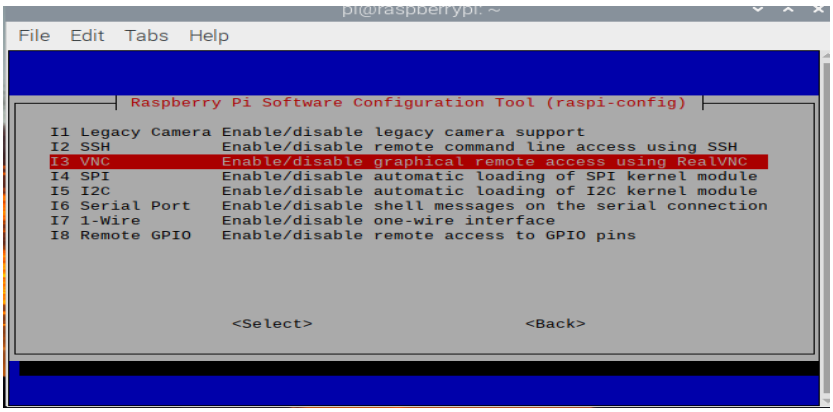


Figure 48:interface vnc.

- **Connexion locale au serveur vnc :**

Pour se connecter en VNC depuis un PC connecté sur le même réseau que le nano-ordinateur, lancer VNC Viewer, indiquer l'adresse IP de la Raspberry Pi, accepter les recommandations de précautions, introduire son login et son mot de passe. On accède alors au bureau à distance, qui lors de la première connexion demande à faire quelques mises à jour. Suite à un redémarrage on peut profiter du bureau à distance du nano-ordinateur.

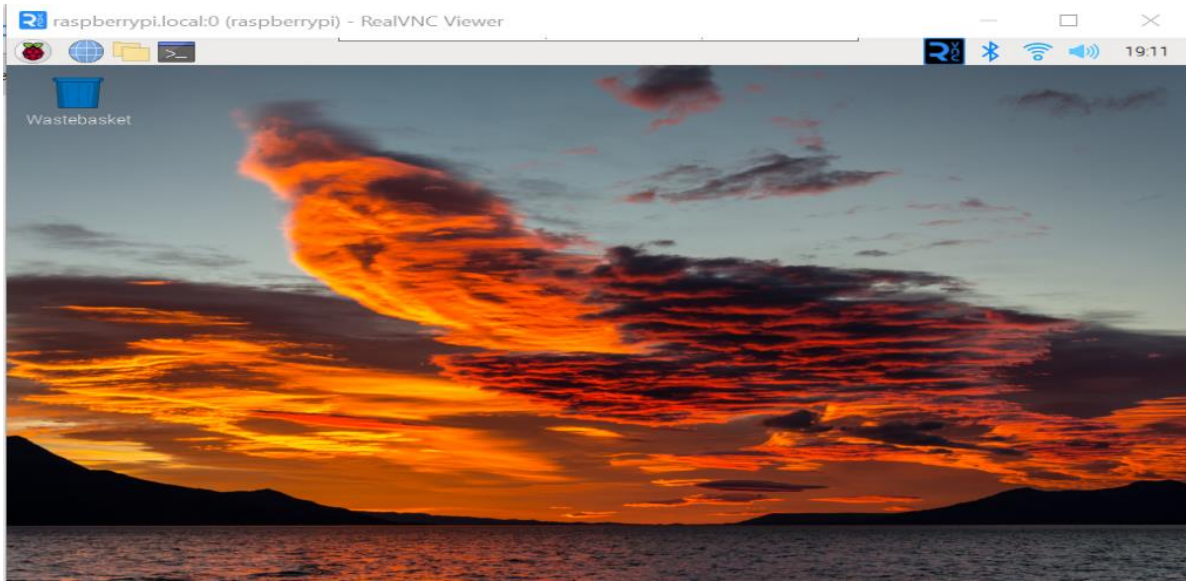


Figure 49:Bureau à distance de la Raspberry Pi via RealVNC.

- **Serveur DHCP et SSH :**

On met la carte SD dans le Raspberry Pi éteint, ensuite on l'alimente avec le câble USB-C. Le Raspberry Pi se connecte de base sur un serveur DHCP pour récupérer une adresse IP.

L'idéal est de faire ce premier démarrage sur votre BOX. Il suffit alors de se connecter sur la box pour voir l'adresse IP qui a été affectée au port Ethernet (ici 169.254.130.186)

Il existe cependant deux autres solutions pour se passer de Box pour affecter une adresse IP au port Ethernet :

- On branche un écran, un clavier, une souris sur le Raspi et l'on configure le port Ethernet depuis l'interface graphique
- On change directement un fichier de configuration pour figer une adresse IP Statique. Cela nécessite d'avoir un PC sous Linux avec un lecteur de carte SD.

Pour avoir accès au terminal Linux du Raspi depuis un PC portable et sans avoir à brancher un écran/clavier/souris, le plus rapide est de passer par une communication SSH. Un client SSH comme Putty est installé sur les postes de l'IUT:

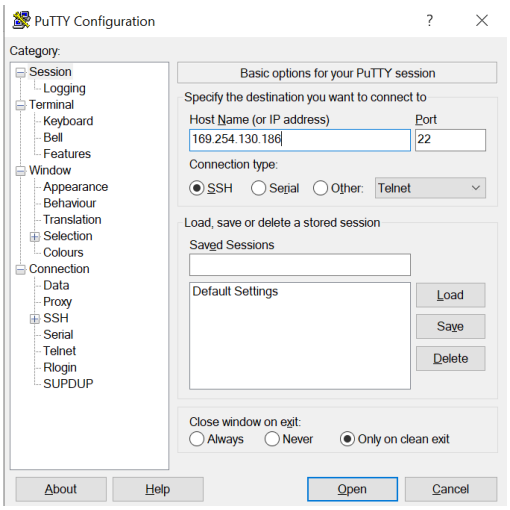


Figure 50: putty configuration avec adresse IP du raspberry pour accéder à CMD du raspberry pi4.

On accepte la clé de cryptage (Accept) et l'on arrive sur le terminal Linux du Raspberry Pi depuis son PC portable. L'inconvénient, c'est qu'il ne s'agit d'une interface texte, moins conviviale qu'une interface graphique. Nous verrons dans un autre article comment se connecter sur l'interface graphique à distance avec Remote Desktop ou VNC.

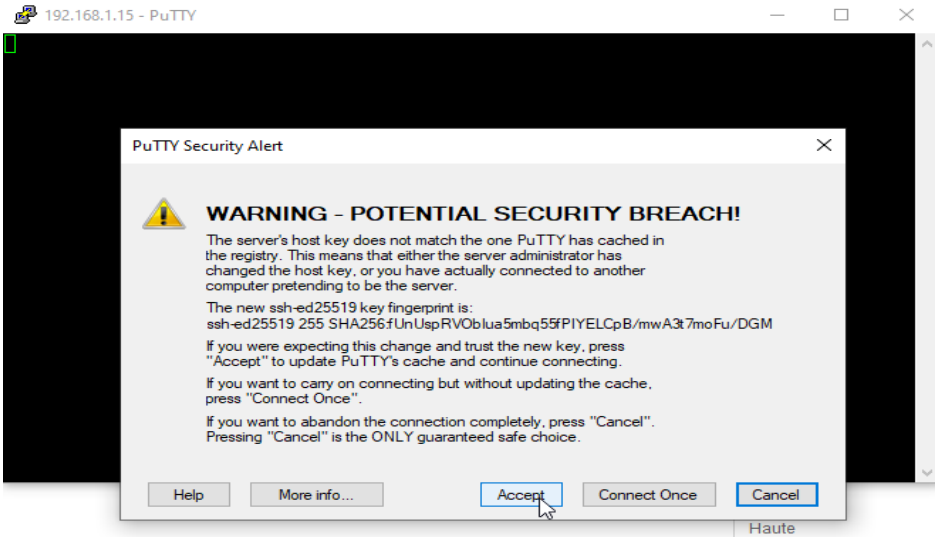


Figure 51:demander acceptation pour accéder raspberry pi4 .

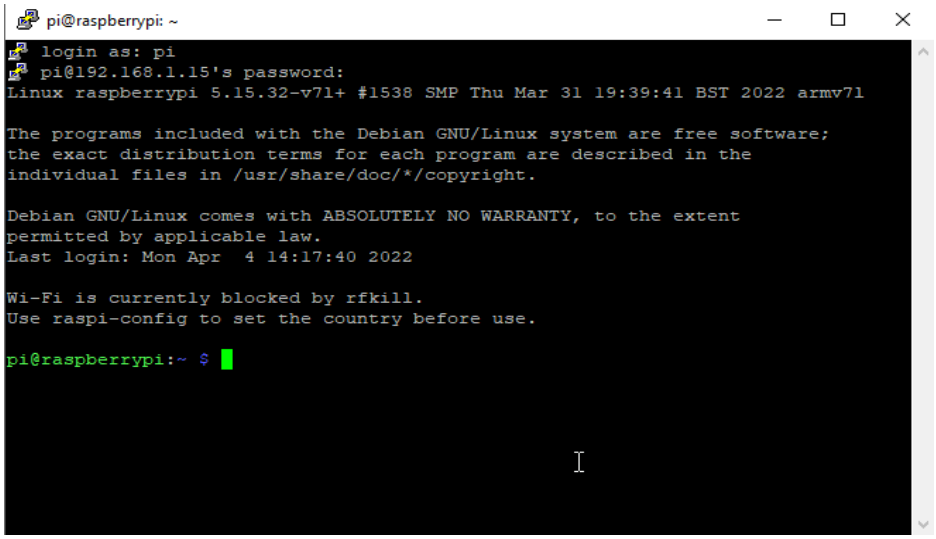


Figure 52:interface du pi@raspiberrypi..

On peut lancer la commande de mise à jour des paquets de la distribution linux en faisant :

sudo apt update

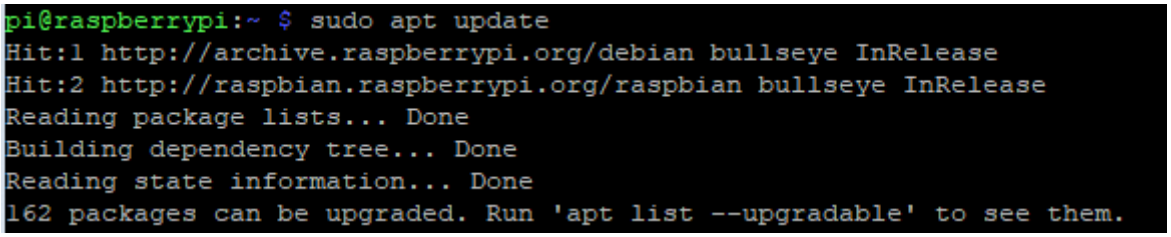


Figure 53:Mise a jours des paquets.

L'upgrade de la distribution se fait en faisant :

sudo apt upgrade (Cliquer sur Y pour valider l'upgrade)

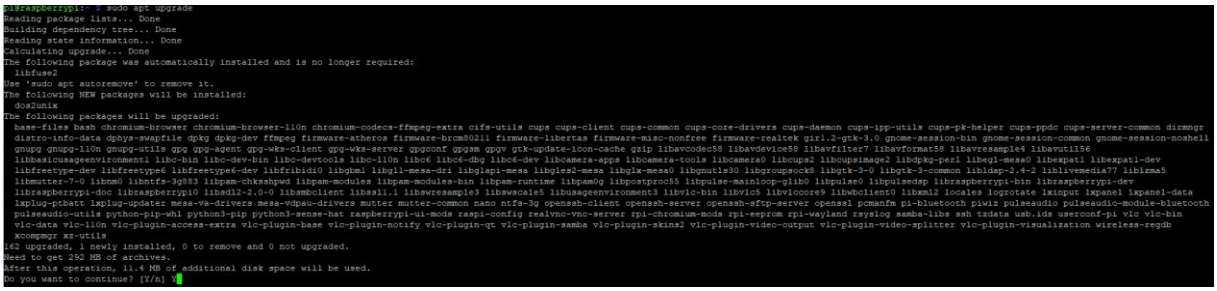


Figure 54:suite de la mise à jour des paquets.

Votre Raspberry Pi est maintenant à jour, nous pouvons y déployer le Remote Desktop et le VNC pour simplifier sa programmation.

- ✓ VNC sur Raspberry Pi

- ✓ Activation du serveur VNC sur le Raspi
- ✓ Depuis un terminal ou une liaison SSH:
  - Lancer la commande : sudo raspi-config
- ✓ Il faut dans un premier activer le serveur VNC sur le Raspberry Pi
  - Interface Options -> Enable graphical remote access using RealVNC

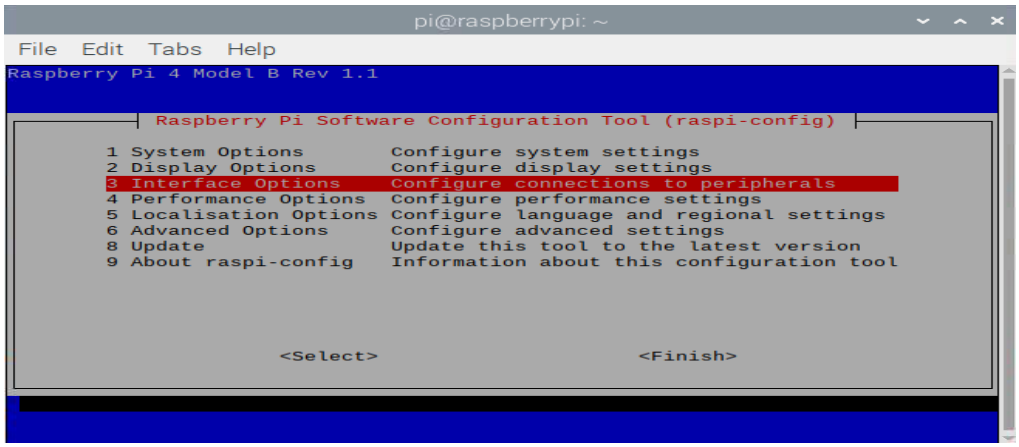


Figure 55:Accéder aux interface du raspberrypi.

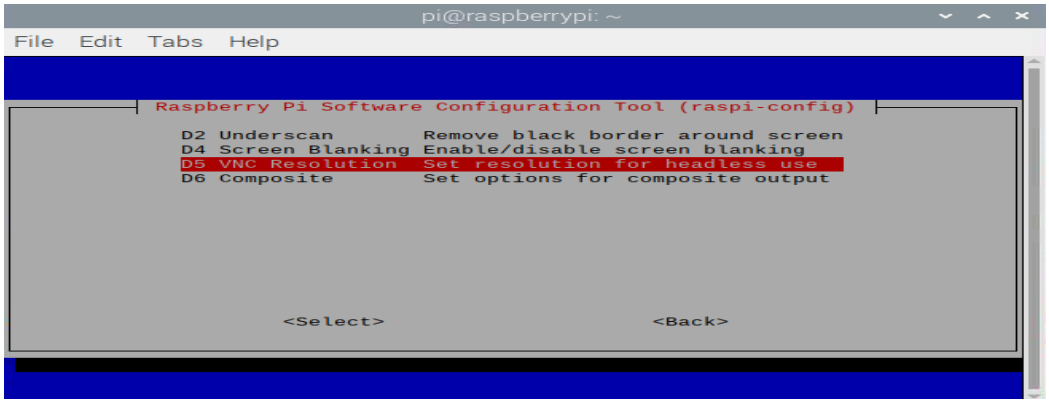


Figure 56:Activer VNC.

On peut régler la résolution du serveur VNC dans Display Option -> Set resolution for headless use

- A choisir en fonction de la résolution de l'écran sur lequel on développe (la résolution du VNC doit être plus petite que celle de l'écran).

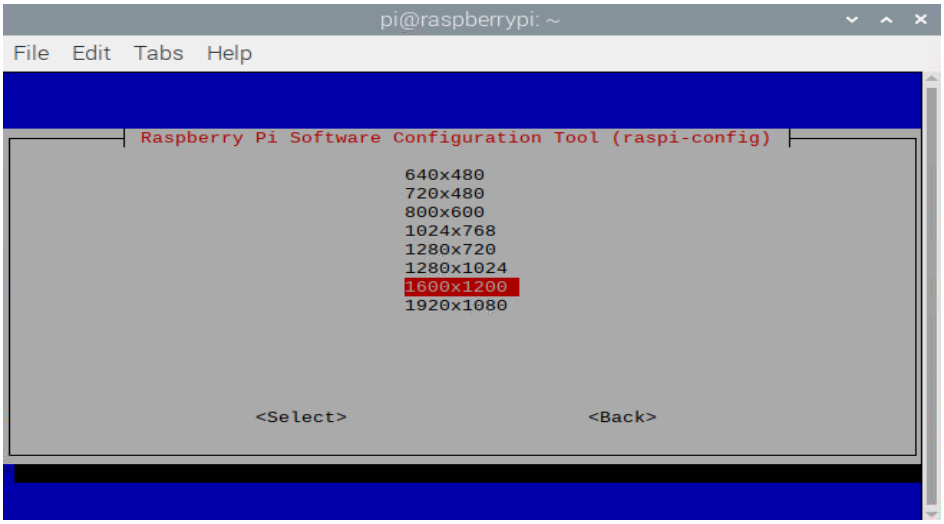


Figure 57:Résolution pour écran du raspberry pi .

Lorsque l'on valide les modifications, un reboot du Raspi se fait automatiquement. Client VNC Windows.

Le client VNC le plus facile à mettre en œuvre est VNC Viewer Client RealVNC VNC Viewer On place l'adresse IP du Raspberry PI et l'on appuie sur Entrée. (Ne pas cliquer sur Ouvrir une session).

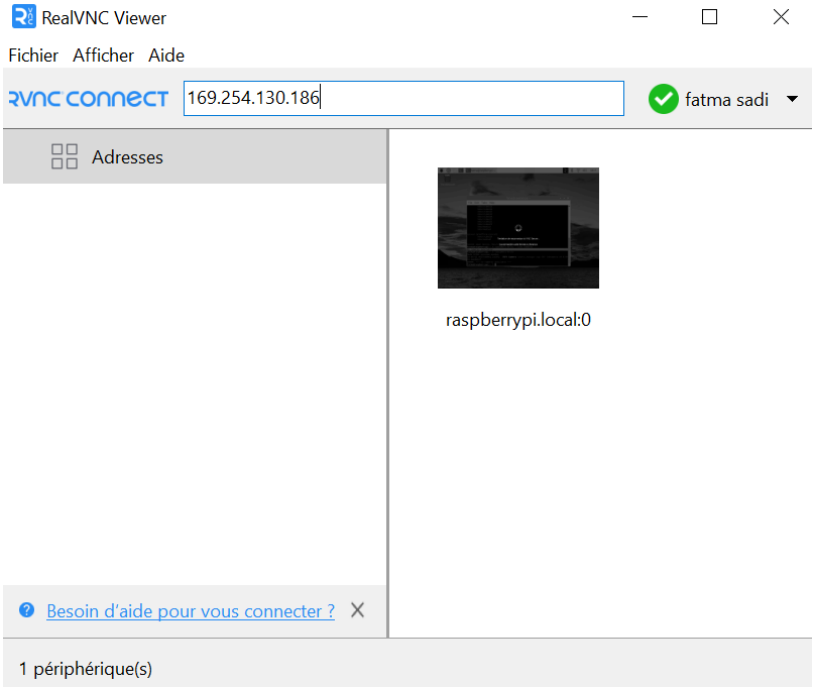


Figure 58:vnc viewer.

On renseigne :

- le user : pi
- le mdp : rasp (si c'est votre mdp)

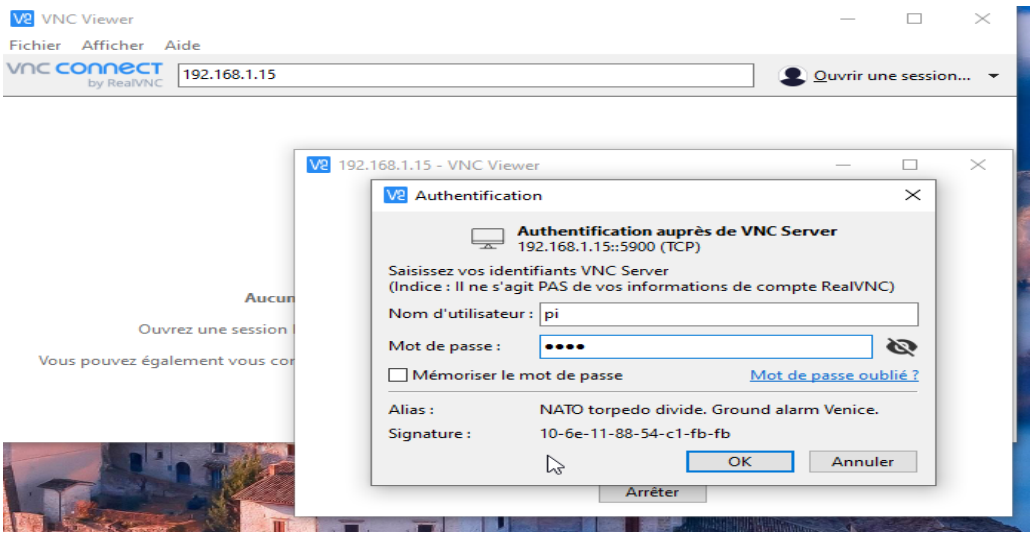


Figure 59:fenêtre saison nom d'utilisateur et mot de passeOn arrive sur le bureau, il faut à nouveau se loger avec le bon mdp (rasp).

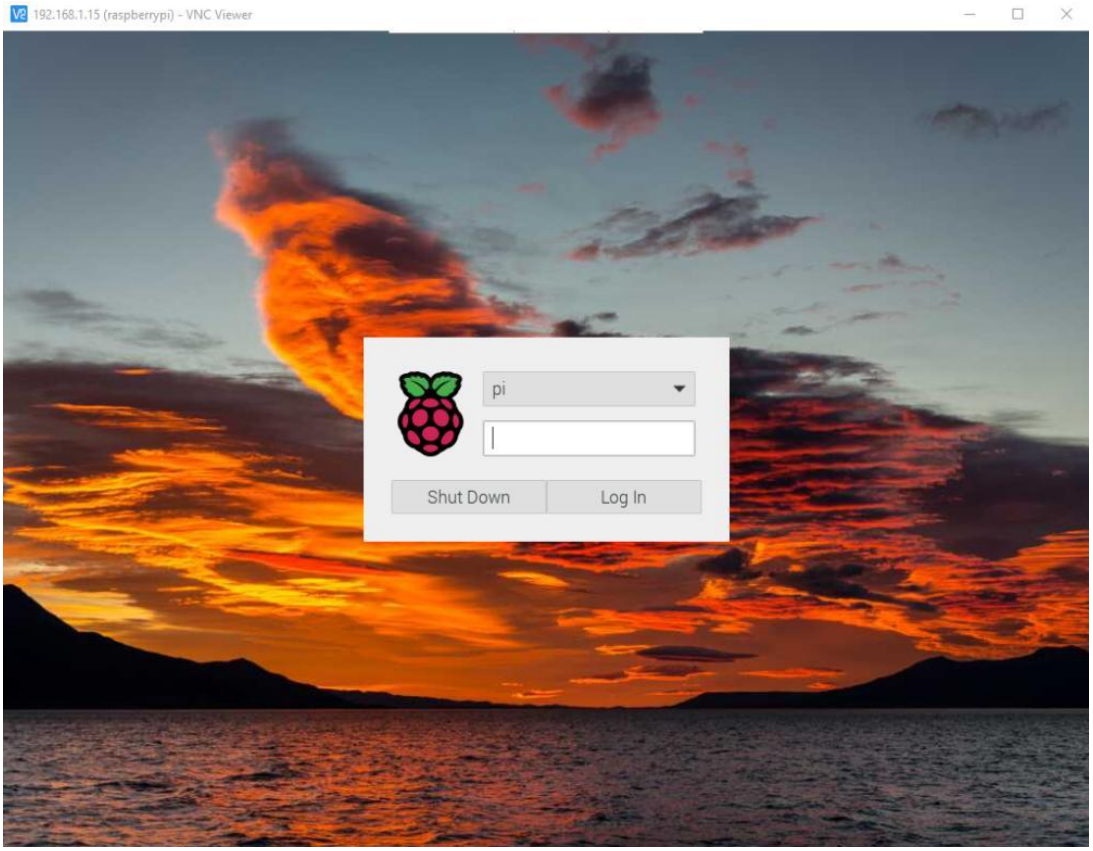


Figure 60:bureau du Raspberrypi.

4. Les Bibliothèque nécessaire à installer :

- **Les étapes qu'on suivi pour installer OPEN CV dans la carte Raspberry pi 4 :**

Étape 1 : Installer les dépendances :

- **Mise à jour des packages existants :**

Exécutez la commande suivante pour mettre à jour et mettre à niveau les packages de votre système :

```
tinhinane@raspberrypi:~ $ sudo apt-get update
Hit:1 http://security.debian.org/debian-security bullseye-security InRelease
Hit:2 http://deb.debian.org/debian bullseye InRelease
Hit:3 http://deb.debian.org/debian bullseye-updates InRelease
Hit:4 http://archive.raspberrypi.org/debian bullseye InRelease
Reading package lists... Done
```

sudo apt-get update

```
Reading package lists... Done
tinhinane@raspberrypi:~ $  sudo apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libfuse2 libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  bsdextrautils bsdutils chromium-browser chromium-browser-l10n
  chromium-codecs-ffmpeg-extra eject fdisk fonts-opensymbol ghostscript
  gstreamer1.0-alsa gstreamer1.0-plugins-base gstreamer1.0-x less libblkid1
  libc-bin libc-dev-bin libc-devtools libc-l10n libc6 libc6-dbg libc6-dev
  libdav1d4 libfdisk1 libglib2.0-0 libglib2.0-bin libglib2.0-data libgs9
  libgs9-common libgstreamer-gl1.0-0 libgstreamer-plugins-base1.0-0
  libjavascriptcoregtk-4.0-18 libjuh-java libjurt-java liblibreoffice-java
  libmount1 libreoffice libreoffice-base libreoffice-base-core
  libreoffice-base-drivers libreoffice-calc libreoffice-common
  libreoffice-core libreoffice-draw libreoffice-gtk3 libreoffice-help-common
  libreoffice-help-en-gb libreoffice-impress libreoffice-java-common
  libreoffice-l10n-en-gb libreoffice-math libreoffice-nlpsolver
  libreoffice-report-builder libreoffice-report-builder-bin
  libreoffice-script-provider-bsh libreoffice-script-provider-js
  libreoffice-script-provider-python libreoffice-sdbc-firebird
```

sudo apt-get upgrade

```
After this operation, 1,155 kB disk space will be freed.
Do you want to continue? [Y/n] Y
```

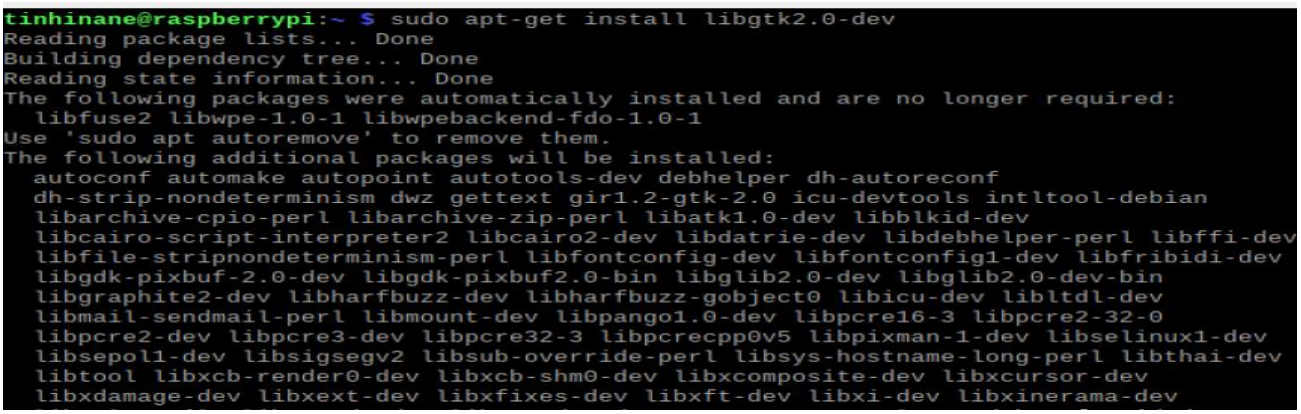
```
Processing triggers for libc-bin (2.31-13+rpt2+rpi1+deb11
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for gnome-icon-theme (3.12.0-3) ...
Processing triggers for shared-mime-info (2.0-1) ...
Processing triggers for mailcap (3.69) ...
Processing triggers for fontconfig (2.13.1-4.2) ...
Processing triggers for desktop-file-utils (0.26-1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for gnome-menus (3.36.0-1) ...
```

- **Installation des packages Image I/O :**

Pour bénéficier de support pour différents formats de fichiers image, installez-les packages nécessaires à l'aide de :

sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev

sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev



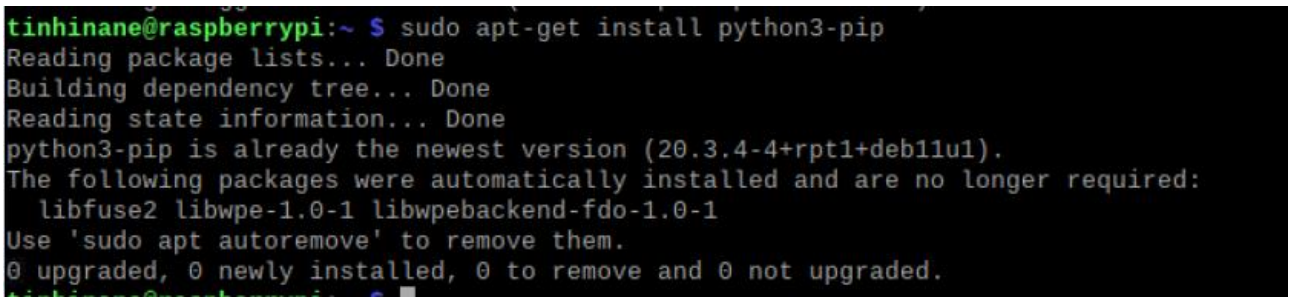
- **Configuration des packages d'E/S vidéo :**

Pour gérer différents formats de fichiers vidéo et travailler avec des flux vidéo, utilisez les commandes ci-dessous :

sudo apt-get install libxvidcore-dev libx264-dev

sudo apt-get install libgtk2.0-dev

Dépendances supplémentaires pour l'optimisation d'OpenCV : Pour une optimisation accrue des opérations OpenCV, installez ces dépendances supplémentaires :



sudo apt-get install libatlas-base-dev gfortran

Étape 2 : Installation de pip (Outil de gestion des packages) :

Si vous n'avez pas encore installé pip pour Python 3, exécutez la commande ci-dessous :

sudo apt-get install python3-pip

```
tinhinane@raspberrypi:~ $ sudo apt-get install python3-pip
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3-pip is already the newest version (20.3.4-4+rpt1+deb11u1).
The following packages were automatically installed and are no longer required:
  libfuse2 libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

Étape 3 : Installation de la bibliothèque Numpy :

Numpy fournit des fonctionnalités mathématiques et numériques essentielles pour OpenCV. Si vous ne l'avez pas encore installé, utilisez la commande :

pip install numpy

```
tinhinane@raspberrypi:~ $ pip install numpy
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: numpy in ./.local/lib/python3.9/site-packages (1.26.4)
```

Étape 4 : Accès à OpenCV dans le dépôt Raspbian par défaut :

Pour localiser OpenCV dans le dépôt Raspbian Buster par défaut, utilisez la commande :

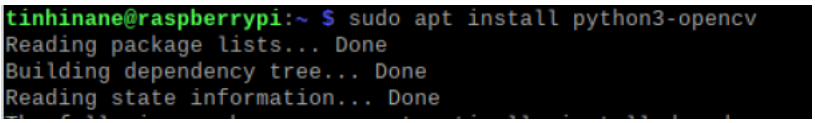
apt list python opencv

```
tinhinane@raspberrypi:~ $ apt list python*opencv*
Listing... Done
python3-opencv-apps/oldstable,oldstable 2.0.2-3 all
python3-opencv/oldstable 4.5.1+dfsg-5 arm64
python3-opencv/oldstable 4.5.1+dfsg-5 armhf
```

```
tinhinane@raspberrypi:~ $ sudo apt-get install libavcodec-dev libavformat-dev libswscale-
dev libv4l-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libfuse2 libwpe-1.0-1 libwpebackend-fdo-1.0-1
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libavutil-dev libswresample-dev
The following NEW packages will be installed:
  libavcodec-dev libavformat-dev libavutil-dev libswresample-dev libswscale-dev
  libv4l-dev
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 7,280 kB of archives.
After this operation, 27.5 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Étape 5 : Installation d'OpenCV :

Exécutez la commande suivante pour installer OpenCV sur un Raspberry Pi.



Étape 6: Vérification de l'installation d'OpenCV :

Pour confirmer l'installation d'OpenCV, utilisez :

apt show python3-opencv

Après l'exécution, vous devriez voir que la dernière version est installée avec succès !

5. Le choix de la base de données :

Pour notre projet de détection d'objets, nous avions besoin d'une base de données adaptée. Bien qu'il existe plusieurs bases de données populaires telles que MS COCO, qui propose 80 classes d'objets, et PASCAL VOC, qui en offre 20, nous avons initialement décidé de créer et d'entraîner notre propre ensemble de données pour répondre à nos besoins spécifiques. Nous avons sélectionné et annoté des objets pertinents à l'intérieur de la maison pour personnaliser notre modèle de détection.

Cependant, nous n'avons pas réussi à entraîner complètement notre modèle avec notre propre base de données en raison de limitations de temps et de ressources. Par conséquent, nous avons décidé d'utiliser la base de données MS COCO, déjà bien établie et riche en annotations, pour l'entraînement initial de notre modèle. Ensuite, nous avons ré-entraîné notre modèle en y ajoutant quelques classes spécifiques supplémentaires, en utilisant nos propres annotations pour compléter les classes manquantes. Cette approche nous a permis de bénéficier des avantages d'une base de données robuste tout en adaptant le modèle à nos besoins particuliers.

6. Prétraitement de donnée :

6.1 Le choix de classe :

Identifier les 15 classes d'objets pertinents dans un environnement domestique parmi les 80 classes disponibles dans MS COCO. Cela pourrait inclure : bouteille, chaise, table à manger, canapé, plante en pot, écran de télévision, personne, laptop, book, fourchette, souris, clavier, cuillère, un verre, couteau.

6.2 Préparer la base de données MS COCO pour YOLOv5 :

Pour préparer la base de données MS COCO pour YOLOv5 sur un Raspberry Pi, suivez ces étapes:

- **Téléchargement des données :** on va commencer par Télécharge les ensembles de données d'entraînement et de validation ainsi que leurs annotations depuis le site officiel de MS COCO. Il faut assurer suffisamment d'espace de stockage disponible sur votre Raspberry Pi pour ces fichiers.
- **Installation des dépendances : on** Installe les outils nécessaires sur la Raspberry Pi. Esnuite Python est déjà installé. On utilisant pip pour installer pycocotools, une bibliothèque Python requise pour manipuler les annotations COCO.
- **Conversion des annotations :** Convertisse les annotations au format YOLOv5 on utilisant « Create RectBox « mais Dans terminal du Raspberry on écrit la commande labelImg .
- **Organisation des données :** ensuite on Organise les données dans une structure de dossier appropriée sur la Raspberry Pi. Les images et leurs fichiers d'annotations convertis doivent être regroupés dans des répertoires distincts pour l'entraînement et la validation.
- **Configuration YAML :** Création ou modifier un fichier YAML pour spécifier les chemins vers les données d'entraînement et de validation, ainsi que le nombre de classes dans votre ensemble de données et leurs noms.
- **Entraînement avec YOLOv5 :** on utilise le script d'entraînement fourni avec YOLOv5 sur votre Raspberry Pi en spécifiant le fichier YAML configuré précédemment. Cela lancera le processus d'entraînement du modèle de détection d'objets.

6.3 Entraînement du Modèle YOLOv5 sur Google Colab :

Pour entraîner le modèle YOLOv5 sur Google Colab, on commence par ouvrir Google Colab et créer un nouveau notebook. On assure que l'environnement d'exécution utilise un GPU en sélectionnant

- **Création d'un Compte Google et Accès à Google Colab :**

Pour utiliser Google Colab, commençant par créer un compte Google si on n'a pas déjà un. Ensuite, on accède à Google Colab avec le site colab.research.google.com.

- **Préparation de l'Environnement Colab :**

a) On ouvre un nouveau notebook Colab.
b) On monte notre Google Drive pour accéder à nos données de formation :

Le code:

```
from google.colab import drive
drive.mount('/content/drive')
```

- **Installation des Dépendances YOLOv5 :**

On Installe les bibliothèques nécessaires pour entraîner YOLOv5 :

Le code :

```
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
!pip install -r requirements.txt
```

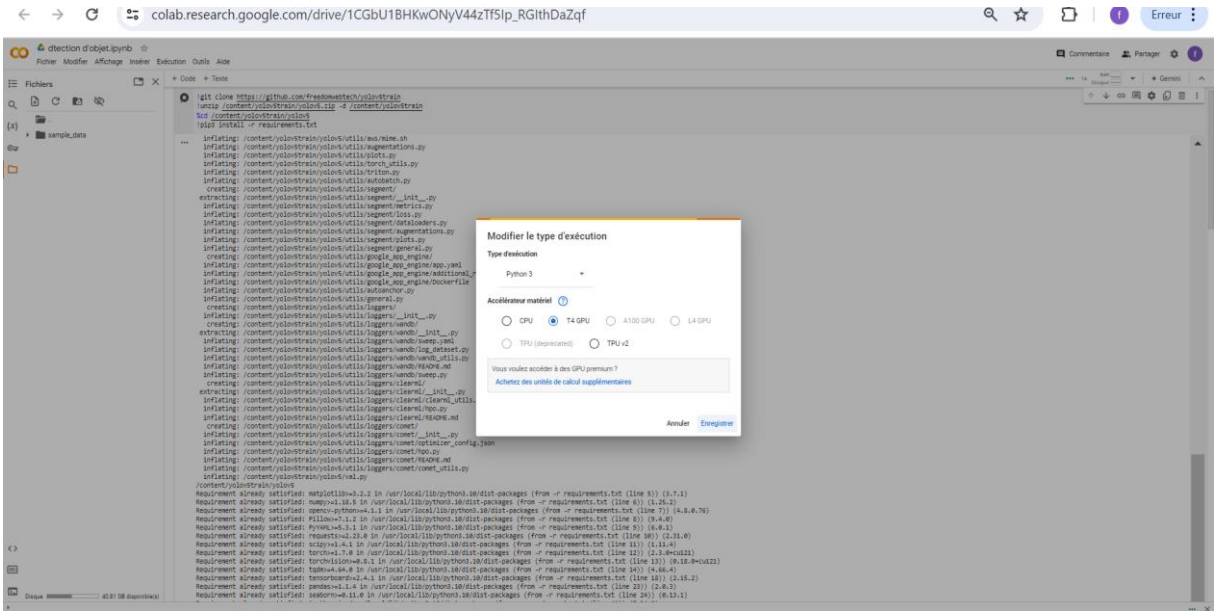


Figure 61:Type d'exécution GPU.

- **Entraînement du Modèle :**

On Prépare nos données et on configure les paramètres d'entraînement dans le fichier data.yaml. Ensuite, on lance l'entraînement avec la commande suivante :

Le code:

```
!python train.py --img 640 --batch 16 --epochs 50 --data data.yaml --weights yolov5s.pt
```

- python train.py :

Cette partie de la commande indique à Python d'exécuter le script qui contient le code principal pour l'entraînement du modèle YOLOv5.

- --img 640 :

Ce paramètre spécifie que les images seront redimensionnées à une taille de 640x640 pixels pendant l'entraînement

- --batch 16 :

Définit la taille du lot (batch size) à 16. Cela signifie que 16 images seront traitées simultanément avant de mettre à jour les poids du modèle. Le choix de cette taille affecte la vitesse d'entraînement et l'utilisation de la mémoire GPU.

- --epochs 50 :

Indique que l'entraînement se déroulera sur 50 époques. Une époque représente une passe complète à travers l'ensemble des données d'entraînement.

- --data data.yaml :

Le fichier YAML data.yaml qui contient les configurations des données d'entraînement et de validation. Ce fichier inclut généralement les chemins vers les répertoires contenant les images et leurs annotations, ainsi que le nombre de classes et les noms de ces classes.

- --weights yolov5s.pt :

Utilise les poids initiaux du modèle YOLOv5 small (yolov5s.pt) pour démarrer l'entraînement. Les poids pré-entraînés sont bénéfiques car ils permettent au modèle de commencer à partir d'une configuration optimisée pour la détection d'objets.

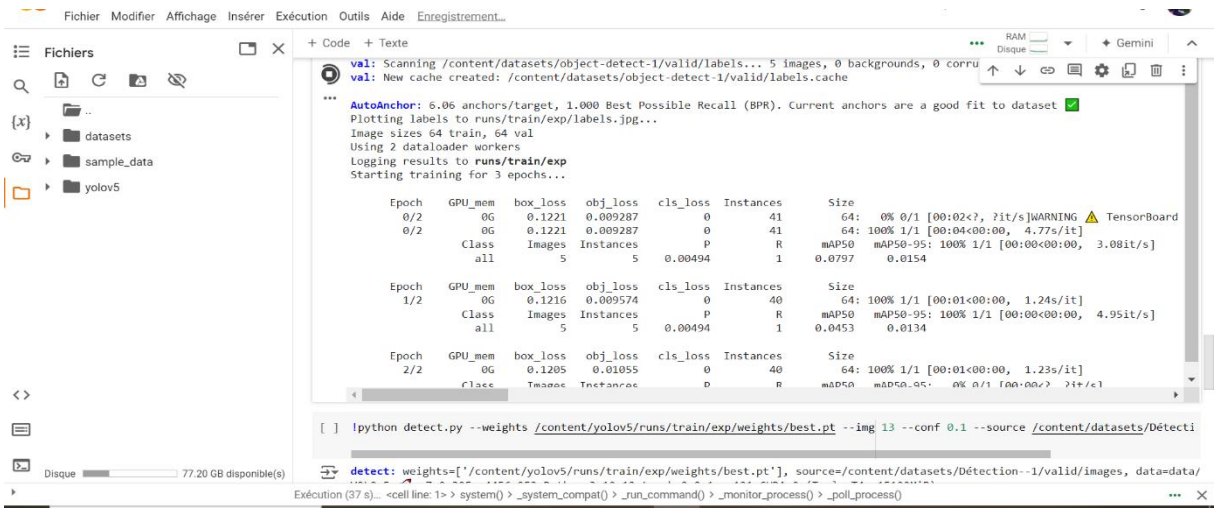


Figure 62: Train YoloV5 dans Google colab.

6.4 Exportation du Modèle Entraîné :

Après l'entraînement, on exporte le modèle pour une utilisation ultérieure sur la Raspberry Pi:

Le code:

```
!python export.py --weights runs/train/exp/weights/best.pt --img 640 --batch 1 --device 0
```

- !python export.py: Exécute le script export.py pour exporter le modèle YOLOv5.
- --weights runs/train/exp/weights/best.pt: Spécifie les poids du meilleur modèle entraîné à utiliser.
- --img 640: Redimensionne les images d'entrée à 640x640 pixels.
- --batch 1: Traite chaque image individuellement (batch size de 1).
- --device 0: Utilise le premier GPU disponible pour l'inférence.

6.5 Prétraitement de l'image de test :

La commande `python detect.py --weights yolov5s.pt --source 0` exécute le script de détection d'objets YOLOv5 avec les poids du modèle `yolov5s.pt`, utilisant la première caméra disponible comme source vidéo.

Le code :

```
!python detect.py --weights yolov5s.pt --source 0
```

7. Test et résultat :

Nous avons testé les performances du modèle YOLOv5 ré-entraîné sur neuf sous-classes d'objets domestiques, sélectionnés à partir du jeu de données MS COCO. Nos résultats montrent une efficacité notable même dans des scénarios complexes, comme illustré dans la figure suivante [59].

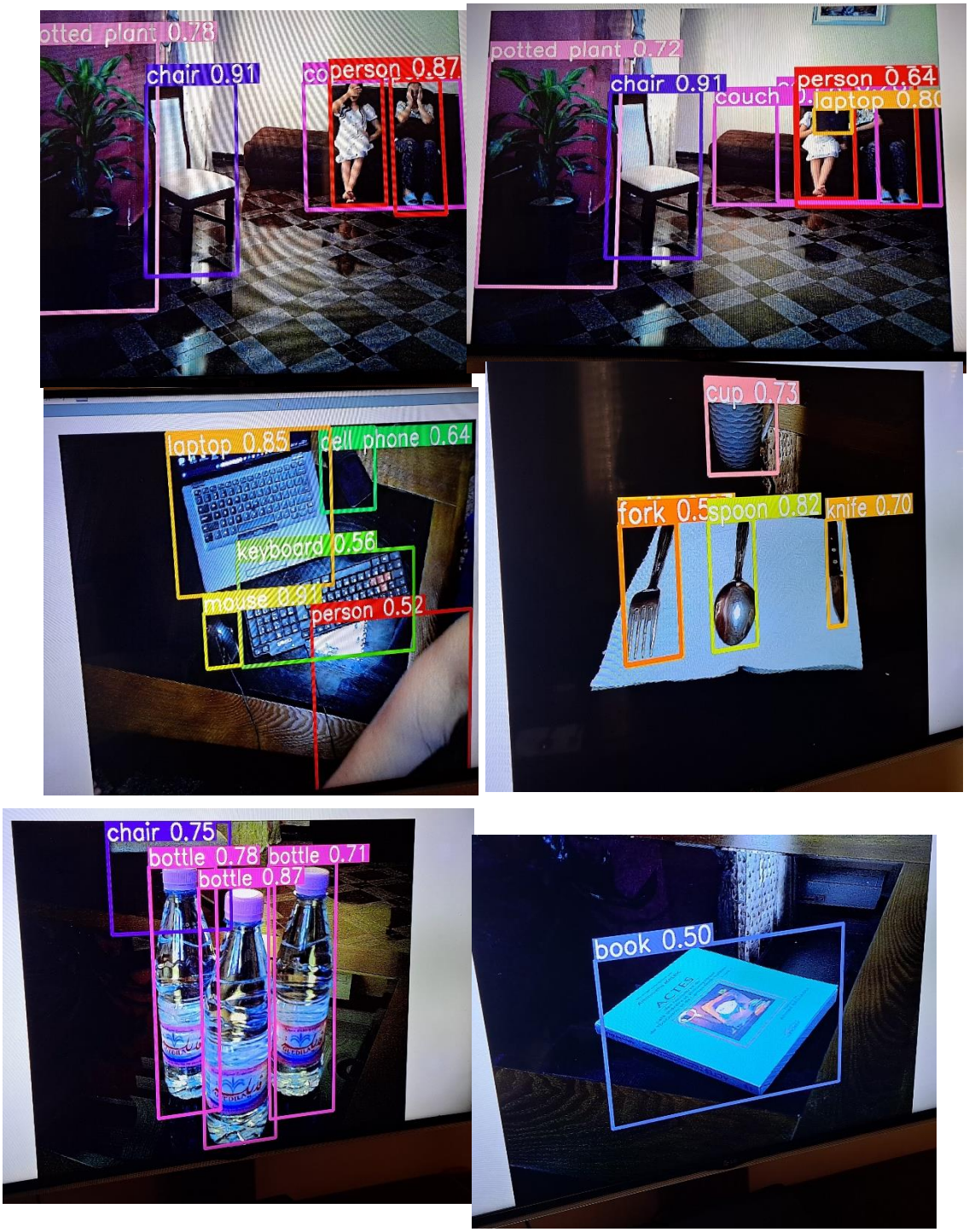


Figure 63: Résultat Obtenu de Détection par Yolov5.

YOLOv5 est capable de détecter des objets même lorsqu'ils sont situés à une certaine distance de la caméra. Voici dans la figure 63 :

Figure 64: Détection d'objets par model yolov5 à distance.

a) Précision :

La précision est le pourcentage de détections correctes. Il met en évidence l'exactitude des prédictions. Il se calcule par ce ratio :

TP = les objets correctement détectés.

FP = les objets détectés à tort.

$$précision = \frac{vrai\ positif}{vrai\ positif + faux\ positif}$$

b) Rappel :

Le rappel est un indicateur qui mesure la capacité du modèle à prédire l'ensemble des résultats attendus. Calculer par [56]:

FN = les objets réels non détectés par le modèle

$$Rappel = \frac{vrai\ positif}{vrai\ positif + faux\ négatif}$$

c) Average Precision (AP) :

Calcule la surface sous la courbe de précision-rappel, fournissant une valeur unique qui englobe les performances de précision et de rappel du modèle

$$\mathrm{AP} = \sum_{n} (R_n - R_{n-1}) P_n$$

d) mAP

Est la précision moyenne sur toutes les classes détectées, obtenue à partir de la courbe Rappel pour chaque classe, fournissant une évaluation globale de la performance du modèle sur l'ensemble des données

$$mAP = \frac{\sum_{q=1}^{Q} AP_{(q)}}{Q},$$

Le résultat de l'apprentissage obtenu avec c'est 100% Parce que objectifs est atteint on arriver détecter les objets, dans le tableau suivant on a affiché tous les résultats obtenus :

Class	***Precision***	***Class***	***précision***
bouteille	87%	Personne	87%
chaise	91%	laptop	85%
table à manger	61.42%	souris	91%
canapé	87.84%	clavier	56%
plante en pot	78%	cuillère	82%
écran de télévision	74.02%	fourchette	56%
couteau	70%	Un verre	73%
Téléphone	64%	Livre	50%

Tableau 2 : Résultats de détection du système.

Les résultats montrent que YOLOv5 a une précision globale de 93%, ce qui indique la proportion d'objets correctement détectés par rapport à tous les objets réels présents dans les images testées. Cette précision varie selon les classes d'objets : par exemple, les chaises et les souris sont détectées avec une précision d'environ 91%, tandis que les livres et les claviers le sont à 50% et 56% respectivement. Pour améliorer ces résultats, il est recommandé d'enrichir l'ensemble de données en ajoutant plus d'exemples variés et d'utiliser des techniques d'augmentation des données. Une évaluation continue et une analyse des erreurs de détection sont essentielles pour optimiser la performance du modèle et atteindre ses objectifs de détection avec précision.

6. Discussion :

Dans ce dernier chapitre, nous avons expliqué comment configurer une carte Raspberry Pi 4, installer Python et utiliser Google Colab pour entraîner le modèle YOLOv5 pour la détection d'objets.

Ces étapes forment la base d'un système de détection d'objets en temps réel, adaptable aux besoins spécifiques de chaque projet. Nous avons détaillé la configuration de la Raspberry Pi 4, y compris l'installation du système d'exploitation, la mise à jour des paquets et l'optimisation des paramètres système. Cette préparation garantit que la Raspberry Pi 4 peut exécuter des algorithmes complexes comme YOLOv5 malgré ses ressources limitées.

Pour l'entraînement du modèle, nous avons utilisé Google Colab, qui offre des GPU gratuits, accélérant ainsi le processus d'entraînement.

Nous avons décrit comment préparer les données, configurer l'environnement et exécuter l'entraînement. Une fois le modèle entraîné, nous avons transféré les poids sur la Raspberry Pi pour effectuer la détection d'objets en temps réel, en installant les dépendances nécessaires et en optimisant le script de détection pour fonctionner efficacement sur le matériel limité.

Ces étapes peuvent être adaptées pour répondre à des besoins spécifiques, comme la détection de nouveaux objets en ré-entrainant le modèle avec des données supplémentaires.

La flexibilité de la Raspberry Pi et de Python permet de développer des solutions personnalisées pour diverses applications de détection d'objets.

Conclusion

En conclusion, notre parcours à travers notre travail a mis en lumière l'impact considérable de l'intelligence artificielle, notamment dans le domaine de la détection d'objets en temps réel. Nous avons observé comment la vision par ordinateur, combinée à des modèles de pointe tels que YOLOv5, a ouvert la voie à des applications innovantes et prometteuses, de la surveillance à la robotique.

L'utilisation d'un système embarqué Raspberry Pi 4 comme plateforme pour déployer notre système de détection d'objets en temps réel a démontré la polyvalence et l'accessibilité de cette technologie. Nous avons vu comment YOLOv5, avec son architecture optimisée pour des ressources limitées, s'est avéré être un choix judicieux pour des applications nécessitant un traitement en temps réel sur des plates-formes embarquées.

À travers nos explorations, nous avons abordé les défis et les opportunités rencontrés lors de l'intégration de YOLOv5 sur Raspberry Pi 4. De la compréhension des principes fondamentaux de l'intelligence artificielle à l'analyse détaillée des modèles de détection d'objets, notre travail a offert une perspective holistique sur ce domaine en constante évolution.

En structurant notre mémoire en quatre chapitres distincts, nous avons pu explorer chaque aspect de notre sujet en profondeur, fournissant ainsi une base solide pour comprendre et mettre en œuvre notre système de détection d'objets en temps réel.

En fin de compte, notre objectif était de fournir une compréhension approfondie de la détection d'objets en temps réel avec YOLOv5 sur Raspberry Pi 4, tout en ouvrant la voie à de nouvelles opportunités dans le domaine de la vision par ordinateur embarquée. La majorité des objectifs tracés dans de ce travail ont été atteints, mais il reste toujours des perspectives et des améliorations possibles qui peuvent encore être réalisé dans le futur, telles que :

- Utiliser un autre model de détection d'objet en temps réel tel que le model SSD et faire une comparaison avec d'autre model de détection en temps réel.
- Changer le backbone de yolov5 utiliser MobileNet et comparé le résultat obtenu avec le DarkNet.
- Essayer d'entrer le model YOLO sur une autre base de données qui contient plusieurs classes pour donner plus de performance à notre système.
- Utiliser les algorithmes d'amélioration de la qualité des images pour améliorer la détection.
- Mettre à jour le modèle régulièrement.

Nous espérons que notre travail servira de point de départ pour de futures innovations et recherches dans ce domaine passionnant et en constante évolution.

Bibliographique

[1] Russell, S., & Norvig, P. (2010). Intelligence artificielle: Avec plus de 500 exercices. Pearson Education France.
[2] Meghnagi, R. R. (2022). Transposition et adaptation d'un système de contrôle par intelligence artificielle au sein d'une unité de préparation centralisée d'injectables destinés à la pédiatrie (Doctoral dissertation).
[3] Talbi, I. (2021, December 24). Qu'est-ce que la segmentation d'images
[4] Talend. (n.d.). Qu'est-ce que l'intelligence artificielle et pourquoi est-ce important ?
[5] Amazon Web Services. (2023). Computer Vision
[6] Marie-Laurence. (2024, 2 avril). Computer vision. LePont
[7] Talbi, I. (2021, December 24). Qu'est-ce que la segmentation d'images
[8] Amazon Web Services. (2023). Qu'est-ce que la vision par ordinateur ?
[9] Kejriwal, K. (2023b, juillet 21). Reconnaissance d'image Vs. Vision par ordinateur : quelles sont les différences ?
[10] Kejriwal, K. (2023b, juillet 21). Reconnaissance d'image Vs. Vision par ordinateur : quelles sont les différences ?
[11] Hammouche, Y. (2021). La reconnaissance de mouvement dans une séquence d'images.
[12] Comprendre l'informatique. (2021, 1 décembre). Introduction au Machine Learning / Apprentissage automatique [Vidéo]. YouTube.
[13] Sur, F. (2023). Enseignement de l'apprentissage automatique à destination d'ingénieur· e· s généralistes. Academic Journal of Civil Engineering,
[14] Russell, S., & Norvig, P. (2020). Intelligence artificielle : Une approche moderne (4e éd.). Pearson Education.
[15] Kassel, R. (2023, 9 novembre). Apprentissage Non Supervisé : principe et utilisation. Formation Data Science | DataScientest.com.
[16] Université Badji Mokhtar-Annaba. (2020). Techniques analyse Big Data [Cours en ligne]. Télé-enseignement Faculté de Technologie.
[17] Université Badji Mokhtar-Annaba. (2020). Techniques analyse Big Data [Cours en ligne]. Télé-enseignement Faculté de Technologie.

[18] Boughaba, M., & Boukhris, B. (2016/2017). L'apprentissage profond (Deep Learning) pour la classification et la recherche d'images par le contenu Université d'Ouargla.
[19] Senani, S. (2019). Réseaux de neurones convolutionnels pour la détection précoce de la rétinopathie diabétique. Mémoire de Master, Université Mouloud Mammeri.
[20] Noukas, O. (2023, 1 juillet). Etude Comparative des CNNs et de L'algorithme K-NN en mammographie.
[21] Kassel, R. (2023b, novembre 9). Deep Neural Network : qu&rsquo ; est-ce qu&rsquo ; un réseau de neurones profond ? Formation Data Science | DataScientest.com.
[22] School IA. (2024, 5 février). Comprendre les Recurrent Neural Networks : principes et applications.

[23] BENADJAOUD, I. (2022). Utilisation des Réseaux Antagonistes Génératifs pour le traitement des images médicales ophtalmologiques : cas de la rétinopathie diabétique (Mémoire de maîtrise). Université A/Mira de Béjaia, Faculté des Sciences Exactes, Département d'Informatique.

[24] Elias, G. (2024, 12 juin). What is deep learning ?

[25] Elias, G. (2024, 12 juin). What is deep learning ?
[26] Labomedia. (2020). Comment configurer le nombre de couches et de nœuds dans un réseau neuronal
[27] Boughaba, M., & Boukhris, B. (2017). L`apprentissage profond (Deep Learning) pour la classification et la recherche d'images par le contenu (Mémoire de Master Professionnel, Université Kasdi Merbah Ouargla
[28] ABIDI, A., & SLIMANI, M. (2017). Suivi d'objets en mouvement (Mémoire de fin d'étude, Université Ahmed Draia – Adrar.

[29] ABIDI, A., & SLIMANI, M. (2017). Suivi d'objets en mouvement (Mémoire de fin d'étude, Université Ahmed Draia – Adrar.

[30] Yedjour, H. (2010). Détection de contours et suivi d'objet dans une séquence d'images par les réseaux de neurones impulsionnels (Mémoire de Magister, Université Mohamed Boudiaf (USTO) Oran

[31] Yedjour, H. (2010). Détection de contours et suivi d'objet dans une séquence d'images par les réseaux de neurones impulsionnels (Mémoire de Magister, Université Mohamed Boudiaf (USTO) Oran

[32] Robert, J. (2020, June 25). Convolutional Neural Network: Tout ce qu'il y a à savoir. DataScientest.

[33] SEKKIL, H. M., & MEBROUKI, M. (2021). Etude comparative entre les différentes architectures des réseaux de neurones convolutifs (CNNs) pour la détection de la fatigue du conducteur (Doctoral dissertation, Directeur: Melle. Imane NEDAJR/Co-Directeur: M. MEGNAFI Hichem).

[34] CHEBBAH, H., & KETAMI, S. (2020). Détection et reconnaissance d'objets à partir des images issues d'une mono-caméra, UMMTO.

[35] Selmane, R., & Delil, S. (2021). Détection et reconnaissance d'objets en temps réel en utilisant l'algorithme YOLOv4 (Master's UMMTO).

[36] Selmane, R., & Delil, S. (2021). Détection et reconnaissance d'objets en temps réel en utilisant l'algorithme YOLOv4 (Master's UMMTO). [38] Girshick, R. "Fast R-CNN." IEEE International Conference on Computer Vision, 2015.

[39] Selmane, R., & Delil, S. (2021). Détection et reconnaissance d'objets en temps réel en utilisant l'algorithme YOLOv4 (Master's UMMTO).

[40] CHEBBAH, H., & KETAMI, S. (2020). Détection et reconnaissance d'objets à partir des images issues d'une mono-caméra, UMMTO.

[41] Selmane, R., & Delil, S. (2021). Détection et reconnaissance d'objets en temps réel en utilisant l'algorithme YOLOv4 (Master's UMMTO).

[42] CHEBBAH, H., & KETAMI, S. (2020). Détection et reconnaissance d'objets à partir des images issues d'une mono-caméra

[43] CHEBBAH, H., & KETAMI, S. (2020). Détection et reconnaissance d'objets à partir des images issues d'une mono-caméra

[44] Blent Team. (2022, June 8). YOLO: détection sur les images avec TensorFlow. Blent.ai.

[45] Terven, J., Córdova-Esparza, D., & Romero-González, J. (2023b). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction.

[46] Terven, J., Córdova-Esparza, D., & Romero-González, J. (2023b). A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction .

[47] Blent Team. (2022, June 8). YOLO: détection sur les images avec TensorFlow. Blent.ai.

[48] Atam, A. (2021, September 25). Architecture of YOLOv3. OpenGenus IQ: Learn Algorithms, DL, System Design.

[49] Manika. (2024, March 21). The ultimate guide to YOLO3 architecture, ProjectPro

[50] Bochkovskiy, A., Liao, H.-Y. M., & Wang, C.-Y. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection.

[51] Solawetz, J. (2020, June 15). How to train a custom object detection model with YOLO v5. Towards Data Science.

[52] Harzallah, H. (2023). Conception et réalisation d'un système pour assister les malvoyants (Master's thesis). Université Mohamed Khider

[53] Solawetz, J. (2020, June 15). How to train a custom object detection model with YOLO v5. Towards Data Science.

[54] Solawetz, J. (2020, June 15). How to train a custom object detection model with YOLO v5. Towards Data Science.

[55] Solawetz, J. (2020, June 15). How to train a custom object detection model with YOLO v5. Towards Data Science.

[56] Gur Arie, L. (2022, March 14). The practical guide for Object Detection with YOLOv5 algorithm. Towards Data Science

[57] Gur Arie, L. (2022, March 14). The practical guide for Object Detection with YOLOv5 algorithm. Towards Data Science

[58] Benosman, M. O., & Benhammou, S. (2020). Conception et réalisation d'un système de télésurveillance en temps réel à base de Raspberry (Doctoral dissertation).

[59] Ouakline, N., & Benchaba, A. (2018). Conception et réalisation d'une maison intelligente via Raspberry Pi 3 (Mémoire de fin d'étude, UMMTO).

[60] Ahmed, A. (2016). Conception, étude et réalisation d'un système domotique à base d'une carte Arduino Méga et d'une carte Raspberry pi 2 (Doctoral dissertation, Université Mouloud Mammeri).

[61] Creactivit. (2023, December 5). Évolution des Raspberry Pi de leur création à aujourd'hui. CRÉACTIVIT.

[62] Contributeurs aux projets Wikimedia. (2024b, mai 29). Raspberry Pi.

Oxatis. (2016). Raspberry Pi 4 Modèle B caractéristiques.

[63] Jérémy Robert. (2023, avril 17). Python Focus sur le langage le plus populaire. DataScientest.

[65] Kassel, R. (2022, September 1). OpenCV: tout savoir sur le principal outil de Computer Vision. DataScientest.

[66] Eleyehou, D. (2024, February 8). Google Colab : la force du cloud pour l'apprentissage automatique. DataScientest.

[67] Pytorch, A. D. I. (2018). Pytorch.

[68] Miele, V., Dray, S., & Gimenez, O. (2021). Deep Learning.

[69] Douarre, C., Rousseau, D., & Chareyron, D. (2021, 11 octobre). Exemples d'applications de l'apprentissage profond. Laboratoire Angevin de Recherche en Ingénierie des Systèmes (LARIS), Université d'Angers et Laboratoire d'InfoRmatique en Image et Systèmes d'information (LIRIS), Lyon.

[70] L'IA et la vision par ordinateur. (2023, 9 novembre). Parlons Sciences. / tos SE. (2022, août 11). Atos Computer Vision Platform - Modèles d'IA pré-entrainés & sur-mesure. Atos. / Bouvier, A. (2020, 14 novembre). La voiture autonome ; : progrès techniques et perspectives. La Jaune et la Rouge.

[71] Haseebullah. (2024). Détection d'objets sur un jeu de données personnalisé avec images, vidéos et flux en direct. Fiverr.
[72] Debusschere, V., Petrusev, A., Rigo-Mariani, R., Hadjsaid, N., Esfandiari, B., & Nock, R. (2022). Catégories d'apprentissage automatique.
[73] Hairy, P. (2021, 10 octobre). Les réseaux de neurones récurrents pour les séries temporelles - MetalBlog. MetalBlog.
[74]. Gundamotoko. (2019, 12 juin). Réseau antagoniste génératif (Generative adversarial Network) - Deep learning -. Kongakura.
[75] Robert, J. (2024, 31 janvier). Convolutional Neural Network : Tout ce qu&rsquo ; il y a à savoir. Formation Data Science | DataScientest.com.
[76] Mendaz, K. (2019). Topologie d'un réseau multicouche (MLP). Dans K. Mendaz, PolyCopie de cours commande avancée « Logique floue et réseaux de Neurone ' application à l'électrotechnique'.
[77] ResearchGate. (2016). Résultat de la segmentation par soustraction de fond : (a) Image courante, (b) Image du fond. Adapté de ResearchGate.
[78] Adapté de "The Dense Estimation of Motion and Appearance in Layers", par Michael Black et Ronan Fablet.
[79] Mesbah, F. (2021). Détection d'objets par Deep Neural Network à l'aide du modèle YOLO en temps réel. Mémoire de fin d'études Master, Université 8 Mai 1945 – Guelma, Faculté des Mathématiques, d'Informatique et des Sciences de la Matière, Département d'Informatique
[80]] R. Girshick et autre (2014) « Rich feature hierarchies for accurate object detection and semantic segmentation » IEEE Conference on Computer Vision and Pattern Recognition.
[81] R. Girshick « Fast R-CNN »(2015)Conférence internationale de l'IEEE sur la vision par ordinateur.
[82] S. Ren, K. He, R. Girshick, and J. Sun. (2015)« Faster R-CNN: Towards real-time object detection with region proposal networks ». dans Conférence NIPS.
[83] Huang, C., Zhou, Y., & Xie, X. (2024). Intelligent Diagnosis of Concrete Defects Based on Improved Mask R-CNN. Applied Sciences.
[84] MDPI. (2019). An Improved Single Shot Multibox Detector Method Applied in Body Condition Score for Dairy Cows. Animals
[85] Sp, M. (2017, 19 octobre). Applying YOLO for traffic light color detection - Monish SP - Medium.
[86] Sp, M. (2017, 19 octobre). Applying YOLO for traffic light color detection - Monish SP - Medium.
[87] Sp, M. (2017, 19 octobre). Applying YOLO for traffic light color detection - Monish SP - Medium.
[88] Rosyking Lumbanraja, F. (2022). Processing Images with YOLO Detection System. Dans Aristoteles, A., Syarif, A., Sutyarso, S., & Hidayatullah, A. (Éds.), Identification of Human Sperm based on Morphology Using the You Only Look Once Version 4 Algorithm.
[89] Derakhshani, M. M., Masoudnia, S., Shaker, A. H., & Araabi, B. N. (2019). Assisted Excitation of Activations: A Learning Technique to Improve Object Detectors. ResearchGate.
[90] Panchal, B. Y., Joshi, M., Shah, R. K., Desai, J., & Shah, A. (2023). Deep Learning in Healthcare Informatics. In Deep Learning in Healthcare Informatics (pp. 17-18). ResearchGate.
[91] Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934, 2-7.

[92] Tan, M., Pang, R., & Le, Q. V. (2019). EfficientDet: Scalable and efficient object detection. arXiv.
[93] Tsang, S. (2023b, juin 11). Brief Review : YOLOv5 for Object Detection - Sik-Ho Tsang - Medium.
[94] Zhang, J., Meng, Y., Yu, X., & Tian, J. (2023). PANet: (a) FPN, (b) bottom-up path, (c) adaptive feature pooling, (d) detection branch, (e) fusion layer [Figure]. Dans MBAB-YOLO: A Modified Lightweight Architecture for Real-Time Small Target Detection.
[95] Tsang, S. (2023b, juin 11). Brief Review : YOLOv5 for Object Detection - Sik-Ho Tsang - Medium.
[96] UK-Raspberry Pi 4 Model B, BCM2711 SOC, 2GB DDR4 RAM, USB 3.0, POE enabled. (s. d.). Dzduino Online Store.
[97] Raspberry Pi 4 Model B 4GB SBC Satın al kompent. (s. d.). Fikirleri Hayata Geçirmek Için Komponent Ve Elektronik Pazarı - Kompent.com.
[98] Google Colab. Adapté de "Google Colab - Google Search".
[99] Learning PyTorch : Modules. Adapté de "Learning PyTorch : Modules", par Wei, D. (2024, 27 février). Medium.

Résumé :

Notre travail présente une étude approfondie sur l'utilisation de YOLOv5 sur le Raspberry Pi 4 pour la détection d'objets en temps réel. Nous avons mis en évidence l'impact significatif de l'intelligence artificielle et de la vision par ordinateur, notamment dans le domaine de la détection d'objets, grâce à des modèles comme YOLOv5. Ce modèle démontre la polyvalence et l'accessibilité du Raspberry Pi 4 en tant que plateforme pour déployer des systèmes de détection d'objets, malgré ses limitations en termes de puissance de traitement et de mémoire RAM.

Notre étude détaille le processus d'entraînement du modèle YOLOv5, son intégration sur le Raspberry Pi 4, et les étapes nécessaires pour obtenir une détection précise et rapide. Nous avons examiné les résultats des tests réalisés, évalué les performances du système, et proposé des pistes d'amélioration pour l'avenir. Parmi ces pistes, nous suggérons l'utilisation d'autres modèles de détection, l'exploration de nouveaux backbones, l'entraînement sur des bases de données plus vastes, et l'adoption d'algorithmes d'amélioration de la qualité des images.

En conclusion, ce travail offre une base solide pour la compréhension et la mise en œuvre de systèmes de détection d'objets en temps réel avec YOLOv5 sur le Raspberry Pi 4. Il ouvre la voie à de nouvelles opportunités dans le domaine de la vision par ordinateur embarquée et encourage la poursuite de recherches et d'innovations dans ce domaine en constante évolution.

Abstract:

Our work presents an in-depth study of the use of YOLOv5 on the Raspberry Pi 4 for real-time object detection. We have highlighted the significant impact of artificial intelligence and computer vision, particularly in the field of object detection, thanks to models such as YOLOv5. This model demonstrates the versatility and accessibility of the Raspberry Pi 4 as a platform for deploying object detection systems, despite its limitations in terms of processing power and RAM memory.

Our study details the process of training the YOLOv5 model, its integration on the Raspberry Pi 4, and the steps required to achieve accurate and fast detection. We have examined the results of the tests carried out, evaluated the system's performance, and proposed avenues of improvement for the future. These include the use of other detection models, the exploration of new backbones, training on larger databases, and the adoption of image quality enhancement algorithms.

In conclusion, this work provides a solid basis for understanding and implementing real-time object detection systems with YOLOv5 on the Raspberry Pi 4. It opens up new opportunities in the field of embedded computer vision, and encourages further research and innovation in this constantly evolving field.