الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة أبي بكر بلقايد – تلمسان –
Université Aboubakr Belkaïd – Tlemcen –
Faculté de TECHNOLOGIE

MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Électronique

Spécialité : Instrumentation

Par : Miloudi Adel Hani

Sujet

Conception et implémentation d'une application de détection d'objets à l'aide de TensorFlow

Soutenu publiquement en juin 2023, devant le jury composé de :

Mr NEMMICHE AHMED	MCB	Université de Tlemcen	Président
Mr BENADDA BELKACEM	Professeur	Université de Tlemcen	Examinateur
Mr MOULAI KHATIR NASSIM	MCB	Université de Tlemcen	Encadreur
HADJ ABDELKADER AMINE	Professeur	Université de Tlemcen	Co-Encadreur

Année universitaire : **2022/2023**

Dédicace

À ma mère*, source inépuisable d'amour et de soutien. Je t'offre humblement ce travail en reconnaissance de ta présence bienveillante et de tes encouragements indéfectibles.*

À la mémoire de mon père *regretté, dont l'héritage guide ma voie. Cette thèse est un hommage vibrant à son influence durable.*

À mes trois frères*, compagnons de mes plus beaux souvenirs, je vous remercie pour votre soutien inconditionnel. Cette dédicace vous est offerte avec amour et gratitude.*

À mon encadreur, le Dr Moulai Khatir Nassim*, pour la qualité de ses orientations, de ses conseils avisés, pour sa patience et sa disponibilité tout au long de cette période de travail intense. Je lui suis profondément reconnaissant pour son précieux accompagnement.*

Cette réussite, je la partage avec ma famille, ma source de force et d'inspiration, Merci.

Remerciements

Avant tout, j'exprime ma reconnaissance envers **Allah**, le Tout-Puissant, pour m'avoir accordé le courage et la patience nécessaires afin de mener ce travail à son terme.

Je tiens à exprimer mes plus sincères remerciements à mon encadrant de thèse, Monsieur **MOULAI KHATIR Ahmed Nassim**. Votre encadrement, votre expertise et votre soutien indéfectible tout au long de ce parcours ont été d'une valeur inestimable.

Je souhaite également exprimer ma sincère reconnaissance envers mon co-encadrant de thèse, **HADJ ABDELKADER AMINE**. Vos encouragements ont été essentiels pour mener à bien ce travail. Votre expertise dans le domaine et vos précieux conseils ont été d'une grande valeur.

Je tiens à exprimer mes sincères remerciements aux membres du jury pour l'immense honneur qu'ils m'ont accordé en consacrant leur temps à lire et évaluer ce travail avec attention.

Je tiens à remercier chaleureusement Monsieur **NEMMICHE AHMED**, président du jury de ma thèse, pour l'attention qu'il a bien voulu porter à ce travail en acceptant de le présider et de le discuter. Votre soutien et vos conseils précieux pendant mes études en Master ont été d'une aide inestimable.

Je souhaite pareillement exprimer ma gratitude envers Monsieur **BENADDA BELKACEM**, membre du jury et examinateur de ma thèse, pour l'intérêt qu'il a porté à ce travail en acceptant de l'examiner.

Je tiens à exprimer ma reconnaissance envers ma famille et mes amis pour leur soutien et leurs encouragements tout au long de ce parcours. Leur confiance inébranlable en moi a été une source constante de motivation.

Table des matières

Table des figures

Liste des tableaux

Liste des sigles et acronymes

IA Intelligence Artificielle

DL Deep Learning

ANN Aritificial Neural Networks

CNN Convolutional Neural Networks

RCNN Region-based Convolutional Neural Networks

MNIST Modified National Institute of Standards and Technology database

SIFT Scale-Invariant Feature Transform

HOG Histogram of Oriented Gradients

LBP Local Binary Patterns

DPM Deformable Part-based Models

GPU Graphics Processing Unit

CPU Central Processing Unit

IPS Image Par Second

SBC Single Board Computer

RPi Raspberry Pi

SVM Support Vector Machine

IoU Intersection over Union

mAP mean Average Precision

ReLU Rectified Linear Unit

RoI Region of Interest

ILSVRC ImageNet Large Scale Visual Recognition Challenge

VOC Visual Object Classes challenge

MS-COCO Microsoft Common Objects in Context

OS Operating System

API Application Programming Interface

DSP Digital Signal Processing

ASIC Application Specific Intergrated Circuit

IoT Internet of Things

CHAPITRE 1

Introduction générale

Contexte

L'avènement de la révolution numérique et l'explosion des données disponibles ont considérablement renforcé le rôle de l'Intelligence Artificielle (**IA**) dans le paysage technologique actuel. Parmi les domaines de l'IA qui suscitent le plus d'intérêt, on trouve la vision par ordinateur, qui vise à conférer aux machines la capacité de "voir" et de comprendre les images et les vidéos de manière similaire à celle des humains. Au sein de la vision par ordinateur, l'une des tâches primordiales est la détection d'objets, qui consiste à identifier et localiser des objets spécifiques dans des images ou des vidéos. La détection d'objets présente de nombreuses applications pratiques, telles que la surveillance vidéo, l'analyse de données médicales, la reconnaissance faciale, la lecture des plaques d'immatriculation, la détection des feux de circulation et la conduite autonome, pour n'en citer que quelques-unes. Cependant, cette tâche est complexe et exige l'utilisation de techniques avancées de traitement d'images et d'apprentissage profond (Deep Learning). Ces derniers temps, l'utilisation de TensorFlow Lite sur une Raspberry Pi pour la détection d'objets a gagné en popularité en combinant les fonctionnalités puissantes du framework TensorFlow avec la polyvalence de la Raspberry Pi. Cette combinaison permet de déployer efficacement et à moindre coût des systèmes de détection d'objets dans divers scénarios réels, notamment dans le domaine de la vidéosurveillance. Alors dans ce travail, nous nous intéressons justement à un système permettant détecter des objets, plus précisément des personnes, et ce système est applicable pour un procédé de vidéo surveillance.

Problématique

Les systèmes de vidéosurveillance traditionnels reposent principalement sur la surveillance manuelle, qui présente des inconvénients tels que la fastidiosité, la propension aux erreurs et l'inefficacité. Il est donc impératif de disposer de solutions automatisées et intelligentes capables de détecter avec précision des objets dans des flux vidéo et de les suivre de manière fiable. À cet égard, la détection d'objets en utilisant TensorFlow Lite sur une Raspberry Pi représente une solution prometteuse pour relever ces défis, car la détection d'objets représente un domaine dynamique qui connaît une évolution constante dans nous jours, où de nouvelles méthodes et techniques émergent régulièrement, et l'avènement de l'apprentissage profond a engendré une révolution dans le domaine de la détection d'objets. Les réseaux de neurones convolutifs (**CNN**) ont été largement adoptés pour cette tâche, du fait de leur aptitude à apprendre des caractéristiques discriminantes

directement à partir des images brutes.

Objectifs

L'objectif principal de cette mémoire est de concevoir et de mettre en œuvre une application de détection d'objets en utilisant la bibliothèque TensorFlow Lite. TensorFlow Lite est une version optimisée et légère du framework TensorFlow, une bibliothèque d'apprentissage automatique open-source développée par Google. L'application que nous développerons reposera sur l'utilisation de réseaux de neurones convolutifs, qui représentent actuellement les techniques les plus performantes en matière de détection d'objets. Notre but principal consiste à créer un système de détection d'objets performant et précis en utilisant des modèles pré-entraînés TensorFlow Lite spécialement optimisés pour les appareils à ressources limitées, tels que la Raspberry Pi. Nous prévoyons de déployer ce système de détection d'objets sur une Raspberry Pi équipée d'un module caméra Pi, afin de créer une solution de vidéosurveillance intelligente et automatisée.De plus, la vidéo de surveillance pourra être consultée depuis un écran LCD 5 pouces installé avec la Raspberry Pi et depuis un site web accessible à partir de réseaux local.

Organisation de la mémoire

Cette mémoire est organisée en cinq chapitres :

1. **Chapitre 1 «Introduction Génerale»** :
   Fournit une vue d'ensemble de la détection d'objets à l'aide de TensorFlow Lite sur un Raspberry Pi et traite de l'application de la détection d'objets dans la vidéosurveillance. De plus, énonce le problème et les objectifs de la mémoire. Enfin, il décrit la structure de la mémoire.
2. **Chapitre 2 «l'état de l'art»** :
   Le Chapitre 2 fournit un état de l'art de détection d'objets et de la vidéosurveillance, ses applications et ses défis, et aussi présente une revue de la littérature sur les méthodes de détection d'objets et son historique, en se concentrant sur les réseaux de neurones convolutifs et les approches d'apprentissage profond.
3. **Chapitre 3 «outils matériels et logiciels pour la réalisation du système»** :
   Est consacré à la présentation des composants matériels et logiciels qui nous ont été nécessaires pour mener à bien notre projet. Nous commencerons par présenter la

Raspberry Pi de manière générale en mettant l'accent sur sa fonctionnalité matérielle et logicielle. Nous nous intéresserons après aux autres matérielles nécessaires comme le module caméra Pi et l'écran 5 pouces type LCD TFT. Ensuite en introduit les outils logiciels nécessaires pour le projet.

4. **Chapitre 4 «Réalisation du système»** :

   Dans le Chapitre 4, nous décrirons en détail la conception de l'application de détection d'objets à l'aide de TensorFlow Lite et les autres bibliothèques. Nous commencerons par une conception détaillée, par un schéma bloc et un organigramme. Ensuite, nous décrirons les étapes nécessaires pour préparer notre système. En plus, nous expliquerons le code de notre programme en détails. Enfin, nous présenterons les résultats expérimentaux de l'application de détection d'objets.

5. **Chapitre 5 «Conclusion génerale»** :

   Enfin, le Chapitre 5 marque la conclusion de cette mémoire en résumant les contributions majeures de ce projet de fin d'étude et en proposant des orientations pour les travaux futurs dans le domaine de la détection d'objets avec une Raspberry Pi. Nous mettrons en évidence l'importance de l'application de détection d'objets proposée dans le contexte de la vidéosurveillance. De plus, nous aborderons les limites de notre projet et identifierons les parties de notre projet qui pourraient bénéficier d'améliorations à l'avenir pour augmenter l'efficacité de notre application.

CHAPITRE 2

L'état de l'art

2.1 Introduction

Identifier les objets dans une image ou séquence d'images est essentiel pour interagir avec notre environnement. Bien que cela soit facile pour les humains et les animaux, c'est difficile pour les ordinateurs. Pour comprendre les scènes visuelles, il y a trois sous-problèmes : la classification, la localisation des objets et la détection d'objets. La classification consiste à étiqueter l'objet dominant dans une image. La localisation des objets ajoute la détermination de sa position dans l'image. La détection d'objets implique d'étiqueter tous les objets dans une image, même s'il y en a plusieurs de la même catégorie et de déterminer leur position. C'est un domaine de recherche actif avec de nombreuses variations.

La détection d'objets est de plus en plus utilisée dans de nombreux domaines, notamment la sécurité, la vidéosurveillance, la reconnaissance de formes, la reconnaissance de visage, l'analyse d'images médicales, etc. Ce qui nous intéresse pour notre application, c'est la vidéosurveillance, qui utilise des caméras vidéo pour surveiller des lieux publics ou privés. La vidéosurveillance est utilisée pour diverses applications, particulièrement la sécurité des bâtiments, la surveillance des infrastructures, la gestion du trafic et la prévention de la criminalité.

Dans ce chapitre, nous nous intéresserons justement à la détection d'objets et les systèmes de vidéosurveillance. Par ailleurs, nous commencerons tout d'abord par définir les réseaux de neurones convolutifs (CNN ou ConvNet) car ils sont le pilier des algorithmes de détection moderne. Ensuite, on va approfondir dans la détection d'objet. Nous allons jeter un bref coup d'œil à son histoire avant de voir par la suite, les concepts important de la détection d'objets et nous examinons les différentes approches de la détection d'objets basées sur les réseaux de neurones convolutifs (CNN), et aussi leurs métriques. Après, on va passer une revue dans la vidéosurveillance pour améliorer la sécurité et la gestion des infrastructures. Etc. Nous évaluons par ailleurs les avantages de la détection d'objets, en soulignant les difficultés et les contraintes associées à cette technologie, nous mettrons en évidence ses défis et ses limites.

2.2 Les réseaux de neurones convolutifs (CNNs)

Les réseaux neuronaux convolutifs (CNNs ou ConvNets) sont considérés comme la forme la plus remarquable d'intelligence artificielle basée sur les réseaux neuronaux (ANNs). Ils jouent un rôle clé dans le domaine de l'apprentissage profond (DL) et sont largement

utilisés pour résoudre des tâches complexes de reconnaissance des formes basées sur des images[48]. Inspirés par les champs récepteurs présents dans le cortex visuel des animaux, les CNNs détectent des stimuli spécifiques tels que les contours. En traitement d'images, on peut obtenir des résultats similaires en utilisant des techniques de filtrage par convolution[19]. Un CNN caractéristique se compose de trois types de couches : les couches de convolution (convolutional layers), les couches de pooling ou mise en commun (pooling layers) et les couches entièrement connectées (fully connected layers). Ces couches sont accumulées pour former un réseau neuronal capable de reconnaître et de détecter des formes. Par exemple, une architecture de CNN simplifiée peut être utilisée pour classer des images manuscrites MNIST[10], comme illustré dans la figure 2.2.1.

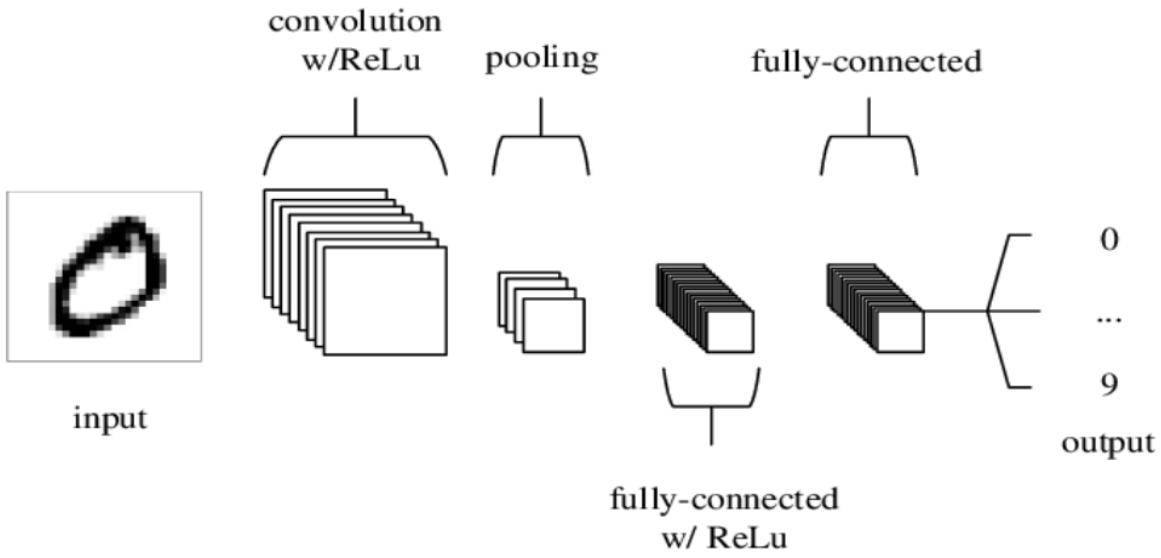


Figure 2.2.1 – Une architecture CNN, composée de cinq couches[48]

Dans le passé, les CNN ont été délaissés en raison du manque de puissance de calcul disponible, et ont été remplacés par des machines à vecteurs de support (SVM). Cependant, le développement de GPU puissants au cours de la dernière décennie a ravivé l'espoir quant aux capacités de perception des CNN. Aujourd'hui, les CNN sont largement utilisés comme approche par défaut pour résoudre de nombreux problèmes de traitement d'images.

2.2.1 Fonctionnement d'un CNN

Un CNN se compose de plusieurs couches d'opérations, chacune avec sa propre fonctionnalité. CNN prend une image d'entrée et la transmet à la première couche. Les informations sur les entités sont propagées à travers les couches masquées. Pour chaque couche, une fonction d'activation (voir Section 2.2.1.4) effectue une activation élément par élément sur la sortie produite par la couche précédente. La sortie de la dernière

couche est ensuite comparée à la sortie cible. Ce processus s'appelle la passe avant (forward pass en anglais).

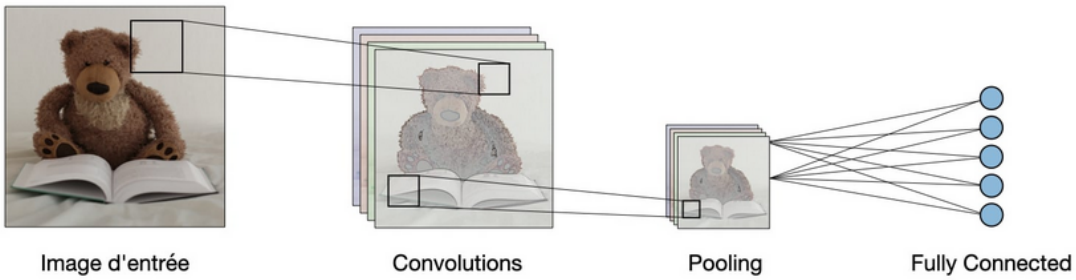


Figure 2.2.2 – Les couches d'un CNN traditionnel[6].

2.2.1.1 Couche convolutionnelle (CONV)

Dans les réseaux neuronaux convolutifs (CNN), une couche de convolution joue un rôle essentiel. Les paramètres de cette couche se concentrent sur des noyaux (ou filtres) de petite taille $n \times n$. La couche de convolution applique chaque filtre ou noyau à travers les dimensions spatiales de l'entrée, générant ainsi une carte d'entités 2D (feature map) ou une carte d'activation (activation map) après le calcul de la fonction d'activation. En général, l'opération de convolution est définie comme une intégrale qui mesure le degré de superposition entre une fonction g lorsqu'elle est décalée sur une autre fonction f. Cette mesure est exprimée par la formule suivante :

$$s(t) = (f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \tag{2.2.1}$$

Où f et g sont les fonctions arbitraires continues dans le domaine t, la fonction $g(t)$ peut être considérée comme un filtre appliqué au signal $f(t)$. Dans le cas des images, le signal d'entrée est bidimensionnel et échantillonné. Si I représente l'image d'entrée et K représente un filtre, la convolution est alors définie par l'équation 2.2.2.

$$S(i,j) = (I * K)(i,j) = \sum_{m}^{\infty}\sum_{n}^{\infty} I(m,n)K(i-m, j-n) \tag{2.2.2}$$

Les définitions données dans les équations 2.2.2 et 2.2.1 sont toutes deux des réalisations de ce que l'on appelle mathématiquement la convolution, mais normalement, lorsque la convolution est décrite dans le cadre de l'apprentissage profond(DL), la fonction dont il est question est la fonction de corrélation croisée(cross-correlation). La corrélation croisée possède des propriétés similaires à celles de la convolution, à la différence près que la fonction de corrélation croisée n'est pas commutative. La fonction de corrélation croisée

est définie comme suit :

$$S(i,j) = (I * K)(i,j) = \sum_{m}^{\infty} \sum_{n}^{\infty} I(i+m, j+n) K(m,n) \tag{2.2.3}$$

qui est presque la même fonction que celle décrite dans l'équation 2.2.2 mais sans refléter le noyau. Tout comme dans le cas unidimensionnel, la fonction définie dans l'équation 2.2.4 peut être décrite comme la correspondance de l'image d'entrée à l'image de sortie où le noyau a été appliqué à plusieurs endroits de l'image. Un exemple simple de le mappage effectuée par la fonction de corrélation croisée est donné dans la figure 2.2.3.

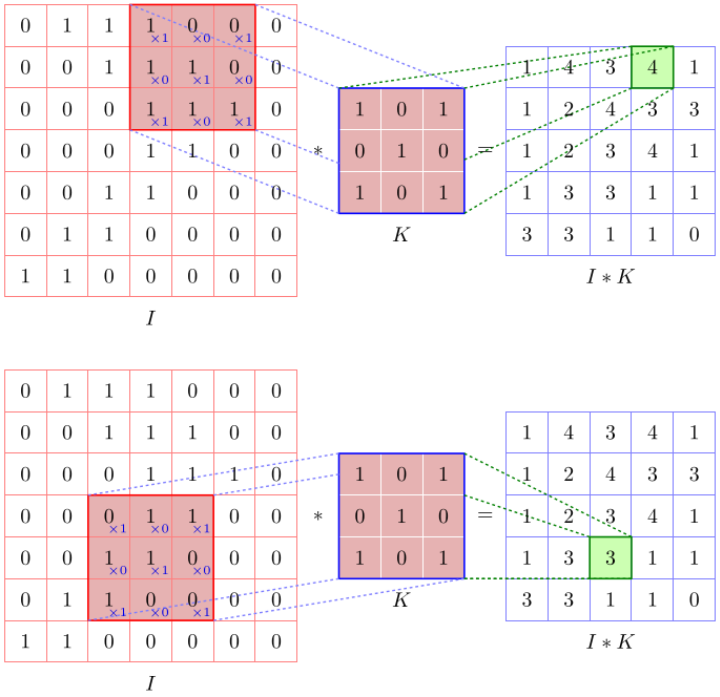


Figure 2.2.3 – Illustration simple de la manière dont le noyau K glisse sur I afin de générer la sortie.

La taille de l'image résultante dépend non seulement de la largeur et de la hauteur du noyau, mais aussi de deux paramètres appelés stride et padding. Stride est défini comme étant le nombre de pixels que le noyau se déplace entre la production de chaque élément en sortie. La fonction comprenant les paramètres stride est définie comme suit :

$$S(i,j) = (I * K)(i,j) = \sum_{m}^{\infty} \sum_{n}^{\infty} I(i \times s + m, j \times s + n) K(m,n) \tag{2.2.4}$$

Pour maintenir la taille de l'image, on utilise des coussinets(padding) pour gérer les cas où le noyau est proche des limites de I. En remarque que le padding n'est pas explicitement affiché dans l'équation, mais il est utilisé pour ajouter des lignes et des colonnes supplémentaires aux données d'entrée afin qu'elles puissent être traitées par le

filtre. Il existe quelques options différentes pour réaliser le coussin d'une image, l'une d'entre elles consiste à ajouter des zéros aux limites de I afin d'empêcher la sortie de diminuer en largeur et en hauteur. Alors la couche convolutionnelle d'un réseau de neurones est une combinaison de plusieurs filtres convolutionnels, dont les valeurs matricielles sont traitées comme des paramètres neuronaux. La répétition successive des couches convolutives avec certains autres types de couches comme la couche de mise en commun (pooling) se traduit par un réseau neuronal convolutif (CNN)[48]

2.2.1.2 Couche de Pooling (POOL)

Les couches de mise en commun (Pooling) sont un autre type de couche couramment utilisé dans les réseaux neuronaux convolutifs. Elles réduisent simplement l'échantillonnage de chacune des cartes de caractéristiques créées par une opération de convolution ; pour la taille de pooling la plus couramment utilisée de 2, cela implique de mapper chaque section 2×2 de chaque carte de caractéristiques soit à la valeur maximale de cette section, dans le cas du max-pooling, soit à la valeur moyenne de cette section, dans le cas du average-pooling. Dans le cas d'une image $n \times n$, cela signifie que l'ensemble de l'image est mis en correspondance avec une image de taille $\frac{n}{2} \times \frac{n}{2}$ [71]. La figure 2.2.4 illustre ceci.

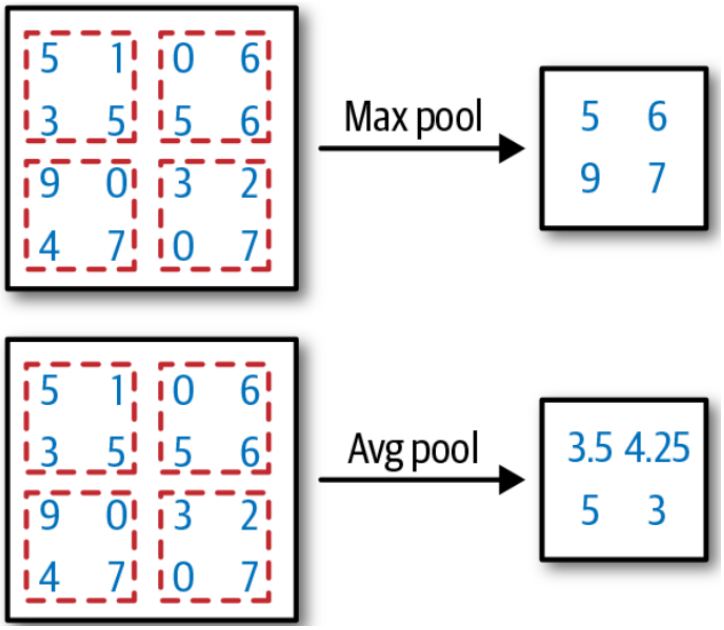


Figure 2.2.4 – Une illustration de max et de moyenne mise en commun avec une entrée de 4×4 ; chaque patch de 2×2 est mappé soit sur les valeurs moyennes soit sur les valeurs maximales pour ce patch[71]

Le principal avantage du pooling est computationnel : en réduisant l'image pour qu'elle contienne un quart des pixels de la couche précédente, le pooling diminue à la fois le nombre de poids et le nombre de calculs nécessaires pour entraîner le réseau d'un facteur

de 4 ; cela peut être encore amplifié si plusieurs couches de pooling sont utilisées dans le réseau, comme cela a été le cas dans de nombreuses architectures au début des CNN. L'inconvénient du pooling, bien sûr, est que seulement un quart des informations peuvent être extraites de l'image sous-échantillonnée. Cependant, le fait que les architectures ont montré une très forte performance sur les benchmarks en reconnaissance d'images malgré l'utilisation du pooling suggérait que, même si le pooling faisait perdre aux réseaux des informations sur les images en diminuant la résolution des images, les compromis en termes de vitesse de calcul accrue en valaient la peine. Néanmoins, le pooling était considéré par beaucoup comme une astuce qui fonctionnait juste mais qui devrait probablement être abandonnée[71].

2.2.1.3 Couche Fully Connected (FC)

Les couches entièrement connectées (FC) sont appliquées aux entrées préalablement aplaties, où chaque entrée est connectée à tous les neurones de la couche. Les couches entièrement connectées sont habituellement localisé à la fin des architectures CNN et peuvent être utilisées pour maximiser des objectifs tels que les scores de catégorie.[6].

2.2.1.4 Les Fonctions d'activation

1. **Unité linéaire rectifiée** :

   La couche d'unité linéaire rectifiée (ReLU en anglais), est une fonction d'activation appliquée à tous les éléments du volume dans le réseau. Son objectif est d'introduire des non-linéarités dans le réseau[71]. Les différentes variantes de ReLU sont résumées dans le tableau ci-dessous :

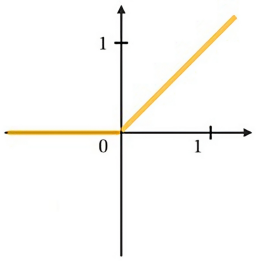
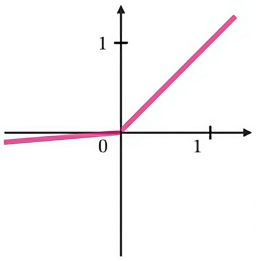
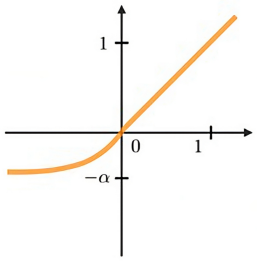
ReLU	Leaky ReLU	ELU
$g(z) = \max(0, z)$	$g(z) = \max(\epsilon z, z)$ with $\epsilon \ll 1$	$g(z) = \max(\alpha(e^z - 1), z)$ with $\alpha \ll 1$
		
• Complexités non-linéaires interprétables d'un point de vue biologique	• Répond au problème de *dying ReLU*	• Dérivable partout

Table 2.2.1 – Les couche d'unité linéaire rectifiée commun[6]

2. **Softmax** :

   L'étape softmax peut être considérée comme une extension de la fonction logistique qui prend en entrée un vecteur de scores $x \in R^n$ et renvoie un vecteur de probabilités $p \in R^n$ grâce à une fonction softmax située à la fin de l'architecture [71]. Sa définition est la suivante :

$$p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} \quad \text{où} \quad p_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

Figure 2.2.5 – Définition de la fonction Softmax

2.2.2 Différents extracteurs de fonctionnalités CNN

Les extracteurs de fonctionnalités ou extracteurs de caractéristiques CNN sont principalement utilisés dans le domaine du traitement d'images et de la vision par ordinateur. Ils sont conçus pour prendre des images en entrée et effectuer diverses tâches telles que la classification d'objets, la détection d'objets et la segmentation d'images. Les extracteurs de caractéristiques CNN présentent un avantage par rapport aux réseaux neuronaux traditionnels, car leur entrée est constituée d'images. Les couches des extracteurs de caractéristiques CNN sont organisées en trois dimensions : largeur, hauteur et profondeur. La profondeur fait référence au nombre de cartes d'entités présentes dans chaque couche. Un extracteur de caractéristiques standard se compose généralement de quatre types de couches : couche de convolution, activation ReLU, couche de regroupement (pooling) et couche entièrement connectée (FC). Les extracteurs de caractéristiques sont également utilisés comme sous-réseaux (backbone) pour les algorithmes de détection d'objets, tels que le détecteur d'objets SSD[40] ou YOLO[57]. Au fil du temps, les architectures CNN ont évolué pour améliorer les performances en matière de classification d'objets et d'autres tâches liées à la vision par ordinateur. Parmi les extracteurs de caractéristiques CNN populaires, on trouve LeNet-5, AlexNet, VGG-16 et ResNet. Voici un tableau donnant un aperçu de quelques architectures CNN :

Architecture	Année de publication
LeNet-5	1998
AlexNet	2012
ZFNet	2013
VGG-16	2014
GoogLeNet	2014
Inception-v3	2015
ResNet	2015
DenseNet	2016
MobileNet	2017
NASNet	2017
MobileNet-v2	2018
CSPDarknet-53	2020

Table 2.2.2 – Quelque architecture CNN

2.3 Détection d'objet

La détection d'objets est une tâche importante de vision par ordinateur qui consiste à détecter des instances d'objets visuels d'une certaine classe (comme les humains, les animaux ou les voitures) dans des images numériques ou dans une vidéo. L'objectif de la détection d'objets est de développer des modèles et des techniques computationnels qui fournissent l'une des connaissances les plus fondamentales nécessaires aux applications de vision par ordinateur : quels objets sont où ? Les deux métriques les plus significatives pour la détection d'objets sont la précision (y compris la précision de classification et la précision de localisation) et la vitesse. Ces dernières années, le développement rapide des techniques d'apprentissage profond -ou en profondeur-[36] (DL) a grandement favorisé les progrès de la détection d'objets, conduisant à des avancées remarquables et la propulsant

à un point chaud de recherche avec une attention sans précédent. La détection d'objets est maintenant largement utilisée dans de nombreuses applications du monde réel, telles que la conduite autonome, la vision robotique, la vidéo-surveillance, etc.

2.3.1 Historique de la détection d'objets

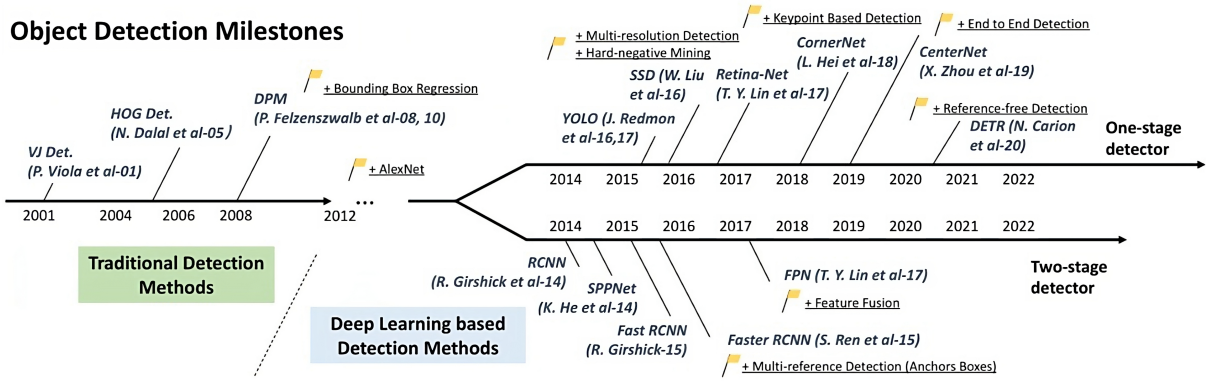


Figure 2.3.1 – La feuille de route de la détection d'objet[79]

Au cours des 20 dernières années, la détection d'objets a progressé à travers deux périodes principales : la période traditionnelle avant 2014 et la période basée sur l'apprentissage profond après 2014 (voir Figure 2.3.1).

Selon que l'apprentissage profond est utilisé ou non, les méthodes de détection d'objets peuvent être divisées en deux catégories : les méthodes artisanales basées sur les fonctionnalités[11, 31, 70] et les méthodes basées sur l'apprentissage profond[23, 28, 38, 40]. Au cours de la première décennie du 21e siècle, les méthodes de production traditionnelles basées sur les fonctionnalités sont devenues courantes. Au cours de cette période, de nombreux descripteurs de caractéristiques d'image célèbres et réussis ont été proposés (par exemple SIFT[41], HOG[8], Haar[68], LBP[49] et DPM[17]). Basées sur ces descripteurs de fonctionnalités et des classificateurs classiques tels que SVM et AdaBoost, ces méthodes connaissé un grand succès. Cependant, à partir de 2010, les performances de détection d'objets se sont stabilisées. Bien que de nombreuses méthodes soient toujours fournies, l'amélioration des performances est relativement limitée. Dans le même temps, l'apprentissage profond a commencé à montrer des performances supérieures dans certains domaines de la vision par ordinateur tels que la classification des images[34, 63, 65]. En 2012, avec des données d'image volumineuses (ImageNet[9]), le réseau CNN profond (appelé AlexNet[34]) réalise les meilleures performances de détection sur ImageNet ILSVRC2012, qui surpasse la deuxième meilleure méthode de 10,9 % sur le chalenge ILSVRC-2012 . Avec le grand succès de l'apprentissage en profondeur dans la classification

d'images[34, 63, 65], les chercheurs ont commencé à explorer comment améliorer les performances de détection d'objets grâce à l'apprentissage profond. Ces dernières années, la détection d'objets basée sur le deep learning a fait de grands progrès[23, 28, 59]. La précision moyenne de détection d'objets (mAP) sur PASCAL VOC2007[14] augmente significativement de 58 % (basé sur RCNN avec AlexNet[23]) à 86 % (basé sur Faster RCNN avec ResNet[31]). Actuellement, les méthodes de pointe de détection d'objets profonds sont basées sur des réseaux de neurones à convolution profonde (CNN)[23, 27, 38, 59].

2.3.2 Les concepts importants de détection d'objets

Dans cette section, quelques concepts importants utilisés dans la détection d'objets sont brièvement introduits.

2.3.2.1 Région d'intérêt (RoI)

Une région d'intérêt (ROI) est une zone spécifique à l'intérieur d'une image ou d'une trame vidéo qui est d'un intérêt particulier pour l'analyse ou le traitement. Dans le contexte de la détection d'objets, une ROI est souvent définie à l'aide d'une boîte englobante, qui décrit la région contenant l'objet(s) d'intérêt. Une région d'intérêt, est une région rectangulaire d'entrée susceptible de contenir des objets. Ces propositions peuvent être générées par certains algorithmes externes, tels que la recherche sélective[67], la détection de boîte de contour (Edge Box Detection)[78], ou via un Réseau de proposition de région(RPN). Une boîte englobante(Bounding Box) est représentée par un vecteur 4×1 contenant son emplacement central, sa largeur et sa hauteur $(\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{h})$ ou par les coins supérieur gauche et inférieur droit du rectangle $(\mathbf{x_{min}}, \mathbf{y_{min}}, \mathbf{x_{max}}, \mathbf{y_{max}})$. Chaque boîte englobante de l'image est accompagnée d'un score d'objectivité ou de confiance pour la probabilité que la boîte contienne l'objet.

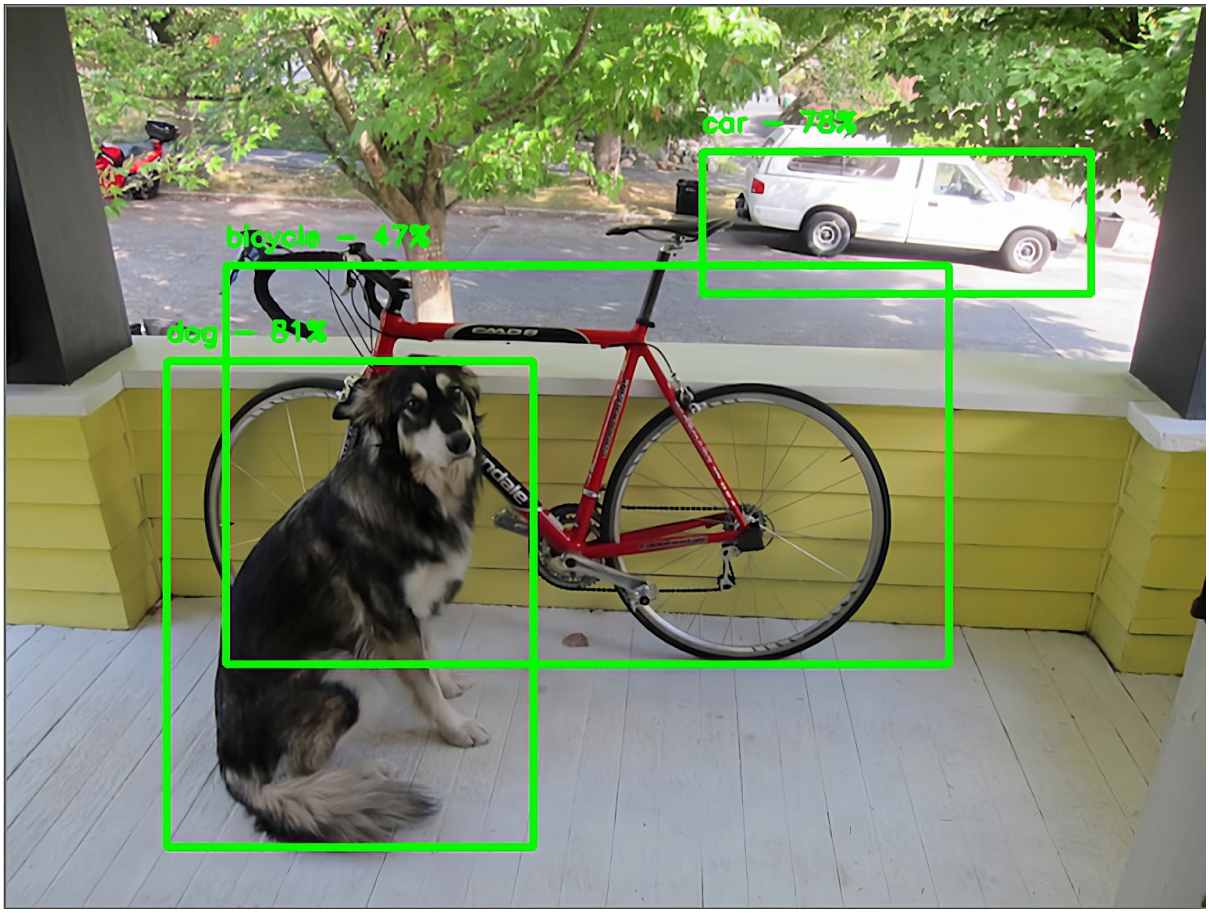


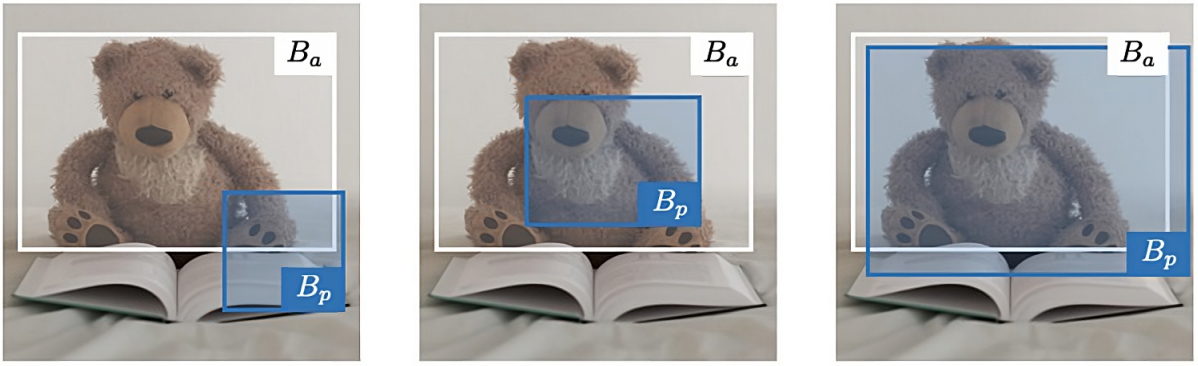
Figure 2.3.2 – RoI dans une image[57]

2.3.2.2 Intersection sur Union (IoU)

L'intersection sur l'Union (IoU), également connue sous le nom Intersection over Union en anglais, est une métrique basée sur l'indice de Jaccard[25] qui quantifie la qualité de la localisation d'une boîte englobante prédite $\mathbf{B_p}$ par rapport à la boîte englobante de vérité au sol -référence- (ground truth) $\mathbf{B_a}$. En d'autres termes, IoU est une mesure qui quantifie dans quelle mesure deux boîtes englobantes se chevauchent, il est défini comme suit :

$$IoU(B_p, B_a) = \frac{\text{Zone de chevauchement}}{\text{Zone de l'Union}} = \frac{B_p \cap B_a}{B_p \cup B_a} \tag{2.3.1}$$

Par convention, on a toujours $IoU \cup [0, 1]$, et la prédiction d'une boîte englobante B_p est considérée comme satisfaisante si on a $IoU(B_p, B_a) \geq 0.5$.



$\text{IoU}(B_p, B_a) = 0.1$ $\text{IoU}(B_p, B_a) = 0.5$ $\text{IoU}(B_p, B_a) = 0.9$

FIGURE 2.3.3 – IoU dans une image[6]

2.3.2.3 raffinement de la boîte englobante

La régression ou raffinement des boîtes englobantes (bounding box regression) est une technique populaire utilisée pour améliorer ou prédire les boîtes de localisation dans les méthodes récentes de détection d'objets. La plupart des détecteurs d'objets modernes utilisent des régresseurs de boîte englobante qui sont formés pour prédire le décalage $\mathbf{\Delta}(\mathbf{x}, \mathbf{y}, \mathbf{w}, \mathbf{h})$ entre la boîte de région d'entrée et la boîte de vérité au sol (ground truth). S'il existe un régresseur pour chaque classe d'objets, on parle de régression spécifique à la classe, et si toutes les classes ont un régresseur, on parle de régression indépendante de la classe. Les régresseurs de boîte englobante sont souvent accompagnés de classificateurs de boîte englobante, également appelés scores de confiance ou seuils de confiance (confidence threshold), pour estimer la confiance en la présence d'objets dans la boîte. Les classificateurs peuvent également être spécifiques à une classe ou indépendants d'une classe.[29]

2.3.3 Ensembles de données et métriques de détection d'objets

2.3.3.1 Ensembles de données de détection d'objets

Un jeux de données (datasets en anglais) pour la détection d'objets est une collection d'images et de leurs informations de localisation et d'étiquetage correspondantes utilisées pour entraîner un modèle de détection d'objets. Il peut être utilisé pour développer et évaluer des détecteurs d'objets dans des images pour cela la création de jeux de données plus grands avec moins de biais est essentielle pour développer des algorithmes de détection avancés. Un certain nombre de jeux de données de détection bien connus ont été publiés au cours des 10 dernières années, notamment les jeux de données des défis PASCAL VOC[14, 15] (par exemple, VOC2007, VOC2012), le défi de reconnaissance visuelle à grande échelle ImageNet (par exemple, ILSVRC2014)[61], le défi de détection MS-COCO[39], l'ensemble

de données Open Images[4, 35], Objects365[62], etc. Les statistiques de ces ensembles de données sont données dans le tableau 2.3.1. La figure 2.3.4 montre quelques exemples d'images de ces ensembles de données. La figure 2.3.5 montre les améliorations de la précision de détection sur les ensembles de données VOC07, VOC12 et MS-COCO de 2008 à 2021.

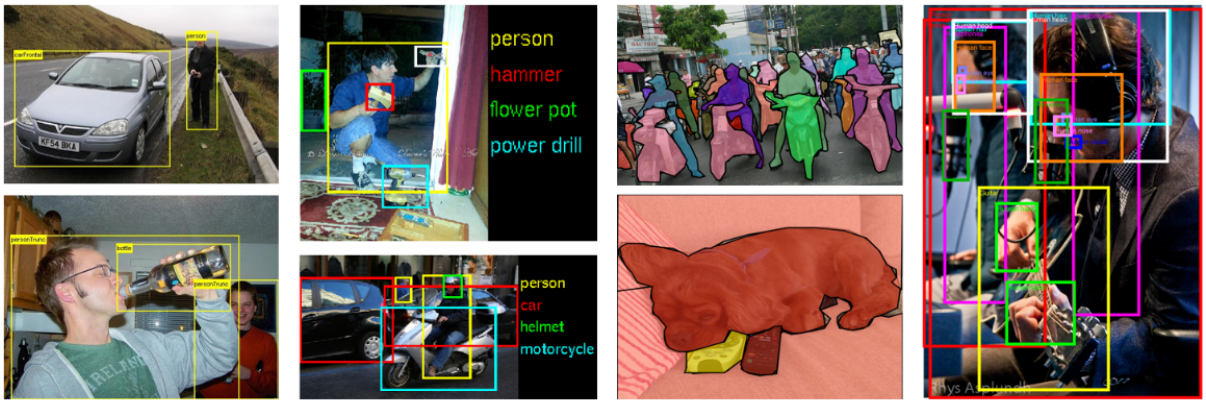


Figure 2.3.4 – Quelques exemples d'images et d'annotations dans (a) PASCAL-VOC07, (b) ILSVRC, (c) MS-COCO et (d) Open Images[79]

Dataset	train		validation		trainval		test	
	images	objects	images	objects	images	objects	images	objects
VOC-2007	2,501	6,301	2,510	6,307	5,011	12,608	4,952	14,976
VOC-2012	5,717	13,609	5,823	13,841	11,540	27,450	10,991	-
ILSVRC-2014	456,567	478,807	20,121	55,502	476,688	534,309	40,152	-
ILSVRC-2017	456,567	478,807	20,121	55,502	476,688	534,309	65,500	-
MS-COCO-2015	82,783	604,907	40,504	291,875	123,287	896,782	81,434	-
MS-COCO-2017	118,287	860,001	5,000	36,781	123,287	896,782	40,670	-
Objects365-2019	600,000	9,623,000	38,000	479,000	638,000	10,102,000	100,000	1,700,00
OID-2020	1,743,042	14,610,229	41,620	303,980	1,784,662	14,914,209	125,436	937,327

Table 2.3.1 – Quelques ensembles de données de détection d'objets bien connus et leurs statistiques[79]

2.3.3.2 Métrique de la détection d'objets

Comment pouvons-nous évaluer l'exactitude d'un détecteur ? Cette question peut avoir différentes réponses à différents moments. Dans les premières recherches sur la détection, il n'y avait pas de métriques d'évaluation largement acceptées sur l'exactitude de la détection. Par exemple, dans les premières recherches sur la détection des piétons[8], le "taux de manquement par rapport aux faux positifs par fenêtre (FPPW)" était couramment utilisé comme métrique. Cependant, la mesure par fenêtre peut être faussée et ne permet pas de prédire les performances de l'image complète[12]. En 2009, le benchmark de détection des piétons Caltech a été introduit[11, 12] et depuis lors, la métrique d'évaluation est passée de FPPW à faux positifs par image (FPPI). Ces dernières années, l'évaluation la plus

fréquemment utilisée pour la détection est "Précision moyenne (AP)", qui a été introduite à l'origine dans VOC2007. L'AP est définie comme la précision de détection moyenne sous différents rappels et est généralement évaluée de manière spécifique à une catégorie. La précision moyenne (mAP) moyennée sur toutes les catégories est majoritairement utilisée comme métrique finale de performance. Pour mesurer l'exactitude de la localisation de l'objet, l'intersection sur union (IoU) entre la boîte prédite et la vérité terrain est utilisée pour vérifier si elle est supérieure à un seuil prédéfini, disons 0,5. Si oui, l'objet sera identifié comme "détecté", sinon "manqué". Le 0,5-IoU mAP est alors devenu la métrique de facto pour la détection d'objets.

Après 2014, en raison de l'introduction des ensembles de données MS-COCO, les chercheurs ont commencé à accorder plus d'attention à l'exactitude de la localisation des objets. Au lieu d'utiliser un seuil IoU fixe, MS-COCO AP est moyenné sur plusieurs seuils IoU entre 0,5 et 0,95, ce qui encourage une localisation d'objet plus précise et peut être d'une grande importance pour certaines applications réelles.

2.3.4 Les algorithmes de détection d'objets moderne

Dans cette section, nous examinerons la détection d'objets sous plusieurs angles, nous allons décrire les détecteurs clés de la période moderne, en utilisant leur temps d'émergence et leurs performances pour mettre en évidence la technologie motrice derrière eux (voir Figure 2.3.5)

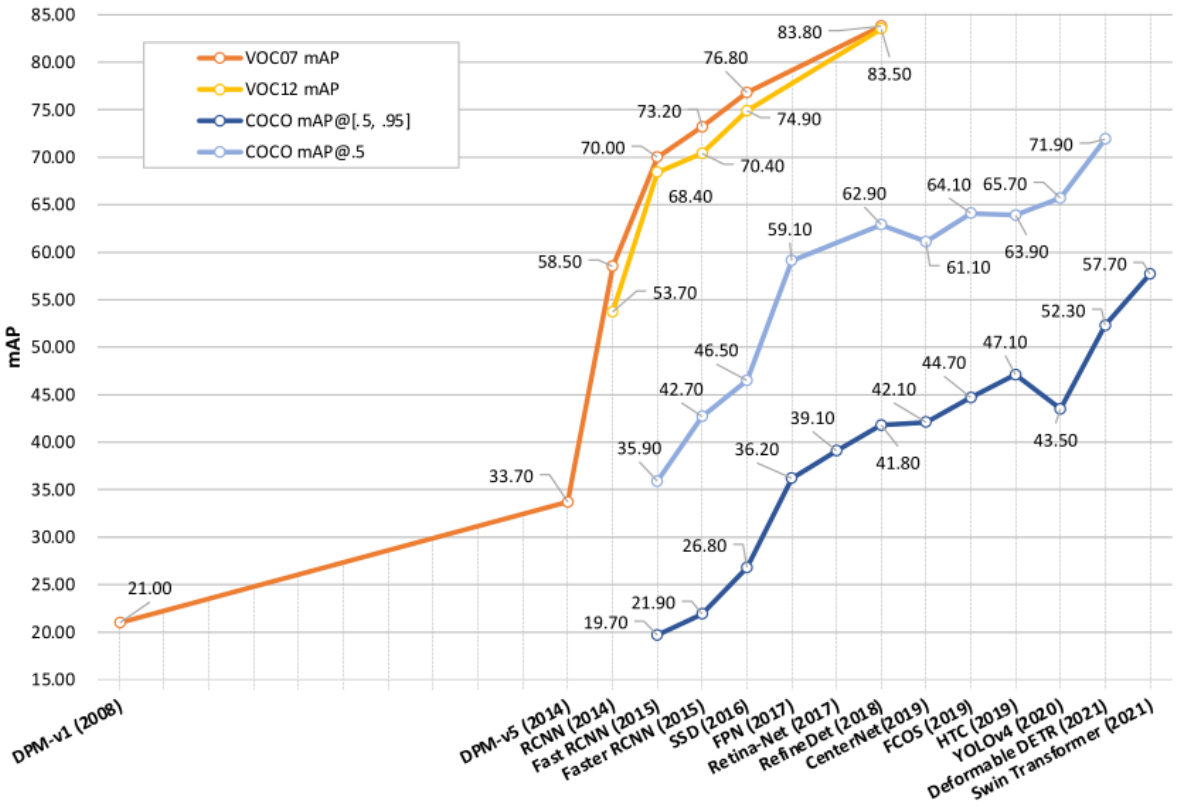


Figure 2.3.5 – Amélioration de la précision de la détection d'objets sur Ensembles de données VOC07, VOC12 et MS-COCO[79]

2.3.4.1 détecteurs à deux étages basés sur CNN

Alors que les performances des fonctionnalités artisanales devenaient saturées, la recherche sur la détection d'objets a atteint un plateau après 2010. En 2012, le monde a vu la renaissance des réseaux de neurones convolutifs[34]. Comme un réseau de convolution profonde est capable d'apprendre des représentations de fonctionnalités robustes et de haut niveau d'une image, une question naturelle se pose : pouvons-nous l'introduire à la détection d'objets ? R. Girshick et al. ont pris la tête pour briser les impasses en 2014 en proposant les régions avec des fonctionnalités CNN (RCNN)[22, 23]. Depuis lors, la détection d'objets a commencé à évoluer à une vitesse sans précédent. Il existe deux groupes de détecteurs à l'ère de l'apprentissage en profondeur : les "détecteurs à deux étages" et les "détecteurs à une étape", L'approche en deux étapes traite la détection d'objets comme un processus en plusieurs étapes. Étant donné une image d'entrée, certaines propositions d'objets possibles sont d'abord extraites. Ensuite, ces propositions sont classées dans des catégories d'objets spécifiques par le classificateur entraîné. Les avantages de ces méthodes peuvent être résumés comme suit : (1) Réduire le nombre de propositions à mettre dans le classificateur suivant. Par conséquent, la vitesse de détection peut être accélérée. (2) La phase de génération de proposition peut être considérée comme une technique d'initiation.

Sur la base de propositions d'objets possibles, le classificateur peut se concentrer sur des tâches de classification avec peu d'influence de fond pendant la phase d'apprentissage. Par conséquent, cela peut améliorer la précision de la détection. Parmi ces méthodes en deux étapes, la famille RCNN comprend RCNN[23], SPPnet[28], Fast RCNN[21] et Faster RCNN[59], sont très représentatives.

1. **Méthode RCNN** :

   R-CNN signifie Réseau de neurones convolutifs basé sur la région. Le concept clé derrière la série R-CNN est les propositions régionales. Les propositions de région sont utilisées pour localiser des objets dans une image, elle commence par l'extraction d'un ensemble de propositions d'objets (Boîtes candidates d'objet ou en anglais object candidate boxes) par recherche sélective[67]. Ensuite, chaque proposition est redimensionnée à une image de taille fixe et introduite dans un modèle CNN pré-entraîné sur ImageNet (disons, AlexNet[34]) Pour extraire des caractéristiques. Enfin, des classificateurs à machines à vecteurs de support (SVM) linéaires sont utilisés pour prédire la présence d'un objet dans chaque région et pour reconnaître les catégories d'objets. RCNN donne un important coup de pouce aux performances sur VOC07, avec une grande amélioration de la précision moyenne (mAP) de 33,7% (DPM-v5[16]) à 58,5%.

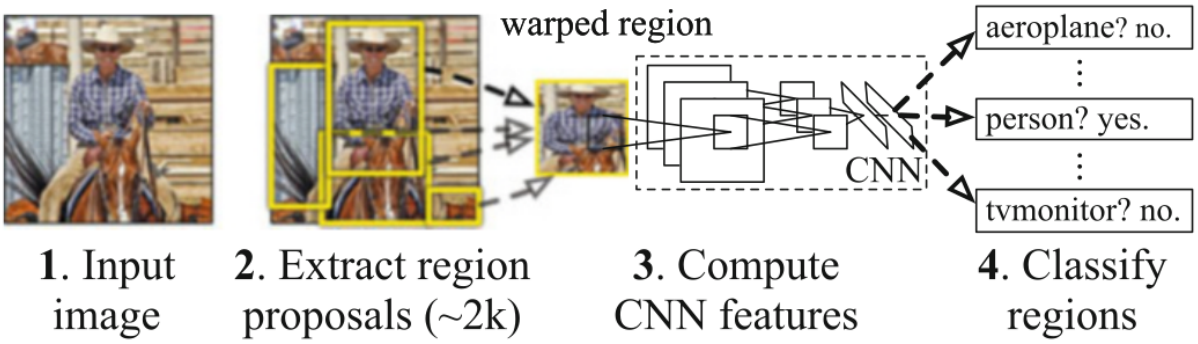


Figure 2.3.6 – L'architecture du RCNN. Il se compose de trois étapes : génération de la proposition, calcul des caractéristiques CNN et classification de la proposition[23].

   Bien que RCNN ait fait de grands progrès, ses inconvénients sont évidents : les calculs de caractéristiques redondants sur un grand nombre de propositions superposées (plus de 2000 RoI à partir d'une image) conduisent à une vitesse de détection extrêmement lente (14s par image avec GPU). Plus tard la même année, SPPNet[28] a été proposé et a résolu ce problème.

2. **Fast RCNN** :

   En 2015, R. Girshick a proposé le détecteur Fast RCNN[21], qui est une amélioration

supplémentaire de R-CNN et SPPNet[23, 28]. Fast RCNN nous permet de former simultanément un détecteur et un régresseur de boîte englobante sous les mêmes configurations de réseau(framework). Sur l'ensemble de données VOC07, Fast RCNN a augmenté le mAP de 58,5% (RCNN) à 70,0% tout en ayant une vitesse de détection plus de 200 fois plus rapide que R-CNN. La figure 2.3.7 illustre l'architecture du Fast RCNN . En règle générale, Fast RCNN calcule d'abord toutes les couches convolutives de l'image entière. Pour chaque proposition, Fast RCNN utilise une couche de mise en commun(Pooling) des ROI pour extraire les cartes de fonctionnalités(feature map) de taille fixe des cartes de caractéristiques de la dernière couche convolutive, puis transmet les cartes de caractéristiques de taille fixe à deux couches entièrement connectées(FC layers),et enfin génère deux branches apparentées avec un fonctionnement entièrement connecté pour la classification d'objets et la régression de boîtes. Pour la classification d'objets, il y a c+1 sorties par softmax, où c représente le nombre de classes d'objets. Pour la régression des boîtes, il a 4c sorties, où chaque quatre sorties correspondent au décalage des boîtes par classe. La couche de mise en commun des ROI déforme les cartes de caractéristiques des propositions d'objets dans des cases spatiales de taille fixe (par exemple, 7×7) et utilise l'opération de mise en commun maximale(max-pooling) pour calculer les réponses des caractéristiques dans chaque case.

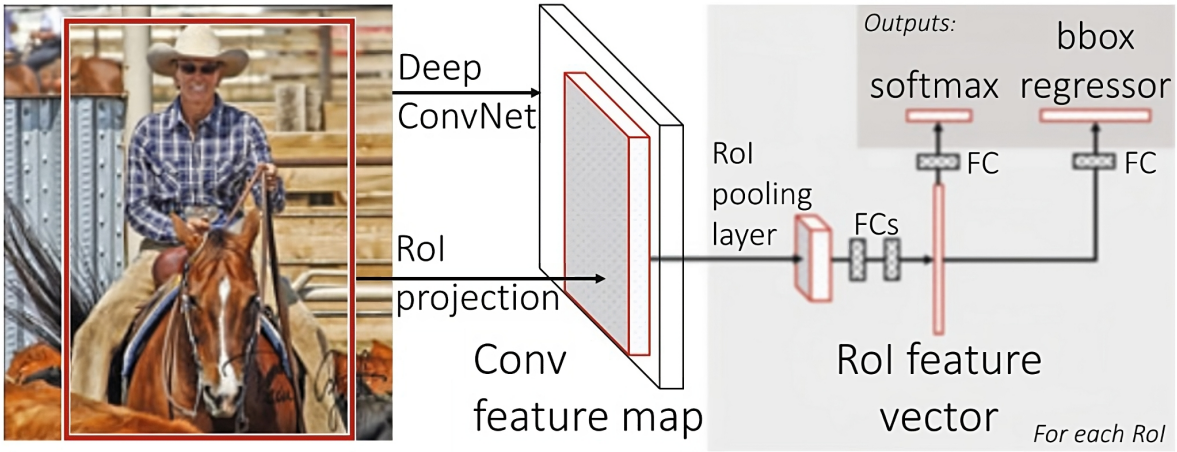


Figure 2.3.7 – L'architecture de Fast RCNN. Comparé à RCNN et SPPnet, il rejoint la classification et la régression dans un cadre unifié[21].

Le Fast R-CNN est plus efficace et précis que le R-CNN en termes de calcul, car les propositions de région sont générées à partir de la carte des caractéristiques (Feature map) au lieu de l'image d'origine. Cependant, les propositions de région étaient toujours détectées avec la méthode de recherche sélective lente. Bien que

Fast-RCNN intègre avec succès les avantages de R-CNN et SPPNet, sa vitesse de détection est encore limitée. Faster R-CNN[58] était la prochaine étape.

3. **Faster RCNN** :

   Pour la génération de propositions, RCNN[23], SPPnet[28] et Fast RCNN[21] sont tous basés sur la recherche sélective. La recherche sélective[67] utilise les fonctionnalités artisanales et adopte les stratégies de regroupement hiérarchique pour capturer toutes les propositions d'objets possibles. Généralement, il s'exécute à 2 s par image sur le CPU commun. Le réseau de détection de Fast RCNN peut fonctionner à environ 100 ms par image sur le GPU. Ainsi, la génération de propositions de Fast RCNN prend plus de temps que le réseau de détection de Fast RCNN. Bien que la recherche sélective puisse également être réimplémentée sur le GPU, l'extraction de propositions est toujours isolée du réseau de détection de Fast RCNN. Ainsi, l'extraction de proposition de région devient le bottleneck de Fast RCNN sur la détection d'objet. Alors en 2015, S. Ren et al. ont proposé le détecteur Faster RCNN[58, 59] peu de temps après le Fast RCNN. Faster RCNN est le premier détecteur d'apprentissage en profondeur quasi-temps réel (COCO mAP@.5=42.7%, VOC07 mAP=73.2%, 17 IPS avec ZF-Net[76]). La principale contribution de Faster-RCNN est l'introduction du réseau de proposition de région (RPN) qui permet des propositions de région presque gratuites. De R-CNN à Faster RCNN, la plupart des blocs individuels d'un système de détection d'objets ont été progressivement intégrés dans un cadre d'apprentissage unifié de bout en bout.

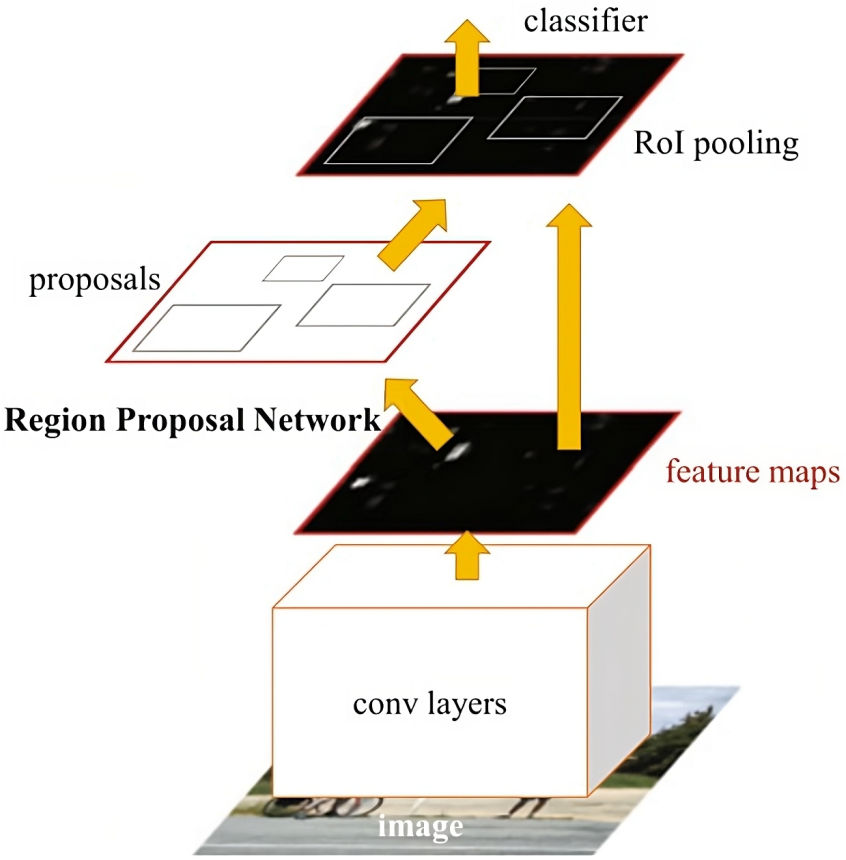


Figure 2.3.8 – L'architecture de Faster RCNN. La génération de propositions (RPN) et la classification des propositions (Fast RCNN) sont intégrées dans un cadre unifié[31].

La figure 2.3.8 montre l'architecture du réseau Faster RCNN. Il se compose de deux modules. Le premier, appelé réseau de proposition de région (RPN), est utilisé pour extraire les propositions d'objets candidats. L'autre module est le Fast RCNN, qui est utilisé pour classer ces propositions dans des catégories spécifiques et prédire avec plus de précision les emplacements des propositions. Les deux modules partagent le même sous-réseau de base. D'une part, RPN peut générer les propositions d'objets candidats à l'aide de caractéristiques convolutives profondes ; il peut améliorer la qualité de l'emplacement des propositions. D'autre part, Faster RCNN est un cadre de bout en bout avec la perte multitâche. Comparé à Fast RCNN, Faster RCNN peut atteindre de meilleures performances de détection avec beaucoup moins de propositions. RPN glisse un petit réseau sur la couche de sortie du réseau de base. Le petit réseau se compose d'une couche convolutive 3×3 et de deux couches convolutives 1×1 apparentées pour la régression et la classification des boîtes. La classification par boîte est indépendante de la classe. Pour chaque emplacement de fenêtre coulissante, RPN prédit plusieurs propositions basées sur les ancres de différents rapports d'aspect et échelles. Bien que Faster RCNN ait brisé le goulot d'étranglement(bottleneck) de la vitesse de Fast RCNN, il y a encore une redondance

de calcul à l'étape de détection ultérieure. Par la suite, diverses améliorations ont été proposées, notamment RFCN[7] et Light head RCNN[37].

2.3.4.2 détecteurs à une étape basés sur CNN

Différentes du processus en deux étapes, les méthodes en une étape visent à prédire simultanément la catégorie et l'emplacement de l'objet. Par rapport aux méthodes en deux étapes, les méthodes à une étape ont une vitesse de détection beaucoup plus rapide et une précision de détection comparable.Ils sont appréciés par les appareils mobiles avec des fonctionnalités en temps réel et faciles à déployer, mais leurs performances souffrent sensiblement lors de la détection d'objets denses et petits. Parmi les méthodes en une étape, YOLO[57], SSD[40] et RetinaNet[38] sont les méthodes représentatives.

1. **Méthode You Only Look Once (YOLO)** :

   YOLO a été proposé par R. Joseph et al. en 2015. C'était le premier détecteur à une étape de l'ère de l'apprentissage en profondeur[57]. YOLO est extrêmement rapide : une version rapide de YOLO fonctionne à 155 ips(image par seconds) avec VOC07 mAP=52,7 %, tandis que sa version améliorée fonctionne à 45 ips avec VOC07 mAP=63,4 %. YOLO suit un paradigme totalement différent des détecteurs à deux étapes : appliquer un seul réseau neuronal à l'image complète. Ce réseau divise l'image en régions et prédit simultanément les boîtes englobantes et les probabilités pour chaque région. YOLO[57] divise l'image d'entrée en grilles k $\times$ k. Chaque cellule de la grille prédit l'évolution de l'image. Chaque cellule de la grille prédit B boîtes de délimitation(bounding box) avec des scores d'objectivité et c probabilités de classe conditionnelles. Les prédictions de chaque boîte englobante sont (x, y, w, h, s), où (x, y, w, h) indique l'emplacement de la boîte englobante et s est le score d'objectivité de confiance de la boîte englobante. La couche de sortie a donc une taille de n $\times$ n $\times$ (5B + c). Sur la base de la prédiction de la boîte englobante et la prédiction de la classe correspondante, YOLO peut simultanément donner la probabilité de l'objet et la probabilité de la catégorie de chaque image.

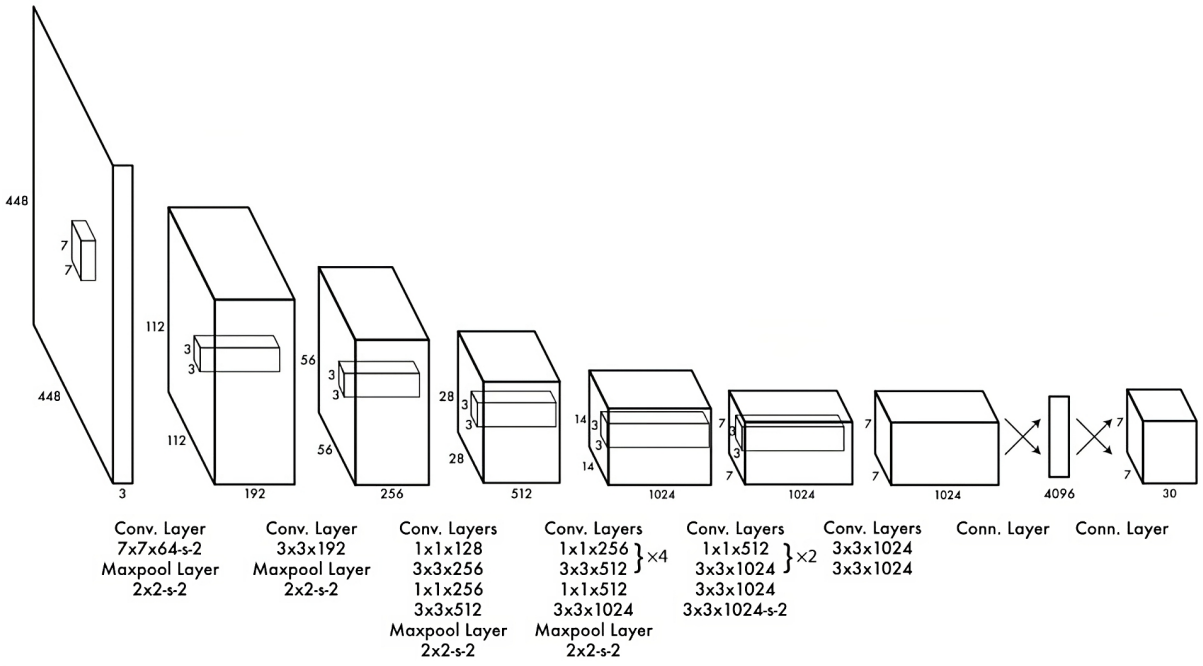


Figure 2.3.9 – L'architecture de YOLO[57].

La figure 2.3.9 montre l'architecture de YOLO. l'architecture de YOLO. Elle se compose de 24 couches convolutives et de 2 couches complètement connectées(FC layer). Pour accélérer la vitesse de détection, les couches convolutives 1×1 alternées sont utilisées dans certaines couches intermédiaires.La taille de l'image d'entrée de YOLO est fixe (448×448). Dans PASCAL VOC, B est fixé à 2 et c à 20. Ainsi, la couche de sortie de PASCAL VOC a une taille de 7×7×30. Malgré son amélioration considérable de la vitesse de détection, YOLO souffre d'une baisse de la précision de localisation par rapport aux détecteurs à deux étapes, en particulier pour certains petits objets. Les versions ultérieures de YOLO[5, 55, 56] et le SSD proposé plus tard[40] ont accordé plus d'attention à ce problème. Récemment, YOLOv7[69], un travail de suivi de l'équipe YOLOv4, a été proposé. Il surpasse la plupart des détecteurs d'objets existants en termes de vitesse et de précision (allant de 5 IPS à 160 IPS) en introduisant des structures optimisées comme l'attribution dynamique d'étiquettes et la reparamétrisation de la structure du modèle.

2. **Méthode Single Shot MultiBox Detector (SSD)** :

   SSD (Single Shot MultiBox Detector)[40] a été proposé par W. Liu et al. en 2015. C'est un détecteur d'un seul coup pour la détection d'objets génériques. Le réseau de base est basé sur VGG16[63]. Après le réseau de base, plusieurs couches convolutives sont ajoutées pour générer d'autres couches convolutives (c.-à-d. Conv6-Conv11) de différentes résolutions. Ensuite, il utilise plusieurs couches convolutives de différentes résolutions pour prédire des objets de différentes échelles. Plus précisément, pour

la couche convolutive de sortie d'une résolution donnée, il utilise d'abord un filtre convolutif 3 × 3 pour générer les nouvelles cartes de caractéristiques, puis prédit les scores des catégories d'objets et les emplacements des objets sur les nouvelles cartes de caractéristiques. Il performe la détection d'objets dans les images à l'aide d'un seul réseau neuronal profond. Cette approche est simple et efficace par rapport à d'autres méthodes qui nécessitent plusieurs étapes de traitement. SSD discrétise l'espace de sortie des boîtes de délimitation en un ensemble de boîtes par défaut sur différents rapports d'aspect et échelles par emplacement de carte de caractéristiques.Lors de la prédiction, le réseau génère des scores pour la présence de chaque catégorie d'objet dans chaque boîte par défaut et ajuste la boîte pour mieux correspondre à la forme de l'objet. De plus, le réseau combine les prédictions de plusieurs cartes de caractéristiques de résolutions différentes pour traiter naturellement des objets de tailles différentes.

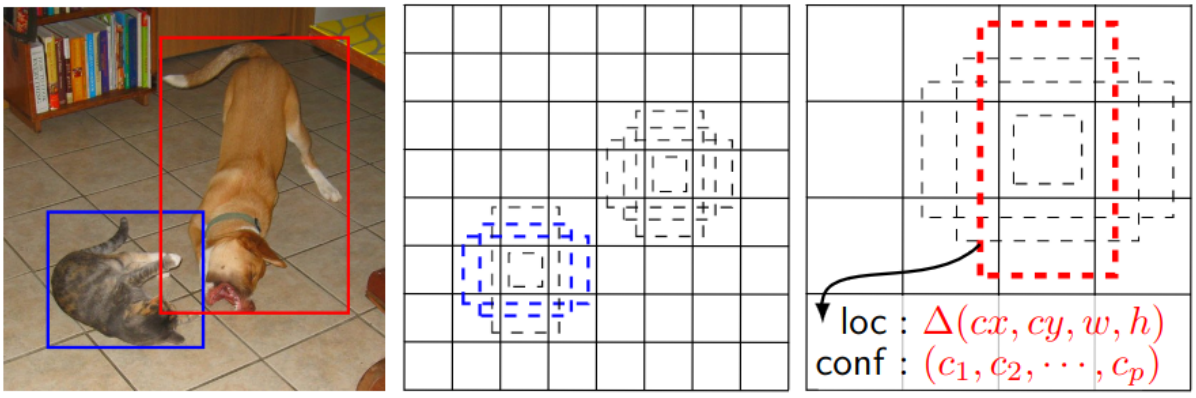


Figure 2.3.10 – Le framework de SSD[40].

Depuis la figure 2.3.10 on a : (a) Le SSD n'a besoin que d'une image d'entrée et de boîtes de vérité pour chaque objet pendant l'apprentissage. De manière convolutive, nous évaluons un petit ensemble (par exemple, 4) de boîtes par défaut de différents rapports d'aspect à chaque emplacement dans plusieurs cartes de caractéristiques à différentes échelles (par exemple, 8 × 8 et 4 × 4 dans (b) et (c)). Pour chaque boîte par défaut, nous prédisons les décalages de forme et les confiances pour toutes les catégories d'objets ((c1, c2, - - - , cp)). Au moment de l'apprentissage, nous commençons par faire correspondre ces boîtes par défaut aux boîtes de la vérité terrain. Par exemple, nous avons fait correspondre deux boîtes par défaut aux boîtes de vérité de terrain. Par exemple, nous avons fait correspondre deux boîtes par défaut avec le chat et une avec le chien, qui sont traitées comme positives et le reste comme positives. sont traitées comme positives et les autres comme négatives.[40]

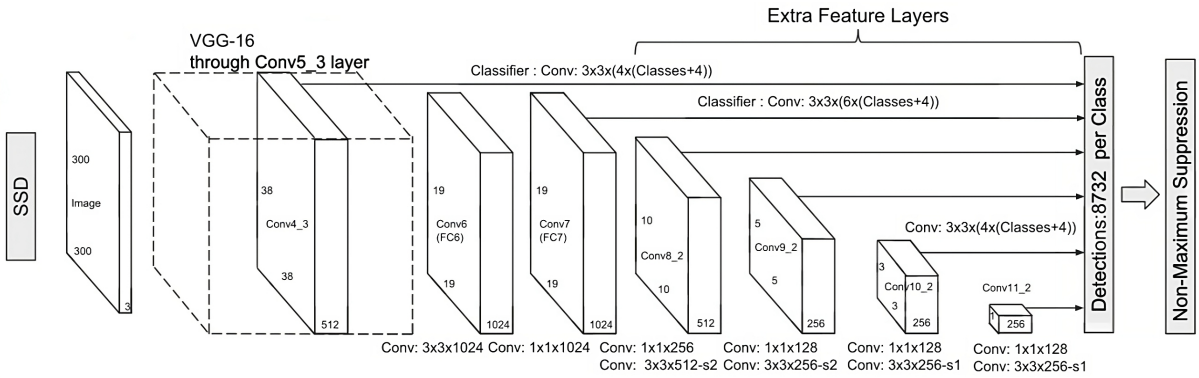


Figure 2.3.11 – L'architectures de SSD avec le backbone VGG16[40].

D'après le schéma ci-dessus, nous pouvons voir que l'architecture SSD s'appuie sur l'ancienne architecture VGG-16 mais ignore les couches entièrement connectées. VGG-16 a été utilisé comme réseau intégrateur en raison de ses hautes performances dans les tâches de classification d'images de haute qualité et de sa popularité [1]. Au lieu de la couche VGG entièrement connectée d'origine, un ensemble de couches convolutionnelles auxiliaires (de conv6) est ajouté, permettant l'extraction de caractéristiques à plusieurs échelles et réduisant progressivement la taille d'entrée de chaque couche suivante.

3. **Méthode de CenterNet** :

   En 2019, X. Zhou et al ont proposé le modèle CenterNet[77].Ce modèle suit un paradigme de détection basé sur les points clés, mais se distingue en éliminant les étapes de post-traitement coûteuses, ce qui permet d'obtenir un réseau de détection entièrement intégré de bout en bout (end-to-end). CenterNet considère chaque objet comme un seul point, représentant son centre, et prédit toutes ses caractéristiques (comme la taille, l'orientation, l'emplacement, la pose, etc.) par rapport à ce point central de référence. Le modèle est à la fois simple et élégant, et il est capable d'intégrer des tâches telles que la détection d'objets 3D, l'estimation de la pose humaine, l'apprentissage du flux optique, l'estimation de la profondeur et d'autres tâches dans un seul cadre. Malgré cette approche de détection concise, CenterNet parvient à obtenir des performances de détection comparables (COCO mAP@.5=61,1%).

1. le réseau VGG-16 a été utilisé comme base, mais d'autres réseaux devraient également donner de bons résultats, comme MobileNet, qui est utilisé pour notre cas

2.4 La vidéosurveillance

Avec le besoin croissant de sécurité dans le monde d'aujourd'hui, les technologies de surveillance ont évolué pour devenir plus intelligentes et sophistiquées. La surveillance vidéo intelligente, également connue sous le nom d'analyse vidéo ou d'analyse de contenu vidéo (VCA), implique l'utilisation d'algorithmes avancés et de techniques d'apprentissage automatique ou l'apprentissage en profendeur pour analyser les données vidéo en temps réel, extrayant des informations et des idées significatives du contenu visuel. Cet section explorera le concept de surveillance vidéo intelligente, ses applications, ses avantages et ses défis potentiels.

Figure 2.4.1 – Exemple de caméra de sécurité publique couramment vue.

2.4.1 Histoire de la vidéosurveillance

Naturellement, le mot surveillance est étroitement lié à la sûreté, à la sécurité, à la vie privée et à l'observation. La surveillance peut être utilisée à des fins positives ou négatives. C'est l'un des éléments les plus importants dans les films hollywoodiens du FBI ou d'espionnage, sous la forme de différents dispositifs d'enregistrement audio/vidéo sophistiqués. Dans cette mémoire, le scénario se limite à la vidéosurveillance à des fins de sécurité publique. Traditionnellement, la vidéosurveillance est également connue sous le nom de télévision en circuit fermé (CCTV). Bien que la télévision en circuit fermé soit définie du point de vue du matériel. Pendant de nombreuses décennies, le système de vidéosurveillance était simplement constitué de moniteurs connectés à des caméras. Les caméras sont la pierre angulaire du système de surveillance depuis plus de 70 ans. Le premier système de vidéosurveillance de l'histoire de l'humanité est installé à

Peenemünde, en Allemagne, en 1942, pour observer le lancement des fusées V-2[13]. Plus tard, ce type de vidéosurveillance a été utilisé aux États-Unis lors des essais de bombes atomiques. La première vidéosurveillance à domicile a été développée à New York en 1969. Elle a été inventée par Marie Van Brittan Brown, une infirmière qui ne se sentait pas en sécurité dans son quartier où le taux de criminalité était élevé. Cependant, il n'existe pas de solution de stockage adaptée et les utilisateurs doivent surveiller l'écran en permanence. C'est plus comme regarder avec l'œil depuis un télescope électronique. Ainsi, le marché de la vidéosurveillance ne peut se développer qu'après les années 1970, lorsque la technique du magnétoscope (magnétoscope) est développée. Dans les années 1990, le multiplexage numérique a été développé, offrant la possibilité d'enregistrer simultanément à partir de plusieurs caméras, ainsi que de réaliser des enregistrements accélérés et en mode mouvement uniquement. Cette avancée a permis de réaliser des économies de temps et d'argent, ce qui a conduit à une utilisation croissante de la vidéosurveillance (CCTV).[60]. En 1996, la première caméra IP a été inventée. Cette caméra pourrait envoyer et recevoir des informations sur des réseaux informatiques. Cela a conduit à des webcams ultérieures et a également marqué le début du déclin de la vidéosurveillance. Ainsi, le terme (vidéosurveillance) est plus souvent utilisé ces dernières années que CCTV. Après 2001, en particulier l'attaque du 911, les programmes de reconnaissance faciale et d'autres avancées numériques sont devenus une priorité plus élevée. Les caméras de surveillance basées sur Internet deviennent de plus en plus courantes. De nos jours, les systèmes de vidéosurveillance sont beaucoup plus avancés. Grâce à Internet et aux techniques sans fil, la vidéosurveillance peut désormais être utilisée et surveillée partout dans le monde. Des composants plus avancés tels que l'identification de personne, l'analyse de trajectoire ou la détection de mouvement ou détection d'événement sont intégrés dans les serveurs de post-traitement. Comme discuté précédemment, nous entrons dans l'oreille de la vidéosurveillance intelligente.

2.4.2 Applications de la surveillance vidéo intelligente

La surveillance vidéo intelligente trouve des applications dans divers domaines, notamment la sécurité publique, les transports, le commerce de détail, les infrastructures critiques et la sécurité résidentielle et le comptage des sujets (personnes/objets). Dans le domaine de la sécurité publique, la surveillance vidéo intelligente peut être utilisée pour la gestion des foules, la détection et l'alerte sur les comportements anormaux et l'identification des menaces potentielles dans les espaces publics, tels que les aéroports, les

gares et les stades[3]. Dans le secteur des transports, elle peut aider à la surveillance du trafic, au suivi des véhicules et à la détection des accidents ou incidents sur les routes[26]. Dans le commerce de détail, la surveillance vidéo intelligente peut aider à l'analyse du comportement des clients, au comptage des personnes et à la détection du vol à l'étalage ou d'autres violations de sécurité[45].Dans les infrastructures critiques, comme les centrales électriques ou les installations de traitement de l'eau, l'utilisation de la surveillance vidéo intelligente présente des avantages en termes de renforcement de la sécurité périmétrique, de surveillance des zones restreintes et de détection des tentatives d'intrusion.[32]. Enfin, en matière de sécurité résidentielle, la surveillance vidéo intelligente peut être utilisée pour surveiller les maisons, détecter les accès non autorisés et fournir un contrôle d'accès à distance aux propriétaires[64].

2.4.3 Avantages de la surveillance vidéo intelligente

La surveillance vidéo intelligente offre plusieurs avantages par rapport aux méthodes de surveillance vidéo traditionnelles. Tout d'abord, elle offre des capacités de surveillance et d'analyse en temps réel, permettant une réponse proactive aux menaces potentielles pour la sécurité[26]. Deuxièmement, elle peut réduire les fausses alarmes en filtrant les événements non pertinents et en générant des alertes précises uniquement pour les incidents pertinents, minimisant la charge de travail de l'opérateur et améliorant l'efficacité de la réponse[3]. Troisièmement, la surveillance vidéo intelligente peut fournir des informations et des données précieuses pour l'analyse et les enquêtes post-incident, aidant à identifier les modèles et les tendances pour la planification future de la sécurité[74]. De plus, elle peut améliorer la conscience de la situation en s'intégrant à d'autres systèmes de sécurité, tels que le contrôle d'accès ou la détection d'intrusion, créant ainsi un écosystème de sécurité complet[32]. Enfin, la surveillance vidéo intelligente peut potentiellement dissuader la criminalité et améliorer la sécurité publique en créant une présence de sécurité visible et proactive[64].

2.4.4 Technologies de surveillance intelligent actuelles

La technologie de surveillance intelligente actuelle utilise des techniques de `DL` et d'intelligence artificielle (`IA`) pour analyser les données collectées et prendre des décisions en temps réel. Par exemple, les algorithmes de `Deep Learning` peuvent être formés pour détecter automatiquement des objets ou des actions spécifiques, tels que des véhicules suspects, des visages, des mouvements ou d'autres éléments d'intérêt dans les images et

les vidéos de surveillance.

1. **Système de surveillance intelligent basé sur CCTV** :

   CCTV a connu trois générations : télévision analogique en circuit fermé (CCTV), conversion numérique-analogique (D/A) et technologie de vision par ordinateur. Les systèmes 1G ont été développés dans les années 1960 et utilisaient un dispositif numérique à couplage de charge (CCD) pour capturer des images, tandis que les systèmes 2G étaient basés manuellement et utilisés pour reconnaître, suivre et identifier des objets. Les systèmes de surveillance intelligents 1G ont les caractéristiques avantageuses des performances visuelles. Les systèmes de surveillance 1G ne suffisent pas pour le traitement automatisé en raison de la conversation des signaux numériques et analogiques, de l'enregistrement et du stockage vidéo numérique, de la suppression des artefacts et de la compression vidéo. Les systèmes 2G se concentrent sur le système multimodal automatisé, les systèmes de distribution intelligents, les plateformes multisensorielles et la fusion de données ; le raisonnement probabiliste a été adopté dans cette phase[64, 74].

   Les systèmes 3G fonctionnent sur une zone étendue avec des informations plus précises, mais manquent d'intégration d'informations et de communications. L'analyse de contenu visuel basée sur la vision par ordinateur est utilisée pour augmenter l'efficacité[64, 74].

2. **Système de surveillance intelligent basé sur IP camera** :

   La surveillance IP est la version numérisée et en réseau de la télévision en circuit fermé (CCTV). Dans les systèmes de surveillance IP, les caméras IP enregistrent des séquences vidéo et distribuent le contenu résultant aux réseaux IP[32]. Voici un système qui offre des avantages supplémentaires :

   — Amélioration des capacités de visualisation et de contrôle à distance. N'importe qui sur le réseau peut voir la vidéo de n'importe quelle caméra connectée au réseau.

   — Le stockage IP permet de stocker les données n'importe où géographique.

   — Plus facile à distribuer. Par exemple, des images de suspects, Prêt pour une distribution immédiate.

   — Possibilité de se connecter au courrier électronique et à d'autres systèmes de communication afin qu'elle puisse être envoyée automatiquement.

3. **Système de surveillance intelligent basé sur microcontrolleur** :

   Les microcontrôleurs sont des composants électroniques programmables qui

combinent un processeur, une mémoire et des interfaces d'entrée/sortie dans un seul circuit intégré. En raison de leur faible coût, de leur faible consommation d'énergie et de leur flexibilité de programmation, ils sont couramment utilisés dans diverses applications, y compris les systèmes de surveillance intelligente[75].

4. **Système de surveillance intelligent basé sur Raspberry Pi** :
   Un système de surveillance basé sur Raspberry Pi est une solution de surveillance qui utilise le Raspberry Pi, un ordinateur monocarte basé sur un processeur ARM, pour capturer, traiter et transmettre des données de surveillance. Raspberry Pi est largement utilisé dans diverses applications, y compris les systèmes de surveillance, en raison de sa taille compacte, de sa polyvalence et de sa faible consommation d'énergie[50]. Le Raspberry Pi peut être connecté à Internet, permettant aux données de surveillance d'être transmises en temps réel ou stockées dans le cloud ou dans un serveur pour un accès à distance[44].

2.4.5 Défis de la surveillance vidéo intelligente

Bien que la surveillance vidéo intelligente présente de nombreux avantages, elle doit également relever des défis qui doivent être pris en compte pour une mise en œuvre efficace. L'un des défis est le besoin de ressources informatiques importantes pour l'analyse vidéo en temps réel, ce qui peut être coûteux et nécessiter une infrastructure robuste[26]. Un autre défi est le risque potentiel d'atteinte à la vie privée, car la surveillance vidéo peut capturer des données personnelles et soulever des questions éthiques concernant l'utilisation et la conservation des données[3]. De plus, l'exactitude et la fiabilité des algorithmes d'analyse vidéo peuvent être affectées par des facteurs tels que les conditions d'éclairage, la qualité de la caméra et les occlusions, ce qui peut entraîner des faux positifs ou négatifs[33]. Il est également nécessaire de disposer d'un personnel qualifié pour exploiter et gérer les systèmes de surveillance vidéo intelligents, notamment des analystes de données et des opérateurs de sécurité[32]. Enfin, il peut y avoir des défis juridiques et réglementaires liés à la protection de la vie privée des données, à la protection des données et à la conformité aux lois et règlements locaux[64].

2.5 Conclusion

Dans ce chapitre, nous avons introduit les réseaux de neurones convolutifs (CNNs) et on a présenté les techniques de détection d'objets utilisant l'apprentissage en profondeur

plus particulièrement les CNNs. Nous avons discuté des avantages de ces techniques pour améliorer la précision et la rapidité de la détection d'objets dans les images ou vidéo. Nous avons également analysé les défis liés à l'utilisation de l'apprentissage en profondeur pour la détection d'objets, tels que la nécessité de grandes quantités de données d'entraînement et la complexité des modèles, nous avons aussi introduit la vidéosurveillance, on a examiné les systèmes actuels de la vidéosurveillance et aussi leurs défis dans la société moderne. Dans le chapitre suivant, nous explorerons les outils matériels et logiciels nécessaires à la réalisation de notre système de détection d'objets appliqué pour une vidéosurveillance intelligent à base de Raspberry Pi.

CHAPITRE 3

outils matériels et logiciels pour la réalisation de système

3.1 Introduction

Après avoir entamé la théorie derrière notre travail, nous présenterons dans ce chapitre les différents composants matériels et logiciels nécessaires à la réalisation du projet. Nous commencerons par décrire les spécifications matérielles requises pour le projet, telles que la Raspberry Pi, la caméra Pi. On donne une description de l'écran TFT LCD utilisé comme moniteur. Nous expliquerons également les avantages de ces choix que nous avons faits en termes de modèle pour chaque composant, ainsi que les raisons de ces choix. Ensuite, nous aborderons les aspects logiciels du projet, en détaillant les différentes bibliothèques et les frameworks que nous avons utilisés pour le développement. Nous expliquerons par ailleurs le choix de langage de programmation utilisé dans notre projet. Enfin, nous terminerons ce chapitre avec une conclusion qui sommaire ce chapitre.

3.2 Conception du système

La conception que nous avons imaginée repose sur la base d'une Raspberry Pi, une caméra Pi, une page web et une base de données locale. Le but ici est de créer une page web dans laquelle les personnes détecté sont capturées avec l'horodatage de cette capture. Ces informations sont synchronisées sur la page web. La surveillance peut-être soit faite dans la page web ou par l'écran LCD TFT. Le personnel responsable aura toutes les informations sur l'état de la surveillance et pourra facilement consulter la vidéo de surveillance depuis l'écran installé avec la Raspberry Pi s'il est présente ou à travers le site web. Le schéma présenté en figure 3.2.1 nous donne une vue générale de système à réaliser et, cela nous permet d'obtenir une meilleure compréhension de l'ensemble du fonctionnement du système.

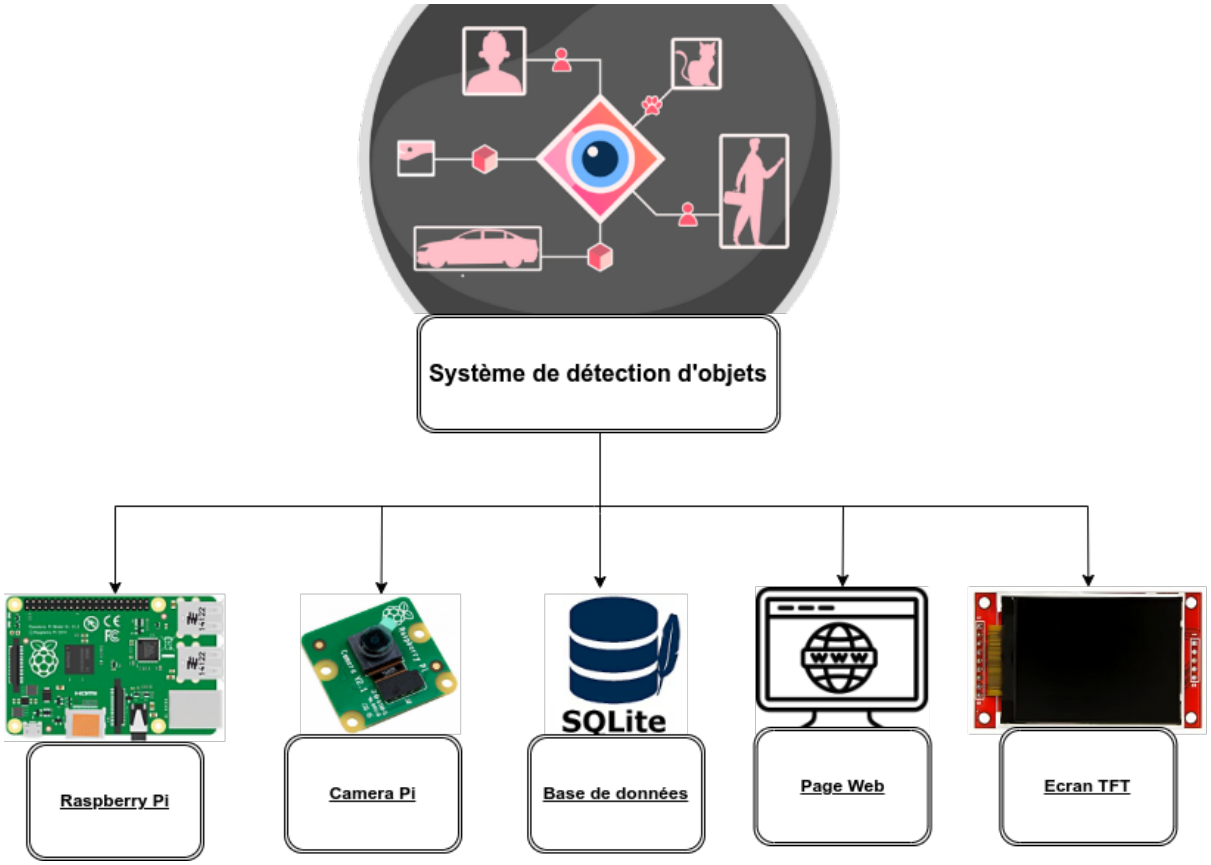


Figure 3.2.1 – Vue générale de système à réaliser.

Afin d'expliquer en détail les composants du système, nous choisissons de les diviser en deux parties principales : la partie matérielle et la partie logicielle

3.3 Partie matérielle

Dans cette partie, nous définissons les éléments électroniques qui caractérisent le système de détection d'objet.

3.3.1 La Raspberry Pi

La Raspberry Pi est une série de petits ordinateurs monocartes (SBC) abordables qui ont été développés au Royaume-Uni par la fondation Raspberry Pi. Le premier Raspberry Pi a été publié en 2012, et depuis, plusieurs modèles ont été introduits avec une puissance et des fonctionnalités croissantes.

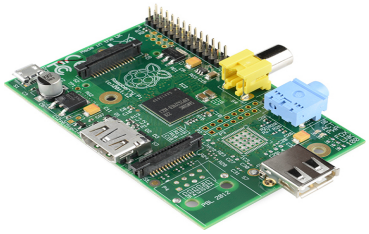
La Raspberry Pi est conçu pour être un ordinateur polyvalent et peu coûteux qui peut être utilisé pour une large gamme de projets, notamment la robotique, l'automatisation domestique, les centres multimédias et même comme ordinateur de bureau. Il fonctionne sur une variété de systèmes d'exploitation, y compris des distributions basées sur le

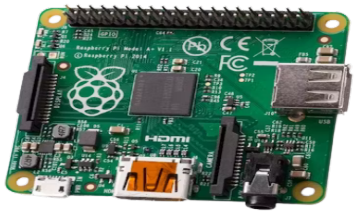
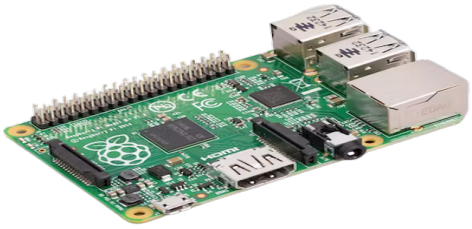
système d'exploitation Linux comme Raspbian, ainsi que Windows 10 et d'autres. Le Raspberry Pi est connu pour sa petite taille, sa faible consommation d'énergie et son faible coût, ce qui le rend accessible à un large éventail de personnes, y compris les étudiants, les amateurs et les professionnels. Ses broches GPIO (Entrées/Sorties à usage général) lui permettent également d'interagir avec une large gamme de capteurs, de moteurs et d'autres composants électroniques, en faisant un choix populaire pour les projets électroniques DIY et même professionnelle.

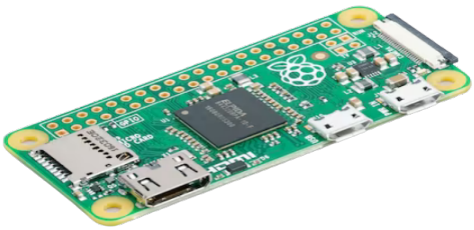
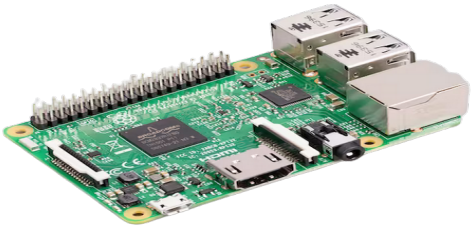
Dans l'ensemble, la Raspberry Pi est devenu une plateforme populaire pour l'éducation, l'expérimentation et l'innovation dans le monde de l'informatique et de l'électronique.

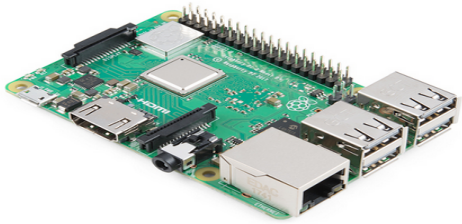
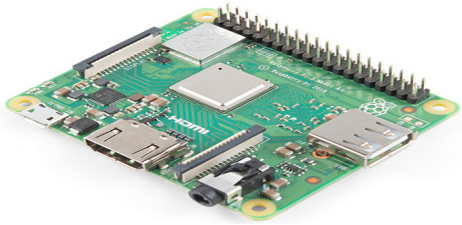
3.3.2 Les modèles du Raspberry Pi

Les modèles Raspberry Pi sont de petits ordinateurs qui se présentent sous différents formats, tailles et spécifications. Ils sont identifiables par leur famille (1-4, 400, Zero) et leur nom de modèle (A, B, A+, B+). Les noms de modèles indiquent si la carte dispose d'Ethernet et s'il s'agit d'une version améliorée d'une précédente. Le dernier modèle de Raspberry Pi est la Raspberry Pi 4, qui possède deux ports micro-HDMI et peut avoir 1 Go, 2 Go, 4 Go ou 8 Go de RAM. Les Raspberry Pi 2 et 3 ont des processeurs quad-core, tandis que les Raspberry Pi 1 et Zero ont des processeurs monocœur. Les modèles Raspberry Pi A et A + manquent de fentes pour cartes Ethernet et MicroSD[42].

Le modéle	La carte	Déscription
Raspberry Pi Model A (Sorti en 2012)		➤ CPU : ARMv6Z 700 MHz ➤ Nombre de cœurs : 1 ➤ RAM : 256 Mo

Raspberry Pi Model B (Sorti en 2012)		➤ CPU : ARMv6Z 700 MHz ➤ Nombre de cœurs : 1 ➤ RAM : 512 Mo
Raspberry Pi Model A+ (Sorti en 2014)		➤ CPU : ARMv6Z 700 MHz ➤ Nombre de cœurs : 1 ➤ RAM : 512 Mo
Raspberry Pi Model B+ (Sorti en 2014)		➤ CPU : ARMv6Z 700 MHz ➤ Nombre de cœurs : 1 ➤ RAM : 512 Mo

Raspberry Pi 2 Model B (Sorti en 2015)		➤ CPU : ARMv7-A 900 MHz ➤ Nombre de cœurs : 4 ➤ RAM : 1 Go
Raspberry Pi Zero (Sorti en 2015)		➤ CPU : ARMv6Z 1 GHz ➤ Nombre de cœurs : 1 ➤ RAM : 512 Mo
Raspberry Pi 3 Model B (Sorti en 2016)		➤ CPU : ARMv8-A 1.2 GHz ➤ Nombre de cœurs : 4 ➤ RAM : 1 Go

Raspberry Pi Zero W (Sorti en 2017)		➤ CPU : ARMv6Z 1 GHz ➤ Nombre de cœurs : 1 ➤ RAM : 512 Mo
Raspberry Pi 3 Model B+ (Sorti en 2018)		➤ CPU : ARMv8-A 1.4 GHz ➤ Nombre de cœurs : 4 ➤ RAM : 1 Go
Raspberry Pi 3 Model A+ (Sorti en 2018)		➤ CPU : ARMv8-A 1.4 GHz ➤ Nombre de cœurs : 4 ➤ RAM : 512 Mo

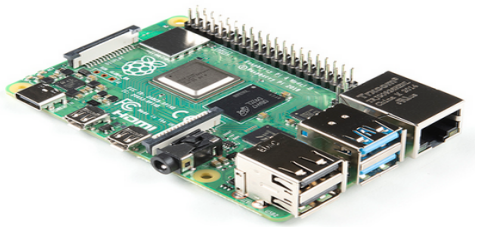
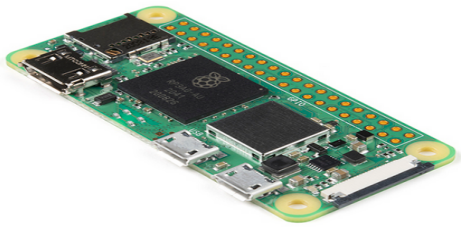
Raspberry Pi 4 Model B (Sorti en 2019)		➤ CPU : ARMv8-A 1.5 GHz ➤ Nombre de cœurs : 4 ➤ RAM : 2/4/8 Go
Raspberry Pi Zero 2 W (Sorti en 2021)		➤ CPU : ARMv8-A 1 GHz ➤ Nombre de cœurs : 4 ➤ RAM : 512 Mo
Raspberry Pi 400 (Sorti en 2021)		➤ CPU : ARMv8-A 1.8 GHz ➤ Nombre de cœurs : 4 ➤ RAM : 4 Go

Table 3.3.1 – Les modèles du Raspberry Pi

En 2022, La fondation Raspberry Pi a sorti un nouveau microcontrôleur appelé

Raspberry Pi Pico. Il est construit à l'aide de la puce de microcontrôleur RP2040 conçue par Raspberry Pi au Royaume-Uni[52].

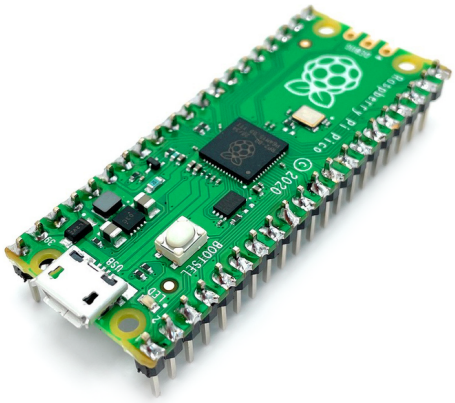


Figure 3.3.1 – Raspberry Pi Pico.[52]

Les microcontrôleurs sont distincts des ordinateurs monocartes, car ils ne fonctionnent pas avec un système d'exploitation et sont généralement programmés pour effectuer une seule tâche. La Raspberry Pi Pico est une carte microcontrôleur plurivalente et puissante adaptée à une variété d'applications, en raison de sa taille compacte et de son coût abordable, cela en fait une option attrayante tant pour les amateurs que pour les fabricants.

3.3.3 Le Raspberry Pi 3 type B+

La Raspberry Pi 3 B+ a été lancé par la Raspberry Pi Fondation le 14 mars 2018. Il s'agit d'une version avancée du modèle Raspberry Pi 3 B introduit en 2016[52]. La Raspberry Pi 3 B+ est une excellente édition de la gamme des Pis modernes, il n'est pas surprenant que ce soit la planche de choix pour la plupart des projets de bricolage et professionnels ! La carte dispose d'un processeur 1,4 GHz, Bluetooth, WiFi, 4 ports USB et d'une excellente réputation.

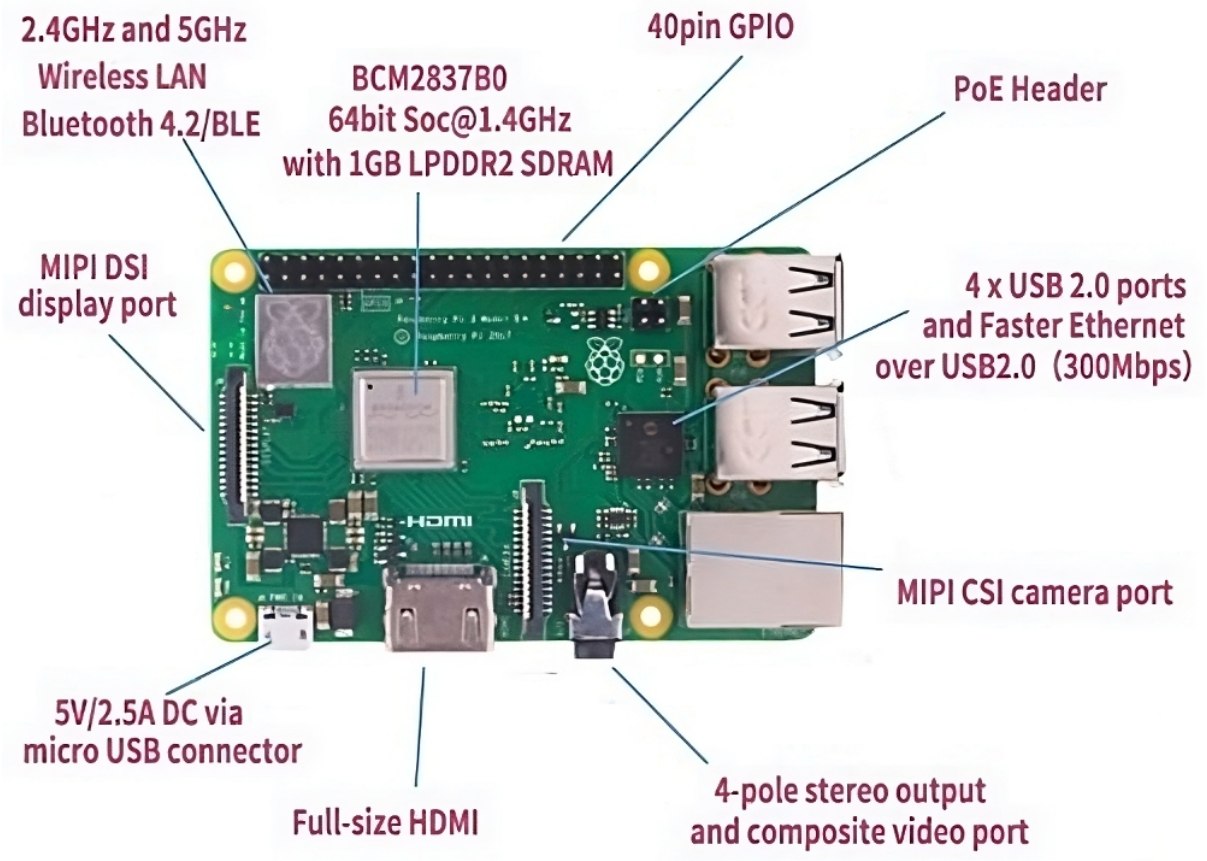


Figure 3.3.2 – Raspberry Pi 3B+[66]

3.3.3.1 Les composantes du RPi 3B+

Voici un tableau récapitulant les composants du Raspberry Pi 3 B+ :

La composante	Déscription
SoC	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz
CPU	ARMv8-A type Cortex-A53 64 bit @ 1.4GHz
Nombre de cœurs	4
GPU	Broadcom VideoCore IV
Mémoire	1GB LPDDR2 SDRAM
Accès	En-tête GPIO étendu à 40 broches
Audio et Video	➤ 1 × HDMI pleine taille ➤ Sortie stéréo 4 pôles et port vidéo composite ➤ Port d'affichage MIPI DSI

Ports USB	Ports USB 2.0 x4
Ethernet	Gigabit Ethernet
Prise en charge de la carte SD	Format Micro SD pour le chargement du système d'exploitation et stockage de données
Alimentation d'entrée	➤ 5 V/2.5A DC par un cable MicroUSB ➤ 5V DC via l'en-tête GPIO ➤ Power over Ethernet (PoE)
Interface camera	Interface camera type CSI
Connectivité sans fil	Il est livré avec des installations WiFi IEEE 802.11 et Bluetooth 4.2 qui n'étaient pas présentes dans les versions précédentes de Raspberry Pi 1 et 2

Table 3.3.2 – Les composantes du Raspberry Pi

3.3.3.2 Connexion des composantes

Les composants ou accessoires de base nécessaires pour une Raspberry Pi et comment les connecter pour le configurer et le faire fonctionner.

— **Une alimentation micro-USB**

Le courant de sortie est compris entre 2 A et un maximum de 2,5 A, puisque le courant dépend de la charge tirée par la Raspberry Pi et du 5 V avec le connecteur micro USB. L'alimentation officielle de la Raspberry Pi est le choix recommandé, car elle peut répondre aux besoins de la Raspberry Pi sans problème, ou acheter une bonne alimentation.

Figure 3.3.3 – Alimentation pour Raspberry Pi

— **La carte SD**

Les ordinateurs Raspberry Pi utilisent une carte micro SD, à l'exception des tout premiers modèles qui utilisent une carte SD pleine taille. la plupart des Raspberry Pis n'ont pas de stockage interne, s'appuyant plutôt sur des cartes microSD amovibles comme disques de démarrage principaux pour le système d'exploitation et de stockage de fichiers. En général, le manque de stockage interne maintient le coût de la Raspberry Pi à un niveau historiquement bas. Les cartes MicroSD sont un moyen de stockage rentable. Au moment d'écrire ces lignes, des cartes micro SD hautes performances de 32 Go peuvent être achetées pour 10 $ sur AliExpress ou Amazon. Il est donc obligatoire d'en avoir une carte de bonne qualité et très performante pour ne pas ralentir le Pi, L'idéal est d'avoir une carte au standard SDHC (classe 10)

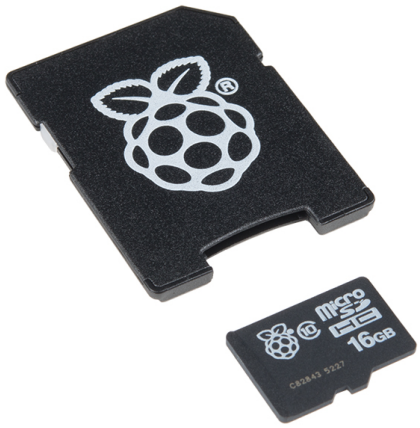


Figure 3.3.4 – Une carte SD pour Raspberry Pi

— **Le clavier et la souris**

Bien qu'un clavier et une souris soient requis pour installer le système d'exploitation et configurer la Raspberry Pi, ils peuvent être considérés comme des accessoires optionnels, car la plupart des gens possèdent déjà ces équipements chez eux. Cependant, il est toujours recommandé de les avoir au moins pendant la phase initiale de configuration de la Raspberry Pi, puisqu'on peut les négliger si on utilise le RPi à distance.

Figure 3.3.5 – Clavier et une souris pour Raspberry Pi

— **Les en-têtes GPIO**

L'en-tête Raspberry est la clé de sa capacité à s'interfacer avec le monde réel. Le RPi utilise un 40 broches ou un 26 broches selon le modèle et il est important de comprendre comment ces broches sont disposées et étiquetées.

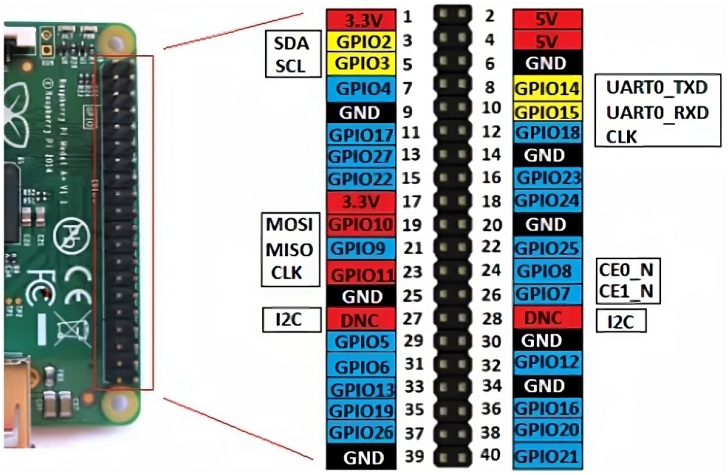


Figure 3.3.6 – Les en-têtes GPIO de Raspberry Pi 3B+.[43]

L'en-tête GPIO fournit les options d'alimentation et d'interface suivantes :

— 3.3V (sur 2 broches)

— 5V (sur 2 broches)

— Masse (sur 8 broches)

— Entrée et sortie à usage général (GPIO)

— PWM (modulation de largeur d'impulsion)

— I2C

— I2S

— SPI

— Connexion en série (UART)

L'en-tête fournit 17 broches qui peuvent être configurées comme entrées et sorties. Par défaut, elles sont toutes configurées en entrées, sauf GPIO 14 et 15. Pour utiliser ces broches, vous devez indiquer au système s'il s'agit d'entrées ou de sorties. Cela peut être réalisé de plusieurs façons, et cela dépend de la façon dont vous avez l'intention de les contrôler. Ceux-ci permettent de connecter une vaste gamme de capteurs, moteurs, LED et accessoires au RPi. Pour notre cas, on va les utiliser pour connecter le RPi avec un écran TFT.

3.3.3.3 Programation du Raspberry Pi 3B+

La Raspberry Pi 3B+ est une plateforme polyvalente pour apprendre la programmation, développer des projets IoT et exécuter diverses applications. Voici un aperçu de la programmation sur la Raspberry Pi 3B+ :

— **Langages de programmation** :

Le Raspberry Pi 3B+ prend en charge une variété de langages de programmation, notamment Python, C/C++, Java, JavaScript, PHP et Ruby. En raison de sa simplicité, de sa lisibilité et de sa polyvalence, Python est le langage le plus populaire pour la programmation Raspberry Pi.

— **Environnements de développement intégrés (IDE)** :

de nombreux IDE sont disponibles pour la programmation sur le Raspberry Pi 3B+. Les options courantes incluent IDLE (IDE Python), Geany (IDE multilingue), Visual Studio Code (IDE multiplateforme) et Vim...

— **Programmation GPIO** :

Le Raspberry Pi 3B+ est doté de broches d'entrée/sortie à usage général (GPIO) qui permettent à l'utilisateur de connecter et de contrôler des périphériques externes tels que des LED, des capteurs et des moteurs. Des langages de programmation tels que Python ou C/C++ peuvent être utilisés pour contrôler les broches GPIO.

— **Programmation Internet des objets (IoT)** :

il est possible d'interagir avec les capteurs et les actionneurs et de connecter le Raspberry Pi 3B+ à Internet, ce qui en fait une plate-forme idéale pour la construction de projets IoT.

— **Analyse et visualisation des données** :

Raspberry Pi 3B+ peut être utilisé pour l'analyse et la visualisation des données à l'aide d'outils tels que les bibliothèques de Python.

— **Intelligence artificielle (IA)** :

Il est possible d'utiliser le Raspberry Pi 3B+ pour la programmation de l'IA, notamment l'apprentissage automatique et l'apprentissage en profondeur, en utilisant des bibliothèques telles que TensorFlow, Keras et OpenCV pour créer des applications d'IA.

Dans l'ensemble, Raspberry Pi 3B+ est une plate-forme polyvalente, permettent de créer des applications innovantes sont infinies.

3.3.4 La caméra Pi

Les modules de caméra dédiés aux Raspberry sont de petites caméras peu coûteuses qui peuvent être connectées à la carte d'ordinateur Raspberry Pi pour permettre la capture d'images et de vidéos. Il existe plusieurs modules de caméra disponibles pour Raspberry Pi, chacun avec ses propres caractéristiques et spécifications. Il existe de nombreux modules de caméra officiels Raspberry Pi. Le modèle original de 5 mégapixels est sorti en 2013, il a été suivi d'un module d'appareil photo version 2 de 8 mégapixels qui est sorti en 2016. Le dernier modèle d'appareil photo est le module d'appareil photo 12 mégapixels version 3 qui est sorti en 2023[52]. En plus du module de caméra officiel, il existe également divers modules de caméra tiers disponibles pour Raspberry Pi. Certains de ces modules offrent des fonctionnalités supplémentaires telles que l'imagerie infrarouge, la vision nocturne et le zoom motorisé.

Figure 3.3.7 – La camera Pi de Raspberry Pi[52]

Les modules de caméra Raspberry Pi peuvent être utilisés pour une variété d'applications, y compris les systèmes de sécurité domestique, la surveillance, la photographie accélérée, etc. Ils sont également populaires parmi les amateurs et les éducateurs à des fins d'enseignement et d'expérimentation.

3.3.5 Écran TFT LCD

L'écran TFT LCD (Thin-Film-Transistor Liquid Crystal Display) est un type d'écran plat qui utilise la technologie des transistors à couche mince pour contrôler les cristaux liquides qui composent l'affichage[73]. Il s'agit d'un type de panneau LCD qui utilise la technologie des transistors à couches minces pour fournir des images plus nettes et plus

lumineuses[73]. Les panneaux TFT ont également tendance à être en plastique, ce qui signifie qu'ils sont moins susceptibles de se casser ou de se fissurer lors d'un impact avec le sol ou un autre objet dur.

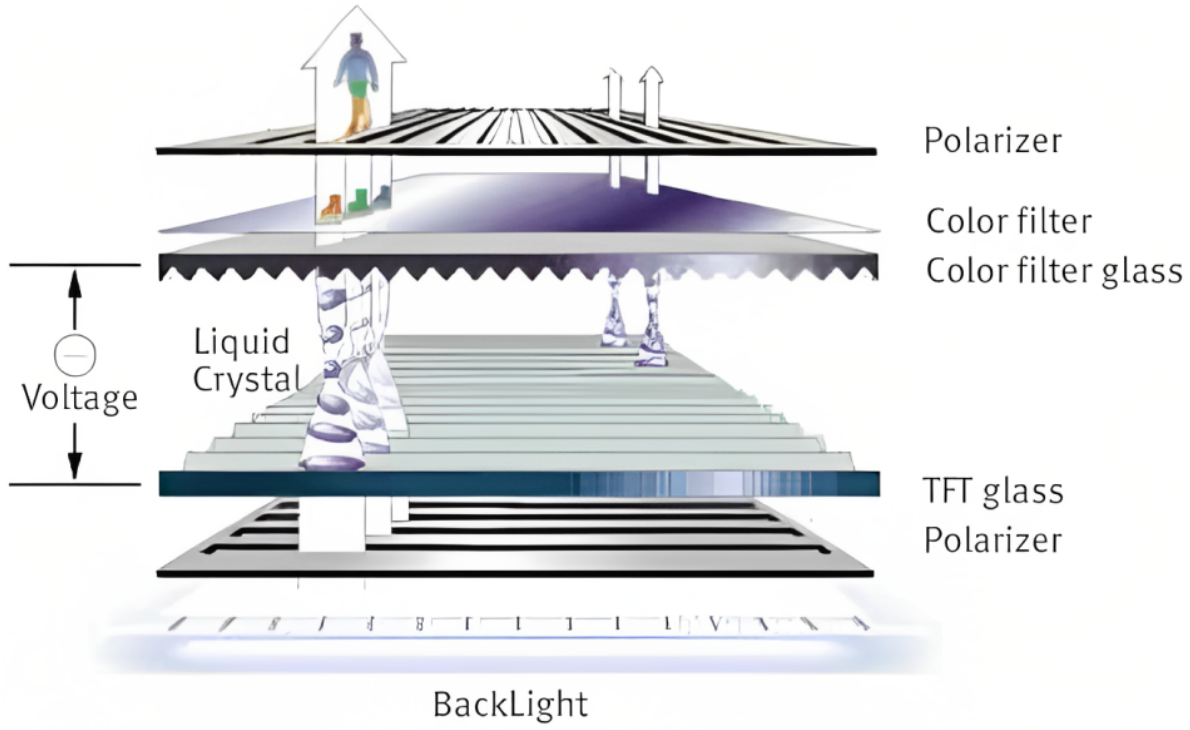


Figure 3.3.8 – La structure de l'écran TFT LCD[73]

Les écrans LCD TFT sont largement utilisés dans de nombreux appareils électroniques, notamment les smartphones, les ordinateurs portables, les téléviseurs et les appareils photo numériques, en raison de leur haute définition, de leur faible consommation d'énergie et de leurs angles de vision larges.

La version 5 pouces de l'écran LCD TFT est une option d'affichage populaire à utiliser avec les cartes Raspberry Pi. Cet écran est conçu pour se connecter directement aux broches GPIO du la Raspberry Pi et peut être utilisé comme affichage principal ou comme affichage secondaire dans une configuration à deux moniteurs. L'écran LCD TFT de 5 pouces a généralement une résolution de **800 × 480 pixels**, ce qui convient à l'affichage de graphiques et de textes de base. L'écran est également tactile sur certains modèles, permettant aux utilisateurs d'interagir avec la Raspberry Pi à l'aide de gestes tactiles. L'un des avantages de l'utilisation de l'écran LCD TFT de 5 pouces avec Raspberry Pi est qu'il peut être programmé à l'aide de divers langages de programmation, notamment Python, C++ et Java. Cela permet de développer facilement des applications personnalisées qui peuvent tirer parti des capacités tactiles de l'écran si elles sont disponibles.

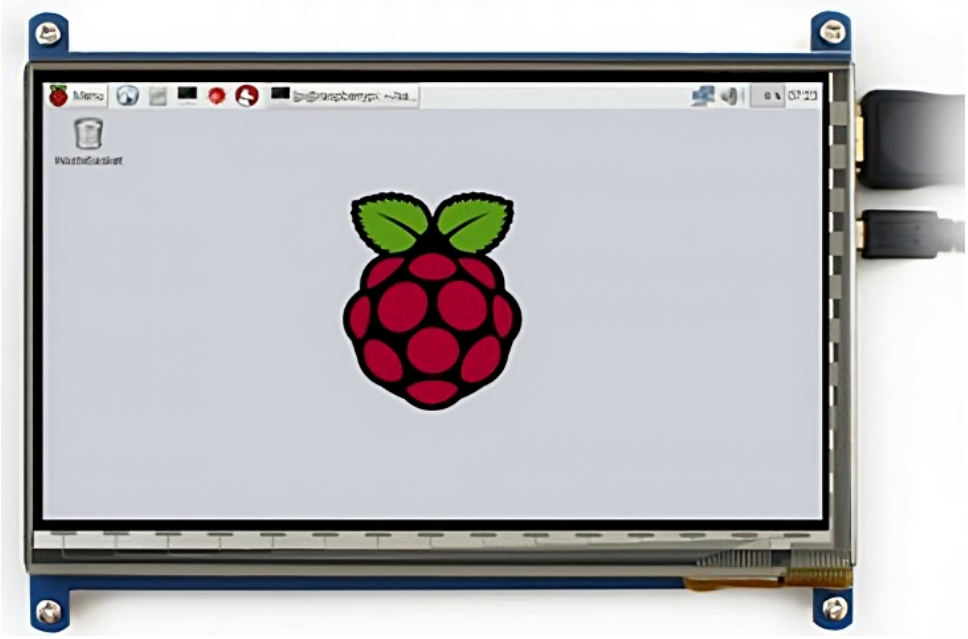


Figure 3.3.9 – L'écran TFT LCD 5 pouces

3.3.5.1 Fonctionnalités

- Taille en diagonale : 5 pouces
- Résolution : 800x480 (RGB)
- Taille du module : 120,7 mm (Largeur) x 75,8 mm (Heuteur)
- Taille de pixel : 0,135 mm x 0,135 mm
- Type d'interface : RGB 24 bits

Dans l'ensemble, l'écran LCD TFT de 5 pouces est une option d'affichage polyvalente et économique pour les projets Raspberry Pi qui nécessitent un écran compact à faible consommation d'énergie. Il est idéal pour les projets nécessitant des graphiques de base et un affichage de texte, ainsi que des interfaces utilisateur tactiles si nécessaire.

3.4 La partie logicieles du système

Cette section présente les bibliothèques requises pour le projet. Ces librairies permettent d'utiliser l'IA de manière simple et ergonomique, de communiquer avec une base de données locale, et d'afficher les données sur une page web. Ces bibliothèques sont utilisées via le langage de programmation Python, qui est également présenté.

Bibliothéque	Déscription
TensorFlow Lite	Bibliothèque pour l'apprentissage automatique conçue pour déployer des modèles sur des appareils mobiles, et embarqués
OpenCV	Bibliothèque pour traitement de vidéo et d'images
NumPy	Bibliothèque pour calcules multidimensionnel hautes performances
SQLite	Moteur de base de données
Flask	Bibliothèque puissant pour créer des applications Web complexes
Pillow	Bibliothèque pour traitement et conversion d'images

Table 3.4.1 – Les Bibliothéques éxterne utilisé dans le projet

3.4.1 Le language Python

Python est considéré comme l'un des langages de programmation les plus simples à apprendre et les plus agréables à utiliser. Le code Python est facile à lire et à écrire, il est concis mais pas obscur. Il est également très expressif, ce qui signifie que l'on peut souvent écrire moins de lignes de code Python que ce qui serait nécessaire pour une application équivalente écrite en C++ ou Java. Guido van Rossum, l'inventeur du langage, a utilisé le système d'exploitation Amoeba pour scripter Python à la fin des années 1989 et au début des années 1990.[72]. Guido van Rossum dirige toujours en tant que concepteur et développeur Python[20]. Depuis sa sortie au public en 1991, Python a rassemblé un grand nombre de groupes enthousiastes qui incorporent des éducateurs, des informaticiens et des programmeurs professionnels en tant qu'utilisateurs. Python est un langage multiplateforme : en général, le même programme Python peut être exécuté sur des systèmes Windows et de type Unix tels que Linux, BSD et Mac OS X, simplement en copiant le ou les fichiers qui composent le programme vers le machine cible, sans "construction" ou compilation nécessaire[20]. Il est possible d'écrire des programmes Python qui utilisent des fonctions spécifiques à la plate-forme, mais ce n'est généralement

pas nécessaire, car la quasi-totalité de la bibliothèque standard Python et la plupart des bibliothèques tierces sont complètement et de manière transparente indépendante de la plate-forme. Python est l'un des langages de programmation les plus anciens, il est considéré comme l'un des langages les plus essentiels qu'une personne doit apprendre si elle souhaite poursuivre dans le domaine de la programmation. Le langage de programmation Python est libre et orienté objet. Le principal avantage de la programmation en Python est sa vaste bibliothèque et son support communautaire remarquable.

Figure 3.4.1 – Logo de la language de programmation Python.

La programmation Python est actuellement en demande et son utilisation ne cesse d'augmenter. Comme nous pouvons voir comment l'intelligence artificielle prend actuellement tout en charge, la demande de programmation Python sera donc très élevée à l'avenir car c'est l'un des meilleurs langages à utiliser pour développer l'IA et l'apprentissage automatique et en profondeur.

3.4.1.1 Avantages du langage Python

Voici quelques-uns des avantages de l'utilisation de Python :

- **Facile à apprendre** :

  Python a une syntaxe simple et facile à apprendre. Cela en fait un excellent choix pour les débutants qui débutent dans la programmation ou pour les étudiants.

- **Puissant et flexible** :

  Python est un langage puissant et flexible qui peut être utilisé pour une grande variété de tâches. Cela ne se limite pas au développement Web ou à la science des données, mais elle peut être même utilsé dans l'intelligence artificielle, le développement de logiciels, l'automatisation et les scripts.

- **Libre et open source** :

  Python est un langage libre et open source. Cela signifie qu'il est accessible à tous gratuitement.

- **Grande communauté** :

  Python possède une communauté importante et active d'utilisateurs et de

développeurs. Cela signifie qu'il existe une multitude de ressources disponibles pour vous aider à apprendre et à utiliser Python

- **Bibliothèques et frameworks** :

  Python a une large gamme de bibliothèques et frameworks qui permettent de écrire de code rapidement et efficacement, notamment NumPy, Pandas, Flask et TensorFlow...

3.4.1.2 Bibliothèques intégrées (built-in) utilisées avec Python

Les bibliothèques intégrées dans Python sont des outils essentiels pour les développeurs car elles offrent une vaste gamme de fonctionnalités qui améliorent l'expérience de programmation et facilitent la création d'applications complexes. Les bibliothèques intégrées de Python offrent une multitude de fonctionnalités, des types de données de base aux opérations mathématiques avancées, en passant par la connectivité réseau, etc. Ces bibliothèques sont conçues pour fonctionner de manière transparente avec le langage, ce qui permet aux développeurs de créer facilement des applications complexes avec un minimum d'effort. En plus, ces bibliothèques sont incluses dans l'installation standard de Python, l'utilisateur peut donc commencer à les utiliser immédiatement sans avoir à installer de packages ou de modules supplémentaires qui exigent la connexion internet. Certaines bibliothèques intégrées couramment utilisées dans Python incluent :

- **Module 'os'** :

  La bibliothèque **'os'** fournit un moyen d'utiliser les fonctionnalités dépendant du système d'exploitation, telles que la création, la lecture et la suppression de fichiers et de répertoires, l'accès aux variables d'environnement et l'exécution de commandes externes.

- **Module 'sys'** :

  La bibliothèque 'sys' donne accès aux variables utilisées ou maintenues par l'interpréteur et aux fonctions qui interagissent fortement avec l'interpréteur.

- **Module 'datetime'** :

  La bibliothèque **'datetime'** fournit des classes pour travailler avec les dates et les heures, ce qui est particulièrement utile pour les applications qui nécessitent une planification, une gestion des événements et d'autres opérations liées au temps.

- **Module 'time'** :

  Le module de temps (**'time'**) dans Python est une bibliothèque intégrée qui fournit des fonctionnalités pour travailler avec des opérations liées au temps, y compris

l'obtention de l'heure actuelle, l'attente d'un certain temps et la conversion entre différents formats d'heure.

➤ **Module 'multiprocessing'** :

Le module de multitraitement (**'multiprocessing'**) de Python est une bibliothèque intégrée qui permet le traitement parallèle et l'exécution simultanée de plusieurs processus, fournissant des fonctionnalités pour la gestion des processus, la communication inter-processus, la mémoire partagée et la synchronisation.

➤ **Module 'threading'** :

Le package de threading en Python fournit une API de haut niveau pour créer et gérer des threads. Il est construit au-dessus du module `_thread` de niveau inférieur, qui fournit un accès de bas niveau aux primitives de threading fournies par le système d'exploitation. Un thread est un flux d'exécution séparé. Cela signifie que le programme aura deux choses qui se produiront en même temps. Ce module fournit un certain nombre de fonctionnalités qui facilitent la création et la gestion des threads avec le language Python.

En conclusion, Python est un langage de programmation puissant, polyvalent et facile à apprendre qui a gagné en popularité parmi les développeurs de tous niveaux. Sa simplicité, sa polyvalence, sa vaste gamme de bibliothèques et sa communauté active sont quelques-unes des raisons de son adoption généralisée. Avec sa popularité croissante, Python devrait continuer à dominer le paysage des langages de programmation dans les années à venir.

3.4.2 Présentation de la librairie TensorFlow

L'apprentissage automatique et en profondeur sont des domaines compliquées, mais heureusement, il existe des ressources qui simplifient son exécution. Une ressource pertinente à cet égard est le framework TensorFlow de Google. TensorFlow est une infrastructure d'apprentissage automatique et en profondeur open source largement utilisée par les développeurs et les spécialistes des données pour créer et déployer des modèles d'apprentissage en profondeur ou automatique. Il a été développé par l'équipe Google Brain et rendu public en 2015. Depuis lors, il est devenu l'un des frameworks d'apprentissage automatique les plus populaires, utilisé par des entreprises comme Airbnb, Intel et Uber. Dans cette partie, nous explorerons les avantages de l'utilisation de TensorFlow pour l'apprentissage automatique et en profondeur et pourquoi c'est un choix préféré parmi les développeurs et les scientifiques des données[54].

Figure 3.4.2 – La bibliothéque TensorFlow

3.4.2.1 Fonctionnement de TensorFlow

TensorFlow fonctionne en définissant, optimisant et exécutant des calculs mathématiques à l'aide de graphiques de flux de données .« dataflow graphs »

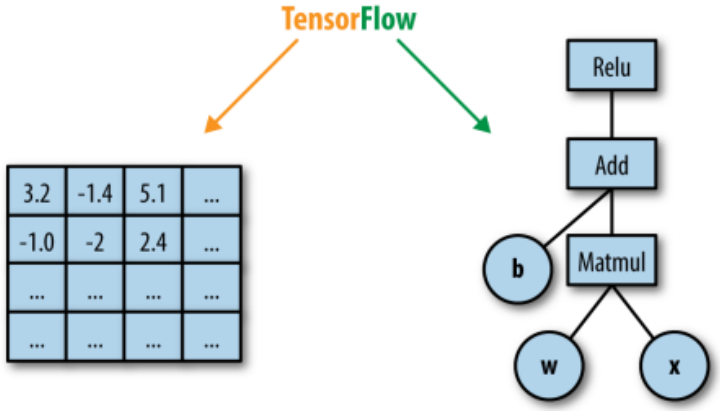


Figure 3.4.3 – Un graphe de calcul de flux de données. Les données sous forme de tenseurs circulent à travers un graphe d'opérations de calcul qui composent nos réseaux de neurones profonds.[30]

Les diagrammes de flux de données ce sont des constructions qui décrivent comment les données sont envoyées via un graphe ou un ensemble de nœuds de traitement. Chaque nœud du diagramme représente une opération mathématique. Chaque connexion entre les nœuds est un tableau multidimensionnel de données appelé **tenseur**. L'interaction avec TensorFlow se fait via le langage Python facile à apprendre et à utiliser. Le langage permet d'exprimer facilement la manière dont les abstractions de haut niveau sont combinées. Les nœuds et les tenseurs TensorFlow sont des objets Python. L'application TensorFlow elle-même est une application Python. Cependant, les opérations mathématiques elles-mêmes ne sont pas effectuées en Python, la bibliothèque de transformation accessible depuis TensorFlow est écrite sous la forme d'un binaire C++ hautes performances[30]. Le langage Python achemine simplement le trafic entre différents éléments et vous permet de les relier avec des abstractions de haut niveau[30]. Les programmes conçus avec TensorFlow

peuvent être exécutés sur diverses plateformes, telles qu'un ordinateur personnel, un cluster dans le Cloud, des appareils mobiles iOS ou Android, ainsi que des processeurs centraux et graphiques. Google offre également la possibilité d'utiliser ses propres puces TPU (Tensor Processing Unit) sur son Cloud pour bénéficier d'une accélération considérable[54].

3.4.2.2 C'est qoui un tenseur ?

Tous les calculs associés à TensorFlow impliquent l'utilisation de tenseurs. Un tenseur est un vecteur/matrice de n-dimensions représentant des types de données. Les valeurs d'un tenseur contiennent des types de données identiques avec une forme connue, et cette forme est la dimensionnalité de la matrice. Un vecteur est un tenseur unidimensionnel ; une matrice est un tenseur bidimensionnel. Un scalaire est un tenseur de dimension zéro.

Dans le graphe, les calculs sont rendus possibles grâce aux interconnexions des tenseurs. Les opérations mathématiques sont effectuées par le nœud du tenseur, tandis que l'arête d'un tenseur explique les relations entrée-sortie entre les nœuds. Ainsi, TensorFlow prend en entrée une matrice/tableau n-dimensionnelle (connue sous le nom de tenseurs), qui circule à travers un système de plusieurs opérations et ressort en sortie. D'où le nom TensorFlow. Un graphe peut être construit pour effectuer les opérations nécessaires à la sortie[30].

3.4.2.3 Avantages de TensorFlow

— **Facilité d'utilisation** :

TensorFlow est intuitif et convivial, ce qui facilite le développement de modèles d'apprentissage automatique pour les développeurs et les scientifiques axés sur les données. Il offre une architecture modulaire et adaptable qui permet aux utilisateurs de personnaliser leurs modèles selon leurs besoins spécifiques. TensorFlow prend également en charge plusieurs langages de programmation, notamment Python, Java et C++, ce qui facilite la participation d'un plus grand nombre de développeurs.

— **Évolutivité** :

TensorFlow est hautement évolutif, permettant aux utilisateurs de faire évoluer leurs modèles d'apprentissage automatique pour gérer de grands ensembles de données et des tâches gourmandes en ressources de calcul. Il peut fonctionner sur une seule machine, ainsi que sur des clusters de centaines ou de milliers de machines, ce qui le rend idéal pour créer des modèles qui nécessitent d'énormes quantités de données et de ressources informatiques. Cette évolutivité permet également des temps de

formation plus rapides et une meilleure précision du modèle.

— **Prise en charge d'un large éventail de plates-formes** :
TensorFlow est un cadre d'apprentissage automatique multiplateforme qui prend en charge divers systèmes d'exploitation, notamment Windows, Linux, macOS et des plates-formes mobiles et SBC, comme Android et iOS et Raspberry Pi. Cette polyvalence permet aux développeurs et aux scientifiques des données de créer et de déployer leurs modèles sur plusieurs appareils et plates-formes.

— **Vaste communauté et documentation complète** :
TensorFlow dispose d'une communauté vaste et active de développeurs et de scientifiques des données qui contribuent à son développement, en fournissant une assistance, des didacticiels et d'autres ressources. Le framework contient également une documentation complète, y compris des références d'API, des didacticiels et des exemples de code, ce qui permet aux utilisateurs de démarrer facilement et d'apprendre à l'utiliser efficacement.

TensorFlow est une infrastructure d'apprentissage automatique et en profondeur puissante et polyvalente qui offre plusieurs avantages par rapport aux autres infrastructures. Sa facilité d'utilisation, son évolutivité, sa prise en charge d'un large éventail de plates-formes et sa documentation complète en font un choix préféré parmi les développeurs et les scientifiques des données. Alors que le domaine de l'IA continue de se développer, TensorFlow est susceptible de rester à l'avant-garde des infrastructures d'apprentissage automatique et de DL, alimentant le développement d'applications innovantes et percutantes dans toute une gamme d'industries.

3.4.2.4 TensorFlow Lite

TensorFlow Lite est une version allégée du framework TensorFlow conçu pour exécuter des modèles d'apprentissage automatique et de DL sur des appareils mobiles et intégrés. Il a été publié par Google en 2017 et est depuis devenu un choix populaire parmi les développeurs pour créer des applications mobiles et IoT qui nécessitent de l'IA sur l'appareil[54].

Figure 3.4.4 – Logo de la bibliothéque TensorFlow Lite.

TensorFlow Lite offre plusieurs avantages par rapport à la version complète de TensorFlow, notamment :

— **Taille et empreinte mémoire réduites** :

Taille et empreinte mémoire réduites : TensorFlow Lite est optimisé pour les appareils mobiles et embarqués, avec une taille plus petite et une empreinte mémoire réduite. Cela le rend plus adapté à une exécution sur des appareils aux ressources limitées, tels que les smartphones, les appareils IoT et les microcontrôleurs.

— **Temps d'inférence plus rapides** :

TensorFlow Lite comprend une gamme d'optimisations qui lui permettent d'effectuer des inférences plus rapidement que la version complète de TensorFlow. Ces optimisations incluent la quantification, qui réduit la précision des paramètres du modèle, et la fusion du noyau, qui combine plusieurs opérations en un seul noyau pour une exécution plus rapide.

— **Prise en charge de l'accélération matérielle** :

TensorFlow Lite exploite les fonctionnalités d'accélération matérielle telles que les GPU, les DSP et les NPU pour accélérer l'exécution des modèles d'apprentissage automatique ou de DL sur les appareils mobiles et intégrés. Il prend également en charge les optimisations spécifiques au matériel telles que le Edge TPU.

— **Intégration facile avec les plates-formes mobiles et IoT** :

TensorFlow Lite prend en charge plusieurs plates-formes mobiles et IoT, notamment Android, iOS, Raspberry Pi et Arduino. Il comprend également des modèles prédéfinis pour des cas d'utilisation courants tels que la classification d'images, la détection d'objets et la reconnaissance vocale.

En résumé, TensorFlow Lite est une version allégée de TensorFlow conçue pour exécuter des modèles d'apprentissage automatique et en profondeur sur des appareils mobiles et intégrés. Sa taille plus petite, son empreinte mémoire réduite, ses temps d'inférence plus

rapides, sa prise en charge de l'accélération matérielle et sa facilité d'intégration avec les plates-formes mobiles et IoT en font un choix populaire parmi les développeurs pour la création d'applications d'apprentissage automatique sur appareil.

3.4.3 Présentation de la librairie OpenCV

OpenCV, connu sous le nom complet de Open Source Computer Vision Library, est une bibliothèque gratuite et open source offrant des fonctions de programmation pour la vision par ordinateur en temps réel. Développé à l'origine par Intel, maintenant maintenu par Willow Garage et Itseez[1]. Cet outil est disponible gratuitement depuis l'an 2000, d'abord sous licence BSD puis sous la licence Apache 2. OpenCV est utilisé dans une variété d'applications, y compris le traitement d'images, la vidéosurveillance, la robotique et l'imagerie médicale[1]. Il fournit un ensemble d'algorithmes qui permettent aux développeurs d'analyser et de manipuler des données visuelles, ce qui en fait un outil populaire dans tous les secteurs. OpenCV est open source, ce qui signifie qu'il est libre d'utiliser, de modifier et de distribuer[1]. OpenCV est écrit en C et C++ mais possède également des liaisons pour d'autres langages tels que Python, Java et MATLAB et d'autres langages et il est disponible pour plusieurs plates-formes, notamment Windows, Linux, macOS et Android.

Figure 3.4.5 – Logo de la bibliothéque OpenCV.

3.4.3.1 Avantages d'OpenCV

L'utilisation d'OpenCV présente de nombreux avantages tels que :

- Il est gratuit et open-source. Cela signifie qu'il est libre d'utiliser et de modifier, et il existe une grande communauté de développeurs qui y contribuent constamment.
- OpenCV est un choix populaire pour les développeurs en raison de sa capacité à fonctionner sur plusieurs plates-formes, notamment Windows, macOS, Linux, iOS et Android. Sa compatibilité multiplateforme permet aux développeurs de déployer leurs applications sur différents appareils[53].

- Il est puissant et polyvalent, OpenCV propose une grande variété de fonctions pour le traitement d'images et de vidéos en temps réel, telles que la détection et la reconnaissance d'objets ainsi que le filtrage d'images. Il est donc un outil très pratique pour les applications nécessitant un traitement d'image rapide[53].
- Il a une communauté nombreuse et active. Cela signifie qu'il existe de nombreuses ressources disponibles pour vous aider à apprendre à utiliser OpenCV, et vous pouvez obtenir de l'aide d'autres utilisateurs si vous en avez besoin.
- C'est bien documenté. La documentation OpenCV est complète et facile à comprendre.

3.4.3.2 Applications d'OpenCV

OpenCV a de nombreuses applications, notamment :

- **Robotique** :

  OpenCV est utilisé en robotique pour effectuer des tâches telles que la détection, le suivi et la localisation d'objets.

- **Automobile** :

  Il est utilisé dans l'industrie automobile pour assister les systèmes d'assistance à la conduite, y compris la détection des voies, la reconnaissance des panneaux de signalisation et la détection des piétons.

- **Médical** :

  Il est utilisé dans les applications d'imagerie médicale, y compris le traitement d'images par rayons X et IRM.

- **Divertissement** :

  Il est utilisé dans l'industrie du film et du jeu pour créer des effets spéciaux, la reconnaissance faciale et le suivi de mouvement.

En résumé, OpenCV est un outil puissant pour les applications de vision par ordinateur. Sa polyvalence et son accessibilité en font un choix populaire pour les développeurs dans tous les domaines, il continue d'évoluer, offrant aux développeurs un large éventail de fonctions et d'algorithmes de traitement d'images. En tant que tel, il restera un outil important pour la vision par ordinateur dans les années à venir.

3.4.4 Présentation de la librairie NumPy

NumPy est une bibliothèque Python gratuite et open source communément utilisée dans la recherche scientifique et technique. C'est la norme de facto pour travailler avec des données numériques en Python, et c'est le composant fondamental de tous les écosystèmes scientifiques Python.Et elle est open source et gratuite, qui facilite le calcul numérique avec Python. NumPy a été créé en 2005 par 'Travis Oliphant' en s'appuyant sur les premiers travaux des bibliothèques Numeric et Numarray[47]. Il est publié sous les termes libéraux de la licence BSD modifiée[46].

Figure 3.4.6 – Logo de la bibliothéque NumPy.

Le package est construit sur le langage de programmation Python et offre une abstraction de niveau supérieur pour la manipulation de tableaux, des matrices et d'autres structures de données numériques. NumPy est écrit partiellement en Python, mais la plupart des parties qui nécessitent un calcul rapide sont écrites en C ou C++ et les tableaux sont stockés à un endroit continu en mémoire contrairement aux listes, afin que les processus puissent y accéder et les manipuler très efficacement, ce qui rend cette bibliothèque très rapide pour les opérations mathématiques[47].

3.4.4.1 Avantages de NumPy

L'utilisation de NumPy présente de nombreux avantages tels que :

- Les opérations mathématiques sur les objets ndarray de NumPy sont jusqu'à 50 fois plus rapides que l'itération sur des listes Python natives à l'aide de boucles. Les gains d'efficacité sont principalement dus au fait que NumPy stocke les éléments du tableau dans un emplacement unique commandé dans la mémoire, éliminant les redondances en faisant en sorte que tous les éléments soient du même type et en utilisant les processeurs modernes.

- ➤ Il offre une syntaxe d'indexation pour accéder facilement à des portions de données dans un tableau.
- ➤ Il contient des fonctions mathématiques intégrées qui permettent la qualité de vie lorsque vous travaillez avec des tableaux et des mathématiques, telles que des fonctions pour l'algèbre linéaire, les transformations de tableaux et les matricielles.
- ➤ Il nécessite moins de lignes de code pour la plupart des opérations mathématiques que les listes Python natives.
- ➤ La documentation NumPy est complète et facile à comprendre.

En fin de compte, Numpy est un composant essentiel du calcul scientifique en Python qui fournit une structure de données de tableau puissante et polyvalente, des fonctions mathématiques efficaces et une variété d'outils pour l'analyse de données, l'IA et la science informatique. Investigation scientifique. Numpy a fondamentalement modifié la manière dont les calculs scientifiques sont effectués, permettant d'effectuer des calculs sur de grands volumes de données plus facilement que jamais. Numpy est crucial pour quiconque manipule des données numériques.

3.4.5 Présentation de la librairie Flask

Flask est un outil qui a gagné en popularité au cours de la dernière décennie. Flask est un micro framework Web écrit en Python qui permet aux développeurs de créer des applications Web avec un minimum de codage. Flask est un framework léger développé par 'Armin Ronacher' en 2010, qui fournit un large éventail de fonctionnalités et d'extensions. L'un des plus grands avantages de l'utilisation de Flask est sa simplicité[18]. Flask est très facile à utiliser et les développeurs peuvent commencer à créer leurs applications en quelques minutes. Flask offre une API simple et raffinée qui facilite la création et la personnalisation des applications pour les développeurs.

Figure 3.4.7 – Logo de la bibliothéque Flask.

3.4.5.1 Avantages de Flask

Voici quelques-uns des avantages de l'utilisation de Flask :

- ➤ Facile à apprendre et à utiliser.
- ➤ Flexible.
- ➤ Bien soutenu par de nombreuses organisations.
- ➤ Grande communauté de développeurs.
- ➤ Flask fournit également une excellente documentation, ce qui permet aux développeurs d'apprendre facilement le framework et de commencer à créer leurs applications.
- ➤ L'une des caractéristiques les plus remarquables de Flask est sa capacité à s'intégrer à d'autres technologies. Flask peut être facilement intégré à d'autres packages Python, tels que NumPy, Flask peut également être intégré à différents serveurs Web, tels qu'Apache et Nginx, ce qui permet aux développeurs de déployer leurs applications dans une variété d'environnements[24].

Dans l'ensemble, Flask est un excellent choix pour les développeurs qui souhaitent créer des applications Web rapidement et facilement. Il est facile à apprendre, à utiliser et à étendre. Il existe une grande communauté de développeurs Flask qui sont toujours prêts à aider. Et il existe de nombreuses extensions disponibles pour Flask qui ajoutent de nouvelles fonctionnalités.

3.4.6 Présentation de la librairie SQLite

SQLite ou SQLite3, c'est un moteur de base de données multiplateforme open source qui prend en charge les requêtes et les transactions SQL. Contrairement aux bases de

données traditionnelles, SQLite ne nécessite pas de serveur pour fonctionner, ce qui le rend idéal pour les applications qui ne nécessitent pas une infrastructure de base de données à grande échelle. SQLite est conçu pour être rapide et efficace, avec un faible encombrement de seulement quelques centaines de kilo-octets, ce qui le rend parfait pour les applications mobiles et embarquées[2]. SQLite est utilisé par les navigateurs Web populaires tels que Firefox et Chrome pour stocker les préférences, les signets et l'historique des utilisateurs.

Figure 3.4.8 – Logo de la bibliothéque SQLite.

La base de code SQLite est prise en charge par une équipe internationale de développeurs qui travaillent sur SQLite à plein temps. Les développeurs continuent d'étendre les capacités de SQLite et d'améliorer sa fiabilité et ses performances tout en maintenant la rétrocompatibilité avec la spécification d'interface publiée, la syntaxe SQL et le format de fichier de base de données[2]. Le code source est absolument gratuit pour quiconque le souhaite, mais un support professionnel est également disponible.

Le projet SQLite a été lancé en septembre année 2000[2]. L'avenir est toujours difficile à prédire, mais l'intention des développeurs est de prendre en charge SQLite jusqu'en 2050[2].

3.4.6.1 Fonctionnalités de SQLite

SQLite offre une multitude de fonctionnalités qui en font un excellent choix pour les petites et moyennes applications. Voici quelques-unes de ses principales caractéristiques :

- **Autonome** :

  SQLite est un moteur de base de données autonome qui ne nécessite aucune dépendance externe ni fichier de configuration. Cela facilite son installation et sa gestion, même pour les utilisateurs non techniques.

- **Sans serveur** :

  SQLite ne nécessite pas de serveur séparé pour fonctionner, ce qui le rend idéal

pour les applications qui ne nécessitent pas une infrastructure de base de données à grande échelle. Il peut fonctionner sur n'importe quelle plate-forme prenant en charge le langage de programmation C, y compris Windows, Mac, Linux et les appareils mobiles et embarqués.

- **Léger** :
  SQLite est un moteur de base de données petit, rapide et efficace qui a un faible encombrement. Il peut gérer des bases de données de plusieurs gigaoctets sans impact significatif sur les performances.
- **Multiplateforme** :
  SQLite est un moteur de base de données multiplateforme qui peut s'exécuter sur n'importe quelle plate-forme prenant en charge le langage de programmation C. Cela facilite le portage d'applications sur différentes plates-formes sans aucune modification de la base de code.

3.4.6.2 Avantages de SQLite

SQLite3 offre plusieurs avantages par rapport aux bases de données traditionnelles comme Oracle et MySQL. Voici quelques-uns de ses principaux avantages :

- **Facile à utiliser** :
  SQLite est facile à utiliser, même pour les utilisateurs non techniques. Il ne nécessite présque aucune configuration et peut être géré à l'aide de simples commandes SQL.
- **Rapide et efficace** :
  SQLite est un moteur de base de données rapide et efficace qui peut gérer de grands ensembles de données sans aucun impact significatif sur les performances. Il est conçu pour être rapide, avec une faible empreinte mémoire, ce qui le rend idéal pour les applications mobiles et embarquées.
- **Portable** : SQLite est un moteur de base de données portable qui peut être facilement porté sur différentes plates-formes sans aucune modification de la base de code. Cela facilite le développement d'applications pouvant s'exécuter sur plusieurs plates-formes.
- **Faible maintenance** : SQLite est un moteur de base de données nécessitant peu de maintenance qui ne nécessite aucune procédure de sauvegarde ou de récupération complexe. Il est conçu pour être fiable et facile à gérer, même pour les utilisateurs non techniques.

En somme, SQLite3 est un moteur de base de données exceptionnel qui peut être utilisé dans diverses applications. Ses fonctionnalités et avantages en font un choix incontournable pour les développeurs qui ont besoin d'un moteur de base de données rapide, fiable et facile à utiliser. En utilisant SQLite3, les développeurs peuvent économiser du temps et des efforts tout en fournissant un moteur de base de données hautes performances à leurs applications.

3.4.7 Présentation de la librairie Pillow

Pillow est une bibliothèque d'imagerie Python open source qui fournit aux développeurs des outils pour travailler avec des images. La bibliothèque Python Pillow est un fork d'une ancienne bibliothèque appelée PIL[51]. PIL signifie Python Imaging Library, et c'est la bibliothèque originale qui a permis à Python de traiter les images. PIL a été abandonné en 2011 et ne prend en charge que Python 2[51]. Pour utiliser la propre description de ses développeurs, Pillow est le fork PIL convivial qui a maintenu la bibliothèque en vie et inclut la prise en charge de Python 3, en plus a été créé pour ajouter plus de fonctionnalités, améliorer les performances et fournir un meilleur support pour les nouvelles versions de Python.

Figure 3.4.9 – Logo de la bibliothéque Pillow.

3.4.7.1 Fonctionnalités de Pillow

Pillow offre une large gamme de fonctionnalités qui permettent aux développeurs de manipuler et d'améliorer les images. Certaines des fonctionnalités clés incluent :

- **Traitement d'image** :

  Pillow peut lire, manipuler et enregistrer des images dans différents formats, notamment JPEG, PNG, BMP, GIF, TIFF et WebP.

- **Filtrage d'image** :

  Pillow dispose d'une large gamme de filtres d'image qui peuvent être utilisés pour améliorer les images, y compris le flou, la détection des contours et la netteté.

- **Transformation d'image** :

  Pillow peut transformer les images de différentes manières, telles que le recadrage, le redimensionnement et la rotation.

- **Améliorations de l'image** :

  Pillow propose une gamme d'outils d'amélioration de l'image, tels que le réglage de la luminosité et du contraste, la balance des couleurs et l'égalisation.

En résumé, Pillow est un outil essentiel pour les développeurs qui travaillent avec des images en Python. Il offre une gamme de fonctionnalités pour les tâches de traitement d'image et fournit une interface facile à utiliser qui simplifie le processus de développement. Avec Pillow, les développeurs peuvent manipuler et améliorer les images de différentes manières, ce qui en fait un choix idéal pour le traitement d'image.

3.5 Conclusion

En conclusion, ce chapitre a présenté en détail les différents composants matériels et logiciels nécessaires à la réalisation de notre projet. Nous avons décrit les spécifications matérielles nécessaires, en soulignant les avantages de chaque choix de modèle pour chaque composant. Nous avons également discuté des composants logiciels du projet, en discutant des différentes bibliothèques et frameworks utilisés, ainsi que du langage de programmation choisi. À travers cette présentation, nous avons fourni toutes les informations pertinentes concernant l'assemblage et l'installation des différentes composantes, ce qui facilitera le fonctionnement complet du système. Dans le prochain chapitre, nous allons passer à la phase de réalisation du projet, en utilisant les éléments présentés dans ce chapitre comme base solide pour la mise en œuvre de notre système.

CHAPITRE 4

Réalisation du système

4.1 Introduction

Une fois que nous avons défini et décrit les différentes composantes logicielles et matérielles qui constituent notre système, nous allons maintenant détailler les étapes de sa réalisation pratique. Dans ce dernier chapitre, nous présenterons en détail les étapes de préparation du système et du programme d'application que nous avons réalisées pour la détection des personnes et leur comptage à l'aide d'un algorithme de suivi d'objet basé sur la détection. Nous commencerons par introduire le système à l'aide d'un schéma fonctionnel et d'un organigramme afin d'obtenir une meilleure visualisation de la structure et du fonctionnement du processus. Enfin, nous conclurons ce chapitre avec des tests pratiques qui illustreront les principes et le fonctionnement de notre système de détection de personnes.

4.2 Conception détaillée du système

Ce système choisit Raspberry Pi comme appareil de calcul principal, en raison de sa taille, qui est suffisamment petite pour être installée dans presque n'importe quel endroit. Le logiciel comprend la base de données SQLite pour stocker les données et un site Web ainsi qu'un écran pour afficher des informations sur l'état de la surveillance, comme les personnes capturées dans la vidéo et leurs horodatages avec un identifiant. Le Raspberry Pi 3B +, la caméra Pi et TensorFlow lite sont utilisés pour la détection d'objets, tandis que les bibliothèques OpenCV et NumPy sont utilisées pour le traitement d'images et de vidéos. La bibliothèque Flask est utilisée pour afficher le flux de surveillance sur la page Web et également pour extraire des données de la base de données afin d'afficher plus de détails sur les données de détection dans un tableau. L'architecture du système est illustrée à la figure 4.2.1.

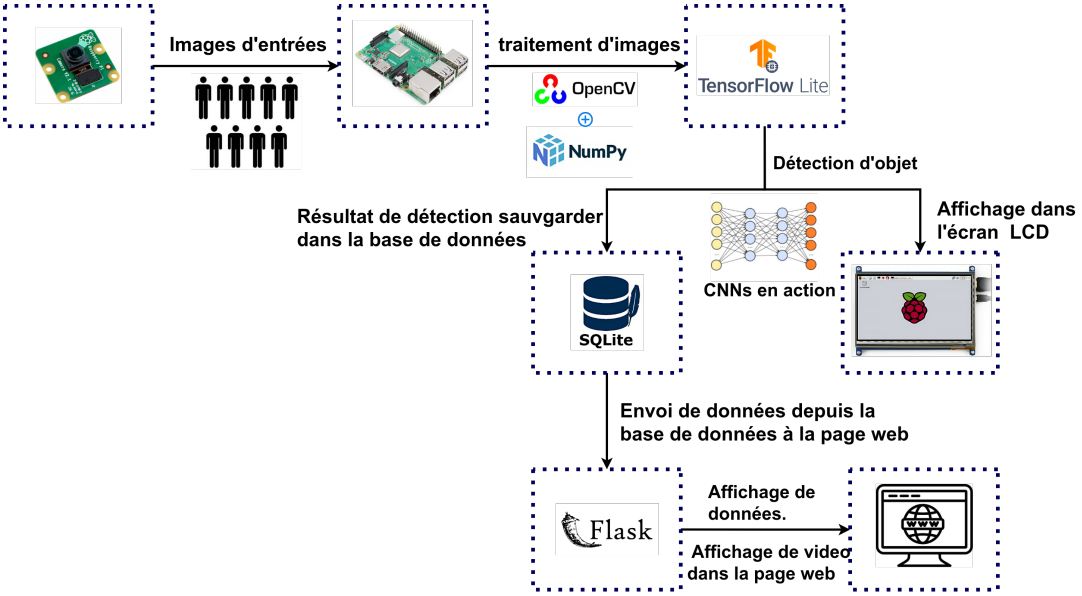


Figure 4.2.1 – Schéma bloc de système de détection.

Pour illustrer davantage le système, un organigramme est présenté à la figure 4.2.2.

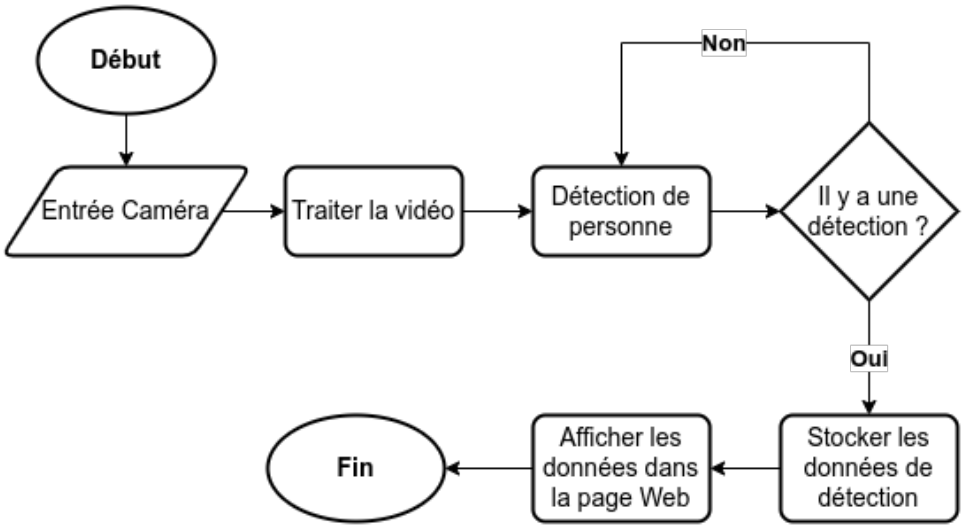


Figure 4.2.2 – L'organigramme de système de détection.

L'organigramme part de la caméra Pi pour récupérer l'image vidéo qui l'alimente vers le Raspberry Pi, où la détection d'objet est déployée à l'aide de TensorFlow lite. Lorsqu'une personne passe devant la caméra, celle-ci déclenche TensorFlow lite (TF lite) pour détecter et classer l'objet. Une fois la personne classée, le résultat est envoyé à la base de données. Si TF lite n'a pas encore classé la personne, il effectuera automatiquement une nouvelle analyse des personnes dans chaque image de la vidéo, jusqu'à ce qu'il puisse classer une personne, puis les données de détection sont stockées dans une base de données et les données de détection sont affichées dans la page Web ainsi que la vidéosurveillance elle-même.

Dans ce qui suit, nous allons détailler la manière ont été programmés les différents blocs

afin que tout soit synchronisé avec chaque composante du système, mais avant cela, on commencera par la phase de préparation.

4.3 Préparation du système

Dans cette partie, en va préparer le système pour commencer à développer notre application de détection de personnes. Tout d'abord, on va installer et configurer le système d'exploitation '**Raspberry Pi OS**'. Ensuite, on va configurer la caméra Pi et l'écran TFT pour qu'on puisse les utiliser avec la Raspberry Pi. En fin, on va installer les bibliothèques nécessaires pour la mise en ouvre de notre application.

4.3.1 Installation et configuration de système d'exploitation

Raspberry Pi OS est la distribution officielle de Raspberry Pi. Anciennement appelé « Raspbian », il a été rebaptisé « Raspberry Pi OS » en 2020. Il s'agit d'un système d'exploitation de type Unix basé sur la distribution Debian Linux pour la famille Raspberry Pi. Il a été développé pour la première fois de manière indépendante en 2012 et est produit comme système d'exploitation principal pour ces cartes depuis 2013, distribué par la Raspberry Pi Foundation.

Figure 4.3.1 – Logo de Raspberry Pi OS.

Raspberry Pi OS suit les noms de version de Debian, donc la dernière version disponible est Raspberry Pi OS Bullseye (Debian 11). Il est hautement optimisé pour Raspberry Pi

avec processeur ARM et fonctionnera sur tous les Raspberry Pi sauf les microcontrôleurs Pico. Il existe d'autres systèmes d'exploitation disponibles pour Raspberry Pi.

4.3.1.1 Installation sur la carte SD

Raspberry Pi Imager est un outil créé par la Fondation Raspberry Pi pour installer de nouveaux systèmes sur l'ordinateur Raspberry Pi. Les cartes SD et les clés USB sont prises en charge, ainsi que les systèmes d'exploitation les plus populaires, alors la Raspberry Pi Imager détecte automatiquement la carte SD pour installer l'OS. Il est également possible de flasher n'importe quelle image personnalisée pour la Raspberry Pi. Une fois Raspberry Pi Imager installé, il peut être utilisé pour flasher n'importe quel système d'exploitation sur une carte SD ou une clé USB. Certains systèmes d'exploitation sont directement pris en charge et peuvent être installés en un clic. Et les autres systèmes d'exploitation peuvent être flashés en utilisant une image personnalisée. Pour installer un système d'exploitation sur une carte microSD, le Raspberry Pi Imager est un outil simple et rapide qui permet de télécharger et d'écrire le fichier image. Il y a trois étapes principales :

— Choisissez le système d'exploitation que vous souhaitez installer.

— Choisissez la carte SD ou le périphérique USB à utiliser.

— Commencez le processus d'écriture.

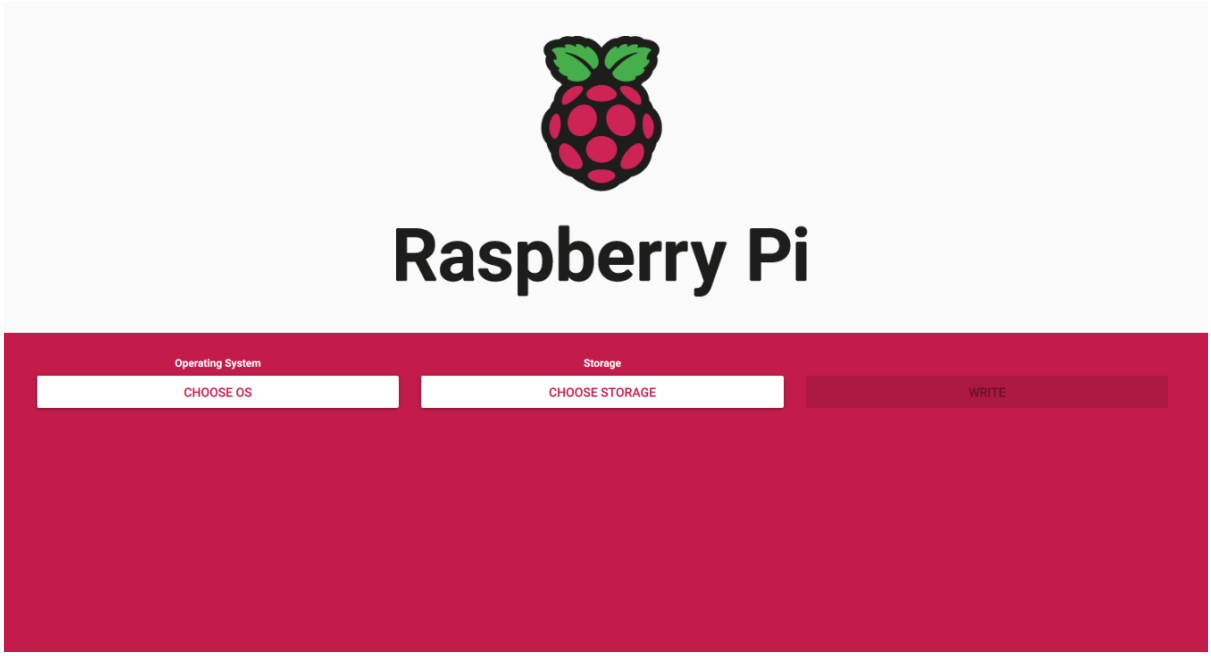


Figure 4.3.2 – Le Raspberry Pi Imager.

Premiérement, Vous devez cliquer sur "Choose OS" et choisir la version que vous souhaitez installer sur la Raspberry Pi. La première option disponible est la version Desktop et c'est

la version recommandée a installé, et vous trouverez les alternatives en cliquant sur le deuxième élément de la liste : Raspberry Pi (other).

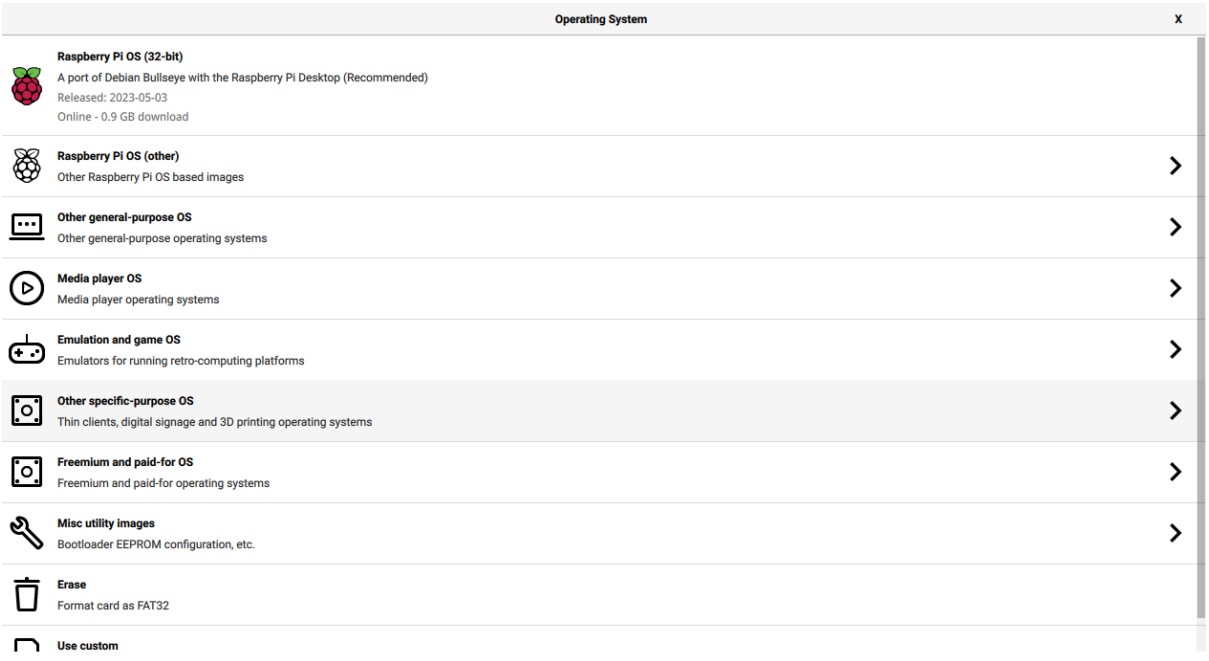


Figure 4.3.3 – Choix d'options d'images dans le Imageur.

Une fois cela fait, vous pouvez choisir la carte SD, en cliquant sur "Choose SD CARD" (en général, vous n'aurez qu'un seul choix).

Figure 4.3.4 – Choix d'options de cartes SD ou USB dans le Imageur.

Quand la carte SD est choisi, maintenant, il faut cliquer sur 'Write' pour démarrer l'installation :

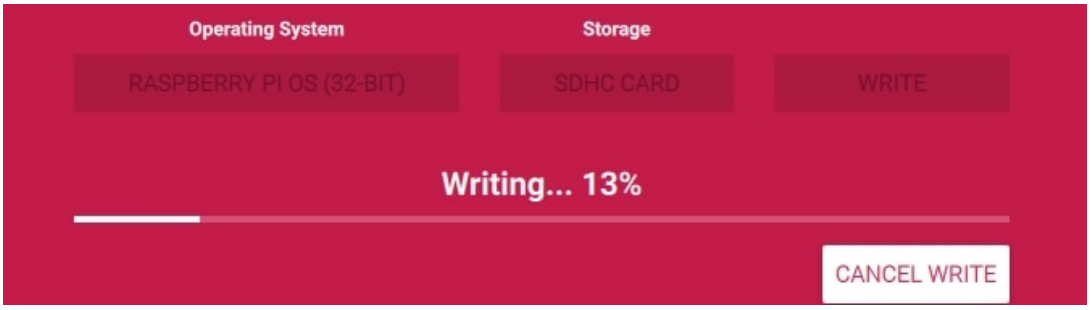


Figure 4.3.5 – Choix d'options d'écréture dans le Imageur.

L'installation prendra quelques minutes. Cela peut prendre plus de temps si vous flashez une image spécifique pour la première fois ou si vous avez une connexion Internet

lente. En effet, Raspberry Imager téléchargera d'abord l'image depuis l'internet, puis la flashera sur le stockage de la carte SD.

4.3.1.2 Premier démarrage sur Raspberry Pi OS

Maintenant, il faut obtenir la carte SD et l'insérer dans la Raspberry Pi, Ensuite, il faut démarrer la Raspberry Pi, avec un écran et un clavier branchés et bien sûr, le RPi est alimenté par une alimentation CC 5V/3A.

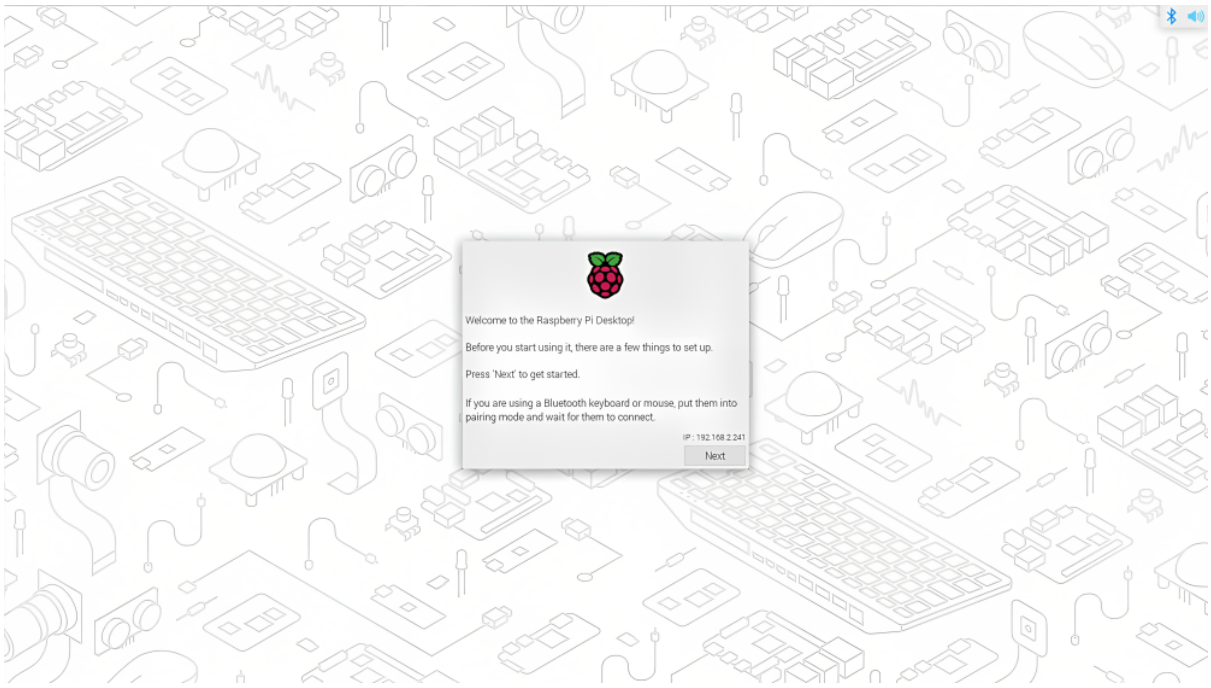


Figure 4.3.6 – Premier démarrage sur Raspberry Pi OS version Bulleye.

Maintenant, il faut simplement suivre l'assistant pour configurer les paramètres principaux, en cliquant sur '**Next**' :

1. Changement de mot de passe et de nom d'utilisateur.

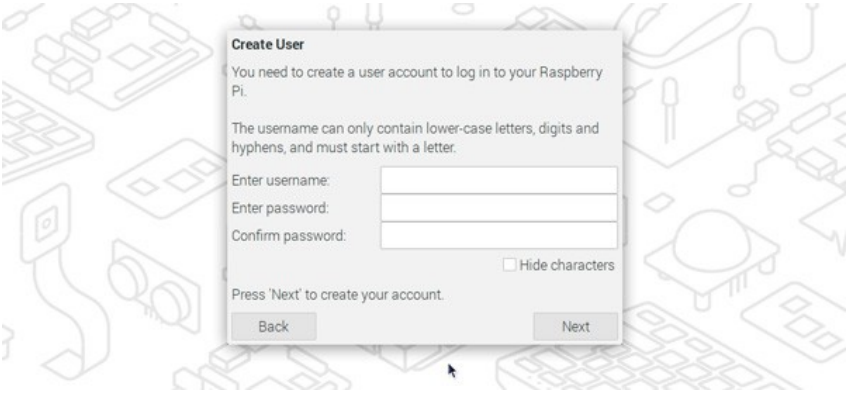


Figure 4.3.7 – Insertion de nouveau nom d'utilisateur et de nouveau mot de passe.

2. Sélectionnez le pays et modifiez la langue par défaut et le layout de clavier.

Figure 4.3.8 – la configuration de la langue et le clavier.

3. Connection au réseau Wi-Fi si nécessaire.
4. Démarrage des mises à jour du système par la commande :
   `sudo apt update && sudo apt full-upgarde`.

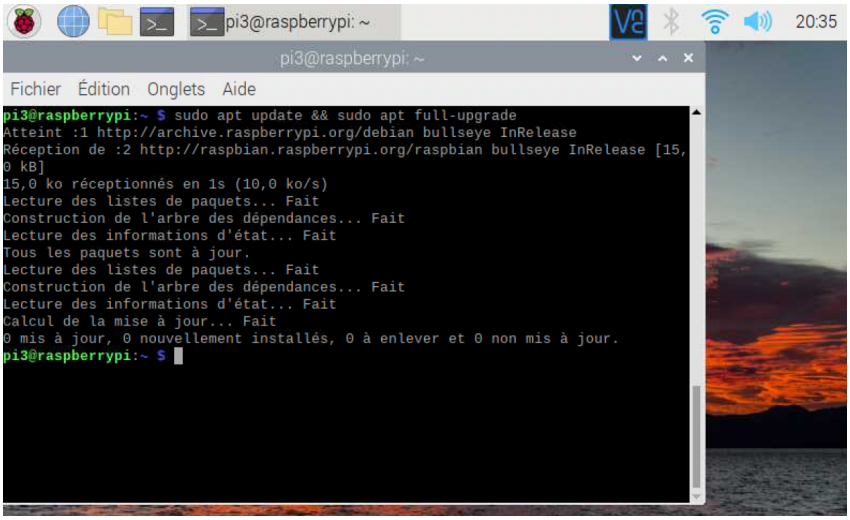


Figure 4.3.9 – La mise à jour complete du systéme.

il faut attendre la fin des mises à jour et redémarrez la Raspberry Pi pour commencer à utiliser la Raspberry Pi.

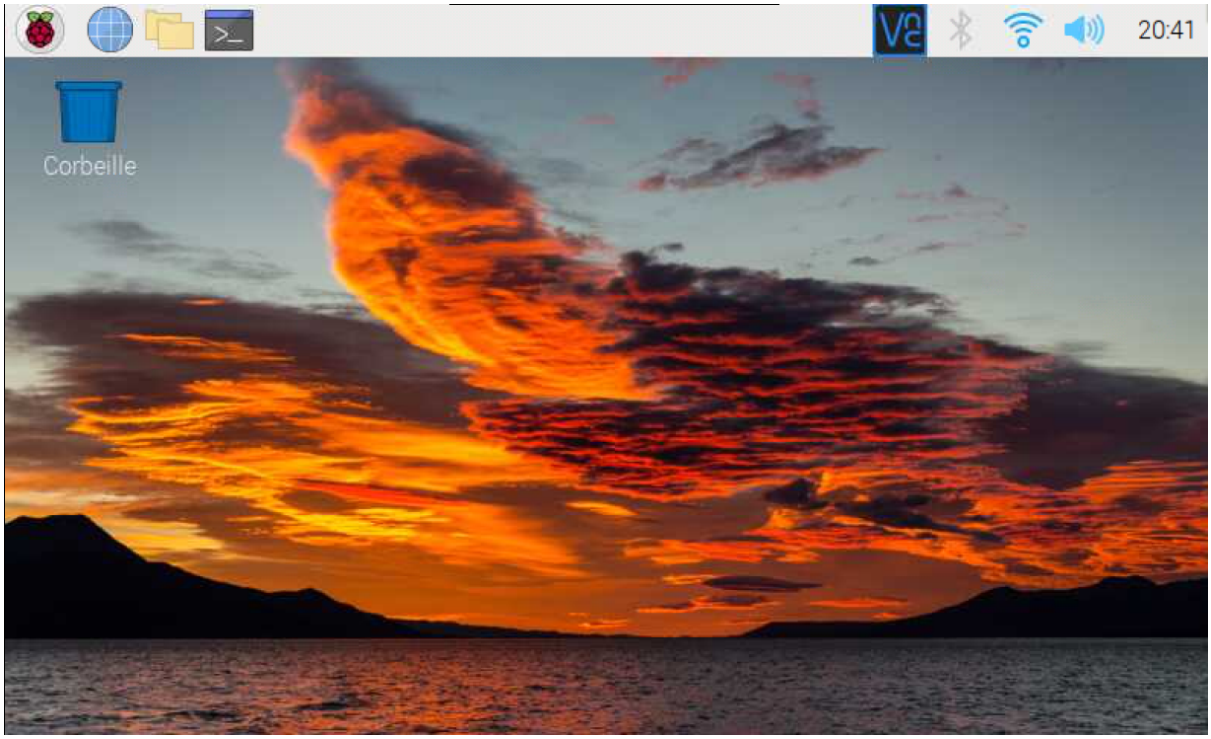


Figure 4.3.10 – Le systéme d'éxploitation Raspberry Pi OS Bullseye.

4.3.1.3 Configuration de la caméra et de l'écran TFT LCD

1. **Configuration de la caméra** :

   Avant d'utiliser le module caméra, il faut d'abord l'activer. Il faut ouvrir un terminal et exécuté la commande '`raspi-config`' pour entrer dans la configuration du système d'exploitation, puis rendez-vous dans **"Interface options"** ensuite **"Camera"** et activez le module caméra. comme illustré dans la figure 4.3.11 :

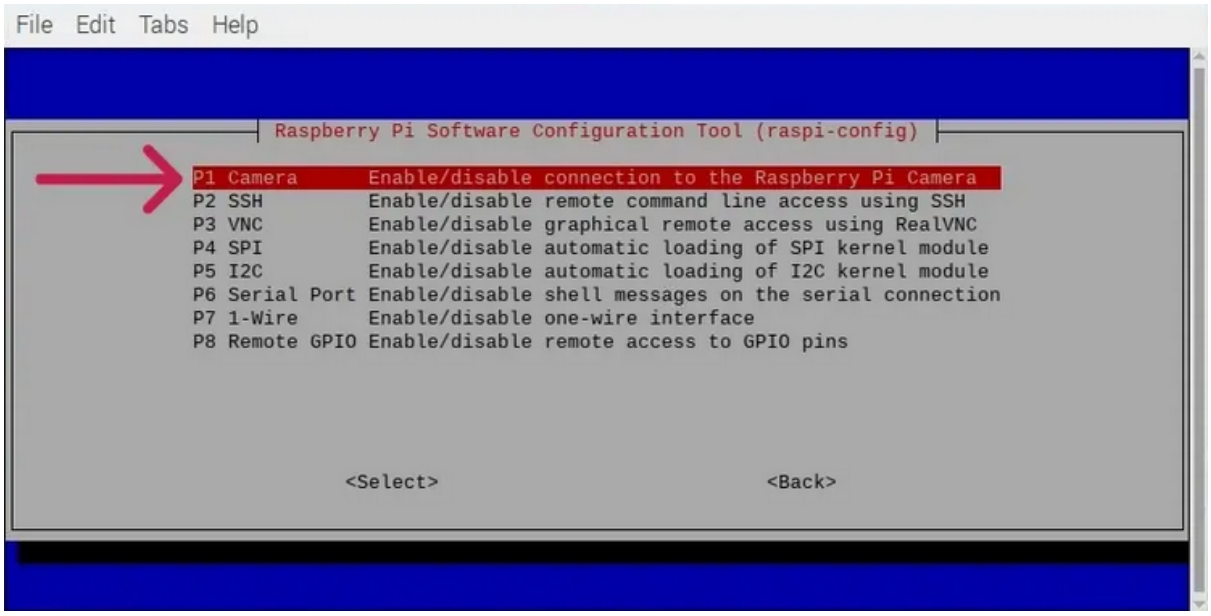


Figure 4.3.11 – Menu de configuration : Partie Interfaces.

Après l'activation de la caméra, il faut redémarrer la Raspberry Pi, et après le redémarrage, la caméra est prête à être utilisée.

2. **Configuration de l'écran TFT LCD** :

   Maintenant, nous devons configurer le Raspberry Pi pour détecter l'écran LCD et également pour reconnaître sa résolution, nous devons donc modifier le fichier de configuration du Raspberry Pi, Le fichier de configuration peut être modifier en utilisant la commande `sudo nvim /boot/config.txt`[1]. Et pour configurer l'écran TFT LCD, il faut simplement ajouter les commandes suivantes dans le fichier *config.txt* :

```
max_usb_current=1
hdmi_group=2
hdmi_mode=87
hdmi_cvt 800 480 60 6 0 0 0
hdmi_drive=1
hdmi_force_hotplug=1
```

   - `max_usb_current=1` : cette commande augmente le courant maximal pouvant être tiré des ports USB du Raspberry Pi.
   - `hdmi_group=2` : cette commande définit le mode HDMI sur le groupe 2. Le groupe 2 est utilisé pour les écrans avec une résolution de 720p ou supérieure.
   - `hdmi_mode=87` : cette commande définit le mode HDMI sur 87. Le mode 87 est utilisé pour les écrans avec une résolution de 800x480.
   - `hdmi_cvt 800 480 60 6 0 0 0` : Cette commande définit les paramètres HDMI pour l'affichage. Les deux premiers nombres (800 et 480) sont la largeur et la hauteur de l'affichage en pixels. Le troisième nombre (60) est le taux de rafraîchissement en Hz. Les nombres restants sont des paramètres de synchronisation utilisés pour générer le signal HDMI.
   - `hdmi_force_hotplug=1` : forcer le Raspberry Pi à allumer la sortie HDMI même si aucun moniteur n'est détecté (peut être nécessaire pour le premier démarrage sur l'écran LCD).

3. **Connexion de l'écran TFT LCD avec la Raspberry Pi** :

1. pour exécuter 'nvim' il faut tout d'abord l'installer avec la commande "sudo apt install -y neovim"

La première étape pour interfacer l'écran LCD avec Raspberry Pi consiste à connecter l'écran LCD avec pi à l'aide des connecteurs GPIO. L'écran est livré avec quatre entretoises d'angle pour le montage. Ensuite, l'écran LCD est placé au-dessus de la Raspberry Pi de manière à ce qu'il glisse dans les emplacements GPIO et que les ports HDMI s'alignent parfaitement de l'autre côté. l'écran LCD n'utilise que 26 broches pour les connexions avec le RPi, donc il faut être prudent lors de la connexion.

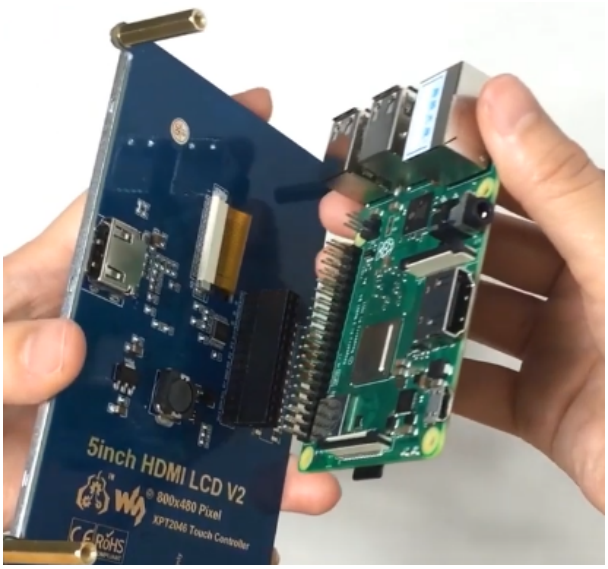


Figure 4.3.12 – Connection de l'écran avec les broches GPIOs.

Après avoir placé l'écran LCD sur le dessus de la Raspberry Pi, il faut connecter le connecteur HDMI fourni avec l'écran, entre le Pi et l'écran LCD, comme indiqué sur la figure ci-dessous :

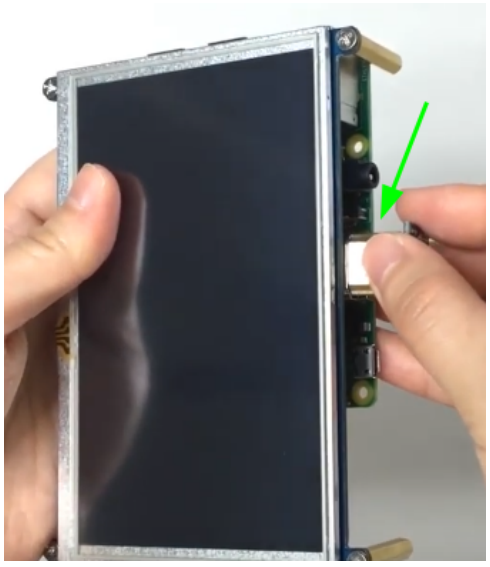


Figure 4.3.13 – Connection de l'écran avec le RPi à travers la connexion HDMI.

Après toutes les étapes précédentes, nous devons alimenter Raspberry Pi et LCD. Il y

a aussi une prise micro USB sur l'écran LCD pour donner une source d'alimentation séparée, mais tant que nous avons une bonne alimentation de 3,0 A pour notre RPi, il n'a pas besoin d'une alimentation séparée pour l'écran LCD.

Figure 4.3.14 – L'écran LCD avec la RPi alimenté par une alimentation 5V.

4.3.1.4 Configuration de l'accès à distance

Vous pouvez vous connecter à un Raspberry Pi à partir d'une autre machine pour quelque raison que ce soit. Mais pour ce faire, nous aurons besoin de connaître son adresse IP. Tout appareil connecté à un réseau local se voit attribuer une adresse IP. Afin de connecter le Raspberry Pi à partir d'une autre machine en utilisant SSH ou VNC, nous devons connaître l'adresse IP du Raspberry Pi. C'est facile si vous avez un écran connecté et qu'il existe plusieurs méthodes pour le trouver à distance à partir d'une autre machine du réseau, l'une de ces méthodes consiste à utiliser la commande ping, comme illustré dans la figure suivante :

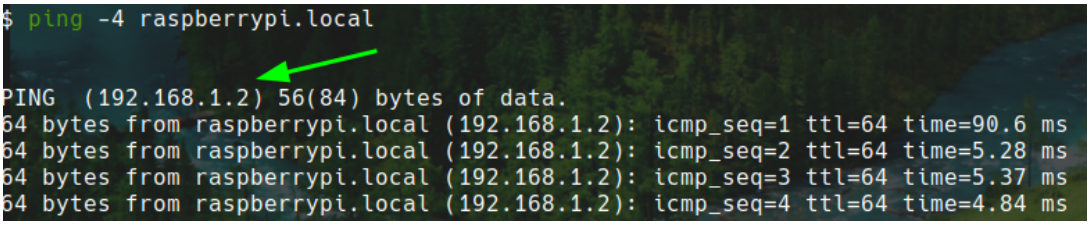


Figure 4.3.15 – L'adresse IP acutelle de notre Rapsberry Pi.

Sur Raspberry Pi OS, le DNS multicast est pris en charge par le service Avahi. Si votre appareil prend en charge mDNS, vous pouvez accéder à votre Raspberry Pi en utilisant son nom d'hôte et le suffixe `.local`. Le nom d'hôte par défaut sur une nouvelle installation de Raspberry Pi OS est '`raspberrypi`', donc par défaut, tout Raspberry Pi

exécutant Raspberry Pi OS répond à la commande indiquée dans la figure précédente. et en exécutant cette commande, on voit que l'adresse IPv4 de notre Raspberry Pi est : `192.168.1.2`

— **Configuration de serveur SSH** :

Il possible d'accéder à distance à la ligne de commande d'un Raspberry Pi depuis un autre ordinateur ou appareil sur le même réseau en utilisant le protocole Secure Shell (SSH). Raspberry Pi OS a le serveur SSH désactivé par défaut. Il peut être activé manuellement depuis le bureau, nous devrions lancer la configuration de Raspberry Pi à partir du menu Préférences, et naviguer vers l'onglet Interfaces et activer le SSH/VNC, comme le montre la figure suivante :

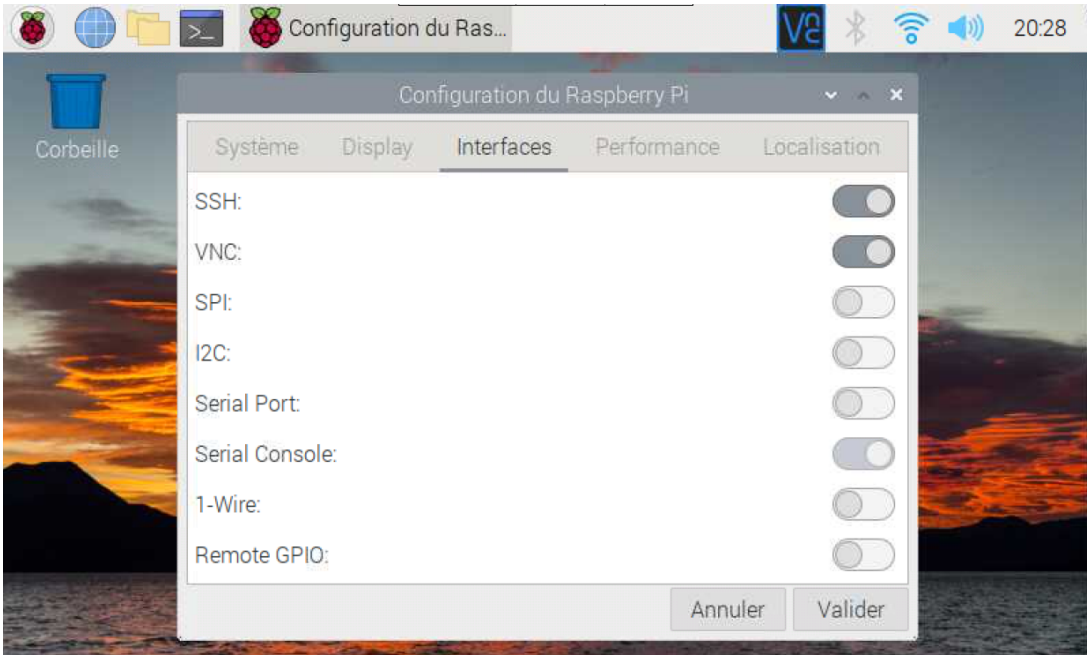


Figure 4.3.16 – Activation de SSH et VNC dans la Rapsberry Pi.

nous pouvons utiliser SSH pour nous connecter à notre Raspberry Pi à partir d'un bureau Linux, d'un autre Raspberry Pi ou d'un Apple Mac sans installer de logiciel supplémentaire. J'ai une machine Linux donc j'ai simplement besoin d'ouvrir une fenêtre de terminal sur mon ordinateur et de taper `ssh non_utilisateur@<adress IP>`, alors dans notre cas en exécute la commande comme illustré dans la figure 4.3.17.

Figure 4.3.17 – Connexion entre mon PC et mon Rapsberry Pi à travers SSH.

— **Configuration de serveur VNC** :

VNC (Virtual Network Computing) est un système de partage de bureau graphique qui vous permet de contrôler à distance l'interface de bureau d'un ordinateur (exécutant le serveur VNC) à partir d'un autre ordinateur ou appareil mobile (exécutant VNC Viewer). VNC Viewer transmet les événements du clavier et de la souris au serveur VNC et reçoit en retour les mises à jour de l'écran. Vous verrez le bureau du Raspberry Pi dans une fenêtre de votre ordinateur ou de votre appareil mobile. Vous pourrez le contrôler comme si vous travailliez sur le Raspberry Pi lui-même. nous avons déjà activé VNC Server, nous pouvons donc l'utiliser maintenant, VNC Server nous donne un accès à distance au bureau graphique qui s'exécute sur le Raspberry Pi, comme si nous étions assis devant. VNC Connect de RealVNC est inclus avec Raspberry Pi OS qui nous donne accès au serveur VNC et nous permet de contrôler notre Raspberry Pi à distance. Le visualiseur RealVNC a été aussi installé sur mon PC, pour pouvoir se connecter avec la Raspberry Pi, et il est configuré comme illustré dans la figure ci-dessous :

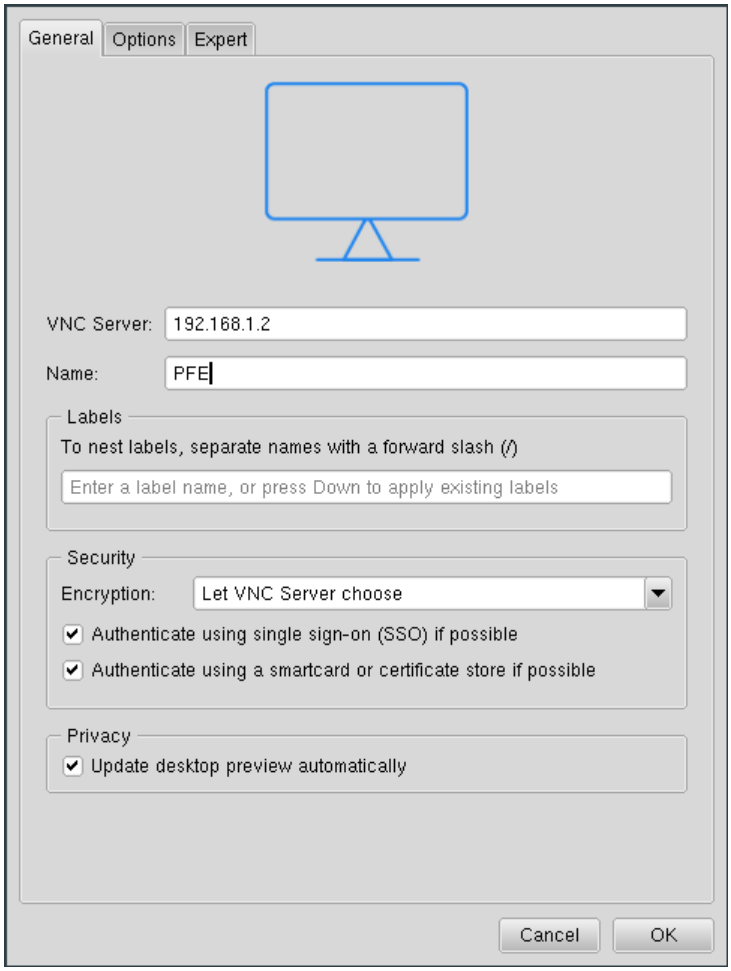


Figure 4.3.18 – Configuration de VNC sur mon PC.

Ensuite, vous pouvez simplement démarrer la connexion en double-cliquant sur le raccourci.

Figure 4.3.19 – L'icon de launcement de VNC sur le PC.

Pour la première fois, vous pouvez avoir un avertissement de sécurité à accepter (la connexion n'est pas cryptée avec VNC). Nous devrons également fournir le nom d'utilisateur et le mot de passe de notre Raspberry Pi. Et c'est tout. Après cela, nous

aurons accès à distance à l'environnement de bureau complet de notre Raspberry Pi.

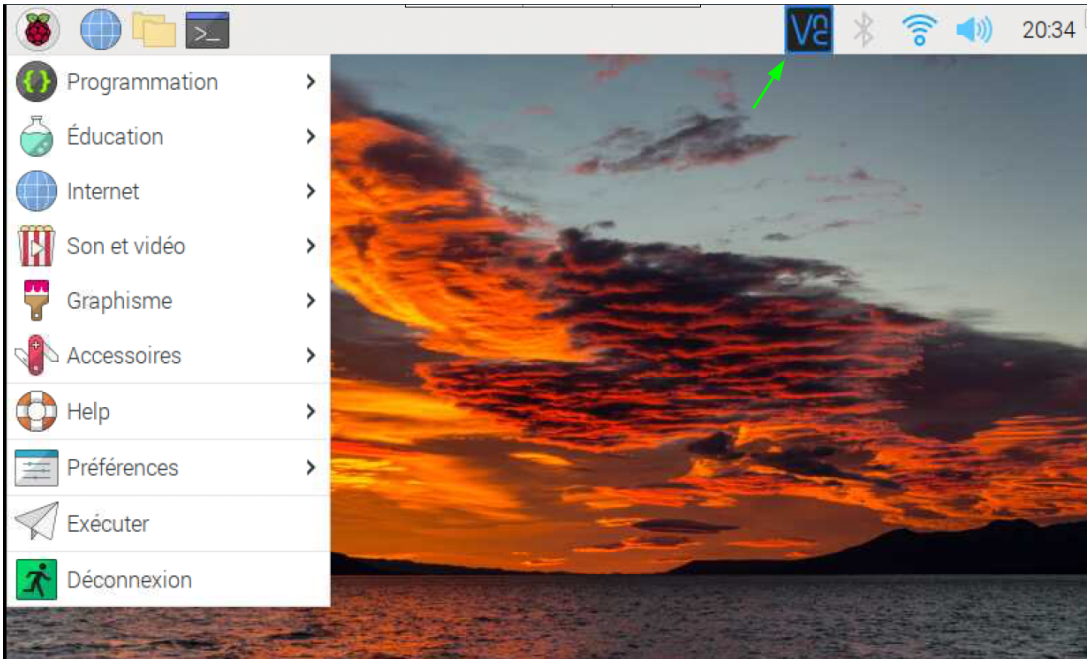


Figure 4.3.20 – Launcement de l'application RealVNC sur mon PC.

L'icône noire aux bordures bleues nous indique que nous sommes connectés à un serveur VNC et qu'il fonctionne.

4.3.1.5 Installation des bibliothèques externe

Python est déjà installé sur Raspberry Pi, mais son utilisation dans notre projet nécessite certaines dépendances externes. La plupart des packages Python pour Raspberry Pi sont disponibles dans les référentiels APT. Un autre outil nommé "PIP" peut également être utilisé pour certains autres modules non inclus dans les référentiels par défaut. PIP est un gestionnaire de packages pour les packages Python, il est préinstallé sur Raspberry Pi OS et permet d'installer des bibliothèques qui ne sont pas disponibles dans les référentiels par défaut, nous l'utiliserons donc ce gestionnaire de packages pour installer les bibliothèques dont nous avons besoin pour le projet, comme illustré dans la figure 4.3.21.

```
pi3@raspberrypi:~ $ pip install opencv-python tflite-runtime numpy pillow flask
Defaulting to user installation because normal site-packages is not writeable
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: opencv-python in ./.local/lib/python3.9/site-packages (3.4.18.65)
Requirement already satisfied: tflite-runtime in ./.local/lib/python3.9/site-packages (2.11.0)
Requirement already satisfied: numpy in ./.local/lib/python3.9/site-packages (1.24.2)
Requirement already satisfied: pillow in /usr/lib/python3/dist-packages (8.1.2)
Requirement already satisfied: flask in ./.local/lib/python3.9/site-packages (2.2.3)
Requirement already satisfied: Werkzeug>=2.2.2 in ./.local/lib/python3.9/site-packages (from flask) (2.2.3)
Requirement already satisfied: Jinja2>=3.0 in /usr/local/lib/python3.9/dist-packages (from flask) (3.1.2)
Requirement already satisfied: itsdangerous>=2.0 in ./.local/lib/python3.9/site-packages (from flask) (2.1.2)
Requirement already satisfied: click>=8.0 in ./.local/lib/python3.9/site-packages (from flask) (8.1.3)
Requirement already satisfied: importlib-metadata>=3.6.0 in /usr/local/lib/python3.9/dist-packages (from flask) (6.0.0)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.9/dist-packages (from importlib-metadata>=3.6.0->flask) (3.13.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from Jinja2>=3.0->flask) (2.1.2)
pi3@raspberrypi:~ $
```

Figure 4.3.21 – Installation des bibliothèques externe avec le gestionnaire PIP [1].

1. Ces packages ont déjà été installés, c'est pourquoi nous avons ce message : Exigence déjà satisfaite

il n'est pas nécessaire d'installer le module SQLite séparément car il est livré par défaut avec Python, il s'agit donc d'un package intégré.

4.4 Déploiement de l'application

Il faut maintenant écrire le code pour l'application de détection d'objets à l'aide de TensorFlow Lite. Le code doit charger le modèle tflite et les données d'entrée, exécuter l'inférence et afficher les résultats de la détection d'objets. Le code est écrit en Python.

4.4.1 Importation des bibliothèques

Cette section est le début de notre programme python et est destinée à déclarer toutes les bibliothèques dont notre système a besoin pour fonctionner correctement.

```python
from tflite_runtime.interpreter import Interpreter
import os
import cv2
import numpy as np
from datetime import datetime
from centroidtracker import CentroidTracker
from flask import Flask, render_template, Response
import threading
import sqlite3
import utils as cm
from PIL import Image
```

Code 4.4.1 – Importaion des bibliothèques.

1. **importation de tflite_runtime** :
   Le package **tflite_runtime** représente une fraction de la taille du package TensorFlow complet et inclut le code minimum requis pour exécuter des inférences avec TensorFlow Lite, principalement la classe Interpreter. Ce petit package est idéal car tout ce que nous voulons faire, c'est exécuter les modèles `".tflite"` et éviter de gaspiller de l'espace disque avec la grande bibliothèque TensorFlow. Alors, cette instruction d'importation `from tflite_runtime.interpreter import Interpreter` apporte la classe Interpreter du module `tflite_runtime`. Cette classe est utilisée pour exécuter des modèles TensorFlow Lite, qui sont optimisés pour fonctionner sur le Raspberry Pi.

2. **importation de cv2** :

   Le module **cv2** est le module principal de la bibliothèque OpenCV, Ce module fournit de nombreuses fonctions de traitement d'images et de vidéos.

3. **importation de numpy** :

   Le module NumPy prend en charge les grands tableaux et matrices multidimensionnels, ainsi qu'une grande collection de fonctions mathématiques de haut niveau pour opérer sur ces tableaux, l'instruction `import numpy as np` importe le module avec le nom 'np'.

4. **importation de centroidtracker** :

   nous avons importé la classe CentroidTracker à partir d'un module personnalisé nommé centroidtracker dans github [2]. Cette classe est utilisée pour suivre le centre de gravité des objets dans les flux vidéo.

5. **importation de flask** :

   L'instruction import apporte la classe Flask, qui est la classe principale pour la création d'applications `Flask`. La fonction `render_template` est utilisée pour rendre les modèles HTML et la classe `Response` est utilisée pour créer des réponses HTTP. L'objet de requête (`request`) est utilisé pour gérer les requêtes HTTP entrantes.

6. **importation de threading** :

   Le module de threading prend en charge le parallélisme basé sur les threads dans notre application. Ce module est utilisé pour exécuter simultanément différentes parties de l'application.

7. **importation de sqlite3** :

   Le module sqlite3 fournit un moteur de base de données léger et sans serveur qui est utilisé dans l'application. Ce module est utilisé pour stocker des données dans une base de données.

8. **importation de utils** :

   l'importation apporte un module personnalisé appelé utils, qui fournit diverses fonctions utilitaires utilisées dans l'application.

9. **importation de PIL** :

   Le module PIL prend en charge l'ouverture, la manipulation et l'enregistrement de nombreux formats de fichiers image différents. Ce module est utilisé pour convertir des images dans l'application.

2. GitHub est un service d'hébergement de référentiel Git basé sur le cloud qui permet aux individus et aux équipes d'utiliser plus facilement Git pour le contrôle de version et la collaboration.

4.4.2 Déclaration des variables

Le code suivant permet de définir les informations capitales qui sont nécessaires pour préparer notre programme pour la détection d'objets à l'aide d'un modèle de détection d'objet à base de CNN pré-entraîné. Les variables et leurs valeurs sont :

```
MODEL_NAME = "all_models/"
GRAPH_NAME = "mobilenet_ssd_v2_coco_quant_postprocess.tflite"
LABELMAP_NAME = "coco_labels.txt"
min_conf_threshold = 0.5
imW, imH = 800, 480

# Get path to current working directory
CWD_PATH = os.getcwd()

# Path to .tflite file, which contains the model that is used for object detection
PATH_TO_CKPT = os.path.join(CWD_PATH,MODEL_NAME,GRAPH_NAME)

# Path to label map file
PATH_TO_LABELS = os.path.join(CWD_PATH,MODEL_NAME,LABELMAP_NAME)
```

(a) Les variables constantes.

```
def main():

    cap = cv2.VideoCapture("testing/output_v3.mp4")

    ct = CentroidTracker()
    objects ={}
    old_objects={}
    curr_ID = 0
    captured_image = np.array([])
    is_captured = False

    leftcount = 0
    rightcount = 0
    obsFrames = 0

    roi_pos_to_left = 0.2
    roi_pos_to_right = 0.8
    is_Left = False
    is_Right = False
```

(b) Les variables non constantes.

Code 4.4.2 – Déclaration des variables.

1. **La variable MODULE_NAME** :

   Il s'agit du nom du répertoire contenant le modèle pré-formé et d'autres fichiers nécessaires à la détection d'objets.

2. **La variable GRAPH_NAME** :

   Il s'agit du nom du fichier de modèle pré-entraîné qui sera utilisé pour la détection d'objet. Il utilise l'architecture d'apprentissage en profondeur MobileNet-V2 comme colonne vertébrale pour l'algorithme de détection **SSD** (Single Shot MultiBox Detector) et est formé sur l'ensemble de données COCO (Common Objects in Context).

3. **La variable LABELMAP_NAME** :

   Il s'agit du nom du fichier contenant les étiquettes des objets que le modèle peut détecter. Dans ce cas, il s'agit des étiquettes du jeu de données COCO.

4. **La variable min_conf_threshold** :

   Il s'agit du seuil de confiance minimum pour les objets détectés. Si un objet a un score de confiance inférieur à cette valeur, il ne sera pas considéré comme une détection valide.

5. **Les variables imW et imH** :

   Ce sont la largeur et la hauteur de la résolution vidéo.

6. **La variable CWD_PATH** :

   Cela obtient le répertoire de travail actuel du programme.

7. **La variable PATH_TO_CKPT** :

   Cela crée le chemin d'accès complet au fichier de modèle pré-formé.

8. **La variable PATH_TO_LABELS** :

   Cela crée le chemin d'accès complet au fichier de mappage d'étiquettes.

9. **La variable cap** :

   Cela initialise l'objet de capture vidéo de la caméra Pi, qui sera utilisé pour capturer des images vidéo en direct pour la détection d'objet.

10. **La variable ct** :

    Cela initialise un suivi d'objet appelé CentroidTracker, qui sera utilisé pour suivre les objets détectés à travers les cadres (frames)

11. **La variable objects** :

    Ceci est un dictionnaire vide pour stocker les détections actuelles.

12. **La variable old_objects** :

    Il s'agit d'un dictionnaire vide pour stocker les détections précédentes des trames précédentes.

13. **La variable curr_ID** :

    Il s'agit de l'identifiant actuel attribué à une nouvelle détection d'objet.

14. **La variable captured_image** :

    Il s'agit d'un tableau NumPy vide pour stocker une image qui a été capturée.

15. **La variable is_captured** :

    Ceci est un boolean pour indiquer si une image a été capturée.

16. **Les variables leftcount, rightcount et is_Left et is_Right** :

    Ces variables sont utilisé pour le comptage des personnes, et pour garder une trace du nombre d'objets se déplaçant vers la gauche et la droite.

17. **La variable obsFrames** :

    Il s'agit d'un compteur permettant de suivre le nombre d'images traitées et utilisées pour le calcul de la direction.

4.4.3 Création de la base de données

La création de la base de données se fait via la base de données SQLite comme indiqué dans le prochain extrait de code :

```
# Create/Connect to the database
conn = sqlite3.connect('data.db', check_same_thread=False)
# Create a cursor object
cur = conn.cursor()
# execute the SQL code
cur.execute('''CREATE TABLE IF NOT EXISTS data_table (
                ID INTEGER PRIMARY KEY,
                Date TEXT,
                Time TEXT,
                Person TEXT);''')
```

Code 4.4.3 – Création de la base de données.

Cet extrait de code nous permet de nous connecter à une base de données SQLite nommée '**data.db**' et de créer une table nommée '`data_table`' dans la base de données si elle n'existe pas déjà. Voici une répartition de ce que fait chaque ligne de code :

1. `conn = sqlite3.connect('data.db', check_same_thread=False)` :

   cette ligne crée une connexion au fichier de base de données SQLite 'data.db'. La méthode `connect()` est utilisée pour établir une connexion à la base de données. Le paramètre `check_same_thread` est défini sur False, ce qui permet à plusieurs threads d'utiliser la même connexion.

2. `cur = conn.cursor()` :

   Cette ligne crée un objet curseur qui peut être utilisé pour exécuter des requêtes SQL sur la base de données. La méthode cursor() est appelée sur l'objet de connexion.

3. `cur.execute('''...''')` :

   La ligne :

   ```
   cur.execute('''CREATE TABLE IF NOT EXISTS data_table (
   ID INTEGER PRIMARY KEY,
   Date TEXT, Time TEXT, Person TEXT);''')
   ```

   exécute une requête SQL pour créer une table nommée '`data_table`' dans la base de données. La clause `IF NOT EXISTS` est utilisée pour s'assurer que la table n'est créée que si elle n'existe pas déjà. Le tableau comporte quatre colonnes : ID, Date, Heure et Personne. La colonne ID est définie comme clé primaire de la table et les autres colonnes sont de type TEXT.

4.4.4 Création de la page web

4.4.4.1 Partie Flask

L'extrait de code suivant définit une application Web utilisant Flask. Il définit donc une application Web simple qui affiche les données d'une base de données SQLite et diffuse du contenu vidéo.

```python
app = Flask(__name__, static_folder='/')

def get_db():

    conn2 = sqlite3.connect('data.db', check_same_thread=False)
    conn2.row_factory = sqlite3.Row
    return conn2

@app.route("/")
def index():
    conn2 = get_db()
    cur2 = conn2.cursor()
    # cur2.execute('BEGIN')
    cur2.execute('SELECT * FROM data_table')

    rows = cur2.fetchall()
    conn2.close()

    return render_template('index2.html', rows=rows)

@app.route('/video_feed')
def video_feed():
    #global cap
    return Response(main(),
                    mimetype='multipart/x-mixed-replace; boundary=frame')
```

Code 4.4.4 – Création de l'application web.

Voici une brève explication de ce que fait le code :

1. `app = Flask(__name__, static_folder='/')` :

   La première ligne crée un objet d'application Flask nommé "app". Il prend deux arguments, le premier étant "name", qui est une variable qui représente le nom du module actuel, et le second est "`static_folder`", qui spécifie l'emplacement des fichiers statiques (par exemple les images) utilisés par l'application. Dans ce cas, les fichiers statiques sont situés dans le répertoire racine ('/').

2. `La fonction get_db()` :

   La fonction "`get_db`" crée une connexion à un fichier de base de données SQLite nommé "data.db" et renvoie l'objet de connexion.

3. `La fonction index()` :

   Le décorateur "`@app.route`" est utilisé pour mapper un point de terminaison d'URL à la fonction index(). Le point de terminaison "`/`" correspond à le contenue de la fonction "index", qui récupère les données de la "`data_table`" depuis la base de données SQLite à l'aide de la fonction "`get_db`" et les affiche sur un fichier de modèle HTML nommé "index2.html" à l'aide de la fonction "`render_template`" qui fait le rendu de la page web.

4. `La fonction video_feed()` :

   Le "`/video_feed`" dans le point de terminaison `@app.route()` correspond à la fonction "`video_feed`", qui renvoie un flux vidéo en utilisant la fonction "**main**" comme source. La fonction "main" est définie plus loin dans ce code.

4.4.4.2 Partie HTML

Dans le fichier HTML, on a trois parties, partie le code HTML lui-même, le code CSS pour Styliser la page web et la partie JavaScript pour recharger dynamiquement le contenu de la table.

— **Code HTML** :

```
<html>
  <head>
    <title>PFE</title>
    <meta http-equiv="Cache-Control" content="no-cache, no-store, must-revalidate">
    <meta http-equiv="Expires" content="0">
  </head>
  <body>
      <div align="center"><h1 class="box">PFE Instrumentation 2023</h1></div>
  <div align="center"><h2 class="box2">Prototype par: Miloudi Adel Hani</h2></div>
  <div align="center"><img src="{{ url_for('video_feed') }}"></div>
<div class="center">
</div>
    <div id="table-div">
    <table >
<tr>
<th width=20> <div align="center">ID</div></th>
<th width=40> <div align="center">DATE</div></th>
<th width=40> <div align="center">TIME</div></th>
<th width=40> <div align="center">PERSON</div></th>
</tr>
      <tbody>
        {% for row in rows %}
    <tr>
        <td><div align="center">{{ row[0] }}</div></td>
        <td><div align="center">{{ row[1] }}</div></td>
        <td><div align="center">{{ row[2] }}</div></td>
        <td><div align="center"><img src="{{ url_for('static', filename=row[3]).lstrip('/') }}" class="image">
</div></td>
    </tr>
        {% endfor %}
      </tbody>
    </table>
  </div>
  </body>
</html>
```

Code 4.4.5 – Partie HTML de la page web.

Le code est un modèle HTML composé de divers éléments utilisés pour créer une page Web. La section du corps contient des en-têtes, des espaces réservés pour les images de la vidéo et une structure de tableau. Les espaces réservés de titre et d'image -de la vidéo- sont centrés à l'aide d'éléments div avec l'attribut align="center". Le tableau comporte quatre colonnes : ID, DATE, TIME et PERSON. Le corps de la table est rempli dynamiquement à l'aide d'une boucle for avec des données provenant du backend Flask. Chaque ligne du tableau contient des cellules de données dont les valeurs proviennent de la ligne de données correspondante. Le code se termine par un corps de fermeture et une balise (tag) HTML, marquant la fin du document HTML. Dans l'ensemble, ce code représente un modèle HTML qui sert notre page Web en affichant un tableau de données, ainsi que des en-têtes, une image et des classes de style supplémentaires. Les parties dynamiques du modèle, telles que l'URL du flux vidéo et les lignes de données, seront remplies avec des valeurs réelles lorsqu'elles seront rendues par le framework Flask à partir de l'application.

— **Code CSS** :

Le code représente la mise en forme CSS de notre modèle HTML. Tout d'abord, il définit une classe appelée "image" qui fixe la largeur et la hauteur des images. L'élément table est stylisé avec des propriétés pour la fusion des bordures, la largeur, la police de caractères et les marges. Une requête média est utilisée pour ajuster la largeur et la marge de la table sur les écrans d'une largeur minimale de 768 pixels. Les cellules d'en-tête de tableau (th) et de données de tableau (td) sont stylisées avec des propriétés pour l'alignement du texte, le rembourrage et les bordures. Les lignes paires du tableau ont une couleur de fond gris clair pour une meilleure lisibilité. Les cellules d'en-tête du tableau ont une couleur de fond bleu foncé avec un texte blanc pour un style d'en-tête contrastant. Il y a des styles supplémentaires pour les éléments ayant les classes "box" et "box2", définissant les bordures, le rembourrage, les couleurs de fond et les coins arrondis, ainsi qu'une classe "center" pour centrer le contenu horizontalement et verticalement. Donc en général, ce code représente les styles CSS associés du modèle HTML. Les styles CSS définissent l'apparence visuelle de ces éléments, y compris les dimensions, les couleurs, les bordures et les alignements. Les styles garantissent que le tableau est correctement formaté, avec des couleurs de ligne alternées, un contenu centré et un en-tête visuellement attrayant. De plus, le CSS fournit un style pour des classes spécifiques, telles que "box" et "box2", qui définissent des boîtes bordées avec des coins arrondis. Le code vise à créer une page Web visuellement agréable et organisée avec un design cohérent.

```
<style>
  .image {
    width: 80px;
    height: 100px;
  }

table {
  border-collapse: collapse;
  width: 80%;
  font-family: Gill Sans Extrabold, sans-serif;
  margin-left: 10%;
  margin-top: 10%;
}

@media screen and (min-width: 768px) {
  table {
    width: 80%;
    margin: 10% auto;
  }
}

th, td {
  text-align: left;
  padding: 8px;
  border: 1px solid #ddd;
}

tr:nth-child(even){background-color: #fAfAfA}

th {
  background-color: #0A0CCB;
  color: white;
}
.box {
  border: 1px solid black;
  padding: 10px;
  border-style: dashed;
  background-color: lightblue;
  border-radius: 10px;
}

.box2 {
  display: inline-block;
  border: 2px solid black;
  padding: 15px;
  border-style: dashed;
  background-color: orange;
  border-radius: 5px;
}
.center {
        display: flex;
        justify-content: center;
        align-items: center;
  }
</style>
```

Code 4.4.6 – Partie CSS de la page web.

— **Code JavaScript** :

Globalement, ce code met en place un mécanisme pour recharger en continu le contenu de notre table avec l'ID "table-div" en utilisant jQuery et JavaScript. Il déclenche automatiquement le processus de rechargement lors du chargement de la page, puis répète le processus toutes les 3 secondes, garantissant que le tableau affiche toujours les informations les plus à jour, sans recharger tout le contenu de la page Web.

```
<script src="templates/jquery.js"></script>
<script>
    function reloadTable() {
          // Use jQuery to reload the table contents
          $('#table-div').load(location.href + ' #table-div');

          // Set a timeout to call this function again after 3 seconds
          setTimeout(reloadTable, 3000);
        }

        // Call the function on page load
        $(document).ready(function() {
          reloadTable();
        });

</script>
```

Code 4.4.7 – Partie JavaScript de la page web.

4.4.5 Partie Principale

Cette partie du programme utilise TensorFlow Lite et OpenCV pour détecter et compté les personnes en temps réel à l'aide d'une caméra Pi. Il dessine un rectangle autour de chaque personne détectée et les étiquette avec un pourcentage de confiance. Il enregistre également l'image sur le disque et stocke son emplacement dans la base de données. De plus, il suit les objets détectés à l'aide d'un tracker centroïde et met à jour leurs centroïdes dans des images consécutives pour identifier les objets qui se sont déplacés dans l'image. Le programme dessine également l'ID et le centroïde de chaque objet suivi sur le cadre de sortie.

4.4.5.1 Initialisation et configuration

```
def main():

   # initialisation des variables nécessaire pour notre programme
    .
    .
    .

    interpreter, labels =cm.load_model(MODEL_NAME,PATH_TO_CKPT,PATH_TO_LABELS)

    # Get model details
    input_details = interpreter.get_input_details()
    output_details = interpreter.get_output_details()

    # Create a new window
    window_name = "Person Detector"
    cv2.namedWindow(window_name, cv2.WINDOW_NORMAL)
```

Code 4.4.8 – Création de la fonction principale et initialisation.

Premièrement, on a défini la fonction main de notre programme, et après l'initialisation des variables nécessaires du programme, le code charge ensuite un modèle d'apprentissage en profondeur à l'aide de l'interpréteur TensorFlow Lite à partir des chemins donnés vers le répertoire du modèle, le modèle tflite et les étiquettes à l'aide de la fonction `load_model()` de notre bibliothèque personnalisée `"utils"`, et le modèle chargé est stocké dans la variable 'interpreter', tandis que les labels sont stockés dans la variable 'labels'.

```
def load_model(model_dir,model, lbl):

    model_path=os.path.join(model_dir,model)
    labels_path=os.path.join(model_dir,lbl)

    interpreter = tflite.Interpreter(model_path)

    interpreter.allocate_tensors()

    labels = load_labels(labels_path)

    return interpreter, labels
```

Code 4.4.9 – Définition de la fonction load_model dans le fichier utils.py.

La fonction load_model charge un modèle TensorFlow Lite et ses étiquettes associées à partir de répertoires spécifiés. Dans l'ensemble, cette fonction prend les noms de fichier du modèle et de l'étiquette et leur répertoire, charge le modèle à l'aide de TensorFlow Lite et renvoie l'interpréteur [3] et les étiquettes chargées avec la fonction `load_labels` qui a également été défini dans le fichier utlis.

```
def load_labels(path):

  with open(path, 'r') as f:
    labels = f.readlines()

  labels = [label.strip() for label in labels]

  return labels
```

Code 4.4.10 – Définition de la fonction load_labels dans le fichier utils.py.

3. un interpréteur est un environnement d'exécution qui charge un modèle d'apprentissage automatique ou DL et vous permet d'exécuter une inférence sur ce modèle. Il est responsable de l'exécution des opérations dans le modèle sur les données d'entrée et de la production de la sortie. En termes simples, vous pouvez considérer l'interpréteur TensorFlow Lite comme un outil qui vous aide à utiliser un modèle d'apprentissage automatique pour faire des prédictions sur de nouvelles données. Pour ce faire, il charge le modèle en mémoire et vous fournit une interface pour interagir avec lui. Vous pouvez définir les données d'entrée, exécuter l'inférence et obtenir la sortie via l'interpréteur

load_labels est utilisé pour charger un fichier d'étiquettes de texte dans une liste de chaînes. L'instruction '`with`' est utilisée pour ouvrir le fichier au chemin donné et lire son contenu. Le fichier est ouvert en mode texte avec le mode de lecture par défaut '`r`'. dans l'ensemble, la fonction est un moyen simple de charger un fichier d'étiquettes de texte dans une liste de chaînes. La méthode `strip()` est utilisée pour supprimer tous les caractères d'espacement indésirables, ce qui peut être nécessaire lorsque vous travaillez avec des modèles d'apprentissage automatique qui s'attendent à ce que les étiquettes soient des correspondances exactes. On revient maintenant dans la fonction principale "main()", les détails d'entrée et de sortie du modèle chargé sont obtenus à l'aide des méthodes '`get_input_details()`' et '`get_output_details()`' de l'interpréteur. Ensuite, une nouvelle fenêtre nommée **"Person Detector"** est créée à l'aide de la fonction OpenCV `namedWindow()`.

4.4.5.2 Détection et suivi des personnes

1. **Création de flux vidéo et traitement d'images** :

```
while True:
    # On the next loop set the value of these objects
    # as old for comparison
    old_objects.update(objects)
    # Grab frame from camera
     _, frame1 = cap.read()

    cv2_im = frame1
    cv2_im_rgb = cv2.cvtColor(cv2_im, cv2.COLOR_BGR2RGB)
    pil_im = Image.fromarray(cv2_im_rgb)

    cm.set_input(interpreter, pil_im)

    # Perform the actual detection by running the model with the image as input
    interpreter.invoke()

    #-------------------- Retrieve detection results
    # Bounding box coords of detected objects
    boxes = interpreter.get_tensor(output_details[0]['index'])[0]
    # Class index of detected objects
    classes = interpreter.get_tensor(output_details[1]['index'])[0]
    # Confidence of detected objects
    scores = interpreter.get_tensor(output_details[2]['index'])[0]

    saved_image_path = 'person_' + str(curr_ID) + '.png'
```

Code 4.4.11 – Création de flux vidéo et traitement d'images.

Cet extrait de code consiste en une boucle continue qui capture les images de la

caméra (plus précisément un flux vidéo dans ce cas). Cette boucle est la boucle principale de notre application. La boucle commence par mettre à jour un ensemble appelé `"old_objects"` avec le contenu d'un autre ensemble nommé `""objects"`. Cela facilite la comparaison dans les itérations ultérieures des trames. Le code récupère ensuite une image à l'aide d'un objet de capture vidéo `"cap.read()"` et la convertit au format approprié pour la détection d'objet. Le cadre est transformé de l'espace colorimétrique BGR en RGB (RVB) et converti en un objet image PIL. Cette image PIL est transmise en entrée à notre interpréteur de modèle de détection d'objet, qui effectue la détection réelle en exécutant le modèle. Les résultats de la détection, y compris les coordonnées de la boîte englobante, les indices de classe et les scores de confiance des objets détectés, sont obtenus à partir de *l'interpréteur*. Enfin, un chemin de fichier est construit pour enregistrer les objets détectés dans la variable `"saved_image_path"`, avec le nom basé sur la valeur de ‘`curr_ID`‘. Ainsi, cet extrait de code permet une détection continue des objets sur un flux de caméra, fournissant des informations sur l'emplacement, la classe et la confiance des objets détectés. on va voir aussi les fonctions définies dans le fichier `utils.py`. la fonction `set_input` est définie dans le fichier utils.py comme suit :

```
def set_input(interpreter, image, resample=Image.NEAREST):
    """Copies data to input tensor."""
    image = image.resize((input_image_size(interpreter)[0:2]), resample)
    input_tensor(interpreter)[:, :] = image
```

Code 4.4.12 – Déclaration de la fonction de set_input dans le fichier utils.py.

La fonction `set_input` prend trois paramètres : ‘`interpréteur`‘, ‘`image`‘, et ‘`resample`‘ (rééchantillonnage). Elle redimensionne l'image d'entrée en utilisant la méthode de redimensionnement de l'objet Image PIL. La taille cible de l'image redimensionnée est déterminée en appelant la fonction ‘`input_image_size`‘. Les données redimensionnées de l'image sont ensuite copiées dans le tenseur d'entrée de l'interpréteur à l'aide de la fonction‘`input_tensor`‘. La fonction ‘`set_input`‘ utilise la syntaxe ‘`[:, :]`‘ pour sélectionner toutes les lignes et colonnes du tenseur d'entrée. En somme, ‘`set_input`‘ prépare l'image d'entrée pour l'exécution du modèle TensorFlow Lite.

Et les fonctions `input_tensor` et `input_image_size` sont défini comme suit :

```python
def input_image_size(interpreter):
    """Returns input image size as (width, height, channels) tuple."""
    _, height, width, channels = interpreter.get_input_details()[0]['shape']
    return width, height, channels

def input_tensor(interpreter):
    """Returns input tensor view as numpy array of shape (height, width, 3)."""
    tensor_index = interpreter.get_input_details()[0]['index']
    return interpreter.tensor(tensor_index)()[0]
```

Code 4.4.13 – Déclaration des fonctions input_tensor et input_image_size dans le fichier utils.py.

Dans l'ensemble, ces fonctions permettent d'accéder facilement aux données de tenseur d'entrée et à leurs informations de forme pour le modèle mis en œuvre à l'aide de TensorFlow Lite.

2. **Détection des personnes dans chaque cadre d'image** :

   Cette partie c'est l'implémentation de système de détection d'objets qui détecte les personnes dans les images vidéo. Le code définit d'abord un seuil de confiance minimum pour que le système de détection d'objets identifie les objets, puis boucle sur tous les objets détectés dans chaque image, ne dessinant que des cadres autour des objets dont la confiance est supérieure au seuil et sont classés comme une « personne ».

```
#rects variable
rects =[]

# Loop over all detections and draw detection box if confidence
# is above minimum threshold
for i in range(len(scores)):
    if ((scores[i] > min_conf_threshold) and (scores[i] ⩽ 1.0)):
        # Look up object name from "labels" array using class index
        object_name = labels[int(classes[i])]
        if object_name == 'person':

            # Get bounding box coordinates and draw box
            # Interpreter can return coordinates that are outside of image dimensions,
            # need to force them to be within image using max() and min()
            ymin = int(max(1,(boxes[i][0] * imH)))
            xmin = int(max(1,(boxes[i][1] * imW)))
            ymax = int(min(imH,(boxes[i][2] * imH)))
            xmax = int(min(imW,(boxes[i][3] * imW)))
            box = np.array([xmin,ymin,xmax,ymax])
            rects.append(box.astype("int"))
            cv2_im = cv2.rectangle(frame1, (xmin, ymin), (xmax, ymax), (10, 255, 0), 2)

            if is_captured:
              .
              .
```

```
if is_captured:
    # Save the image
    roi = frame1[ymin:ymin+ymax, xmin:xmin+xmax]

    saved_image_path =  'person_' + str(curr_ID) + '.png'
    captured_image = cv2.imwrite(saved_image_path, roi)

    print( "captured_image:{} ".format( saved_image_path ) )

    cur.execute(
         "INSERT OR IGNORE INTO data_table (ID, Date, Time, Person) VALUES (?, ?, ?, ?)",
         (curr_ID, datetime.now().date(), datetime.now().strftime("%H:%M:%S"),saved_image_path))

    conn.commit()

    is_captured = False

# Draw label
label = '%s: %d%%' % (object_name, int(scores[i]*100)) # Example: 'person: 72%'
# Get font size
labelSize, baseLine = cv2.getTextSize(label, cv2.FONT_HERSHEY_SIMPLEX, 0.7, 2)
# Make sure not to draw label too close to top of window
label_ymin = max(ymin, labelSize[1] + 10)
# Draw white box to put label text in
cv2_im = cv2.rectangle(frame1, (xmin, label_ymin-labelSize[1]-10),
                    (xmin+labelSize[0], label_ymin+baseLine-10), (255, 255, 255), cv2.FILLED)
# Draw label text
cv2_im = cv2.putText(frame1, label, (xmin, label_ymin-7), cv2.FONT_HERSHEY_SIMPLEX,
                   0.7, (0, 0, 0), 2)
```

Code 4.4.14 – Détection des perssones et affichage des boîtes englobantes.

Le segment de code initialise une liste vide dans la variable `'rects'` pour stocker les coordonnées de la boîte englobante des objets détectés. Il itère ensuite sur les détections, vérifiant si le score de confiance est supérieur à un seuil minimum. Si l'objet détecté est classé comme une personne, le code extrait les coordonnées de la boîte englobante et les redimensionne aux dimensions de l'image d'origine. Un rectangle est tracé autour de l'objet détecté sur le cadre. Si un indicateur de capture

`'is_captured'` est défini comme `True`, le code enregistre la région d'intérêt (ROI) définie par la boîte englobante sous forme d'image et stocke le chemin. Le code insère également l'ID de l'objet, la date, l'heure et le chemin de l'image enregistrée dans une base de données. En outre, il dessine une étiquette pour l'objet détecté, y compris le nom de l'objet et le score de confiance, près du coin supérieur gauche de la boîte englobante. En résumé, ce code illustre la visualisation des détections d'objets, avec un accent particulier sur l'identification et la capture d'images de personnes.

3. **La suivi et le comptage des personnes dans chaque frame** :

```
#update the centroid for the objects
objects = ct.update(rects)
# calculate the difference between this and
# the previous frame
x = DictDiff(objects,old_objects)
# loop over the tracked objects
for (objectID, centroid) in objects.items():

    if is_Left and centroid[0] < roi_pos_to_left*imW:
            leftcount += 1
            is_Left = False
            # is_captured = True

    if is_Right and centroid[0] > roi_pos_to_right*imW:
            rightcount += 1
            is_Right = False
            # is_captured = True
    # draw both the ID of the object and the centroid
    # of the object on the output frame
    textID = "ID {}".format(objectID)
    .
    .
```

```
      textID = "ID {}".format(objectID)
      curr_ID = objectID
    # Draw ID
    cv2.putText(frame1, textID, (centroid[0] - 2, centroid[1] - 2),
                cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
    # Draw circle
    cv2.circle(frame1, (centroid[0], centroid[1]), 4, (0, 255, 0), -1)
# Draw Right:
cv2_im = cv2.putText(frame1,'Right: {0}'.format(rightcount),(10,40),
                     cv2.FONT_HERSHEY_SIMPLEX,1,(250,80,80),2,cv2.LINE_AA)
# Draw Left:
cv2_im = cv2.putText(frame1,'Left: {0}'.format(leftcount),
                     (10,80),cv2.FONT_HERSHEY_SIMPLEX,1,(250,80,80),2,cv2.LINE_AA)
# Draw Detected:
cv2_im = cv2.putText(frame1,'Detected: {0}'.format(len(objects)),(10,120),
                     cv2.FONT_HERSHEY_SIMPLEX,1,(250,110,100),2,cv2.LINE_AA)
# Draw date and time
cv2_im = cv2.putText(frame1, '{0} - {1}'.format(datetime.now().date(),
                                                datetime.now().strftime("%H:%M:%S")),(200,470),
                     cv2.FONT_HERSHEY_SIMPLEX, 1,(255,255,0),2,cv2.LINE_AA)
```

Code 4.4.14 – Suivi et comptage des perssones.

(a) Cette partie du code commence par mettre à jour les centroïdes des objets suivis à l'aide de la fonction `‘ct.update(rects)‘`. La variable `‘rects‘` représente les boîtes englobantes des objets dans le cadre actuel.

(b) Ensuite, le code calcule la différence entre l'ensemble actuel d'objets suivis (`‘objects‘`) et l'ensemble précédent d'objets suivis (‘`old_objects`‘). La fonction `‘DictDiff‘` est une fonction personnalisée dans le fichier principale, qui compare les dictionnaires représentant les objets et renvoie les différences entre eux. Le résultat est stocké dans la variable `‘x‘`. Et elle définit comme cela :

```
# compare the co-ordinates for dictionaries of interest
def DictDiff(dict1, dict2):
   dict3 = {**dict1}
   for key, value in dict3.items():
       if key in dict1 and key in dict2:
               dict3[key] = np.subtract(dict2[key], dict1[key])
   return dict3
```

Code 4.4.15 – Différence entre les coordonnées de centres des objets.

La fonction `DictDiff` prend deux dictionnaires représentant le cadre actuel et précédent de notre algorithme de détection d'objets comme entrées et renvoie un nouveau dictionnaire avec les différences entre les coordonnées centroïdes des objets. La fonction crée d'abord une copie du premier dictionnaire, puis parcourt chaque paire clé-valeur du dictionnaire copié pour vérifier si elle existe dans les deux dictionnaires. Si c'est le cas, la fonction calcule la différence entre les coordonnées centroïdes correspondantes à l'aide de la fonction de soustraction NumPy. Enfin, la fonction renvoie le nouveau dictionnaire avec les différences entre les coordonnées centroïdes des objets. Cette information est ensuite utilisée pour déterminer la direction du mouvement de chaque objet et compter le nombre d'objets qui se déplacent vers la gauche ou vers la droite.

(c) On revient maintenant dans la fonction principale `main()`, le code boucle ensuite sur chaque objet suivi en utilisant la syntaxe `‘for (objectID, centroid) in objects.items()‘`. Cela permet d'accéder à la fois à l'identifiant unique de l'objet **(‘objectID‘)** et à son centroïde **(‘centroid‘)**.

(d) a l'intérieur de la boucle, il y a des instructions conditionnelles qui vérifient si le centroïde de l'objet est sur le côté gauche ou droit du cadre. Si ‘`is_Left`‘ est

vrai et que la coordonnée x du centroïde est inférieure à ‘roi_pos_to_left‘ multiplié par la largeur du cadre (‘imW‘), cela signifie que l’objet a franchi une position prédéfinie sur le côté gauche du cadre. Dans ce cas, la variable ‘leftcount‘ est incrémentée et ‘is_Left‘ est défini sur False pour éviter plusieurs comptages pour le même objet. Une logique similaire s’applique au côté droit du cadre.

(e) Le code dessine ensuite l’ID et le centroïde de l’objet sur le cadre de sortie (‘frame1‘) à l’aide des fonctions OpenCV ‘cv2.putText‘ et ‘cv2.circle‘. L’ID s’affiche sous forme de texte au-dessus du centroïde. Le texte est vert et le centroïde est représenté par un cercle vert.

(f) Après la boucle, le code utilise les fonctions OpenCV ‘cv2.putText‘ pour afficher le nombre d’objets sur le côté droit, le nombre d’objets sur le côté gauche, le nombre total d’objets détectés et la date et l’heure actuelles . Chaque texte est affiché à différentes positions sur le cadre.

4. **La direction des personnes dans chaque frame** :

Ce segment de code est lié au calcul de la direction du mouvement en fonction de la différence des centroïdes des objets suivis.

```
#count number of frames, for direction calculation
obsFrames = obsFrames + 1
#see what the difference in centroids is after every x frames
# to determine direction of movement
if obsFrames % 24 == 0:

    for k,v in x.items():
        if v[0] > 3:
            is_Left = True
            is_captured = True
        elif v[0] < -3:
            is_Right = True
            is_captured = True
```

Code 4.4.16 – La direction du mouvement des personnes.

(a) Le code commence par incrémenter la variable ‘obsFrames‘, qui est utilisée pour garder une trace du nombre de trames traitées. Cette variable aide à déterminer la direction du mouvement après un certain nombre d’images.

(b) La condition ‘if obsFrames % 24 == 0:‘ vérifie si le nombre de trames

traitées est un multiple de 24. nous pouvons ajuster cette valeur pour changer la fréquence de calcul de la direction.

(c) À l'intérieur de la condition, le code itère sur les éléments du dictionnaire `"x"`, qui contient les différences de centroïdes entre les images actuelles et précédentes pour chaque objet suivi.

(d) Pour chaque paire clé-valeur **(‘k, v‘)** dans le dictionnaire ‘x‘, le code vérifie la différence de coordonnée x (`‘v[0]‘`) entre les centroïdes.

(e) Si la différence de coordonnée **x** (`‘v[0]‘`) est supérieure à 3[4] (nombre de pixels), cela suggère que l'objet s'est déplacé de manière significative vers la gauche. Dans ce cas, le drapeau `‘is_Left‘` est défini sur `True`, indiquant un mouvement vers la gauche. Le drapeau `‘is_captured‘` est également défini sur `True`, utilisé pour capturer la personne détectée et pour mettre à jour la base de données.

(f) Inversement, si la différence de coordonnée x (‘v[0]‘) est inférieure à -3, cela implique que l'objet s'est déplacé de manière significative vers la droite. Dans ce cas, le drapeau `‘is_Right‘` est défini sur `True`, indiquant un mouvement vers la droite. Encore une fois, le drapeau `‘is_captured‘` est défini sur True.

Dans l'ensemble, ce segment de code permet de déterminer la direction du mouvement en vérifiant les différences de centroïdes entre les images après un certain nombre d'images (dans ce cas, toutes les 24 images). Si la différence dépasse un seuil (3 dans ce cas), les drapeaux de direction respectifs (`"is_Left"` ou `"is_Right"`) sont définis, et le drapeau `"is_captured"` est également défini sur True.

4. Nous avons choisi le chiffre 3 à titre de comparaison car il s'agit d'un seuil raisonnable pour déterminer si un objet s'est déplacé. Si un objet s'est déplacé de plus de 3 pixels, il est probable qu'il se soit déplacé de manière significative. Si un objet s'est déplacé de moins de 3 pixels, il est possible qu'il se soit déplacé, mais il est également possible que le mouvement soit dû au bruit

4.4.5.3 Launcement de la vidéo et de streaming

```
        # Set the window's property to full screen
        cv2.setWindowProperty(window_name, cv2.WND_PROP_FULLSCREEN, cv2.WINDOW_FULLSCREEN)
        cv2.imshow(window_name, cv2_im)

        # Press 'q' to quit and give the total tally
        if cv2.waitKey(1) == ord('q'):
            break

        ret, jpeg = cv2.imencode('.jpg', cv2_im)
        pic = jpeg.tobytes()

        #Flask streaming
        yield (b'--frame\r\n'
               b'Content-Type: image/jpeg\r\n\r\n' + pic + b'\r\n\r\n')


    cv2.destroyAllWindows()
    cap.release()
    conn.close()
```

Code 4.4.17 – Launcement de la vidéo et de streaming.

Le code définit d'abord la propriété de la fenêtre sur plein écran. Cela fera que la fenêtre remplira tout l'écran. Ensuite, le code affiche l'image sur la fenêtre. La fonction `cv2.imshow()` prend deux arguments : le nom de la fenêtre et l'image à afficher. Le code vérifie alors si l'utilisateur a appuyé sur la touche `q`. Si c'est le cas, le code sort de la boucle. Ainsi, l'application cède l'image au streaming Flask. Cela permettra à Flask de diffuser l'image sur le navigateur de l'utilisateur et d'afficher le flux vidéo sur la page Web. Enfin, le code ferme toutes les fenêtres ouvertes, libère la caméra et ferme la connexion à la base de données, ce code marque la fin de la fonction principale `main()`.

4.4.5.4 Exécution de programme

```python
def run_server():

    app.run(host='0.0.0.0', port=8080, threaded=True, debug=False) # Run FLASK

if __name__ == '__main__':

    server_thread = threading.Thread(target=run_server)
    app_thread = threading.Thread(target=main)

    server_thread.start()
    app_thread.start()

    # wait until thread 1 is finished
    server_thread.join()
    # wait until thread 2 is finished
    app_thread.join()
```

Code 4.4.18 – Exécution de l'application.

D'abord, on a définit une fonction appelée `run_server()`. Cette fonction exécute le serveur Flask. Enfin, on vérifie si le nom du module est égal à la chaîne `'__main__'` avec la l'instruction `if __name__ == '__main__'`. Cette ligne est nécessaire car elle permet d'exécuter le code à l'intérieur du bloc `if` uniquement lorsque le module est exécuté en tant que programme principal.Alors si le module est importé d'un autre module, le code à l'intérieur du bloc if ne sera pas exécuté. Donc si la condition est vérifié, le code crée deux threads : un thread de serveur et un thread d'application. Le thread du serveur exécute le serveur Flask et le thread de l'application exécute la fonction principale. Le code démarre ensuite les threads et attend qu'ils se terminent.

4.5 Tests pratiques et résultats

Dans cette partie, il s'agira de démontrer le fonctionnement pratique de notre système de détection d'objet appliqué pour la vidéosurveillance. Avant de commencer ces tests, nous avons commencé par alimenté notre carte Raspberry Pi, et l'écran LCD est installé avec la RPi, elle est connectée à l'internet, pour que nous puisse accéder au site web, maintenant, en peut démarrer l'application.

4.5.1 Lancement du programme

La première étape par laquelle nous devons commencer est d'abord le lancement de programme avec la commande suivante : `python object_detection_counting_web.py`. quand le programme est lancé, le serveur va s'exécuté et aussi l'application de détection.

4.5.2 Affichage et résultats sur la Page Web

4.5.2.1 Vue d'ensemble de la page web

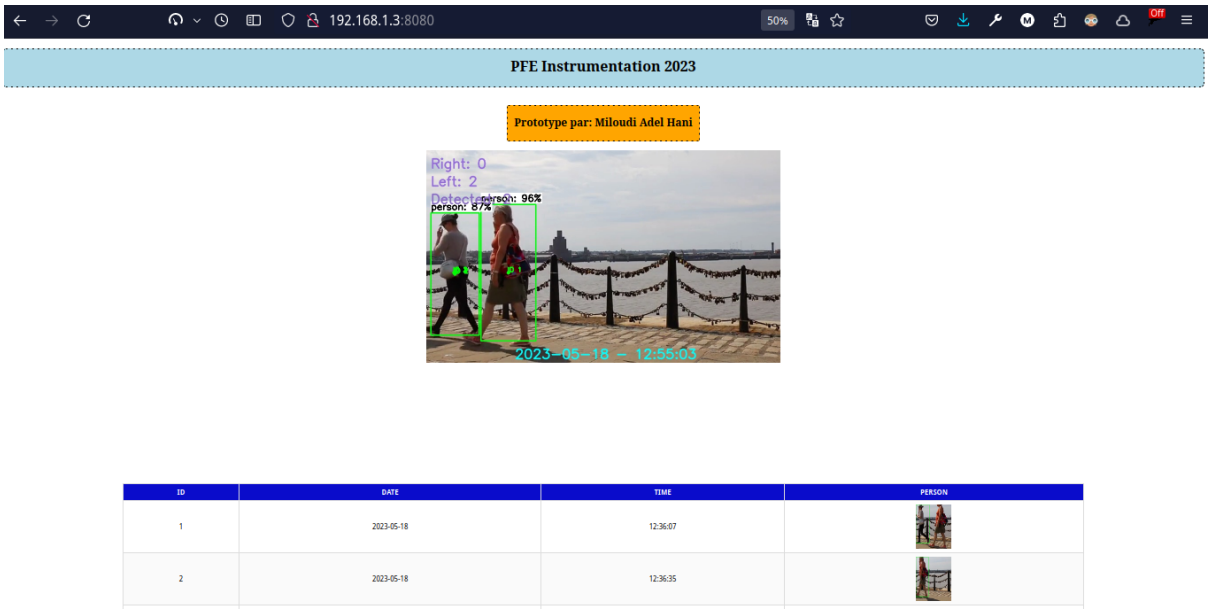


Figure 4.5.1 – Vue d'ensemble de la page web.

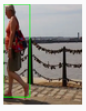
Dans la figure 4.5.1 de système de "Détection d'objets" mentionné précédemment, la page Web est configurée de manière à afficher une vidéo en streaming. Lorsqu'une personne est repérée dans le champ de vision, une boîte entourant la personne apparaît, accompagnée d'un indice de confiance et du nom de la classe (dans ce cas, "personne"). Dans le coin supérieur gauche, des informations sur le nombre de personnes détectées sont fournies en fonction de leur direction, et le nombre de détections dans l'image actuelle est également indiqué. En bas de la page, un tableau affiche des données relatives à la détection, telles que la capture de la personne repérée et un horodatage assorti d'un identifiant pour différencier les personnes détectées.

4.5.2.2 Procédure de détection dans la page web

dans cette partie, nous allons suivre étape par étape comment les données sont affichées dans la page web :

1. **Première et deuxième détection** :

(a) Première et deuxième détection des personnes

ID	DATE	TIME	PERSON
1	2023-05-18	12:36:07	
2	2023-05-18	12:36:35	

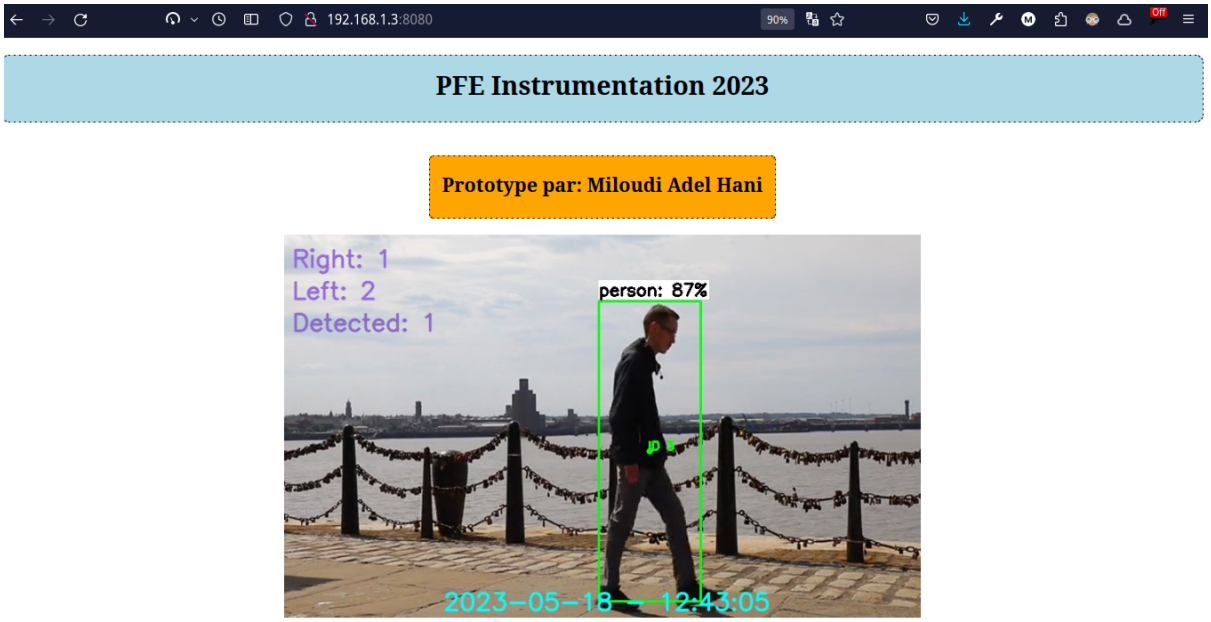
(b) Première et deuxième affichage dans le tableu

Figure 4.5.2 – Première et deuxième détection

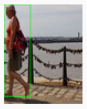
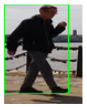
(a) Dès que les deux individus sont apparus dans le champ de vision, le système a immédiatement repéré qu'il s'agissait d'êtres humains et a entouré chacun d'eux d'une boîte de délimitation. Les informations sur le niveau de confiance de l'algorithme ainsi que sur la catégorie d'objet détecté ont également été fournies. Par la suite, leur direction a été calculée ; lorsque les personnes étaient identifiées comme se dirigeant vers la gauche, le compteur du nombre de personnes allant vers la gauche a été augmenté.

(b) Dans le même temps, la base de données a stocké les données nécessaires sur la détection, comme l'image des personnes détectées plus l'horodatage de la détection et un identifiant unique. De plus, ces informations ont été affichées dans le tableau comme indiqué sur la figure.

2. **Troisième détection** :



(a) Troisième détection des personnes

ID	DATE	TIME	PERSON
1	2023-05-18	12:36:07	
2	2023-05-18	12:36:35	
3	2023-05-18	12:42:28	

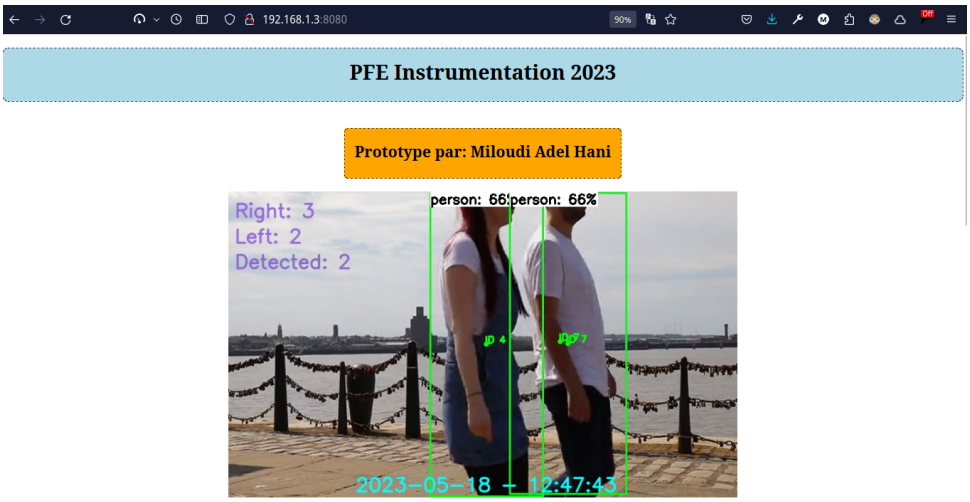
(b) Troisième affichage dans le tableu

Figure 4.5.3 – Troisième détection

(a) Dès que l'individu est apparu dans le champ de vision, le système a immédiatement repéré qu'il s'agissait d'une personne et il a entouré d'une boîte de délimitation. Les informations sur le niveau de confiance ainsi que sur la catégorie d'objet détecté a été également fournies. Par la suite, la direction a été calculée ; lorsque le personne était identifiée comme se dirige vers la droite, le compteur du nombre de personnes allant vers la droite a été augmenté, exactement comme pour la première détection.

(b) Dans le même temps, la base de données stocke les données nécessaires liées à la détection, telles que l'image de personne détectée ainsi que l'horodatage de détection et l'identifiant unique pour cette personne. De plus, ces informations ont été affichées dans le tableau comme indiqué.

3. **Quatrième et cinquième détection** :



(a) Quatrième et cinquième détection des personnes

ID	DATE	TIME	PERSON
1	2023-05-18	12:36:07	
2	2023-05-18	12:36:35	
3	2023-05-18	12:42:28	
4	2023-05-18	12:47:21	
7	2023-05-18	12:47:43	

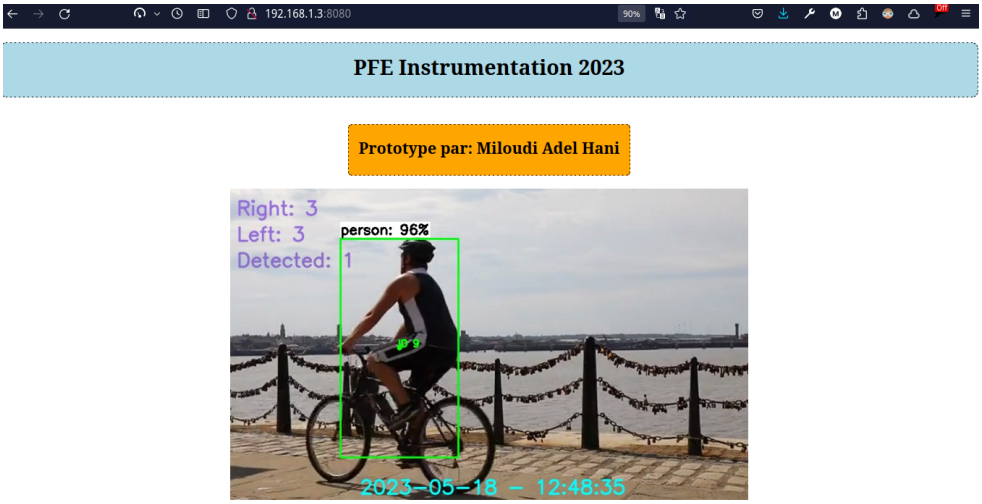
(b) Quatrième et cinquième affichage dans le tableu

Figure 4.5.4 – Quatrième et cinquième détection

(a) Dès que les deux individus sont apparus dans le champ de vision, le système a immédiatement repéré qu'il s'agissait d'êtres humains et a entouré chacun d'eux d'une boîte de délimitation, comme pour les autres détections. En suite, leur direction a été calculée ; lorsque les personnes étaient identifiées comme se dirigeant vers la droite, le compteur du nombre de personnes allant vers la droite a été augmenté pour chaque personne[5]. **(b)** Comme pour les détections précédentes, la base de données stocke les données liées à la détection,et ces informations ont été affichées dans le tableau dans la page web.

5. **Remarque** : quand deux objets sont trous proches, c'est possible qu'un objet ne soit pas suivi correctement, donc l'incrémentation du compteur sera fait pour une seule personne, car la connaissance de la direction est faite à base de suivi d'objet, et notre algorithme de suivi, elle est basée sur la détection, qu'il le rende des fois imprécise, et ce problème est connu comme problème d'occlusion dans la détection et suivi des objets.

4. **Sixième détection** :



(a) Sixième détection de personne

ID	DATE	TIME	PERSON
1	2023-05-18	12:36:07	
2	2023-05-18	12:36:35	
3	2023-05-18	12:42:28	
4	2023-05-18	12:47:21	
7	2023-05-18	12:47:43	
9	2023-05-18	12:48:25	

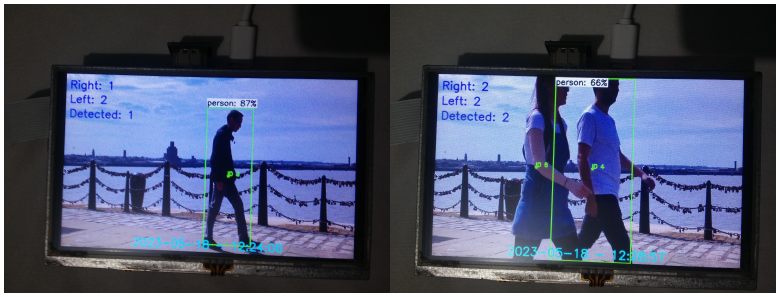
(b) Sixième affichage dans le tableu

Figure 4.5.5 – Sixième détection

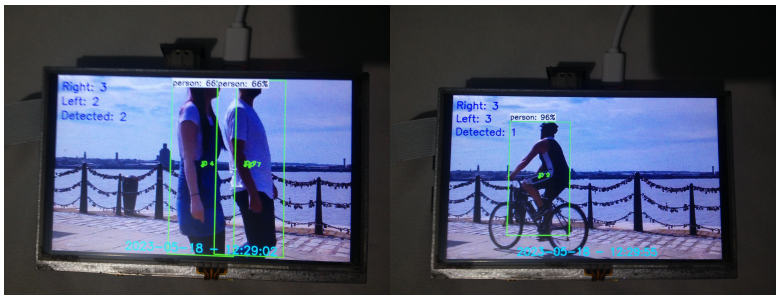
Le programme a détecté l'individu et l'a encadré avec une boîte de délimitation. Le système a ensuite compté la personne comme se dirigeant vers la gauche.Et comme pour les détections précédentes, et les informations de détection ont été affichées dans le tableau dans la page web.

4.5.3 Affichage sur l'écran LCD TFT

(a) 1er et 2éme détections de personne



(b) 3éme et 4éme détections de personne



(c) 5éme et 6éme détections de personne

Figure 4.5.6 – L'affichage dans l'écran LCD TFT

La vidéo est affichée en plein écran dans notre application, car c'est plus pratique. Nous constatons que la vidéo apparaît exactement comme sur le site web, avec les mêmes informations que sur la page web, puisque ce sont la même chose. Ainsi, notre application affiche la vidéo à la fois sur l'écran LCD et sur la page web. L'application affiche la vidéo en plein écran, offrant une plus grande praticité, et c'est plus pratique. Elle reproduit fidèlement la vidéo telle qu'elle apparaît sur le site web, garantissant ainsi la préservation de toutes les informations d'affichage sur la page web. Cette intégration transparente garantit une expérience cohérente pour les utilisateurs, que ce soit sur les écrans LCD ou sur la page web. Alors l'utilisateur bénéficie de la même qualité visuelle et des mêmes informations affichées.

4.6 Conclusion

Au cours de ce chapitre, nous avons abordé de manière approfondie le concept de notre système de détection de personnes qui inclut un algorithme de suivi d'objet utilisé pour le comptage des personnes, qui est appliqué à la vidéosurveillance pour la rendre intelligente et automatisée. Nous avons commencé par présenter le schéma bloc de fonctionnement de notre système, puis nous avons procédé à la préparation du système afin de pouvoir développer le programme de notre application. Enfin, nous avons conclu ce chapitre en effectuant des tests pratiques qui ont démontré le bon fonctionnement du système de détection, accompagnés d'une description des résultats obtenus.

CHAPITRE 5

Conclusion générale et perspectives

Conclusion générale

L'objectif de ce projet consiste à proposer une solution visant à améliorer les systèmes de vidéosurveillance en exploitant l'intelligence artificielle, ce qui permettra d'améliorer la qualité du service de sécurité dans divers établissements. Ainsi, nous avons développé un système de détection polyvalent de personnes, qui peut être déployé facilement et à moindre coût.

Pour la réalisation de ce projet, nous avons utilisé deux éléments fondamentaux. Tout d'abord, nous avons utilisé une carte électronique Raspberry Pi équipée du Wi-Fi, du Bluetooth et de l'Ethernet. Sur cette carte, nous avons connecté un écran LCD TFT de 5 pouces ainsi qu'une caméra Pi. Ensuite, nous avons créé une application Web qui est synchronisée avec une base de données locale. Pour une meilleure clarté, nous avons divisé ce travail en cinq grandes parties importantes :

Introduction générale : Cette partie a présenté une vue d'ensemble de la détection d'objets à l'aide de TensorFlow Lite sur une Raspberry Pi, expose le problème et les objectifs de la mémoire, et décrit la structure de celle-ci.

L'état de l'art : Cette partie présente les Réseaux de Neurones Convolutifs (CNN) en tant que fondement des algorithmes de détection modernes. Nous avons approfondi ensuite la détection d'objets, en explorant son histoire, les concepts importants et les différentes approches basées sur les CNN.

Composants matériels et logiciels : Nous avons présenté les divers composants matériels nécessaires pour mener à bien le projet, en mettant l'accent sur la Raspberry Pi et ses fonctionnalités, ainsi que sur d'autres éléments tels que le module caméra Pi et l'écran LCD TFT de 5 pouces. Ensuite, nous avons introduit les outils logiciels nécessaires, tels que le langage de programmation utilisé pour le développement du projet, ainsi que les bibliothèques pour la mise en œuvre du système de détection d'objets et la création de la page web.

Réalisation du système : Cette partie détaille la conception du système de détection d'objets. Nous avons commencé par la préparation du système, puis explications de la création de notre application en détaillant les différentes parties du code, notamment le programme Python et le code HTML/CSS/JavaScript de la page web. Et on a ajouté les tests pratiques réalisés permettent de démontrer le fonctionnement du système.

Conclusion générale et perspectives : Le dernier chapitre marque la conclusion de cette mémoire en résumant notre projet de fin d'études et en proposant des orientations

et perspectives pour les travaux futurs dans le domaine de la détection d'objets avec une Raspberry Pi. Nous soulignons l'importance de l'application de détection d'objets proposée dans le contexte de la vidéosurveillance, et identifions les limites de notre projet ainsi que les parties qui pourraient bénéficier d'améliorations à l'avenir pour augmenter l'efficacité de notre application.

Perspectives

Dans le cadre de cette étude, nous avons réussi à atteindre l'objectif principal en développant un système de vidéosurveillance intelligent qui peut être intégré à un réseau local. Cependant, il est possible d'améliorer ce travail et d'optimiser certains aspects de notre système. À cet égard, nous proposons quelques perspectives d'amélioration.

1. **Au niveau de matériel disponible** :

   Le prototype actuel est fonctionnel, mais il ne convient pas à une utilisation en temps réel en raison de sa capacité limitée à traiter seulement 2 images par seconde. Cette vitesse de traitement est insuffisante pour une application en temps réel. Afin de résoudre ce problème, il est nécessaire de remplacer la Raspberry Pi 3B+ par la Raspberry Pi 4B ou par le Jetson Nano. Cependant, même cette amélioration ne sera pas suffisante, car il faudra également ajouter l'accélérateur USB Coral de Google. L'accélérateur USB Coral est un périphérique USB doté d'un ASIC spécialisé appelé Edge TPU (Tensor Processing Unit), conçu pour accélérer les calculs d'inférence en apprentissage automatique (ML) et DL. Le coprocesseur Edge TPU est capable d'exécuter 4 000 milliards d'opérations par seconde, tout en ne consommant que 2 watts de puissance. Lorsqu'il est connecté à une Raspberry Pi, il accélère considérablement la vitesse de traitement des modèles d'apprentissage profond. Il est particulièrement compatible avec la Raspberry Pi 4B, bien qu'il puisse également se connecter à d'autres systèmes dotés d'un système d'exploitation. Théoriquement, en utilisant la Raspberry Pi 4B en conjonction avec l'accélérateur USB Coral, nous pourrons augmenter le nombre d'images traitées à plus de 20 ips. Cela permettrait à notre système de fonctionner en temps réel de manière satisfaisante.

2. **Au niveau de logiciel** :

   Notre application fonctionne parfaitement dans le cadre d'une configuration en réseau

local, mais pour assurer une accessibilité dans un réseau couvrant une vaste zone géographique, il est nécessaire de considérer un système IoT (Internet des objets). Un système de surveillance basé sur l'IoT présente de nombreux avantages. Il offre un accès et une surveillance à distance, permettant ainsi une surveillance de n'importe où. La gestion centralisée simplifie l'administration et réduit les interventions physiques. Le stockage dans le cloud, plutôt que dans une base de données locale, permet une gestion et une évolutivité efficaces des données. De plus, il peut être intégré à d'autres appareils intelligents pour des fonctionnalités avancées. Ce système améliore l'accessibilité, l'évolutivité, l'efficacité et l'efficacité de la sécurité.

3. **Au niveau de l'application Web** :

   La page web actuelle est à la fois simple et efficace, mais il est envisageable d'ajouter des fonctionnalités avancées pour l'améliorer davantage. Par exemple, il serait possible d'ajouter une page de connexion/inscription afin de renforcer la sécurité du système et de le protéger contre les tentatives de piratage. Cela apporterait des mesures de sécurité supplémentaires à notre site web. De plus, si notre application était basée sur l'IoT, il serait envisageable d'ajouter la fonctionnalité permettant de sauvegarder le flux vidéo sur un serveur. Cela permettrait de stocker les enregistrements vidéo de manière sécurisée et d'accéder ultérieurement à ces données en cas de besoin.

En général, notre prototype a une quantité infinie d'améliorations possibles, la seule limite que nous ayons est le temps et le budget.

Bibliography

[1] *About*. OpenCV. URL : https://opencv.org/about/ (visité le 25/04/2023).

[2] *About SQLite*. URL : https://sqlite.org/about.html (visité le 26/04/2023).

[3] Yahel E. Appenzeller, Paul S. Appelbaum et Manuel Trachsel. "Ethical and Practical Issues in Video Surveillance of Psychiatric Units". In : *Psychiatric Services (Washington, D.C.)* 71.5 (1er mai 2020), p. 480-486. ISSN : 1557-9700. DOI : 10.1176/appi.ps.201900397. pmid : 31847737.

[4] Rodrigo Benenson, Stefan Popov et Vittorio Ferrari. "Large-Scale Interactive Object Segmentation with Human Annotators". In : *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, p. 11700-11709.

[5] Alexey Bochkovskiy, Chien-Yao Wang et Hong-Yuan Mark Liao. "Yolov4 : Optimal Speed and Accuracy of Object Detection". 2020. arXiv : 2004.10934.

[6] *CS 230 - Pense-bête de Réseaux de Neurones Convolutionnels*. URL : https://stanford.edu/~shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels#filter (visité le 02/04/2023).

[7] Jifeng Dai et al. "R-FCN : Object Detection via Region-Based Fully Convolutional Networks". In : *Proceedings of the 30th International Conference on Neural Information Processing Systems*. NIPS'16. Red Hook, NY, USA : Curran Associates Inc., 5 déc. 2016, p. 379-387. ISBN : 978-1-5108-3881-9.

[8] N. Dalal et B. Triggs. "Histograms of Oriented Gradients for Human Detection". In : *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). T. 1. Juin 2005, 886-893 vol. 1. DOI : 10.1109/CVPR.2005.177.

[9] Jia DENG et al. "Imagenet : A Large-Scale Hierarchical Image Database". In : *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Ieee, 2009, p. 248-255.

[10] Li DENG. "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]". In : *Signal Processing Magazine, IEEE* 29 (1er nov. 2012), p. 141-142. DOI : 10.1109/MSP.2012.2211477.

[11] Piotr DOLLAR et al. "Pedestrian Detection : An Evaluation of the State of the Art". In : *IEEE transactions on pattern analysis and machine intelligence* 34.4 (2011), p. 743-761.

[12] Piotr DOLLÁR et al. "Pedestrian Detection : A Benchmark". In : *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, p. 304-311.

[13] Walter DORNBERGER. *V-2 The Nazi Rocket Weapon*. 1st THUS edition. Ballantine, 1er jan. 1954. ISBN : 978-0-345-06273-4.

[14] Mark EVERINGHAM et al. "The Pascal Visual Object Classes (Voc) Challenge". In : *International journal of computer vision* 88 (2009), p. 303-308.

[15] Mark EVERINGHAM et al. "The Pascal Visual Object Classes Challenge : A Retrospective". In : *International journal of computer vision* 111 (2015), p. 98-136.

[16] Pedro FELZENSZWALB, David MCALLESTER et Deva RAMANAN. "A Discriminatively Trained, Multiscale, Deformable Part Model". In : *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 2008 IEEE Conference on Computer Vision and Pattern Recognition. Juin 2008, p. 1-8. DOI : 10.1109/CVPR.2008.4587597.

[17] Pedro F. FELZENSZWALB et al. "Object Detection with Discriminatively Trained Part-Based Models". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.9 (sept. 2010), p. 1627-1645. ISSN : 1939-3539. DOI : 10.1109/TPAMI.2009.167.

[18] *Flask*. URL : https://www.fullstackpython.com/flask.html (visité le 26/04/2023).

[19] Kunihiko FUKUSHIMA. "Neocognitron : A Hierarchical Neural Network Capable of Visual Pattern Recognition". In : *Neural Networks* 1.2 (1er jan. 1988), p. 119-130. ISSN : 0893-6080. DOI : 10.1016/0893-6080(88)90014-7. URL : https://www.sciencedirect.com/science/article/pii/0893608088900147 (visité le 02/04/2023).

[20] *General Python FAQ*. Python documentation. URL : https://docs.python.org/3/faq/general.html (visité le 25/04/2023).

[21] Ross GIRSHICK. *Fast R-CNN*. Comment : To appear in ICCV 2015. 27 sept. 2015. DOI : 10.48550/arXiv.1504.08083. arXiv : 1504.08083 [cs]. URL : http://arxiv.org/abs/1504.08083 (visité le 31/03/2023). preprint.

[22] Ross GIRSHICK et al. "Region-Based Convolutional Networks for Accurate Object Detection and Segmentation". In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.1 (jan. 2016), p. 142-158. ISSN : 1939-3539. DOI : 10.1109/TPAMI.2015.2437384.

[23] Ross GIRSHICK et al. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In : *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014 IEEE Conference on Computer Vision and Pattern Recognition. Juin 2014, p. 580-587. DOI : 10.1109/CVPR.2014.81.

[24] Miguel GRINBERG. *Flask Web Development*.

[25] Lieve HAMERS et al. "Similarity Measures in Scientometric Research : The Jaccard Index versus Salton's Cosine Formula". In : *Information Processing & Management* 25.3 (1er jan. 1989), p. 315-318. ISSN : 0306-4573. DOI : 10.1016/0306-4573(89)90048-4. URL : https://www.sciencedirect.com/science/article/pii/0306457389900484 (visité le 03/04/2023).

[26] Fangcheng HE. "Intelligent Video Surveillance Technology in Intelligent Transportation". In : *Journal of Advanced Transportation* 2020 (16 nov. 2020), e8891449. ISSN : 0197-6729. DOI : 10.1155/2020/8891449. URL : https://www.hindawi.com/journals/jat/2020/8891449/ (visité le 06/04/2023).

[27] Kaiming HE et al. "Mask R-CNN". In : *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017 IEEE International Conference on Computer Vision (ICCV). Oct. 2017, p. 2980-2988. DOI : 10.1109/ICCV.2017.322.

[28] Kaiming HE et al. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". In : t. 8691. Comment : This manuscript is the accepted version for IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 2015. See Changelog. 2014, p. 346-361. DOI : 10.1007/978-3-319-10578-9_23. arXiv : 1406.4729 [cs]. URL : http://arxiv.org/abs/1406.4729 (visité le 30/03/2023).

[29] Yihui He et al. "Bounding Box Regression With Uncertainty for Accurate Object Detection". In : Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, p. 2888-2897. URL : https://openaccess.thecvf.com/content_CVPR_2019/html/He_Bounding_Box_Regression_With_Uncertainty_for_Accurate_Object_Detection_CVPR_2019_paper.html (visité le 03/04/2023).

[30] Tom Hope. *Learning TensorFlow*.

[31] Xiaoyue Jiang et al., éd. *Deep Learning in Object Detection and Recognition*. Singapore : Springer Singapore, 2019. ISBN : 978-981-10-5151-7 978-981-10-5152-4. DOI : 10.1007/978-981-10-5152-4. URL : http://link.springer.com/10.1007/978-981-10-5152-4 (visité le 02/04/2023).

[32] S. Naga Jyothi et K. Vijaya Vardhan. "Design and Implementation of Real Time Security Surveillance System Using IoT". In : *2016 International Conference on Communication and Electronics Systems (ICCES)*. 2016 International Conference on Communication and Electronics Systems (ICCES). Oct. 2016, p. 1-5. DOI : 10.1109/CESYS.2016.7890003.

[33] Dmitry Kangin. "Intelligent Video Surveillance". In : ().

[34] Alex Krizhevsky, Ilya Sutskever et Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In : *Communications of the ACM* 60.6 (24 mai 2017), p. 84-90. ISSN : 0001-0782. DOI : 10.1145/3065386. URL : https://dl.acm.org/doi/10.1145/3065386 (visité le 30/03/2023).

[35] Alina Kuznetsova et al. "The Open Images Dataset v4 : Unified Image Classification, Object Detection, and Visual Relationship Detection at Scale". In : *International Journal of Computer Vision* 128.7 (2020), p. 1956-1981.

[36] Yann LeCun, Yoshua Bengio et Geoffrey Hinton. "Deep Learning". In : *Nature* 521.7553 (7553 mai 2015), p. 436-444. ISSN : 1476-4687. DOI : 10.1038/nature14539. URL : https://www.nature.com/articles/nature14539 (visité le 28/03/2023).

[37] Zeming Li et al. *Light-Head R-CNN : In Defense of Two-Stage Object Detector*. 22 nov. 2017. DOI : 10.48550/arXiv.1711.07264. arXiv : 1711.07264 [cs]. URL : http://arxiv.org/abs/1711.07264 (visité le 31/03/2023). preprint.

[38] Tsung-Yi Lin et al. "Focal Loss for Dense Object Detection". In : *Proceedings of the IEEE International Conference on Computer Vision*. 2017, p. 2980-2988.

[39] Tsung-Yi LIN et al. "Microsoft Coco : Common Objects in Context". In : *Computer Vision–ECCV 2014 : 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer, 2014, p. 740-755.

[40] Wei LIU et al. "Ssd : Single Shot Multibox Detector". In : *Computer Vision–ECCV 2016 : 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, p. 21-37.

[41] David G. LOWE. "Distinctive Image Features from Scale-Invariant Keypoints". In : *International Journal of Computer Vision* 60.2 (nov. 2004), p. 91-110. ISSN : 0920-5691. DOI : 10.1023/B:VISI.0000029664.99615.94. URL : http://link.springer.com/10.1023/B:VISI.0000029664.99615.94 (visité le 29/03/2023).

[42] Raspberry Pi LTD. *Raspberry Pi*. Raspberry Pi. URL : https://www.raspberrypi.com/ (visité le 19/04/2023).

[43] Simon MONK. *Raspberry Pi Cookbook : Software and Hardware Problems and Solutions*. 3rd edition. Beijing China ; Boston [MA] : O'Reilly Media, 5 nov. 2019. 605 p. ISBN : 978-1-4920-4322-5.

[44] Huu-Quoc NGUYEN et al. "Low Cost Real-Time System Monitoring Using Raspberry Pi". In : *2015 Seventh International Conference on Ubiquitous and Future Networks*. 2015 Seventh International Conference on Ubiquitous and Future Networks. Juill. 2015, p. 857-859. DOI : 10.1109/ICUFN.2015.7182665.

[45] Joachim NIEMEYER, Franz ROTTENSTEINER et Uwe SOERGEL. "Contextual Classification of Lidar Data and Building Object Detection in Urban Areas". In : *ISPRS Journal of Photogrammetry and Remote Sensing* 87 (jan. 2014), p. 152-165. ISSN : 09242716. DOI : 10.1016/j.isprsjprs.2013.11.001. URL : https://linkinghub.elsevier.com/retrieve/pii/S0924271613002359 (visité le 06/04/2023).

[46] *NumPy*. URL : https://numpy.org/ (visité le 26/04/2023).

[47] *NumPy Documentation — NumPy v1.24 Manual*. URL : https://numpy.org/doc/stable/ (visité le 26/04/2023).

[48] Keiron O'SHEA et Ryan NASH. *An Introduction to Convolutional Neural Networks*. Comment : 10 pages, 5 figures. 2 déc. 2015. DOI : 10.48550/arXiv.1511.08458. arXiv : 1511.08458 [cs]. URL : http://arxiv.org/abs/1511.08458 (visité le 02/04/2023). preprint.

[49] T. Ojala, M. Pietikainen et T. Maenpaa. “Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns”. In : *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24.7 (juill. 2002), p. 971-987. ISSN : 1939-3539. DOI : 10.1109/TPAMI.2002.1017623.

[50] Rahul Patil et al. *IoT Enabled Video Surveillance System Using Raspberry Pi*. 21 déc. 2017, p. 7. 1 p. DOI : 10.1109/CSITSS.2017.8447877.

[51] *Pillow*. Pillow (PIL Fork). URL : https://pillow.readthedocs.io/en/stable/index.html (visité le 26/04/2023).

[52] *Raspberry Pi Documentation*. URL : https://www.raspberrypi.com/documentation/ (visité le 19/04/2023).

[53] Team rédac. *OpenCV : tout savoir sur le principal outil de Computer Vision*. Formation Data Science | DataScientest.com. 1er sept. 2022. URL : https://datascientest.com/opencv (visité le 25/04/2023).

[54] Team rédac. *TensorFlow : le framework de Machine Learning de Google*. Formation Data Science | DataScientest.com. 7 jan. 2021. URL : https://datascientest.com/tensorflow (visité le 24/04/2023).

[55] Joseph Redmon et Ali Farhadi. “YOLO9000 : Better, Faster, Stronger”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, p. 7263-7271.

[56] Joseph Redmon et Ali Farhadi. *YOLOv3 : An Incremental Improvement*. Comment : Tech Report. 8 avr. 2018. DOI : 10.48550/arXiv.1804.02767. arXiv : 1804.02767 [cs]. URL : http://arxiv.org/abs/1804.02767 (visité le 31/03/2023). preprint.

[57] Joseph Redmon et al. “You Only Look Once : Unified, Real-Time Object Detection”. In : *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, p. 779-788.

[58] Shaoqing Ren et al. *Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks*. Comment : Extended tech report. 6 jan. 2016. DOI : 10.48550/arXiv.1506.01497. arXiv : 1506.01497 [cs]. URL : http://arxiv.org/abs/1506.01497 (visité le 31/03/2023). preprint.

[59] Shaoqing Ren et al. “Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks”. In : *IEEE transactions on pattern analysis and machine intelligence* 39.6 (juin 2017), p. 1137-1149. ISSN : 1939-3539. DOI : 10.1109/TPAMI.2016.2577031. pmid : 27295650.

[60] Lucy Roberts. “History of Video Surveillance and CCTV”. In : *We CU Surveillance* (2004).

[61] Olga Russakovsky et al. “Imagenet Large Scale Visual Recognition Challenge”. In : *International journal of computer vision* 115 (2015), p. 211-252.

[62] Shuai Shao et al. “Objects365 : A Large-Scale, High-Quality Dataset for Object Detection”. In : *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, p. 8430-8439.

[63] Karen Simonyan et Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 10 avr. 2015. DOI : 10.48550/arXiv.1409.1556. arXiv : 1409.1556 [cs]. URL : http://arxiv.org/abs/1409.1556 (visité le 05/04/2023). preprint.

[64] G. Sreenu et M A Durai. “Intelligent Video Surveillance : A Review through Deep Learning Techniques for Crowd Analysis”. In : *Journal of Big Data* 6 (6 juin 2019), p. 48. DOI : 10.1186/s40537-019-0212-5.

[65] Christian Szegedy et al. *Going Deeper with Convolutions*. 16 sept. 2014. DOI : 10.48550/arXiv.1409.4842. arXiv : 1409.4842 [cs]. URL : http://arxiv.org/abs/1409.4842 (visité le 05/04/2023). preprint.

[66] Raspberry Team. *Raspberry Pi 3 B Connectors*. Raspberry. URL : https://mintwithraspberry.blogspot.com/2018/12/raspberry-pi-3-b-connectors.html (visité le 20/04/2023).

[67] J. R. R. Uijlings et al. “Selective Search for Object Recognition”. In : *International Journal of Computer Vision* 104.2 (1er sept. 2013), p. 154-171. ISSN : 1573-1405. DOI : 10.1007/s11263-013-0620-5. URL : https://doi.org/10.1007/s11263-013-0620-5 (visité le 30/03/2023).

[68] Paul Viola et Michael J. Jones. “Robust Real-Time Face Detection”. In : *International Journal of Computer Vision* 57.2 (1er mai 2004), p. 137-154. ISSN : 1573-1405. DOI : 10.1023/B:VISI.0000013087.49260.fb. URL : https://doi.org/10.1023/B:VISI.0000013087.49260.fb (visité le 29/03/2023).

[69] Chien-Yao WANG, Alexey BOCHKOVSKIY et Hong-Yuan Mark LIAO. "YOLOv7 : Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors". 2022. arXiv : 2207.02696.

[70] Xiaoyu WANG et al. "Regionlets for Generic Object Detection". In : *IEEE transactions on pattern analysis and machine intelligence* 37.10 (oct. 2015), p. 2071-2084. ISSN : 1939-3539. DOI : 10.1109/TPAMI.2015.2389830. pmid : 26353185.

[71] Seth WEIDMAN. "Deep Learning from Scratch : Building with Python from First Principles". In : ().

[72] *What Is Python ? Executive Summary*. Python.org. URL : https://www.python.org/doc/essays/blurb/ (visité le 25/04/2023).

[73] *What Is TFT LCD TV and LCD Monitor Panel ?* URL : https://www.rose-hulman.edu/class/ee/hoover/ece331/old%20stuff/LCD%20Panel%20Info/What%20is%20TFT%20LCD%20TV%20and%20LCD%20Monitor%20Panel.htm (visité le 24/04/2023).

[74] Wei Qi YAN. *Introduction to Intelligent Surveillance : Surveillance Data Capture, Transmission, and Analytics*. Texts in Computer Science. Cham : Springer International Publishing, 2019. ISBN : 978-3-030-10712-3 978-3-030-10713-0. DOI : 10.1007/978-3-030-10713-0. URL : http://link.springer.com/10.1007/978-3-030-10713-0 (visité le 08/04/2023).

[75] Fan YANG, Qi WEI et Fan YANG. "Design of Smart Home Control System of Internet of Things Based on STM32". In : *Journal of Physics : Conference Series* 1972.1 (juill. 2021), p. 012002. ISSN : 1742-6596. DOI : 10.1088/1742-6596/1972/1/012002. URL : https://dx.doi.org/10.1088/1742-6596/1972/1/012002 (visité le 08/04/2023).

[76] Matthew D. ZEILER et Rob FERGUS. "Visualizing and Understanding Convolutional Networks". In : (28 nov. 2013). DOI : 10.48550/arXiv.1311.2901. arXiv : 1311.2901 [cs]. URL : http://arxiv.org/abs/1311.2901 (visité le 31/03/2023). preprint.

[77] Xingyi ZHOU, Dequan WANG et Philipp KRÄHENBÜHL. "Objects as Points". 2019. arXiv : 1904.07850.

[78] C. Lawrence ZITNICK et Piotr DOLLÁR. "Edge Boxes : Locating Object Proposals from Edges". In : *Computer Vision – ECCV 2014*. Sous la dir. de David FLEET et al. Lecture Notes in Computer Science. Cham : Springer International Publishing, 2014, p. 391-405. ISBN : 978-3-319-10602-1. DOI : 10.1007/978-3-319-10602-1_26.

[79] Zhengxia ZOU et al. “Object Detection in 20 Years : A Survey”. In : (18 jan. 2023). Comment : Accepted by Proceedings of the IEEE. DOI : 10.48550/arXiv.1905.05055. arXiv : 1905.05055 [cs]. URL : http://arxiv.org/abs/1905.05055 (visité le 24/03/2023). preprint.

Résumé

Le projet initial était de créer une application de détection d'objets avec TensorFlow sur Raspberry Pi. Cependant, il est devenu un système de vidéosurveillance intelligent utilisant TensorFlow Lite, adapté aux appareils embarqués et mobiles. TensorFlow Lite offre des modèles pré-entraînés pour détecter des personnes, animaux, véhicules, etc. Le système utilise un écran LCD connecté à la Raspberry Pi et une interface web pour la surveillance. Les informations de détection sont affichées sur la page web, incluant un ID unique, l'heure de détection et une capture d'image. Toutes ces données sont stockées localement dans une base de données.

Mots Clés : Vision par ordinateur, CNN, Détection d’objets, Raspberry Pi, Tensorflow

Abstract

The initial project was to create an object detection application using TensorFlow on Raspberry Pi. However, it evolved into an intelligent video surveillance system using TensorFlow Lite, designed for embedded and mobile devices. TensorFlow Lite provides pretrained models for detecting people, animals, vehicles, etc. The system utilizes an LCD screen connected to the Raspberry Pi and a web interface for monitoring. Detection information is displayed on the web page, including a unique ID, detection time, and an image capture. All this data is stored locally in a database.

Keywords: Computer Vision, CNN, Object Detection, Raspberry Pi, TensorFlow

ملخص

كان المشروع الأولي هو إنشاء تطبيق لاكتشاف الكائنات باستخدام TensorFlow على Raspberry Pi ، إلا أنه تطور إلى نظام مراقبة فيديو ذكي باستخدام TensorFlow Lite ، وهو مناسب للأجهزة المدمجة والمتنقلة. يقدم TensorFlow Lite نماذج مدربة مسبقًا لاكتشاف الأشخاص والحيوانات والمركبات وما إلى ذلك. يستخدم النظام شاشة LCD متصلة بـ Raspberry Pi وواجهة ويب للمراقبة. يتم عرض معلومات الكشف على صفحة الويب ، بما في ذلك معرف فريد ووقت الاكتشاف والتقاط الصورة للأشخاص. يتم تخزين كل هذه البيانات محليًا في قاعدة بيانات.

الكلمات المفتاحية: رؤية الكمبيوتر ، CNN ، اكتشاف الأشياء ، TensorFlow، Raspberry Pi