

République Algérienne Démocratique et Populaire
الجمهورية الجزائرية الديمقراطية الشعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليم العالي و البحث العلمي



Université A. Mira de Béjaïa
des Sciences Exactes
Département d'Informatique

MÉMOIRE DE MASTER RECHERCHE

En Informatique

Option : Intelligence Artificielle

Systeme de Détection et de Reconnaissance d'Intrus basé sur l'Intelligence Artificielle

A soutenir le 04/07/2023

Réalisé par :

M^{me} BORDJAHSana

M^{lle} OUARIManel

Mme. BOULAHROUZ Djamila	Université de Béjaïa	Présidente
M. ATMANI Mouloud	Université de Béjaïa	Encadrant
Mme. AIT KACI AZZOU Samira	Université de Béjaïa	Co-encadrante
Mme. KHALED Hayette	Université de Béjaïa	Examinatrice

Promotion : 2022/2023

Dédicace

“

Je dédie ce travail à mes parents, qui ont été toujours là pour moi, quelles que soient les circonstances. À mes frères Syphax et surtout Dylan, mon ange. À mes sœurs Wafa, Hadja et Sirine. À mon très cher mari, qui m'a encouragée à continuer. Et surtout à mes amis, qui ont été présents à mes côtés tout au long de cette période à l'université. Nous avons travaillé tous ensemble, nous nous sommes encouragées et nous nous sommes entraides. À Rany, Anna, Lucy, Jas, Fougou, et surtout à ma binôme Manel.

À tous ceux qui nous sont chers, à vous tous

Merci.

”

Sana

Dédicace

“

Je dédie ce travail à ma grand-mère, celle qui représentait ce que j'avais de plus précieux, la personne qui m'a accompagné tout au long de ma vie et qui a su me construire. Je te remercie pour tout ce que tu m'as donné et pour chaque moment partagé. Je suis profondément déçu car tu ne peux pas être présente à mes côtés en ces moments-là. À mes parents, . À mon frère et ma soeur. À ma meilleure amie et cousine Zouina. À mes oncles et mes tantes . À mes chats. Et surtout à mes amis, qui ont été présents à mes côtés tout au long de cette période à l'université. Nous avons travaillé tous ensemble, nous nous sommes encouragées et nous nous sommesentraïdés. À Ryma, Ahlam, Rany, Anna, Lucy, Foufou, Fianso, Djilil, Imad, Djam, Halim et surtout à ma binôme Sana. À tous ceux qui nous sont chers, à vous tous

Merci.

”

Manel

Remerciements

Tout d'abord, je remercie Allah le tout puissant de m'avoir donné le courage et la patience nécessaires à mener ce travail à son terme.

Nous tenons à remercier tout particulièrement notre encadrante **Mr. ATMANI MOULOU**D, pour l'aide compétente qu'il nous a apporté, pour sa patience et son encouragement. Son œil critique nous a été très précieux pour structurer le travail et pour améliorer la qualité des différentes sections.

Je tiens à remercier également notre co-encadrante **Mme. AIT KACI AZZOU SAMIRA Epse BOUKERRAM** pour son aide immense, la qualité de son suivi ainsi que pour tous les conseils, les renseignements précieux et les informations qu'elle nous a prodigués avec un degré de patience et de professionnalisme sans égal.

Que les membres de jury trouvent, ici, l'expression de nos sincères remerciements pour l'honneur qu'ils nous font en prenant le temps de lire et d'évaluer ce travail.

Pour finir, nous souhaitons remercier toute personne ayant contribué de près ou de loin à la réalisation de ce travail.

Résumé

Notre travail consiste à développer un système de surveillance avancé pour détecter les personnes, reconnaître leurs visages et identifier les intrus. Nous avons exploré différentes méthodes de détection en temps réel en utilisant des outils tels qu'une caméra classique et un système embarqué.

Après avoir analysé les différentes approches, nous avons opté pour la méthode SSD (Single Shot MultiBox Detector), en raison de sa haute précision de 97% et sa vitesse pour la détection des objets. Nous avons adapté et implémenté l'algorithme SSD pour la détection de personnes. Ensuite, nous avons implémenté un algorithme appelé SSD-FID (SSD For Intrusion Detection) de reconnaissance faciale, afin de pouvoir identifier les intrus permettant de distinguer les personnes autorisées des personnes non autorisées. Plusieurs cas d'expérimentation ont été effectués afin de montrer l'efficacité de notre approche.

Mots clés : Système de surveillance, Détection de personnes, Reconnaissance faciale, Identifier les intrus, SSD, Système embarqué, Intrusion.

Abstract

Our work consists of developing an advanced surveillance system to detect individuals, recognize their faces, and identify intruders. We have explored various real-time detection methods using tools such as a classic camera and an embedded system.

After analyzing different approaches, we have chosen the SSD (Single Shot MultiBox Detector) method due to its high accuracy of 97% and its speed for object detection. We have adapted and implemented the SSD algorithm for person detection. Furthermore, we have implemented an algorithm called SSD-FID (SSD for Intrusion Detection) for facial recognition, allowing us to identify intruders and distinguish authorized individuals from unauthorized ones. Multiple experimentation cases have been conducted to demonstrate the effectiveness of our approach.

Keywords : Surveillance system, Person detection, Facial recognition, Identify intruders, SSD, Embedded system.

Table des matières

Dédicace	I
Dédicace	II
Remerciements	III
Introduction générale	1
1 Systeme de surveillance humain	4
1.1 Introduction	5
1.2 Processus de surveillance	5
1.2.1 Détection d'objet	6
1.2.2 Détection humaine	6
1.2.3 Détection faciale	6
1.2.4 Reconnaissance faciale	7
1.3 Systèmes de surveillance	8
1.3.1 Caméras de surveillance	8
1.3.2 Système de surveillance lié à un ordinateur	8
1.3.3 Système de surveillance embarqué	9
1.4 Vers des systèmes embarqués sur ordinateurs monocartes	10
1.4.1 Raspberry Pi	12
1.5 Conclusion	17
2 État de l'Art des méthodes de détection et reconnaissance humaine	18
2.1 Introduction	19
2.2 Travaux connexes	19
2.2.1 Approche basée sur les caractéristiques du visage(Face Feature Based)	19
2.2.2 Approche basée sur les caractéristiques du mouvement (Motion Feature Based)	22
2.2.3 Approche basée sur l'apparence corporelle(Body Appearance Based)	25
2.2.4 Approche Basée sur l'apprentissage en profondeur(Deep Learning Based)	27
2.2.5 Tableau comparatif des solutions lue	33
2.2.6 Synthèse de la comparaison	35
2.3 Conclusion	35
3 Notre proposition : SSD pour la détection d'intrusion	36
3.1 Introduction	37

Table des matières

3.2	Choix de la méthode de détection d'intrusion	37
3.3	Description de la solution proposée	37
3.3.1	Fonctionnement du modèle SSD-FID	38
3.3.2	Détection et reconnaissance en utilisant une caméra	41
3.3.3	Détection et reconnaissance en utilisant un système embarqué (Raspberry Pi4)	42
3.3.4	Algorithme de détection et reconnaissance choisi dans notre solution	45
3.4	Environnement de développement	47
3.5	Expérimentation et discussion des résultats	49
3.5.1	Détection Humain	49
3.5.2	Résultats de détection et reconnaissance faciale	52
3.6	Conclusion	60
	Conclusion et perspectives	61

Table des figures

1.1	Systeme de surveillance	5
1.2	Detection d'objet	6
1.3	Détection humaine	7
1.4	Détection faciale	7
1.5	Reconnaissance faciale	8
1.6	Architecture du processus de traitement	9
1.7	Exemple de type de caméras existantes	9
1.8	Exemple des systèmes embarqués	10
1.9	Les ordinateurs monocartes	10
1.10	La carte Raspberry Pi modèle A (à gauche) et modèle B (à droite)	12
1.11	Une image réel de la Raspberry Pi 4	12
1.13	Raspberry Pi 4B sous un angle.	14
1.14	périphériques-La Raspberry pi 4	15
1.15	Pi4-GPIO	16
1.16	Raspberry Pi OS	17
2.1	Classification des techniques de détection humaine	19
2.2	Approche basé sur les caractéristiques du visage	20
2.3	Modélisation de la couleur de la peau	20
2.4	Détection des parties du visage à l'aide de l'algorithme de Viola Jones	21
2.5	Reconnaissance des visages à l'aide de l'algorithme LBPH	21
2.6	Géométrie du visage	22
2.7	Approche basée sur les caractéristiques du mouvement	22
2.8	Technique de soustraction de l'arrière-plan	23
2.9	Technique de différenciation des trame	23
2.10	Histogram of Oriented Optical Flow (HOOF)	24
2.11	flux optique	24
2.12	Approche basée sur l'apparence corporelle	25
2.13	Histogramme du gradient orienté	26
2.14	CENTRIST	26
2.15	Edgelets	27
2.16	Approche Basés sur l'apprentissage en profondeur	28
2.17	Architecture et composition d'un réseau des neurones convolutifs (CNN)	28
2.18	R_{CNN}	29
2.19	Fast R-CNN	29
2.20	Architecteur du modèle Faster-R-CNN	29
2.21	Architecture du SSD	30
2.22	single shot detector (SSD)	30

2.23	Architecture de la méthode Yolo	31
2.24	YOLO	32
3.1	diagramme de la solution	38
3.2	Diagramme de la methode SSD_FID.	39
3.3	Exemples d'images humain avant le prétraitées.	39
3.4	Exemple d'image après le prétraitement	40
3.5	Afficher l'image avec les rectangles entourant les humaines détectés	40
3.6	Schéma de la solution proposé avec un ordinateur.	42
3.7	Schéma de traitement avec la Raspberry Pi 4	44
3.8	logo du python.	47
3.9	logo du goole colab.	48
3.10	logo du spyder.	48
3.11	logo du Inkscape.	49
3.12	Exemple Detection humain	50
3.13	Exemple de détection humaine pendant la nuit	50
3.14	Personnes detectées et enregistrées individuellement	51
3.15	Exemple reconnaissance facial	53
3.16	Détection des personnes inconnu	54
3.17	Décision	55
3.18	donner un nom a la personne	55
3.19	Visage ajouter a la base de deonnée	55
3.20	Resultat Final	56
3.21	teste sur Alina	57
3.22	Détection de visage inconnu	58
3.23	Décision	58
3.24	Resultat final	58

Liste des tableaux

Liste des algorithmes

1	Algorithme SSD pour la détection des humains	41
2	Algorithme de détection d'humains et reconnaissance faciale	45

Liste des acronymes

ASM	<i>Active Shape Model</i>
CNN	<i>Convolutional Neural Network</i>
HMM	<i>Hidden Markov Model</i>
HOG	<i>Histogram of Oriented Gradients</i>
HOOF	<i>Histogram of Oriented Optical Flow</i>
LBPH	<i>Local Binary Patterns Histograms</i>
PCA	<i>Principal Component Analysis</i>
PDM	<i>Point Distribution Model</i>
REST	<i>Representational State Transfer</i>
SBC	<i>Single Board Computers</i>
SIM	<i>Subscriber Identification Module</i>
SSD	<i>Single Shot MultiBox Detector</i>
SSD-FID	<i>Single Shot MultiBox Detector For intrusion detection</i>
SVM	<i>Support Vector Machine</i>
YOLO	<i>You Only Look Once</i>

Introduction générale

Contexte

La surveillance est devenue un élément essentiel de notre société moderne, conçue pour assurer la sécurité et la protection de nos espaces publics et privés. Elle peut également être utilisée pour l'automatisation des tâches quotidiennes et la réduction des coûts. Les progrès technologiques ont permis le développement de systèmes de surveillance de plus en plus sophistiqués, offrant des fonctionnalités telles que la détection humaine, la détection de visage, la reconnaissance faciale et la détection d'intrusion. L'objectif principal de ces systèmes de surveillances est d'identifier les individus présents dans une zone surveillée, de déterminer s'ils sont connus ou inconnus et de prendre les mesures appropriées en fonction de ces informations. La détection humaine reconnaît la présence d'une personne dans le champ de vision de la caméra, tandis que la reconnaissance faciale vise à identifier spécifiquement une personne en comparant son visage à une base de données de visages connus et détecter les intrus.

Différents chercheurs proposent régulièrement de nouvelles méthodes de détection humaine. Parallèlement, d'autres réalisent des revues et des comparaisons pour recenser quelques approches récentes mentionnées dans le chapitre 2. Dans notre étude, nous avons examiné plusieurs articles et regroupé ces approches en quatre catégories, que nous avons étudiées dans un état de l'art. La première approche repose sur les caractéristiques du visage, notamment la couleur de peau, la texture faciale et la géométrie du visage qui permet de détecter les visages humains avec précision. Ensuite, l'approche basée sur les caractéristiques du mouvement utilise des techniques telles que la différenciation d'images, le flux optique, le vecteur de mouvement, la méthode HOOF et la géométrie spatiale qui est utile dans des scénarios tels que la surveillance vidéo. La troisième approche exploite l'apparence corporelle en utilisant des méthodes comme HOG, CENTRIS, basé sur les edglets et les descriptions des composants. Enfin, l'approche basée sur l'apprentissage en profondeur, une approche moderne et très prometteuse, est constamment améliorée en termes de précision et de vitesse. Elle utilise des méthodes comme les réseaux de neurones convolutifs (CNN), R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN, YOLO et SSD.

Après avoir regroupé ces approches, nous les avons comparées afin de déterminer la méthode la plus efficace en termes de vitesse et de précision. pour notre projet de fin de cycle sur un système de surveillance pour la détection et la reconnaissance de personne et l'identification des intrus basé sur un système embarqué.

Les systèmes de surveillance offrent de nombreux avantages. Ils renforcent la sécurité en détectant les intrusions, découragent les actes criminels et empêchent les accès non autorisés à

des lieux surveillés. De plus, ils permettent de prévenir les activités malveillantes et facilitent l'identification des personnes responsables des infractions

Problématique

Dans la pratique la détection humaine et la reconnaissance faciale jouent un rôle crucial dans la prévention et la résolution de divers problèmes de sécurité. En effet, les intrusions, les cambriolages, les agressions, les accès non autorisés à des lieux surveillés, les accidents avec délit de fuite et toutes activités malveillantes et de même les problèmes d'identification des personnes impliquées dans des infractions sont autant de situations qui nécessitent un système efficace d'identification des intrus. L'utilisation d'un système de surveillance embarqué renforce considérablement cette capacité.

Objectifs

Dans notre projet de fin de cycle, nous développons un système automatisé utilisant la détection de personne et la reconnaissance faciale pour identifier et reconnaître les intrus. Nous utilisons une caméra, un ordinateur et la Raspberry Pi pour examiner en détail le processus de détection de personne et de reconnaissance faciale. Ces systèmes automatisés de surveillance sont largement utilisés dans divers domaines pour renforcer la sécurité et améliorer la gestion des processus. Ils jouent un rôle essentiel dans la vidéosurveillance pour la protection des bâtiments, des lieux publics et des infrastructures critiques. De plus, ils sont utilisés dans le contrôle du trafic routier, la surveillance environnementale, la gestion des installations industrielles, la prévention des incendies et bien d'autres domaines où une surveillance précise et constante est indispensable pour assurer la sécurité et l'efficacité des opérations.

Nous nous concentrerons particulièrement sur l'approche SSD (Single Shot MultiBox Detector), un algorithme de détection d'objets en temps réel qui se base sur des réseaux de neurones convolutifs pour extraire des caractéristiques visuelles à partir des images capturées par les caméras, adapté pour la détection de personnes, une méthode efficace et rapide. Cependant, nous allons adapter cette approche en ajoutant la fonctionnalité de reconnaissance faciale, ce qui nous permettra d'identifier spécifiquement les visages des personnes et d'ignorer tous les autres objets présents dans l'image.

En modifiant l'approche SSD pour inclure la reconnaissance faciale, nous serons en mesure de détecter les personnes et de les identifier individuellement. Nous mettrons également en place un système d'alerte qui nous informera en temps réel de la présence d'intrus détectés dans l'endroit surveiller.

En résumé, notre objectif est de développer un système automatisé d'identification et de reconnaissance des intrus, basé sur la détection de personne et la reconnaissance faciale, en ajoutant une couche supplémentaire à notre approche SSD, appelé SSD_FID (FOR INTROSION DETECTION). Ce système sera doté d'un système d'alerte pour assurer une réponse rapide face à la présence d'intrus, ajouter à la base de données les personnes autorisés d'accéder aux endroits

surveillés s'il ont été identifiés comme intrus pour la première fois et de mettre dans une liste noir les intrus.

Organisation du mémoire

Ce mémoire est organisé en trois chapitres qui abordent différents aspects liés au système de surveillance humain.

Le premier chapitre, intitulé "Système de surveillance humaine", se concentre sur la présentation et la description du système de surveillance, qui met en évidence ses différents domaines tels que la détection d'objet, de personne et de la reconnaissance faciale, ainsi que l'utilisation d'un système embarqué.

Le deuxième chapitre, intitulé "État de l'art", examine les avancées récentes dans le domaine de la détection humaine et de la reconnaissance faciale. Il explore les différentes approches et techniques utilisées, y compris l'approche SSD mentionnée précédemment, et présente un aperçu des recherches et des technologies les plus pertinentes dans le domaine.

Le troisième chapitre, intitulé "Solution proposée et analyse", propose une solution spécifique pour améliorer la détection de personnes et la reconnaissance faciale. Ce chapitre détaille les étapes de mise en œuvre de la solution proposée. Il inclut également une analyse des résultats obtenus, en évaluant l'efficacité et l'exactitude de la détection et la reconnaissance.

Chapitre 1

Systeme de surveillance humain

1.1 Introduction

Dans un monde en constante évolution, la sécurité et la surveillance jouent un rôle crucial dans la protection des personnes et des biens. Les systèmes de surveillance basés sur des systèmes embarqués ou des ordinateurs PC offrent des solutions avancées pour répondre à ces besoins. Ces systèmes intègrent des capteurs, des caméras et des logiciels sophistiqués pour détecter, analyser et enregistrer les activités en temps réel. Que ce soit pour la sécurité résidentielle, la surveillance d'entreprise ou la surveillance publique, ces systèmes offrent une surveillance fiable, une gestion efficace des données et des fonctionnalités avancées telles que la reconnaissance faciale et la détection d'intrusion. Dans cet article, nous explorerons les avantages et les applications d'un système de surveillance basé sur un système embarqué ou un ordinateur PC, et comment il peut améliorer la sécurité dans divers environnements.

1.2 Processus de surviellance

Le système de surveillance humain est une application de la technologie de la vision par ordinateur qui permet de détecter, reconnaître et suivre les mouvements des personnes à l'aide de caméras. Ces systèmes sont utilisés pour la sécurité, la prévention des crimes, le contrôle d'accès et d'autres besoins spécifiques voir figue 1.1. Ils permettent de dissuader les comportements indésirables, de fournir des preuves en cas d'incident et d'améliorer la sécurité des personnes et des biens. Son fonctionnement est basé sur des caméras pour capturer des images et des vidéos de l'environnement. Les algorithmes de traitement d'image sont utilisés pour détecter les mouvements, les formes humaines reconnaître les personnes et leurs actions[4] [20].

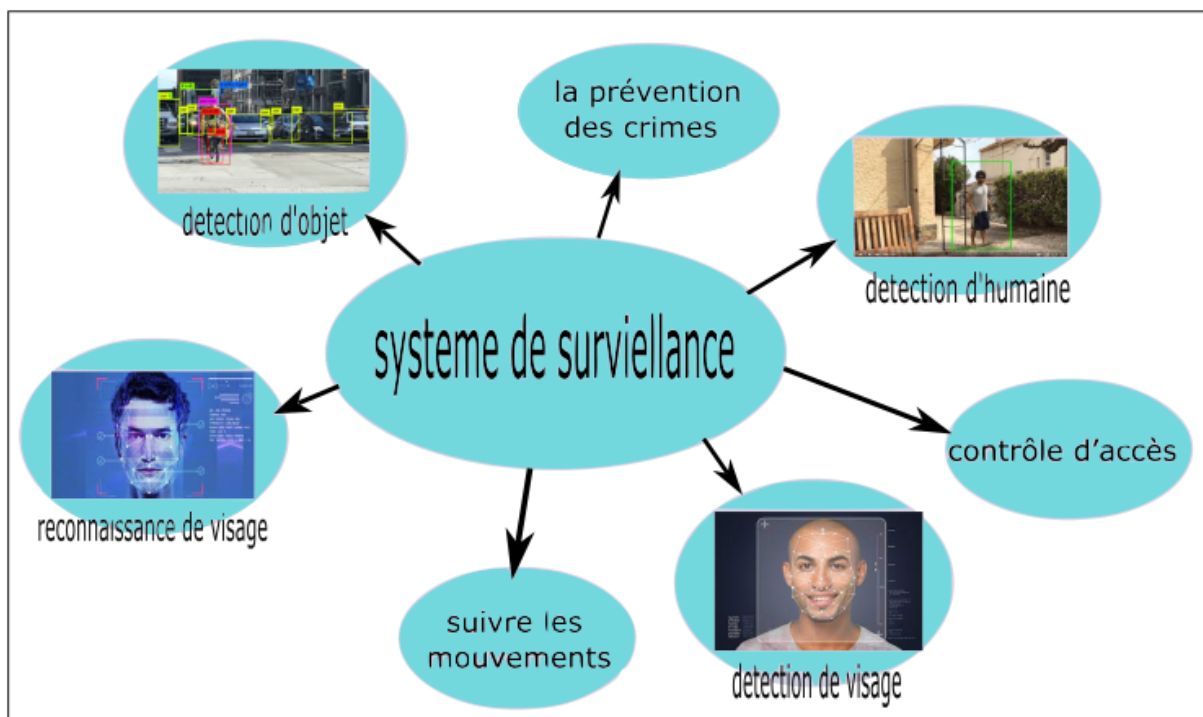


Fig. 1.1 : Systeme de surviellance

1.2.1 Détection d'objet

La détection d'objets est une tâche essentielle en vision par ordinateur. Elle consiste à repérer et identifier des objets spécifiques tels que des humains, des animaux ou des voitures dans des images numériques. La détection d'objets vise à fournir une compréhension approfondie de l'environnement visuel et à permettre aux systèmes informatiques de prendre des décisions et d'interagir de manière plus intelligente avec le monde réel[51][47] [45] [24] [23][34].

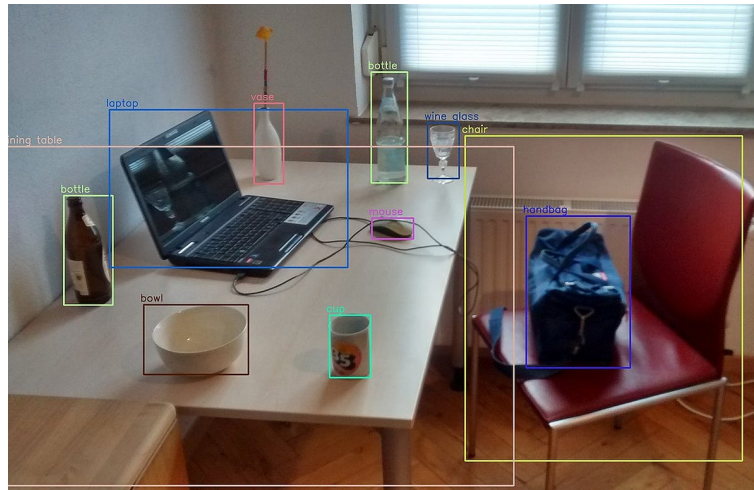


Fig. 1.2 : Détection d'objet

1.2.2 Détection humaine

La détection humaine en temps réel dans une vidéo est une technologie de surveillance qui utilise des algorithmes de vision par ordinateur pour détecter et localiser les humains présents dans une vidéo en direct. Cette technologie est souvent utilisée dans les domaines de la sécurité, de la surveillance des transports, de la surveillance de la foule, de la reconnaissance des comportements et des émotions, entre autres. La détection humaine en temps réel utilise des caméras et des capteurs pour capturer des images en continu, qui sont ensuite traitées par des algorithmes de détection pour identifier les humains et leur position dans la vidéo en direct. Cette technologie permet aux opérateurs de sécurité et aux professionnels de la surveillance de surveiller et de réagir rapidement à des situations potentiellement dangereuses ou suspectes [44] [2] (voir figure 1.3).

1.2.3 Détection faciale

La détection faciale est une technique de reconnaissance de formes qui détermine l'emplacement et la taille d'un visage humain dans une image ou une vidéo. Le but de la détection faciale est d'identifier des personnes dans les images et de les différencier les unes des autres. Les systèmes de détection faciale utilisent souvent des algorithmes de vision par ordinateur pour analyser des images à partir de caméras de surveillance, de téléphones portables ou d'autres sources.

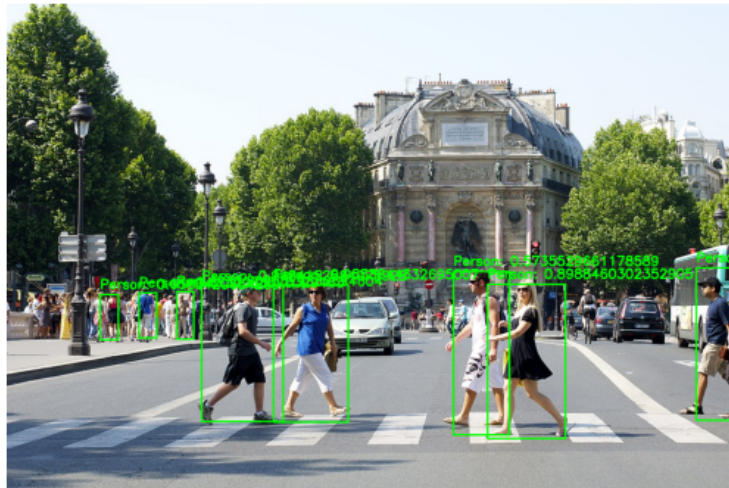


Fig. 1.3 : Détection humaine

Les algorithmes de détection faciale fonctionnent généralement en recherchant des caractéristiques spécifiques du visage, comme les yeux, le nez, la bouche et les oreilles, pour identifier les personnes [18] [42].

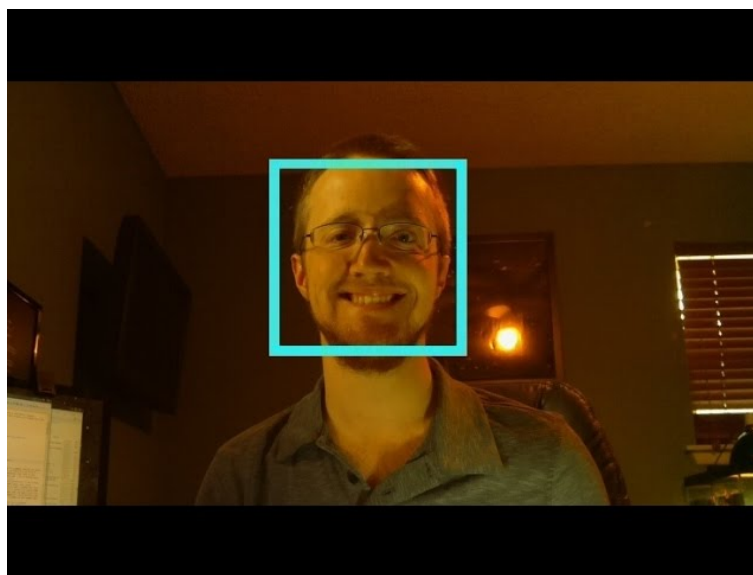


Fig. 1.4 : Détection faciale

1.2.4 Reconnaissance faciale

La reconnaissance faciale est une technologie utilisée pour identifier ou vérifier l'identité d'une personne en analysant et en comparant les caractéristiques uniques de son visage. La reconnaissance faciale repose sur des algorithmes informatiques qui utilisent des caractéristiques spécifiques du visage, telles que la distance entre les yeux, la forme du nez et de la bouche, ainsi que les contours du visage, pour effectuer leur analyse [48] [1].

Cette technologie est largement utilisée dans divers domaines, tels que la sécurité, la gestion des identités, la surveillance et la vérification biométrique. Elle est également utilisée de manière ludique dans des applications telles que les filtres de réalité augmentée sur les réseaux sociaux [31] [27] (voir figure 1.5).

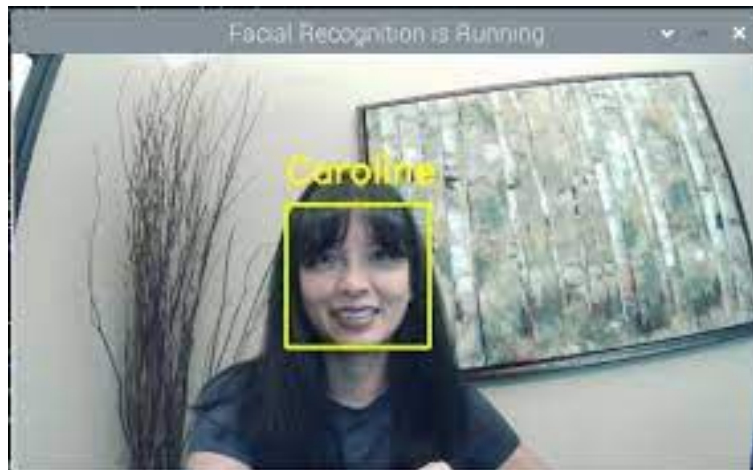


Fig. 1.5 : Reconnaissance faciale

1.3 Systèmes de surveillance

Le processus de traitement automatique des données provenant d'une caméra peut être réalisé à la fois avec une caméra liée à une machine ordinateur ou bien avec une caméra liée à un système embarqué. Il suit une architecture de traitement représenté par la figure suivante 1.6 :

1.3.1 Caméras de surveillance

Une caméra de surveillance est un dispositif électronique conçu pour détecter et enregistrer des informations visuelles ou sensorielles dans le cadre d'un système de surveillance. Elle capture des images ou des vidéos en temps réel, permettant de visualiser et ce qui se passe dans une zone surveillée. Il est généralement utilisé pour surveiller et enregistrer des images, activités, des mouvements ou des événements dans un environnement spécifique.

Ce moyen d'acquisition d'informations peut être fixe (dans les maisons, les entreprises, au bord des routes, etc) et peut être embarqué pour le rendre mobile.

1.3.2 Système de surveillance lié à un ordinateur

Lorsqu'une caméra est connectée à une machine informatique, elle est généralement connectée physiquement à l'ordinateur via un câble ou une interface appropriée. Le flux de données de la caméra est transféré vers un ordinateur où il est traité par un logiciel spécialisé. L'ordinateur agit comme la plate-forme principale de traitement. Les données de la caméra sont capturées et transmises à un logiciel de traitement d'image ou de vision par ordinateur exécuté sur un

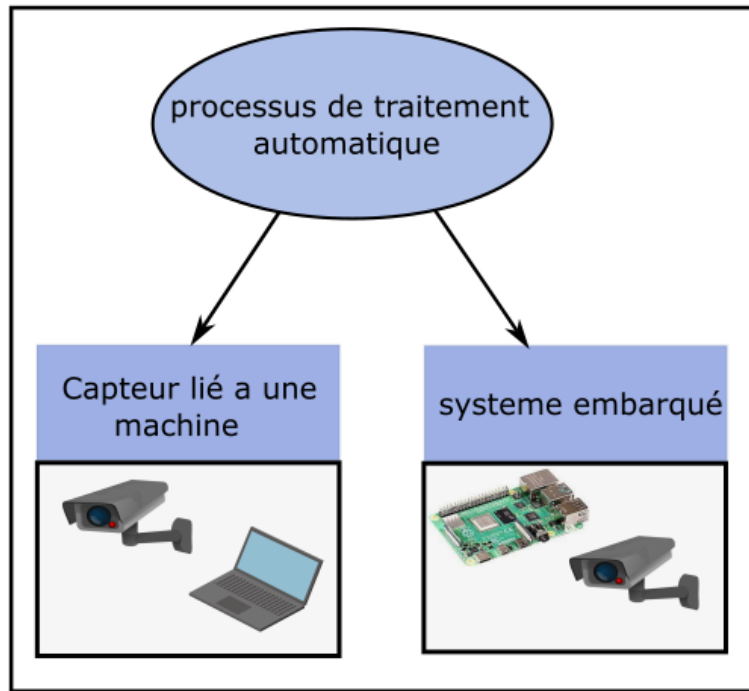


Fig. 1.6 : Architecture du processus de traitement

ordinateur. Ces logiciels peuvent appliquer des algorithmes de détection, de suivi, de reconnaissance ou d'autres techniques de traitement d'images pour extraire des informations spécifiques d'images ou de vidéos (voir figure 1.7).



Fig. 1.7 : Exemple de type de caméras existantes

1.3.3 Système de surveillance embarqué

Les systèmes de surveillance sont de plus en plus présents dans la vie quotidienne. Les systèmes embarqués sont des composants essentiels qui permettent de surveiller et de sécuriser divers environnements. Ces systèmes intègrent des technologies électroniques et informatiques spécialisées pour assurer des fonctionnalités avancées de surveillance, de détection et de traitement des données. Les systèmes de surveillance embarqués peuvent inclure des caméras de

surveillance, des capteurs de mouvement, des capteurs de température, des systèmes de reconnaissance faciale, des systèmes de gestion de vidéosurveillance, des enregistreurs numériques, des dispositifs de stockage et de traitement des données, ainsi que des logiciels spécialisés pour analyser et interpréter les informations recueillies (voir figure 1.8).



Fig. 1.8 : Exemple des systèmes embarqués

1.4 Vers des systèmes embarqués sur ordinateurs monocartes

Les ordinateurs monocartes, également appelés SBC (Single Board Computers), sont des ordinateurs complets qui sont intégrés sur une seule carte électronique. Ils sont compacts et portables, ce qui les rend pratiques pour différentes applications. Les SBC rassemblent tous les composants essentiels tels que le processeur, la mémoire, la connectivité réseau et les ports d'entrée/sortie sur une seule carte. Ils peuvent fonctionner avec différents systèmes d'exploitation et sont utilisés dans divers domaines tels que l'informatique embarquée, l'Internet des objets (IoT), l'automatisation industrielle et les projets éducatifs. Les SBC offrent une grande flexibilité et sont adaptés aux besoins spécifiques des utilisateurs [5]. voir différents exemples d'ordinateurs monocartes dans l'image ci-dessous fig : 1.9 [22]



Fig. 1.9 : Les ordinateurs monocartes

Il existe plusieurs familles de SBC qui se distinguent par leurs caractéristiques et leurs fonctionnalités. Voici quelques-unes des familles les plus courantes :

- **Arduino**

La plateforme Arduino a été créée en 2005 par des étudiants de l'Interaction Design Institute Ivrea en Italie. Depuis lors, de nombreuses variantes d'Arduino ont été développées, telles que l'Arduino Uno, l'Arduino Mega et l'Arduino Nano. Bien qu'Arduino soit principalement reconnu pour ses outils de prototypage électronique, il propose également des SBC compacts et puissants. Les cartes Arduino sont fréquemment utilisées dans des projets électroniques personnalisés, des objets connectés, des systèmes de contrôle, et bien d'autres applications [46].

- **Odroid**

La société coréenne Hardkernel a développé les SBC Odroid. Leur premier modèle, l'Odroid board, a été lancé en 2009. Depuis lors, d'autres modèles populaires tels que l'Odroid-XU4, l'Odroid-C2 et l'Odroid-N2 ont été introduits. Les SBC Odroid sont réputés pour leur puissance de traitement élevée et leur compatibilité avec divers systèmes d'exploitation. Ils sont couramment utilisés dans des projets impliquant des médias, des serveurs, des applications de jeu, et bien d'autres domaines [21].

- **BeagleBone**

BeagleBoard.org a introduit le premier BeagleBone, connu sous le nom de BeagleBone White, en 2011. Depuis lors, d'autres modèles tels que le BeagleBone Black, le BeagleBone Green et le BeagleBone AI ont été développés. La famille BeagleBone propose des SBC dotés de fonctionnalités avancées et d'une connectivité étendue. Ils sont largement utilisés dans des domaines tels que l'industrie, l'automatisation et les projets de robotique, offrant des solutions polyvalentes pour répondre aux besoins spécifiques de ces applications [28].

- **Raspberry Pi**

En 2012, le Raspberry Pi a été lancé avec son premier modèle, le Raspberry Pi 1 Model B. Depuis lors, de nombreux autres modèles ont été développés, tels que le Raspberry Pi 2, le Raspberry Pi 3, le Raspberry Pi 4 et le Raspberry Pi Zero. Les SBC de la famille Raspberry Pi sont extrêmement populaires et largement utilisés. Ils offrent un excellent équilibre entre performances, polyvalence et prix abordable. Les modèles Raspberry Pi sont couramment utilisés dans des projets éducatifs, des systèmes de domotique, des serveurs personnels et bien d'autres applications [5].

- **NVIDIA Jetson**

En 2014, NVIDIA a lancé la famille des SBC NVIDIA Jetson avec le Jetson TK1, suivi en 2015 par le Jetson TX1, en 2017 par le Jetson TX2, et en 2018 par le Jetson Xavier. Par la suite, d'autres modèles tels que le Jetson Nano et le Jetson AGX Xavier ont été introduits. Les SBC de la famille NVIDIA Jetson sont spécialement conçus pour répondre aux besoins des applications d'intelligence artificielle et de traitement d'images. Ils offrent des performances élevées pour le calcul parallèle, ce qui les rend très prisés dans des projets tels que la vision par ordinateur et les drones autonomes, entre autres utilisations [35].

Après avoir évoqué plusieurs modèles d'ordinateurs monocarte, notre attention s'est portée plus spécifiquement sur le Raspberry Pi.

1.4.1 Raspberry Pi

Le Raspberry Pi est ordinateur monocartes peu coûteux, de la taille d'une carte de crédit, développés par la Fondation Raspberry Pi au Royaume-Uni.[5]. La Fondation Raspberry Pi propose plusieurs distributions de systèmes d'exploitation populaires, et le Raspberry Pi peut être programmé dans divers langages tels que Python, C, C++ ,Java et matlab. La caméra Raspberry Pi est un accessoire spécialement conçu pour cet ordinateur, permettant de capturer des images et des vidéos de haute qualité. Elle est utilisée dans de nombreux projets, tels que la surveillance, la robotique, la domotique, la photographie et la capture de vidéos en temps réel. La caméra Raspberry Pi offre une solution abordable et polyvalente pour l'acquisition d'images et de vidéos, répondant ainsi aux besoins des amateurs, des étudiants, des développeurs et des passionnés du Raspberry Pi [30].

a. Modèles Raspberry Pi : il existe differente modèles de Raspberry [36] :



Fig. 1.10 : La carte Raspberry Pi modèle A (à gauche) et modèle B (à droite)

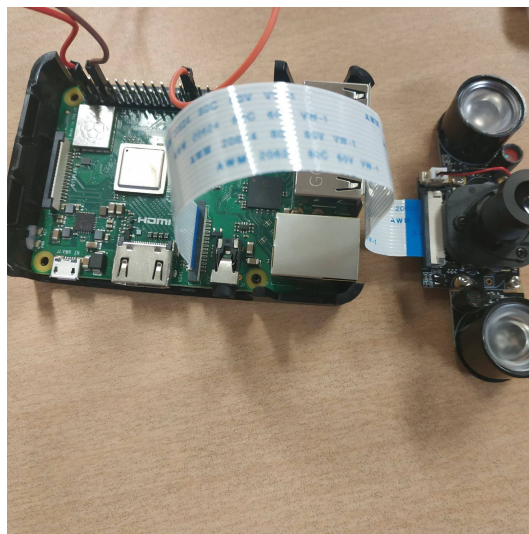


Fig. 1.11 : Une image réel de la Raspberry Pi 4

- **Raspberry Pi Model B**

Est le premier modèle de Raspberry Pi, a été lancé en février 2012. Il s'agissait d'un petit ordinateur de carte unique (SBC) doté d'un processeur ARM11, de 256 Mo de RAM, de ports USB et Ethernet, ainsi que d'un port HDMI. Son prix abordable et sa flexibilité en ont fait un énorme succès. En 2014, le Raspberry Pi Model B a été amélioré avec le lancement du Raspberry Pi Model B+, qui offrait une plus grande capacité de mémoire, plus de ports USB et une meilleure gestion de l'alimentation.

- **Raspberry Pi 2 Model B**

le Raspberry Pi 2 Model B a été introduit en 2015, avec un processeur plus puissant et 1 Go de RAM, ce qui a permis d'accroître les performances et d'ouvrir de nouvelles possibilités pour les projets basés sur Raspberry Pi.[17]

- **Raspberry Pi 3 Model B**

En 2016, le Raspberry Pi 3 Model B a été lancé, apportant des améliorations supplémentaires avec un processeur plus rapide, le support du Wi-Fi et du Bluetooth intégrés, ainsi que la prise en charge de la norme 64 bits. Depuis lors, de nouveaux modèles de Raspberry Pi ont été introduits, tels que le Raspberry Pi 3 Model B+ en 2018

- **Raspberry Pi 4 Model B**

En 2019, une version améliorée du Raspberry Pi, le Raspberry Pi 4 Model B, a été lancée. Cette nouvelle itération a considérablement renforcé les performances, la capacité de mémoire et les fonctionnalités de la plateforme. Il était désormais possible d'obtenir des options de RAM allant jusqu'à 8 Go, offrant ainsi une puissance de traitement accrue pour une variété de projets. Le Raspberry Pi 4 Model B a été accueilli avec enthousiasme par la communauté, ouvrant de nouvelles opportunités pour des applications plus gourmandes en ressources et des utilisations plus avancées de la carte.

Ces différents modèles de Raspberry Pi ont été adoptés par une communauté mondiale, d'éducateurs, de bricoleurs et de professionnels, qui ont exploité leur potentiel pour créer une multitude de projets allant de la robotique à l'informatique embarquée, en passant par les serveurs domestiques, la domotique et bien d'autres domaines. Les modèles de Raspberry Pi continuent d'évoluer, offrant des fonctionnalités de plus en plus puissantes et de nouvelles possibilités d'innovation. Ils ont joué un rôle majeur dans la promotion de l'apprentissage de l'informatique et de la créativité technologique, tout en restant abordables et accessibles pour tous.

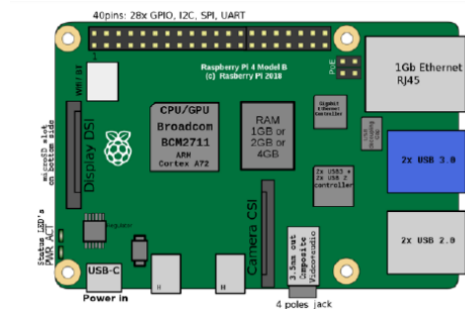
b. Raspberry pi 4 modèle B

le Raspberry Pi 4 Model B est un ordinateur de carte unique SBC qui a intéressé de nombreux passionnés de technologie. Avec son processeur quad-core et des options de mémoire RAM allant jusqu'à 8 Go, le Raspberry Pi 4 offre des performances considérablement améliorées par rapport à ses prédécesseurs. Grâce à ses ports USB 3.0, ses ports micro HDMI prenant en charge la résolution 4K et sa connectivité Wi-Fi et Bluetooth intégrée, ce modèle offre de nombreuses possibilités pour l'apprentissage, la création de projets et le développement de compétences en informatique. Que ce soit pour la programmation, l'exploration de l'électronique ou la mise en place de projets IoT, le Raspberry Pi 4 est un outil abordable et polyvalent qui ouvre de nouvelles

perspectives d'apprentissage et d'expérimentation pour les étudiants [37].
(voir La figure 1.4.1)



(a) Raspberry Pi 4B vue de dessus.



(b) Raspberry Pi 4B schéma

La photo ci-dessous présente une vue au dessous du Raspberry Pi modèle 4B 1.13



Fig. 1.13 : Raspberry Pi 4B sous un angle.

b.1 Caractéristiques

1. **Processeur** : Le Raspberry Pi 4 est équipé d'un processeur Broadcom BCM2711 quad-core ARM Cortex-A72 cadencé jusqu'à 1,5 GHz, offrant une puissance de calcul considérablement accrue par rapport aux modèles précédents.
2. **Mémoire RAM** : Le Raspberry Pi 4 est disponible avec des options de mémoire RAM allant de 2 Go à 8 Go, permettant des performances plus rapides et une gestion efficace des tâches gourmandes en mémoire.
3. **Connectivité** : Le Raspberry Pi 4 dispose de deux ports USB 3.0 pour des transferts de données rapides, ainsi de deux ports USB 2.0. Il est également équipé de deux ports micro HDMI prenant en charge une résolution 4K pour la sortie vidéo. De plus, il dispose d'un port Gigabit Ethernet, d'une prise en charge du Wi-Fi 802.11ac et du Bluetooth 5.0 intégrés pour une connectivité sans fil.
4. **Stockage** : Le Raspberry Pi 4 utilise une carte microSD pour le stockage du système d'exploitation et des données. Il dispose également de ports pour la connexion de périphériques de stockage externes tels que des disques durs ou des clés USB.

5. **GPIO(General Purpose Input/Output)** : Le Raspberry Pi 4 offre un certain nombre de broches GPIO pour permettre la connexion de composants électroniques supplémentaires, ce qui en fait un outil idéal pour l'apprentissage de l'électronique et la création de projets interactifs.[36]
6. **Système d'exploitation** : Le Raspberry Pi 4 est compatible avec une variété de systèmes d'exploitation, notamment Raspbian (maintenant connu sous le nom de Raspberry Pi OS), Ubuntu, et d'autres distributions Linux.
7. **Alimentation** : Le Raspberry Pi 4 nécessite une alimentation de 5V via un adaptateur secteur USB-C. Il est également possible d'utiliser des batteries ou d'autres sources d'alimentation appropriées pour une utilisation portable.

b.2 Périphériques La Raspberry Pi 4 est équipée de divers périphériques et connecteurs qui permettent une grande connectivité et une utilisation polyvalente. Voici certains des périphériques et connecteurs importants présents sur la Raspberry Pi 4 [37] :

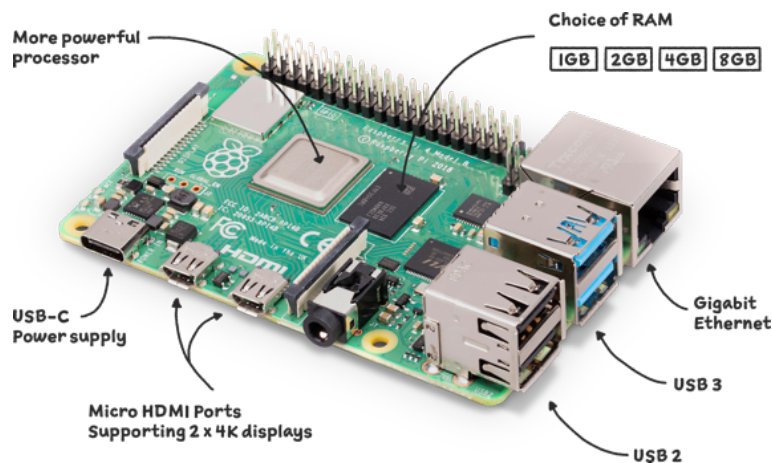


Fig. 1.14 : périphériques-La Raspberry pi 4

- **HDMI** : La Raspberry Pi 4 est équipée de deux ports HDMI, permettant la connexion à des écrans externes pour l'affichage.
- **USB** : Elle dispose de deux ports USB 2.0 et de deux ports USB 3.0, qui permettent de connecter différents périphériques USB tels que claviers, souris, disques durs, etc.
- **Ethernet** : La Raspberry Pi 4 est dotée d'un port Ethernet Gigabit intégré, offrant une connexion filaire haut débit pour la connectivité réseau.
- **Wi-Fi et Bluetooth** : Elle est équipée d'une connectivité sans fil intégrée, prenant en charge le Wi-Fi 802.11ac et le Bluetooth 5.0.
- **GPIO** : La Raspberry Pi 4 dispose de 40 broches GPIO (General-Purpose Input/Output) permettant de connecter et de contrôler divers composants électroniques externes.

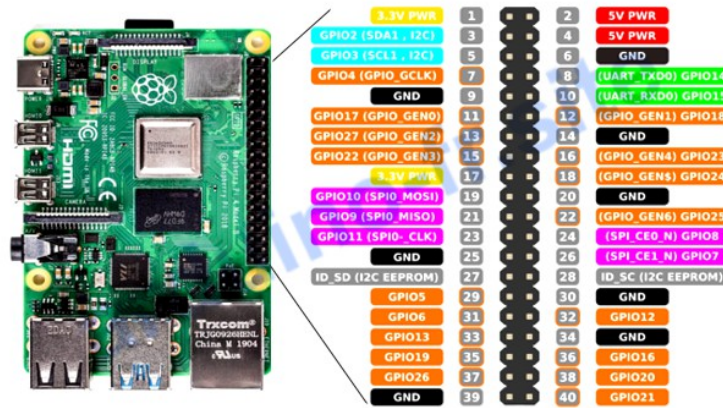


Fig. 1.15 : Pi4-GPIO

- **Lecteur de carte microSD** : Elle utilise une carte microSD pour le stockage du système d'exploitation et des données.
- **Prise audio** : La Raspberry Pi 4 est équipée d'une prise jack 3,5 mm pour la sortie audio.
- **Port d'alimentation** : La Raspberry Pi 4 utilise un connecteur USB-C pour l'alimentation.

b.3 Système D'exploitation

Le système d'exploitation le plus couramment utilisé pour la Raspberry Pi est appelé Raspbian, qui est une distribution basée sur Debian spécialement conçue pour les Raspberry Pi. Cependant, depuis le lancement de la Raspberry Pi 4, le système d'exploitation officiel a été renommé Raspberry Pi OS.

Raspbian Os Raspberry Pi OS est un système d'exploitation open source basé sur Linux spécialement conçu pour les cartes Raspberry Pi. Anciennement connu sous le nom de Raspbian, il est développé par la Fondation Raspberry Pi. Ce système offre une plateforme optimisée pour les projets embarqués et propose une interface utilisateur conviviale. Il est disponible en différentes variantes, dont une version complète avec environnement de bureau graphique, une version légère sans interface graphique, et une version intermédiaire avec une sélection plus large de logiciels préinstallés. Raspberry Pi OS est largement utilisé par la communauté des passionnés de Raspberry Pi et offre une gamme de ressources et de support en ligne pour aider les utilisateurs à exploiter pleinement leur carte Raspberry Pi.

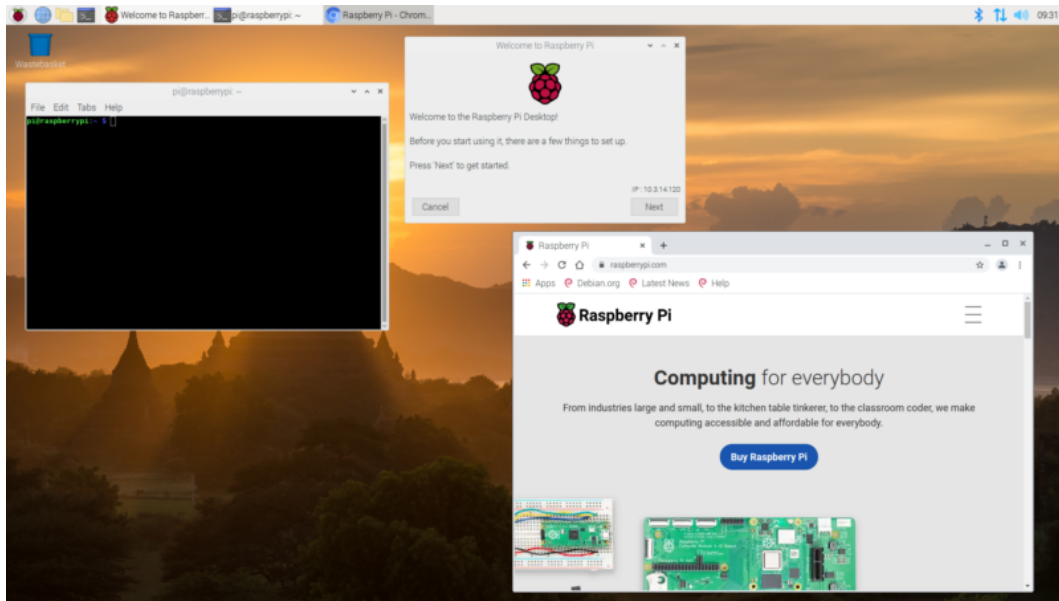


Fig. 1.16 : Raspberry Pi OS

1.5 Conclusion

Nous avons présenté dans ce chapitre les différents modèles qui composent un système de surveillance et plus particulièrement les capteurs et les systèmes embarqués. Le chapitre suivant fera l'objet des différentes méthodes de détection et de reconnaissance faciale.

Chapitre 2

État de l'Art des méthodes de détection et reconnaissance humaine

2.1 Introduction

Le système de surveillance humaine par vidéo est devenu de plus en plus important dans les dernières années. Les vidéos de surveillance sont utilisées pour détecter, suivre et analyser les comportements individuels à des fins de sécurité, tels que la détection de piétons, la reconnaissance de personnes, le comptage de personnes dans les foules denses, etc. Toutefois, l'observation humaine peut être limitée dans les vidéos en temps réel, ce qui nécessite un système de surveillance vidéo automatique.

2.2 Travaux connexes

Des techniques ont été développées dans le but de réaliser une détection humaine et une reconnaissance faciale. La figure 2.1 illustre un récapitulatif de quelques méthodes.

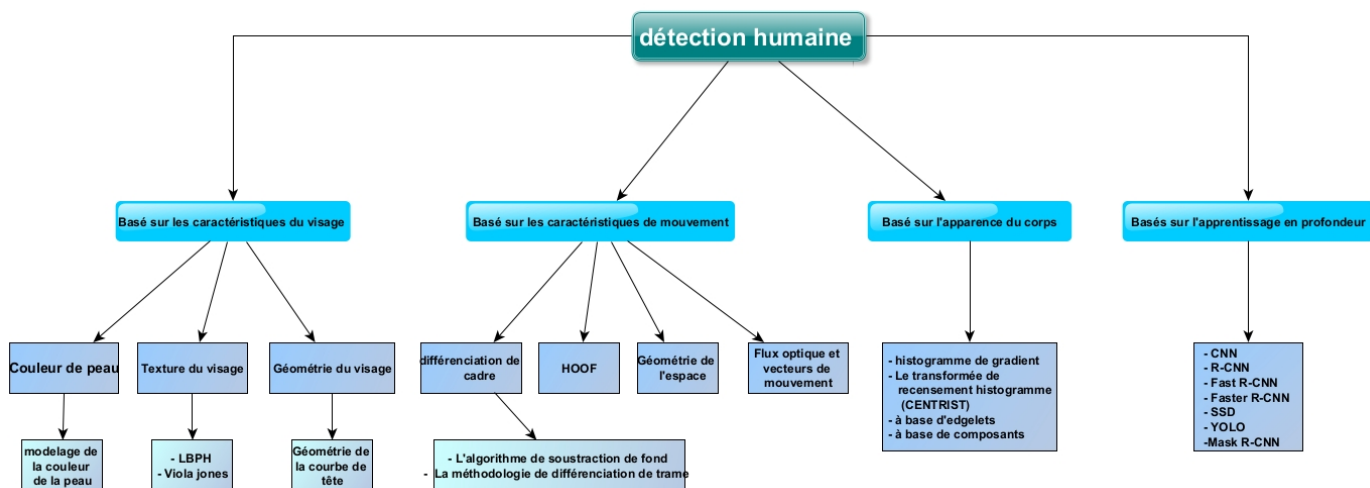


Fig. 2.1 : Classification des techniques de détection humaine

2.2.1 Approche basée sur les caractéristiques du visage (Face Feature Based)

Les détecteurs de personnes qui utilisent les caractéristiques du visage sont capables de déterminer la présence d'une personne dans une image en se focalisant sur son apparence faciale. Les caractéristiques du visage tels que la couleur de peau, la texture du visage et la géométrie faciale voir la figure 2.2. sont utilisées pour classifier la personne comme telle [2]

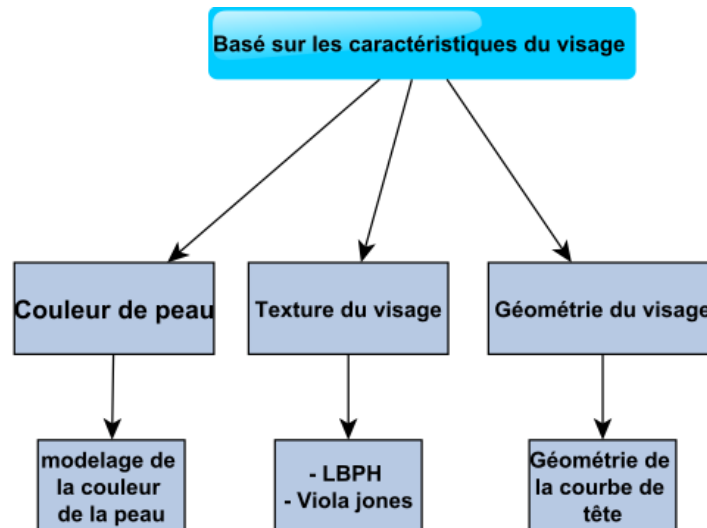


Fig. 2.2 : Approche basé sur les caractéristiques du visage

- **Modélisation de la couleur de la peau** Cette étape consiste à extraire des informations sur la couleur de la peau à partir de l'image du visage. Des techniques telles que la segmentation de la peau sont utilisées pour isoler la région du visage. Des modèles de couleur de la peau peuvent être créés pour classifier les pixels de la peau et différencier les zones de la peau des autres parties du visage voir la figure 2.3. Cette étape est importante car elle permet de caractériser la couleur de la peau pour des applications telles que la détection d'émotions ou l'analyse des teints de peau [2].



Fig. 2.3 : Modélisation de la couleur de la peau

- **Texture du visage (LBPH et Viola-Jones)** Extraire des descripteurs de texture sont extraits à partir de l'image du visage. Une méthode couramment utilisée est le Local Binary Patterns Histograms (LBPH), qui capture les motifs locaux du visage en utilisant des opérateurs de seuillage binaires [49] voir la figure 2.5 [3]. Le détecteur de visage de Viola-Jones est également souvent utilisé pour détecter et extraire des caractéristiques faciales, en utilisant un ensemble de classificateurs basés sur des caractéristiques simples mais

puissantes, telles que les rectangles de Haar. Ces descripteurs de texture permettent de caractériser les détails et les structures du visage, tels que les ridules, les rides ou les textures spécifiques [26] voir la figure 2.4.

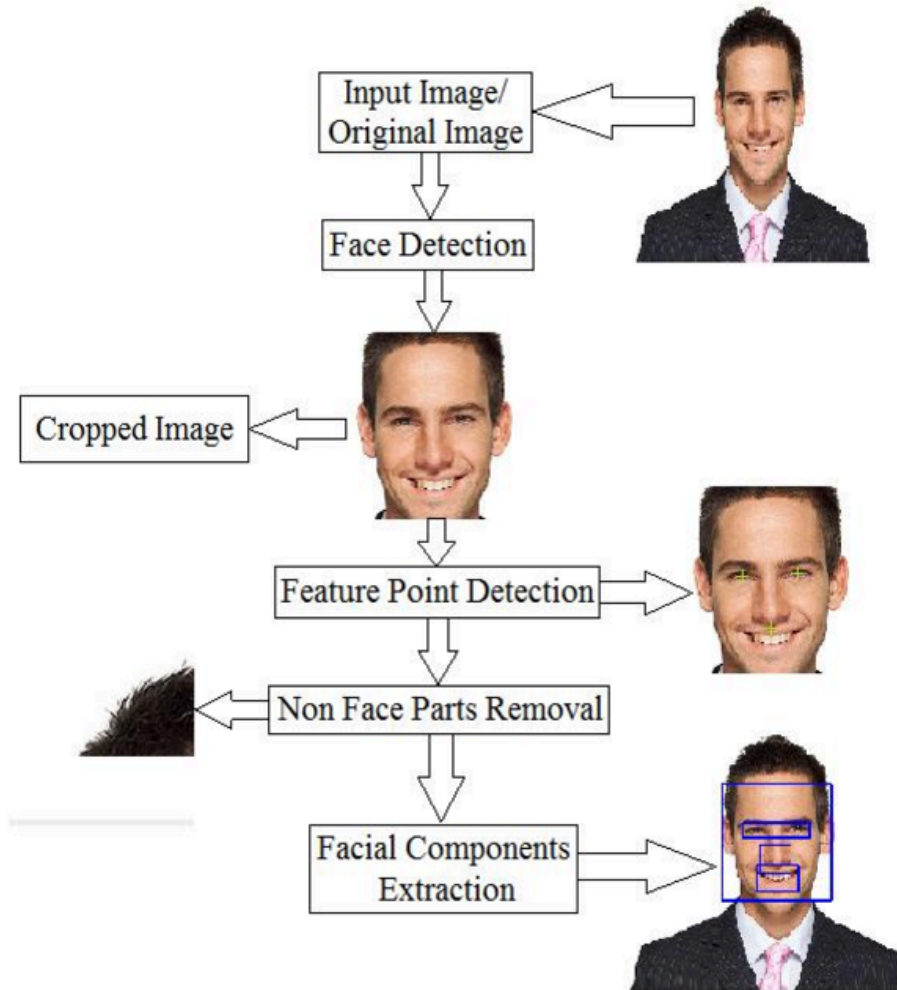


Fig. 2.4 : Détection des parties du visage à l'aide de l'algorithme de Viola Jones

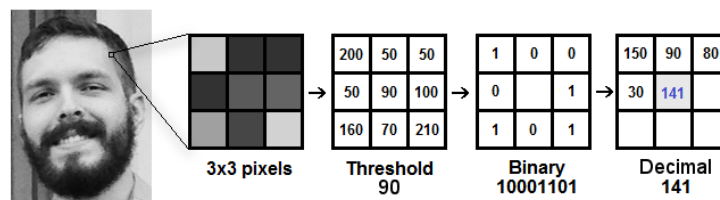


Fig. 2.5 : Reconnaissance des visages à l'aide de l'algorithme LBPH

- **Géométrie du visage (géométrie de la courbe de tête)** Cette étape se concentre sur les caractéristiques géométriques du visage, telles que la position des yeux, du nez, de la bouche, etc. La géométrie de la courbe de tête fait référence à la représentation mathématique des caractéristiques du visage en utilisant des courbes et des points spécifiques. Les techniques de détection des points d'intérêt, comme le détecteur de visage basé sur

les caractéristiques de forme actives (Active Shape Model - ASM), peuvent être utilisées pour localiser et extraire ces caractéristiques géométriques voir la figure 2.6 [25].

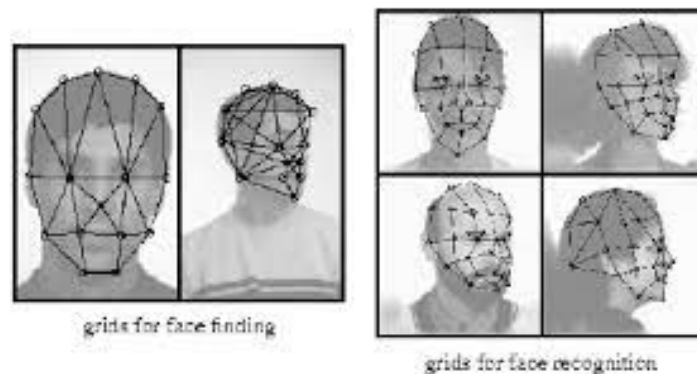


Fig. 2.6 : Géométrie du visage

2.2.2 Approche basée sur les caractéristiques du mouvement (Motion Feature Based)

Les détecteurs humains qui se basent sur les caractéristiques de mouvement analysent les vecteurs de mouvement des pixels dans les images successives pour déterminer la présence de personnes. Il utilise différentes techniques, telles que la différenciation d'images, le flux optique, le vecteur de mouvement, HOOF (Histogram of Oriented Optical Flow), et la géométrie spatiale voir la figure 2.7 suivante.

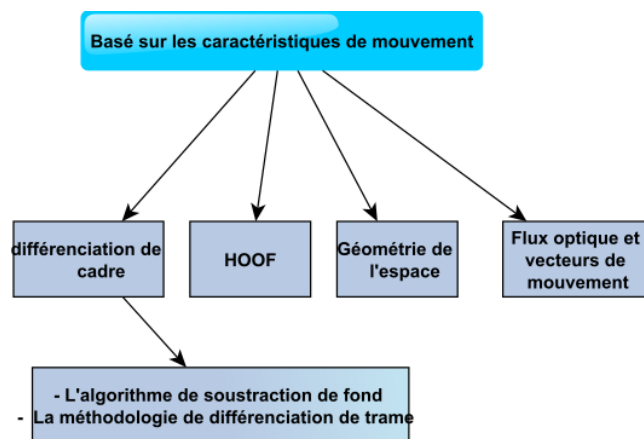


Fig. 2.7 : Approche basée sur les caractéristiques du mouvement

- **Différenciation de cadre** cette étape se concentre sur la détection d'objets en mouvement dans les séquences vidéo qui est une tâche difficile. Les chercheurs se sont intéressés aux méthodologies de différenciation des images basées sur les pixels pour détecter les objets en mouvement à l'aide d'une caméra en temps réel [16]. Ces méthodes consistent

à analyser les différences entre les images pour détecter les objets en mouvement[15]. Ces méthodes sont décrites ci-dessous :

1. Technique de soustraction de l'arrière-plan L'algorithme est utilisé pour trouver les objets en mouvement dans une séquence vidéo. Cependant, l'image résultante peut accumuler du bruit et nécessite donc une étape de post-traitement pour le filtrer. Il est également important que l'image de fond soit régulièrement mise à jour pour qu'elle puisse s'adapter aux conditions changeantes de luminosité et de géométrie [6].

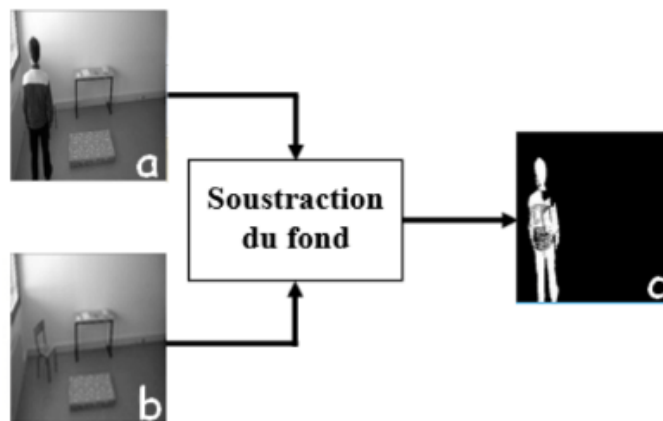


Fig. 2.8 : Technique de soustraction de l'arrière-plan

2. la technique de différenciation des trames est utilisée pour détecter les objets en mouvement à l'aide de la trame capturée par la caméra fixe voir la figure 2.9. Cette méthode est largement adaptative et présente le coût de calcul le plus faible. L'une des limites de cet algorithme est que l'objet doit se déplacer en continu. Si l'objet arrête son mouvement, il devient l'arrière-plan. De plus, cette approche est plus sujette à traiter le bruit et les mouvements dans l'arrière-plan. [6][14].

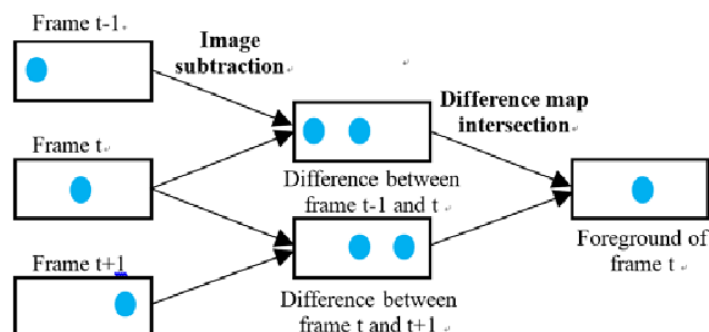


Fig. 2.9 : Technique de différenciation des trame

- **HOOF (Histogram of Oriented Optical Flow)** est une technique qui capture les informations de flux optique et les représente sous forme d'histogrammes orientés. Le flux

optique est une mesure du déplacement apparent des pixels entre les images consécutives. HOOF divise le flux optique en plusieurs directions et calcule les histogrammes des directions du mouvement. Cela permet de capturer les caractéristiques spatio-temporelles du mouvement voir la figure 2.10[19].

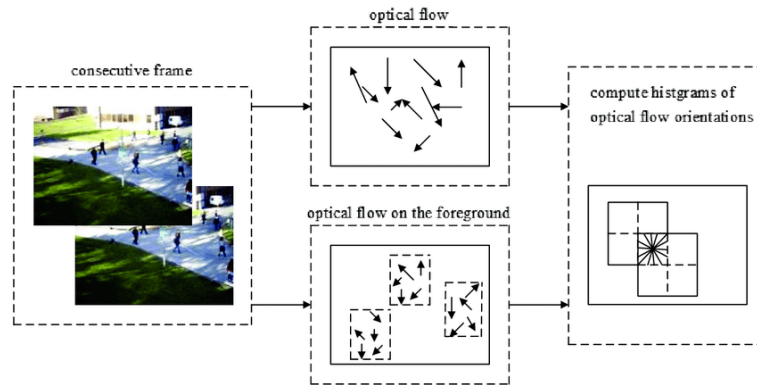


Fig. 2.10 : Histogram of Oriented Optical Flow (HOOF)

- **Flux optique** est un algorithme populaire pour la détection d'objets en mouvement dans les séquences vidéo en temps réel. Il évalue le mouvement entre deux images pour chaque pixel à différents intervalles de temps et produit des vecteurs optiques pour les objets en mouvement. Il est également utilisé pour le suivi d'objets en marquant l'objet d'intérêt dans la première image et en le suivant dans chaque image successive à l'aide du flux optique. Les objets en mouvement sont suivis par la technique d'estimation de vecteur de mouvement qui estime la position de l'objet dans les images consécutives. L'estimation de vecteur de mouvement produit un vecteur 2D en comparant les images consécutives. Cette technique peut éliminer les limitations rencontrées par le flux optique traditionnel. Il peut être utilisé dans de nombreux domaines différents tels que la reconstruction 3D, la reconnaissance d'activité, la surveillance de la circulation, l'imagerie médicale, l'analyse du mouvement humain, etc[45].

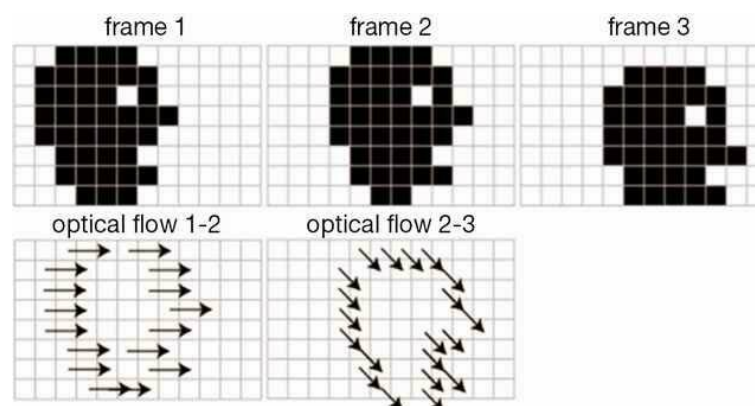


Fig. 2.11 : flux optique

- **Géométrie spatiale** Il existe deux façons de détecter le mouvement d'un corps humain dans une image. Le premier utilise la transformée Bandelet de première génération pour

détecter les objets en mouvement, mais il présente des inconvénients tels que des calculs complexes. La transformée Bandelet de deuxième génération résout ces problèmes en permettant la décomposition d'image basée sur le flux géométrique et l'analyse multi-échelle [2]. Une autre approche utilise la segmentation des régions et la classification de l'apprentissage automatique basée sur le flux optique et la transformée de Bandelet. Ces deux méthodes sont largement utilisées dans les domaines de la surveillance et de la reconnaissance des activités humaines[40].

2.2.3 Approche basée sur l'apparence corporelle(Body Appearance Based)

Les détecteurs de personnes qui se basent sur l'apparence du corps utilisent des caractéristiques telles que la silhouette, la forme du corps, les courbes et les descriptions des composants pour détecter la présence d'un être humain dans une image voir figure 2.18.

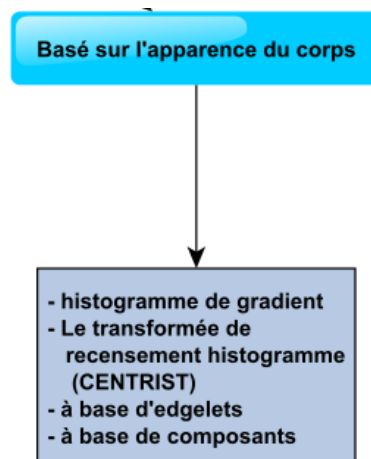


Fig. 2.12 : Approche basée sur l'apparence corporelle

- **HOG** (Histogramme du gradient orienté) est une méthode de représentation des caractéristiques d'une image qui est souvent utilisée pour la détection d'objets[50]. Il est basé sur la distribution des gradients orientés de l'image, qui est calculée en divisant l'image en petites cellules et en calculant les gradients dans chaque cellule[15]. Ces gradients sont ensuite normalisés et regroupés en histogrammes de direction. La méthode HOG a été largement utilisée pour la détection de piétons, la reconnaissance de gestes, la détection de visages et d'autres applications de vision par ordinateur [15][50]

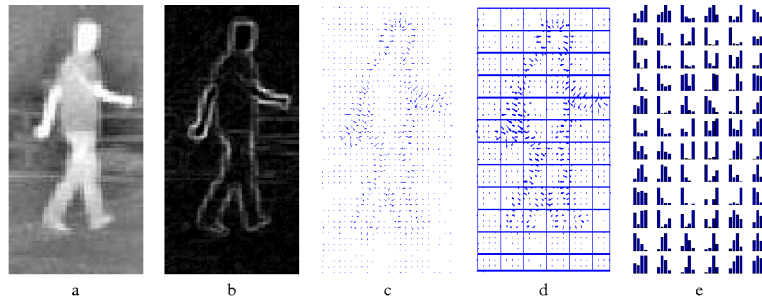


Fig. 2.13 : Histogramme du gradient orienté

- **CENTRIST** utilisée pour extraire des caractéristiques de texture à partir d'images. CENTRIST se base sur une transformation de recensement afin de capturer les propriétés locales de texture de l'image [2]. Par la suite, un histogramme est créé à partir de ces caractéristiques pour représenter l'image. ce dernier a été largement utilisé pour la reconnaissance d'objets, la classification d'images et la reconnaissance de scènes. Des études ont démontré que cette méthode est particulièrement efficace pour la reconnaissance d'objets dans des environnements complexes, tels que des scènes encombrées ou mal éclairées. [7][15][2]

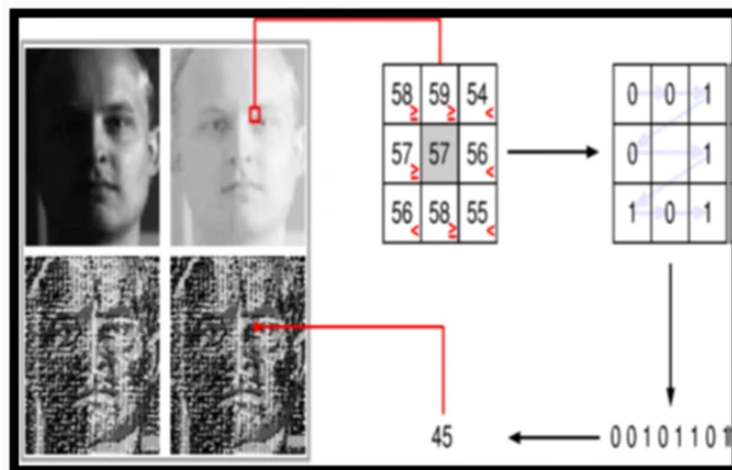


Fig. 2.14 : CENTRIST

- **Edgelets** est une méthode de traitement d'image qui utilise des informations de contours pour extraire des caractéristiques significatives d'une image [15]. Les edgelets sont des segments de contours qui sont détectés à partir de l'image. Ces segments sont ensuite utilisés pour construire un histogramme de distribution de l'orientation des contours [29][32]. Les caractéristiques obtenues peuvent être utilisées pour la détection d'objets, la reconnaissance faciale, la détection de personnes et d'autres tâches de vision par ordinateur. Cette méthode est souvent utilisée en combinaison avec d'autres techniques de traitement d'image pour améliorer la précision de la détection et la robustesse aux conditions environnementales variables [15] [2][32].

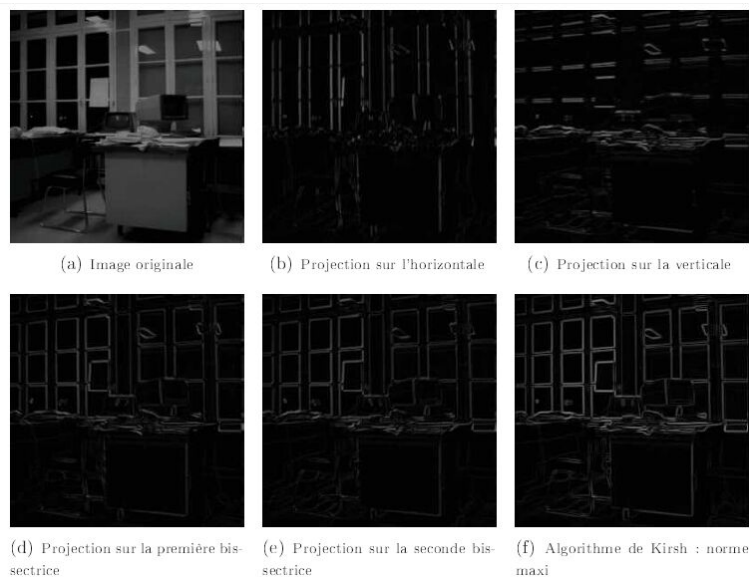


Fig. 2.15 : Edgelets

- **Basé sur les composants** est une méthode qui utilise la forme et la structure des objets pour les détecter dans une image [33] [29]. Cette méthode consiste à détecter des parties d'objet connues comme des composants et à les combiner pour détecter l'objet complet. Les composants sont des parties d'objet qui peuvent être des formes géométriques simples telles que des cercles, des triangles, des carrés, etc.[15][33], ou des parties d'objet plus complexes qui sont caractéristiques de cet objet particulier. Les techniques basées sur les composants ont été utilisées pour la détection d'objets dans des images en utilisant des modèles de forme pré-définis et des caractéristiques telles que la texture, la couleur et la forme pour détecter des objets dans des images.[33][29][15]

2.2.4 Approche Basée sur l'apprentissage en profondeur(Deep Learning Based)

Les détecteurs de personnes qui utilisent l'apprentissage profond se basent sur une architecture convolutive profonde, avec une ou plusieurs couches, pour localiser la présence d'un être humain dans une image avec précisions. Les approches modernes basées sur cette technique peuvent détecter avec précision un seul objet ou plusieurs objets dans une image voir figure 2.16, surpassant ainsi les méthodes traditionnelles de détection d'objets(voir figure 2.16).

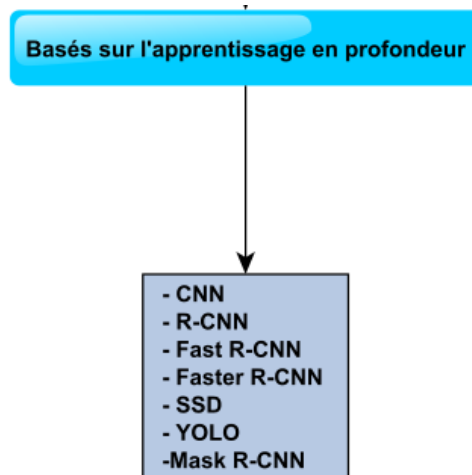


Fig. 2.16 : Approche Basés sur l'apprentissage en profondeur

- **CNN (Les réseaux de neurones convolutifs)** Les CNN sont des modèles de réseaux de neurones profonds utilisés pour l'analyse d'images. Leur structure est inspirée de l'organisation du cortex visuel. Ils sont capables d'extraire des caractéristiques importantes de l'image en utilisant des couches spécifiques telles que la convolution, le pooling, la normalisation et les couches entièrement connectées[41]. Ces détecteurs d'objets basés sur CNN sont divisés en deux catégories : ceux à une étape et ceux à deux étapes.
 - Les détecteurs à une étape utilisent des réseaux de neurones pour prédire directement les positions et les étiquettes de classe des objets dans l'image.
 - les détecteurs à deux étapes extraient d'abord des régions proposées avant d'appliquer un CNN pour détecter les objets

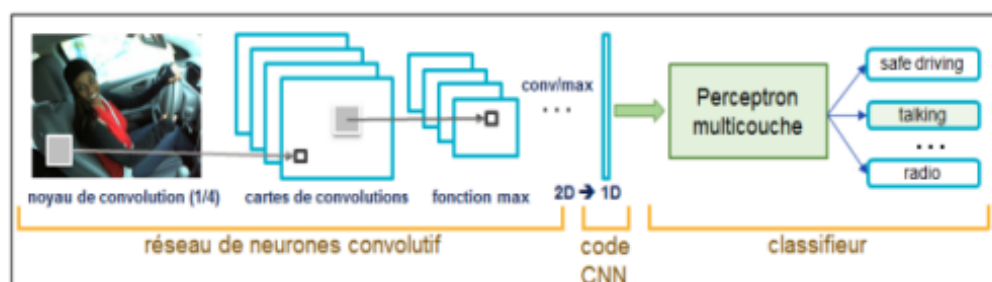


Fig. 2.17 : Architecture et composition d'un réseau des neurones convolutifs (CNN)

- **RNN** Le R-CNN (Regional Convolutional Neural Network) est un réseau de neurones convolutifs qui permet de détecter et de localiser correctement les objets dans une image. Cette méthode est plus efficace que les approches traditionnelles qui nécessitent un grand nombre de régions pour la classification des objets. Cette technique nécessite beaucoup de temps pour former le réseau et requiert également beaucoup d'espace disque pour stocker les cartes de caractéristiques, mais il permet d'obtenir de meilleurs résultats que les méthodes traditionnelles [11].

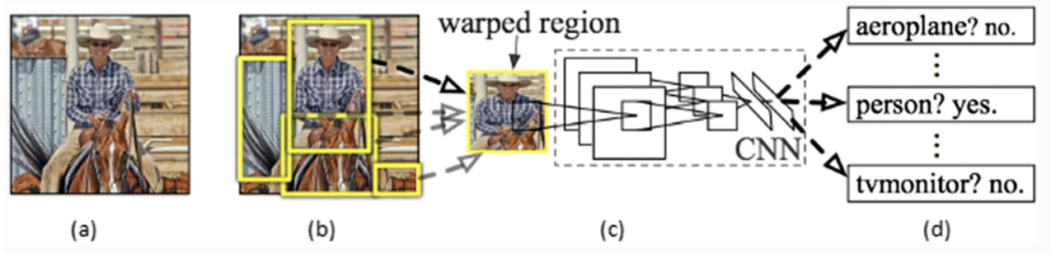


Fig. 2.18 : R_{CNN}

• **Fast R-CNN et Faster R-CNN**

Fast R-CNN (Regional Convolutional Neural Network) est une méthode de détection d'objets basée sur les réseaux de neurones convolutifs. Elle est une extension du modèle R-CNN proposé par Ross Girshick et al[10] qui vise à améliorer l'efficacité et la vitesse de traitement.

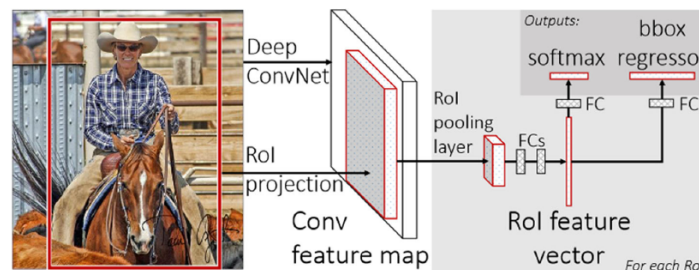


Fig. 2.19 : Fast R-CNN

Pour améliorer davantage la vitesse de traitement, l'équipe de Shaoqing Ren a développé Faster R-CNN [12]. Cette méthode utilise une technique appelée Region Proposal Network (RPN) pour générer les propositions de régions, ce qui la rend plus rapide que Fast R-CNN. Le processus de travail de Faster R-CNN implique la convolution de l'image d'entrée à travers un réseau de neurones, suivi de l'extraction des boîtes d'ancrage par le RPN. Les caractéristiques remodelisées sont ensuite transmises à une couche entièrement connectée qui prédit les étiquettes de classe et les valeurs de la boîte de délimitation. Faster R-CNN est environ 10 fois plus rapide que Fast R-CNN et 250 fois plus rapide que R-CNN, avec un temps de traitement d'environ 0,2 seconde par image [12] [38].

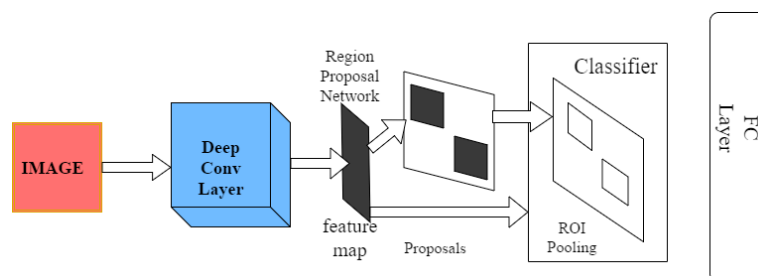


Fig. 2.20 : Architecteur du modèle Faster-R-CNN

- SSD(Single Shot MultiBox Detector)** Contrairement à Faster R-CNN, qui utilise un algorithme de proposition de régions, le SSD utilise des caractéristiques multi-échelles et des boîtes par défaut pour accélérer le processus de détection tout en maintenant une précision similaire[24]. Le SSD se compose d'un réseau de base, de couches de caractéristiques supplémentaires et de couches de prédiction qui prédisent les scores de classification et les coordonnées des boîtes englobantes. Il offre de meilleures performances que le Faster R-CNN[44]. Le modèle MobileNet est une architecture de réseau de neurones conçue pour des tâches de vision par ordinateur sur des appareils mobiles et embarqués. Il vise à réduire la quantité de calcul nécessaire tout en maintenant de bonnes performances de détection. Le modèle MobileNet utilise des opérations de convolution spécifiques, appelées convolutions profondes séparables, pour réduire le nombre de paramètres et d'opérations nécessaires par rapport à d'autres architectures plus lourdes. La méthode SSD présente une approche spécifique pour utiliser le modèle MobileNet-SSD dans un scénario de détection d'objets en temps réel. Ils ont décrit la configuration du modèle, les techniques d'optimisation utilisées pour accélérer les prédictions en temps réel, et une évaluation des performances du modèle sur des ensembles de données de référence, en comparant les résultats avec d'autres approches existantes.[47]

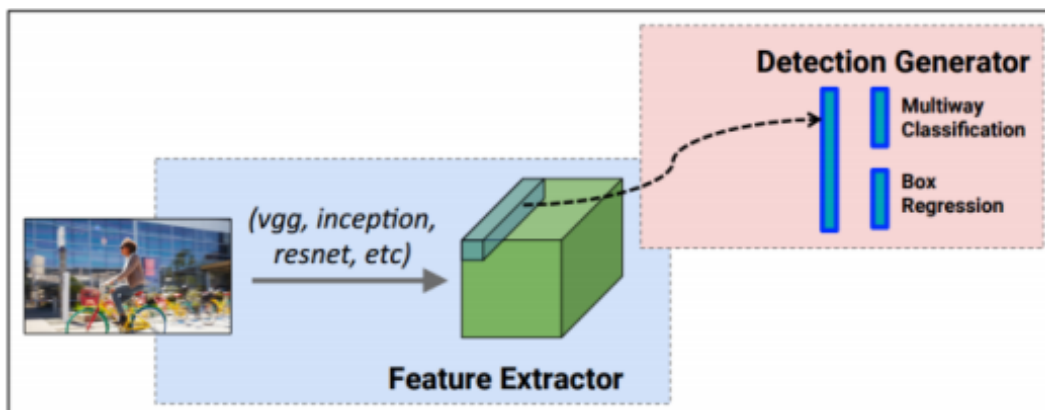


Fig. 2.21 : Architecture du SSD

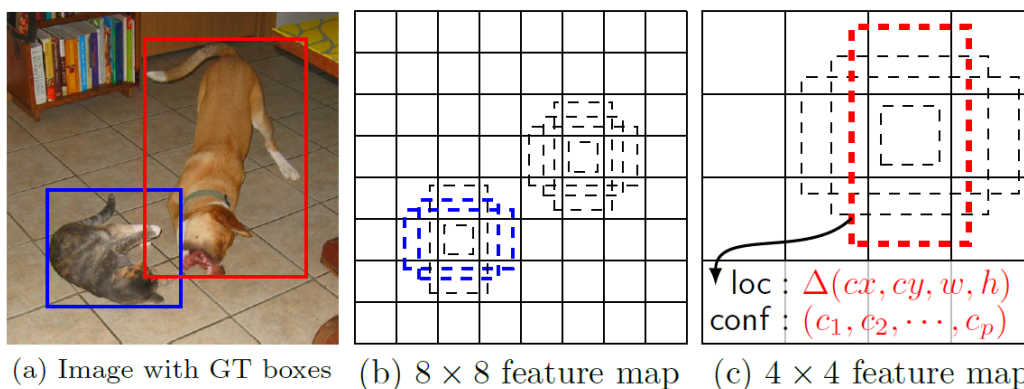


Fig. 2.22 : single shot detector (SSD)

- YOLO(You Only Look Once) :** Le détecteur d'objets YOLO utilise un réseau convolutionnel pour fournir des boîtes de délimitation et des probabilités de classe pour ces boîtes.[8] YOLOv2 est une version améliorée de YOLOv1, utilisant le concept de boîtes d'ancrage et atteignant une précision de 91,2% avec une vitesse de traitement de 67 FPS. YOLOv3 est une version incrémentielle de YOLOv2, utilisant Darknet-53 pour l'extraction des caractéristiques et offrant une détection plus précise des petits objets. YOLO est plus rapide que les autres détecteurs d'objets tels que R-CNN, Fast R-CNN et Faster R-CNN en raison de ses ordres de grandeur plus rapides (environ 45 FPS) [9].

La détection d'objets en temps réel est une tâche clé dans le domaine de la vision par ordinateur et a de nombreuses applications, notamment dans la surveillance vidéo, la conduite autonome et la réalité augmentée. Le modèle YOLO est une approche couramment utilisée pour résoudre ce problème [8].

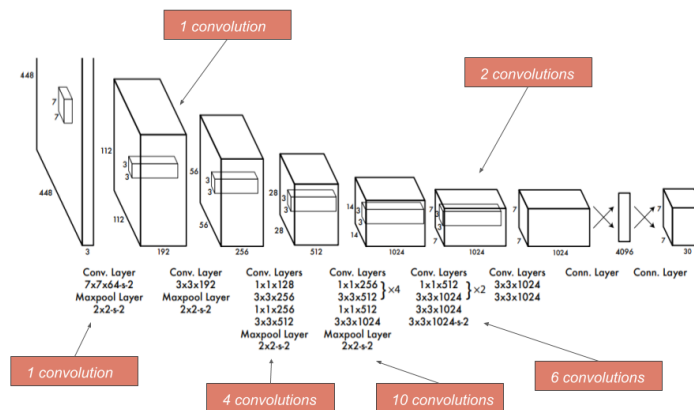


Fig. 2.23 : Architecture de la méthode Yolo

YOLO est une famille de modèles de détection d'objets en temps réel. YOLOv1 a été la première version et a introduit la prédiction d'objets en une seule passe. YOLOv2 a ajouté des améliorations telles que les couches résiduelles et les "anchor boxes". YOLOv3 a utilisé plusieurs échelles de détection pour détecter des objets de différentes tailles. YOLOv4 a intégré des techniques avancées telles que CSPDarknet53 et des blocs PANet. Il existe également d'autres variantes comme YOLOv5, YOLO Nano et YOLOX, chacune apportant des ajustements spécifiques pour améliorer la détection d'objets en temps réel[43].

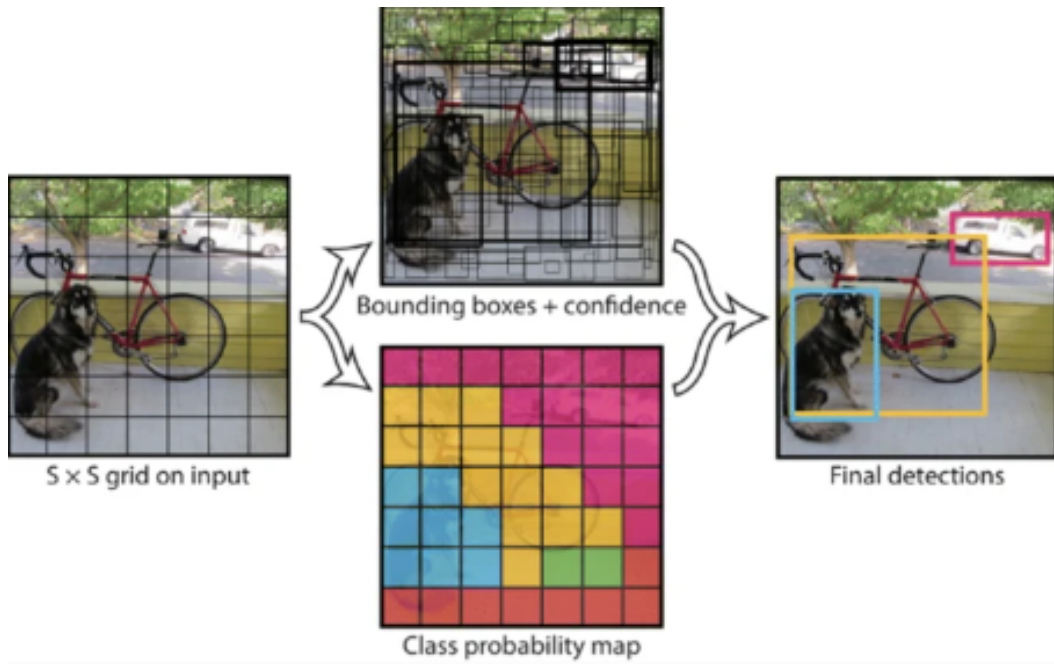
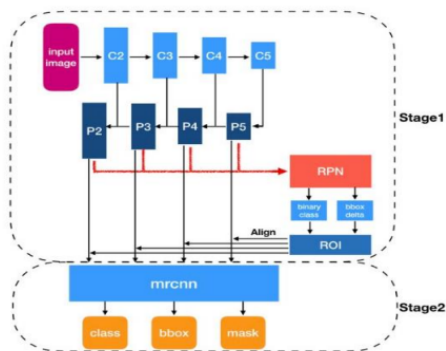


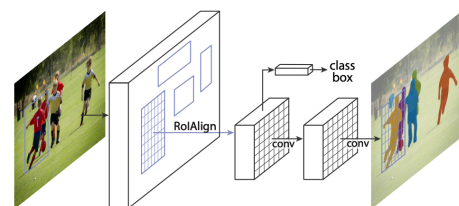
Fig. 2.24 : YOLO

- **Mask R-CNN** D'après les travaux de Kaiming He et al, le masque R-CNN (en anglais «Mask Region Neural Network»)[13] est un détecteur à deux étapes voir figure 2.25a :
 - la première étape scanne l'image et génère des propositions.
 - La deuxième étape classe les propositions et génère des boîtes et des masques de délimitation.

Les deux étapes sont reliées à la structure de base. La structure de base est un réseau neuronal convolutif standard (généralement, ResNet50 ou ResNet101) qui sert d'extracteur de caractéristiques. Les premiers couches détectent les éléments de bas niveau (bords et coins), et les couches suivantes détectent successivement les éléments de plus haut niveau (voiture, personne, ciel...) voir figre 2.25b[13].



(a) Architecture de la méthode Mask R_CNN



(b) Mask R_CNN

2.2.5 Tableau comparatif des solutions lue

Dans le tableau suivant nous récapitulons les différents travaux de détection et de reconnaissance facials, mettant en évidence leurs avantages et inconvénients.

Modalités	Travaux	Méthodes	Avantages	Inconvénients
Basé sur les caractéristiques du visage	MA Ansari et al[2]	Modélisation de la couleur de la peau	Précision raisonnable, résultats satisfaisants	Peut être sensible aux variations de l'éclairage
	Z Arya et al[3]	LBPH et Viola-Jones	Haute précision, apprentissage en profondeur	Requiert des ressources informatiques importantes
	Zankruti Arya Et Vibha Tiwari [25]	Géométrie du visage	Méthodes éprouvées, performances élevées dans certaines situations	Peut être limité à des caractéristiques spécifiques
Basé sur les caractéristiques du mouvement	Khalifa AF et al [15]	Soustraction de l'arrière-plan, Différenciation des trames	Bonne séparation de l'arrière-plan et des objets	Peut être sensible aux variations de l'éclairage
	A Kushwaha et al [19]	HOOF	Bonne détection des mouvements fluides	Peut être sensible au bruit et aux mouvements rapides
	Hengsen Wang et al [45]	Flux optique	Bonne détection des mouvements fluides	Peut être sensible au bruit et aux mouvements rapides
	MA Ansari et al [2]	Géométrie spatiale	Utilisation efficace des ressources matérielles	Peut être sensible aux changements d'arrière-plan
Basé sur l'apparence corporelle	W Zhou et al [50]	HOG	Utilisation répandue, Bonne précision pour la détection, Calculs rapides	Sensible aux variations d'éclairage, Difficulté à détecter des objets petits, Faible robustesse aux occlusions

	M Braik et al [7]	CENTRIST	Efficace dans des environnements complexes, Bonne reconnaissance d'objets	Sensible aux variations d'échelle, Temps de calcul plus long
	HH Nguyen et al [32]	Edgelets	Bonne détection des contours, Utilisation en combinaison avec d'autres méthodes, Bonne précision pour la détection	Sensible aux changements de perspective, Sensible aux occlusions
	C Ning et al [33]	Basé sur les composants	Utilisation de la forme et de la structure des objets, Bonne précision pour la détection	Dépendance de modèles de formes prédéfinis, Limité à des objets avec des composants spécif
Basés sur l'apprentissage en profondeur	G Sreen et al [41]	CNN	- Bonne précision pour la détection - Utilisation de caractéristiques profondes - Réseau de neurones bien établi	- Besoin de beaucoup de données d'entraînement - Temps de calcul plus long
	R Girshick et al [11][10]	R-CNN, Fast R-CNN, Faster R-CNN	- Haute précision de détection - Bonne localisation des objets - Utilisation de régions proposées	- Temps de calcul plus long - Nécessite un grand espace de stockage - Nécessite un grand nombre de régions proposées
	Ayesha Younis et al [47][24]	SSD	- Haute vitesse de traitement - Utilisation de caractéristiques multi-échelles - Moins de régions proposées à traiter - précision de plus de 97%	- Précision légèrement inférieure à Faster R-CNN - Moins précis pour les petits objets

	RK Chandana et al [8]	YOLO	- Très haute vitesse de traitement	- Moins précis que Faster R-CNN et SSD
	Kaiming He et al [13]	Mask R _C NN	- Détection précise des objets	- Temps d'inférence plus long - Sensible à la qualité des données d'entrée

2.2.6 Synthèse de la comparaison

Le tableau 2.1, représente une comparaison entre avantages et inconvénients d'un ensemble de travaux et articles étudiés. Comme nous voyons que tous les travaux ont presque des avantages similaires c'est que tous les systèmes peuvent maîtriser les dangers et sauver des vies humaines et comme inconvénients on a l'insécurité des données partagées et le manque de la fiabilité et ce qui coïncide exactement et confirme notre problématique.

2.3 Conclusion

Nous avons résumé dans ce chapitre les différentes techniques de détection et reconnaissance des êtres humains. D'après le tableau récapitulatif Tab. 2.1 nous optons pour la méthode SSD pour sa vitesse et sa précision. D'où notre choix de cette technique pour la réalisation d'un système de détection et d'identification des intrus dans un système de surveillance embarqué.

Chapitre 3

**Notre proposition : SSD pour la détection
d'intrusion**

3.1 Introduction

L'approche SSD est une méthode de détection d'objet très puissante, notamment pour la détection des êtres humains dans les images. Il s'agit d'un algorithme basé sur un réseau de neurones convolutionnels qui permet de détecter et de localiser les humains dans une image en une seule passe. La variation de choix de méthodes de détection humaine et ça reconnaissance facial rend le domaine de la surveillance très vaste. Plusieurs propositions ont vu le jour, selon les objectifs d'études. Dans ce chapitre, nous mettrons en avant notre proposition de système de surveillance. nous aborderons tout d'abord la problématique étudiée, puis détaillerons les différentes phases de notre méthode de détection d'intrusion.

3.2 Choix de la méthode de détection d'intrusion

La méthode SSD est un choix approprié pour la détection et la reconnaissance de personnes ainsi que pour la détection d'intrusion, en raison de plusieurs facteurs, ainsi que sa vitesse en temps réel, sa précision raisonnable en matière de détection d'objets, sa facilité d'implémentation et sa qualité de modèle pré-entraîné.

3.3 Description de la solution proposée

Notre proposition, dans le cadre de ce projet, consiste à mettre en place un système de détection d'intrusion en utilisant dans différents endroits des caméras de surveillances préinstallées. Ces dernières peuvent être reliées à un ordinateur connecté ou système embarqué sur des systèmes mobiles (voitures, drones, etc).

ce faire, nous nous sommes basées sur la méthode SSD qui offre plusieurs avantages, comme montré dans le chapitre état de l'art, pour servir une base de détection d'intrusion par rapport aux images des personnes autorisées d'entrer dans une zone surveillée. D'autre part, nous avons proposé d'autres fonctionnalités et améliorations à la méthode SSD :

- Fonctionnalité d'alerte et de notification du propriétaire en cas de détection d'intrusion ;
- Ajoute des personnes détectés comme intrus, alors qu'ils n'existent pas dans la base de données de ceux qui sont autorisés d'accéder à la zone surveillées ;
- Mettre dans une liste noire les intrus.

Notre méthode proposée basée sur l'algorithme SSD s'appelle SSD-FID (SSD For Intrusion Detection)

Dans ce travail, nous proposons de développer un système de détection humaine dans un ordinateur PC puis la Raspberry Pi4, qui est un système rapide et précis Avec une architecture puissante en utilisons la méthode SSD. Cependant, Cette dernière soulève certaines questions essentielles quant aux méthodes à combiner, aux traitements à appliquer, ainsi qu'au système embarqué à adopter afin d'améliorer les performances du système. Dans cette proposition, nous

avons modifié la méthode SSD pour l'adopter à la détection et la reconnaissance des personnes afin de détecter l'éventuels. Cette méthode nous l'avons appelées SSD-FID

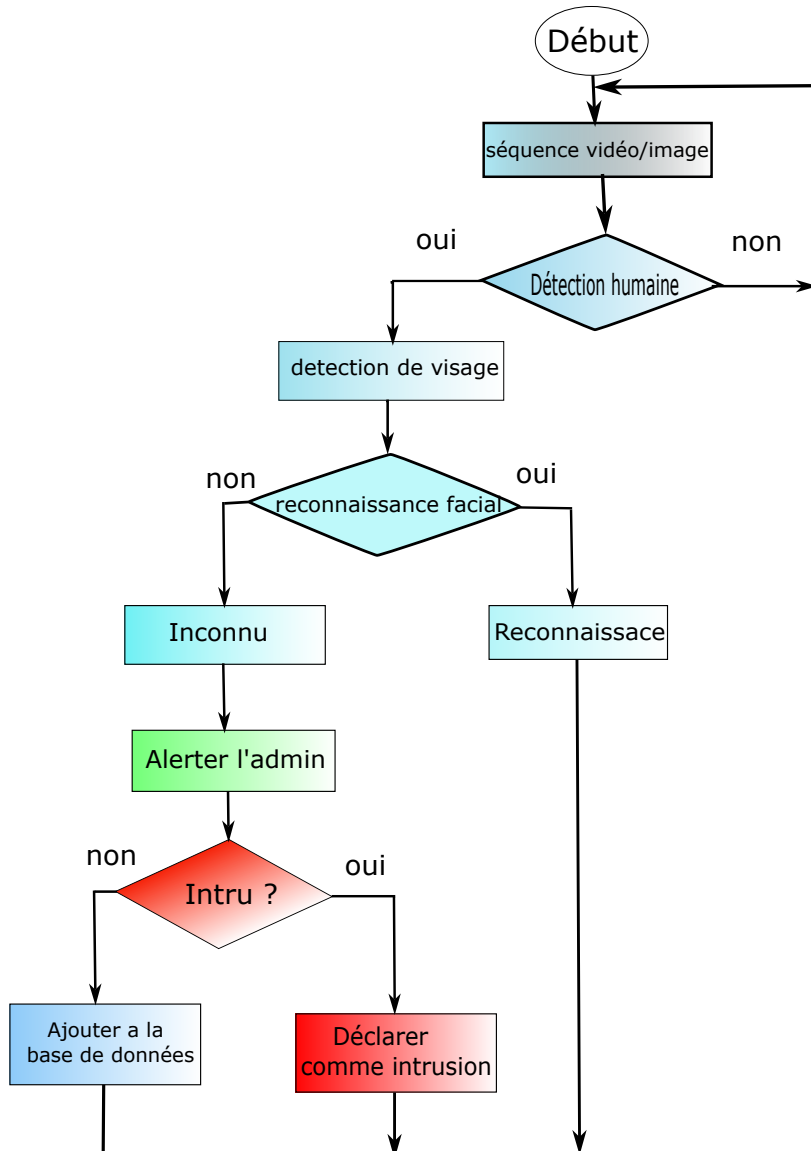


Fig. 3.1 : diagramme de la solution

3.3.1 Fonctionnement du modèle SSD-FID

Dans ce qui suit, nous allons décrire chacune des étapes de fonctionnement de notre modèle SSD_FID :

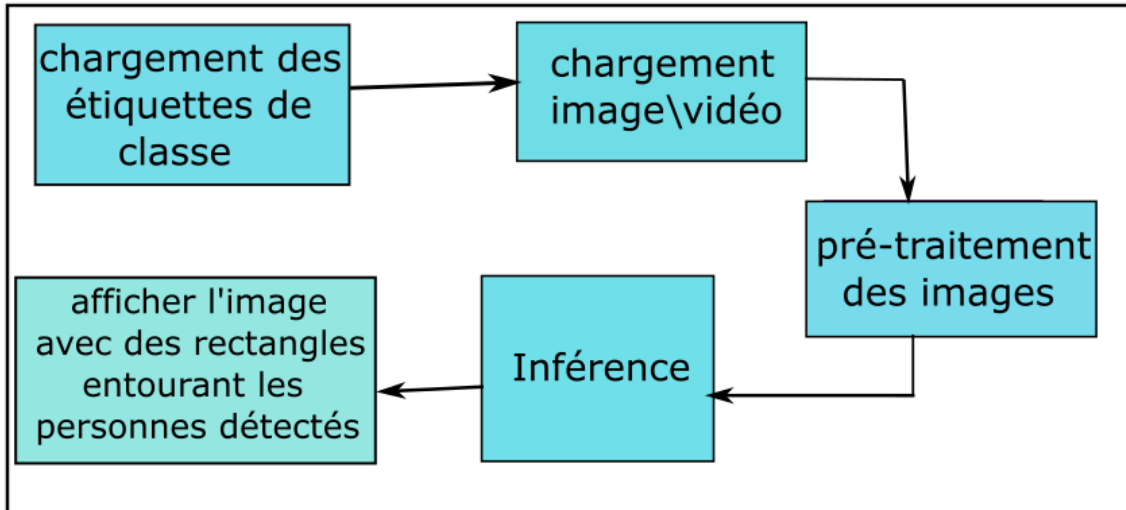


Fig. 3.2 : Diagramme de la méthode SSD_FID.

A/. chargement des étiquettes de classe Consiste à importer des données étiquetées de personnes dans notre programme en utilisant le dataset MSCOCO. Cela permet au modèle de savoir où se trouvent les personnes dans l'image ou la vidéo que nous souhaitons analyser.

B/. chargement de l'image/ vidéo charger et lire une image ou une vidéo à partir de notre dataset(figure 3.3).



Fig. 3.3 : Exemples d'images humain avant le prétraitées.

C. Prétraitement et affichage des images Nous traitons les images d'entrée en l'adaptant aux exigences du modèle, par le redimensionnement de l'image à une taille spécifique, ainsi que la normalisation des valeurs des pixels pour les mettre à l'échelle appropriée, la figure (3.4) illustre les modifications apportées à l'image après le prétraitement.



Fig. 3.4 : Exemple d'image après le prétraitement

D. Inférence Effectuons une inférence sur l'image ou la vidéo donnée en utilisant notre modèle chargé puis renvoyer les résultats des prédictions de détection de personne. Ensuite dessiner des rectangles autour des régions où se trouvent les personnes

E. Afficher l'image avec les rectangles entourant les humains détectés afficher l'image prétraitée avec les rectangles de détection et les étiquettes ajoutés (figure 3.5).



Fig. 3.5 : Afficher l'image avec les rectangles entourant les humains détectés

Algorithme de détection humaine avec la méthode SSD

Nous trouvons ci-dessus l'algorithme des étapes de la méthode SSD dans des sections de vidéo(frames) :

Algorithm 1 : Algorithme SSD pour la détection des humains

```
Entrée : Image d'entrée
Sortie : Régions de détection et catégories

// 1ère étape : chargement des données étiquetées
1 Importer les données étiquetées à partir du jeu de données MSCOCO;

// 2ème étape : chargement de l'image/vidéo
2 Importer une image ou un flux vidéo;

// 3ème étape : prétraitement et affichage de l'image
3 Pré-traiter l'image (réduire la taille, normaliser, etc.); Afficher l'image prétraitée;

// 4ème étape : inférence
4 Effectuer une inférence sur l'image en utilisant le modèle SSD entraîné; Récupérer les
résultats des prédictions pour les régions contenant des humains;

// Sortie
5 Afficher l'image d'origine avec des rectangles dessinés autour des régions où des
humains ont été détectés;
```

3.3.2 Détection et reconnaissance en utilisant une caméra

L'approche SSD de la reconnaissance et de la détection humaines sur un ordinateur implique l'utilisation de modèles spécialisés pour trouver et localiser des personnes dans des images ou des vidéos. Le modèle SSD est formé sur des données indiquant où se trouvent les personnes. Ensuite, lorsque le modèle est utilisé, il analyse l'image ou la vidéo et détecte les personnes en leur attribuant des cadres de délimitation et des scores de confiance. Les cadres de délimitation indiquent où se trouvent les personnes et les scores de confiance indiquent la probabilité que ces détections soient correctes. Les détections valides peuvent être présentées à l'utilisateur, par exemple en encadrant la personne détectée dans l'image. Cela permet une reconnaissance et une détection automatique des personnes, ce qui est utile dans de nombreux domaines, comme la vidéosurveillance ou la recherche d'objets perdus.

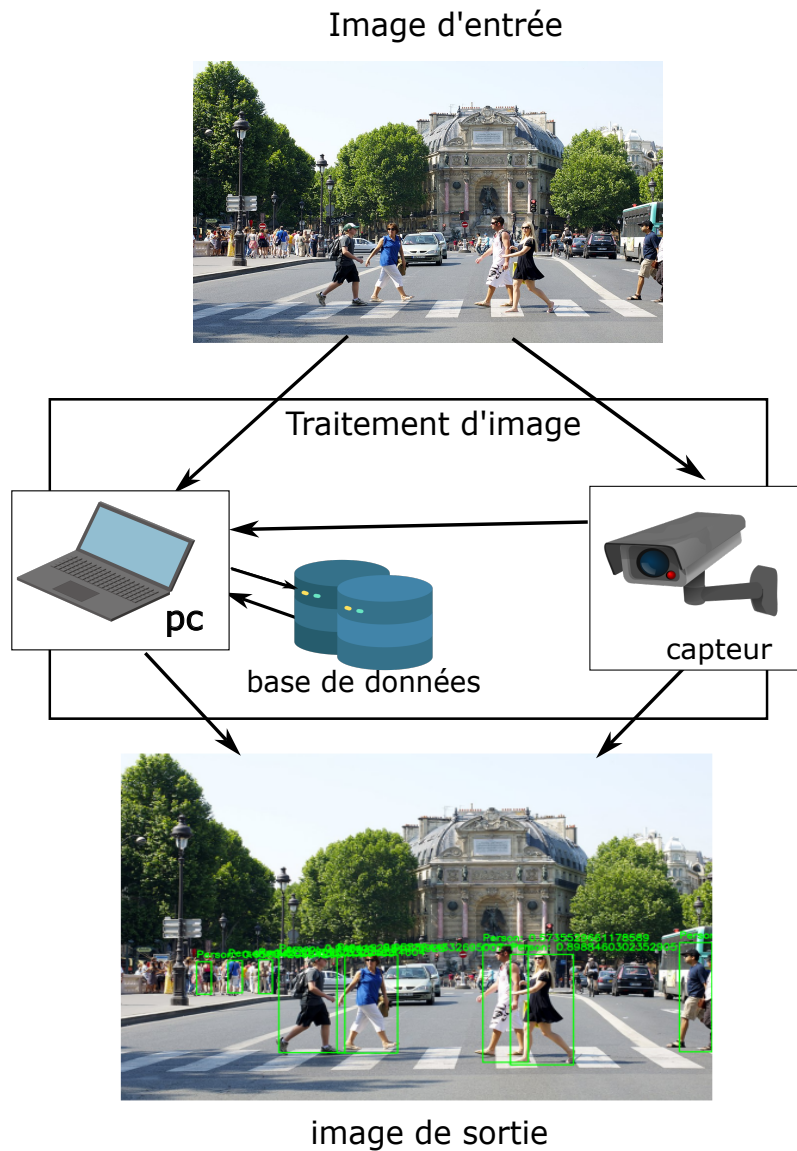


Fig. 3.6 : Schéma de la solution proposée avec un ordinateur.

3.3.3 Détection et reconnaissance en utilisant un système embarqué (Raspberry Pi4)

La reconnaissance faciale et la détection humaine sur un système embarqué tel que le Raspberry Pi 4 en utilisant la méthode SSD-FID sont des techniques permettant d'analyser les images en temps réel. En utilisant la caméra intégrée au Raspberry Pi 4, le modèle SSD est appliqué pour détecter les personnes présentes dans les images en dessinant des rectangles autour d'elles (voir figure 3.7). De plus, un modèle de reconnaissance faciale peut être entraîné pour identifier spécifiquement les individus en comparant leurs caractéristiques faciales avec celles apprises lors de l'entraînement. Ces approches permettent de réaliser des applications embarquées efficaces de reconnaissance et de détection humaine. Nous avons opté pour l'utilisation d'un Raspberry Pi 4 en tant que système embarqué pour plusieurs raisons. Parmi ces raisons nous citons :

1. C'est un appareil compact et facile à transporter, ce qui nous permet de le déployer dans différents endroits selon nos besoins. Il offre une grande flexibilité et une portabilité pratique pour des applications de détection et de reconnaissance humaine sur le terrain.
2. Il dispose de ressources de calcul suffisantes pour effectuer ces tâches de manière autonome. Il est équipé d'un processeur puissant et d'une mémoire suffisante pour traiter les données en temps réel, sans nécessiter une connexion constante à des serveurs externes. Cela signifie que nous pouvons effectuer les analyses et les prédictions directement sur le Raspberry Pi 4, ce qui nous offre une réactivité rapide et une indépendance vis-à-vis des infrastructures externes.
3. la confidentialité des données. nous pouvons conserver les données localement et éviter de les transférer à des serveurs distants. Cela renforce la sécurité et protège la vie privée des individus. Les informations sensibles sont traitées localement, ce qui réduit les risques de piratage ou de divulgation indésirable.
4. C'est système embarqué polyvalent et abordable. Il peut être facilement personnalisé et étendu grâce à sa large gamme d'interfaces et de connecteurs. Nous pouvons intégrer des capteurs, des caméras et d'autres dispositifs pour améliorer les capacités de détection et de reconnaissance humaine du système.

En somme, nous avons choisi le Raspberry Pi 4 comme système embarqué pour la détection et la reconnaissance humaine en raison de sa portabilité, de sa puissance de calcul, de sa confidentialité des données et de sa flexibilité. Cela nous permet d'effectuer des tâches de manière autonome, de protéger la confidentialité des données et de personnaliser le système en fonction de nos besoins spécifiques.

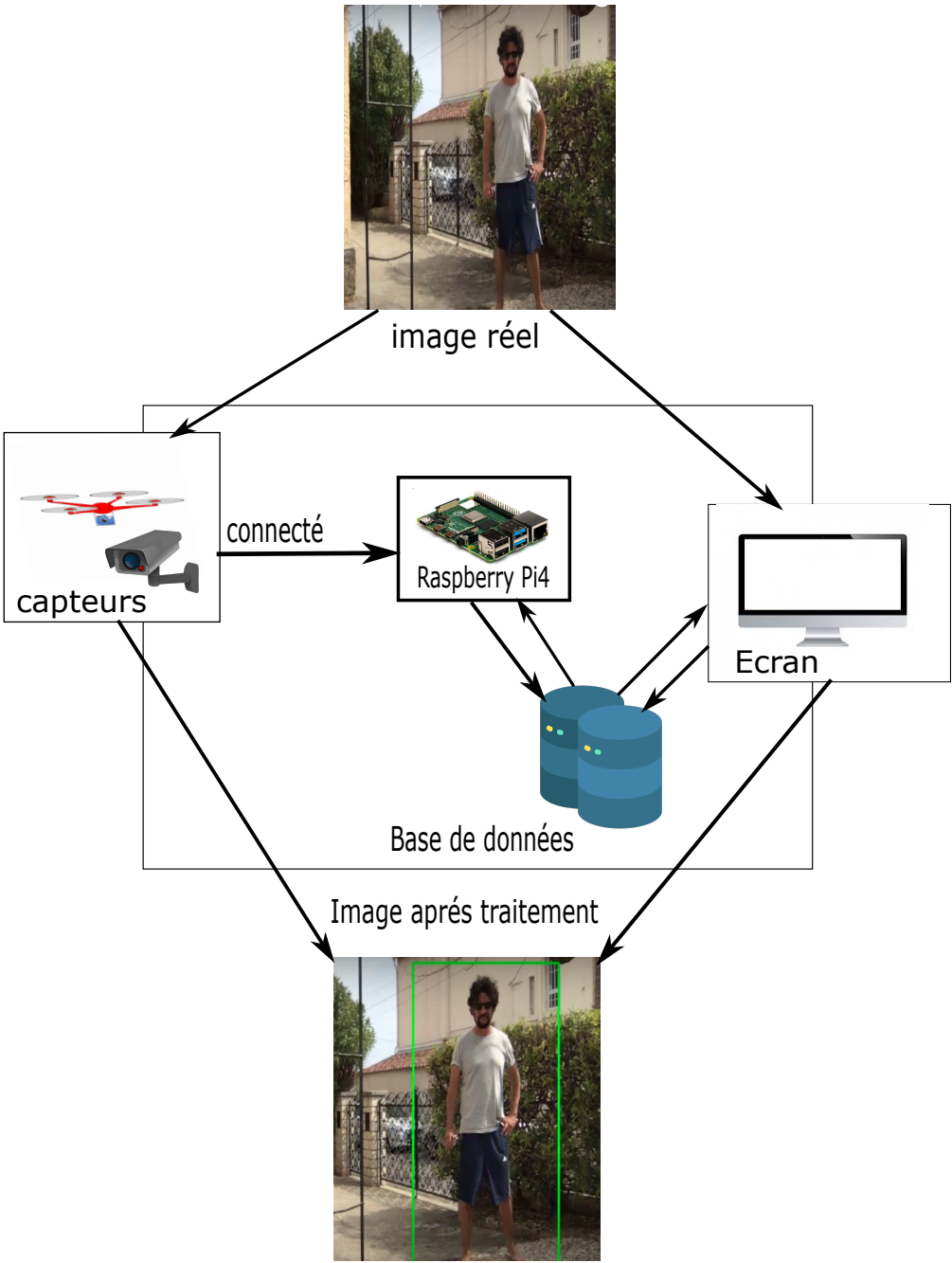


Fig. 3.7 : Schéma de traitement avec la Raspberry Pi 4

3.3.4 Algorithme de détection et reconnaissance choisi dans notre solution

Nous avons introduit dans cet algorithme (2) toutes les étapes et les conditions discutées pour la détection et la reconnaissance proposée dans notre solution.

Algorithm 2 : Algorithme de détection d'humains et reconnaissance faciale

Entrée : Images/vidéo capturées par une caméra connectée à une Raspberry Pi

Sortie : Régions de détection, catégories et reconnaissance faciale

```
// 1ère étape : Capture d'image
1 Capturer l'image à l'aide de la caméra connectée à la Raspberry Pi

// Pour chaque image capturée
2 foreach image capturée do
    // 2ème étape : Détection d'humains
    3 Appliquer la détection d'humains à l'image
    4 if aucun humain détecté then
    5     | Passer à l'image suivante
    6 else
        // 3ème étape : Détection de visages
        7 Appliquer la détection de visages à l'image
        // 4ème étape : Reconnaissance faciale
        8 foreach visage détecté do
            // Effectuer la reconnaissance faciale
            9 if visage reconnu then
            10     | Afficher le nom de la personne
            11 else
                12 if Visage inconnu détecté then
                13     | Afficher("Visage inconnu détecté. Voulez-vous l'ajouter à la base
                | de données? (O/N)")
                14     | Lire la réponse de l'utilisateur dans reponse
                15     | if reponse est "O" then
                16     |     | Demander le nom de l'utilisateur dans nom
                17     |     | AjouterVisage(img, nom)
                18     |     | result ← "Visage ajouté à la base de données"
                19     | else
                20     |     | SauvegarderDansDossierIntrus(img)
                21     |     | result ← "Visage intrus"
                22     |     | result
            | result
        | result
    | result
// Passer à l'image suivante
```

Déscription des étapes de l'algorithme

Nous décrivons dans cette section les étapes de notre algorithme de détection et de reconnaissance :

- image ou vidéo(entrée) : nous capturons l'image à l'aide de la caméra connectée à la Raspberry Pi.
- Détection d'humains : Nous appliquons la détection d'humains à l'image pour détecter la présence de personnes. Si aucun humain n'est détecté, nous passons à l'image suivante.
- Détection de visages : Si des humains sont détectés, nous appliquons la détection de visages pour localiser les visages présents dans l'image.
- Reconnaissance faciale : Pour chaque visage détecté, nous effectuons la reconnaissance faciale en comparant les caractéristiques faciales avec celles apprises lors de l'entraînement.
 - Si un visage est reconnu, nous affichons le nom de la personne correspondante.
 - Sinon, si un visage inconnu est détecté, nous demandons à l'utilisateur s'il souhaite l'ajouter à la base de données.
 - Si oui, nous ajoutons le visage avec le nom fourni par l'utilisateur.
 - Sinon, nous considérons le visage comme intrus et le sauvegardons dans un dossier dédié.
- passer à l'image suivante.

Sauvgarde de l'image dans la BDD de la Raspberry

Pour garantir un traitement rapide et efficace des images ou des vidéos dans notre solution, nous avons identifié plusieurs étapes clés pour enregistrer les images dans une base de données montée sur une carte Raspberry Pi. Voici les étapes à suivre dans notre solution :

- installée une BDD sur notre Raspberry Pi. puis stocker les informations relatives aux visages, telles que le nom de la personne correspondante et l'image elle-même.
- détecté un visage , extraire l'image correspondante
- vérifier si le visage est déjà présent dans la base de données.
 - Si le visage correspondant est trouvé dans la base de données, nous pouvons récupérer le nom associé à ce visage et l'afficher à l'utilisateur.
 - sinon, si le visage détecté est inconnu, nous pouvons demander à l'utilisateur s'il souhaite l'ajouter à la base de données. Si oui, demander à l'utilisateur de fournir un nom associé au visage et procéder à la sauvgarde de l'image et du nom dans la base de données.
 - si non, considérer-le comme un intrus. sauvegarder l'image dans un dossier dédié sur la Raspberry Pi, destiné à stocker les images d'intrus.

3.4 Environnement de développement

Dans le but de programmer notre système, nous utilisons une variété d'outils de mise en œuvre, y compris le langage Python sur la plateforme Google Colab. Cette section traite de ses détails matériels, ainsi que des définitions liées au langage Python.

Langage Python

Python est un langage de programmation interprété, interactif et orienté objet créé par Guido Van Rossum. Il comporte plusieurs modules et permet une manipulation dynamique des types de données. En plus d'être orienté objet, ce langage de programmation permet l'exécution de code procédural ou fonctionnel. Python est portable sur plusieurs systèmes d'exploitation, y compris Linux, Windows et MacOS. Voici les bibliothèques les plus cruciales auxquels nous faisons référence :



Fig. 3.8 : logo du python.

- **OpenCV(Cv2)** : est une bibliothèque open source très populaire pour la vision par ordinateur. Elle fournit des outils et des fonctions puissantes pour le traitement d'images et de vidéos, y compris la détection d'objets, la reconnaissance faciale, le suivi d'objets, etc [39].
- **Tensorflow** : est une bibliothèque d'apprentissage automatique (machine learning) développée par Google. Elle est utilisée pour créer et entraîner des modèles d'apprentissage automatique, y compris des réseaux de neurones, et permet l'inférence et l'application de ces modèles sur de nouvelles données.
- **Os** : fournit des fonctions pour interagir avec le système d'exploitation sous-jacent. Il permet de manipuler des fichiers, d'accéder aux répertoires, de créer des processus, de gérer les variables d'environnement, etc.
- **Face_Recognition** : est une bibliothèque Python qui utilise la vision par ordinateur et l'apprentissage automatique pour la reconnaissance faciale. Elle fournit des fonctionnalités pour détecter et reconnaître les visages dans les images ou les flux vidéo, ainsi que pour extraire et comparer les caractéristiques des visages.

- NumPy : est une bibliothèque Python utilisée pour travailler avec des tableaux. Elle dispose également de fonctions permettant de travailler dans le domaine de l'algèbre linéaire, de la transformée de Fourier et des matrices
- Dlib : est une bibliothèque logicielle polyvalente écrite dans le langage de programmation C++. Sa conception est fortement influencée par les idées de la conception par contrat et de l'ingénierie logicielle basée sur les composants.

Google Colab

GoogleColab est une plateforme de création d'environnements interactifs appelés Colab Notebooks, qui sont des notebooks Jupiter hébergés par Colab. Il permet d'écrire et d'exécuter du code en langage Python, ainsi que de collaborer avec plusieurs personnes dans le même environnement. Il existe également d'autres options utiles, telles que l'importation de données pour l'entraînement de modèles IA, l'accès gratuits au GPU, ou encore l'organisation du notebook en plusieurs sections de codes exécutables,



Fig. 3.9 : logo du goole colab.

Spyder

est un environnement de développement intégré (IDE) open source basé sur Python. Il est inclus dans la distribution Anaconda, une plate-forme populaire de science des données et de développement Python. Spider fournit une interface conviviale pour écrire, tester et déboguer du code Python et exécuter des notebooks Jupyter.



Fig. 3.10 : logo du spyder.

Inkscape

Inkscape est un logiciel de dessin vectoriel gratuit et open source. Il fournit une large gamme d'outils et de fonctions pour créer et éditer des images vectorielles. Contrairement aux images raster basées sur des pixels, les images vectorielles sont composées de formes géométriques qui permettent de les redimensionner sans perte de qualité. Inkscape prend en charge des fonctionnalités telles que le dessin de courbes, la manipulation de chemins, la gestion des calques, les filtres, les effets de texte et la possibilité d'importer et d'exporter des formats de fichiers courants tels que SVG (Scalable Vector Graphics). C'est un outil populaire utilisé par les artistes, les graphistes, les concepteurs de sites Web et d'autres professionnels de la création pour créer des graphiques, des icônes, des logos et des illustrations.



Fig. 3.11 : logo du Inkscape.

3.5 Experimentation et discussion des résultats

Notre approche SSD-FID implique de soumettre chaque étape de notre algorithme de détection et de reconnaissance à des tests rigoureux en utilisant notre méthode. Ainsi, nous vérifions la performance de chaque composant de l'algorithme afin de garantir des résultats fiables et précis. Cette approche nous permet d'optimiser et de valider chaque étape de manière systématique, assurant ainsi la qualité et l'efficacité de notre algorithme dans son ensemble

3.5.1 Détection Humain

Nous utilisons un ensemble d'images pour tester notre modèle afin de voir s'il peut détecter les humains, même dans des situations difficiles. L'algorithme de détection humaine est testé sur différentes images pour évaluer ses performances. Les résultats obtenus sont les suivants 3.12 :

Chapitre 3. Notre proposition : SSD pour la détection d'intrusion



Fig. 3.12 : Exemple Detection humain

L'image suivante 3.13 présente les personnes détectées dans des photos prises la nuit.



Fig. 3.13 : Exemple de détection humaine pendant la nuit

Une fois toutes les personnes détectées dans le cadre, chacune d'entre elles est enregistrée

individuellement dans un dossier appelé "personnes_detectées".

la figure ci dessus 3.14 explique les résultats sur une images après l'exécution

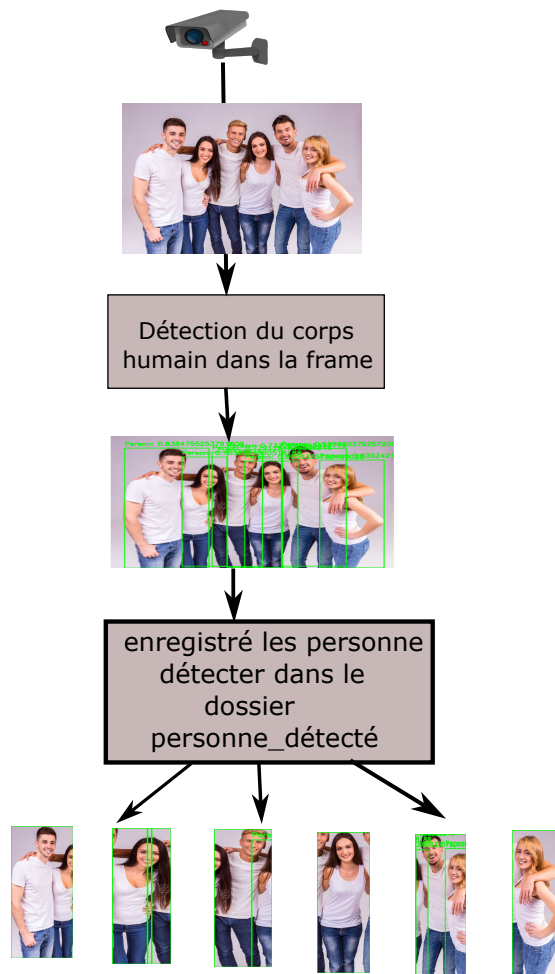


Fig. 3.14 : Personnes détectées et enregistrées individuellement

Code sur la détection humain

Dans cette section nous mettons notre code de détection de personnes :

```
# Importer les bibliothèques nécessaires
import tensorflow as tf
import cv2
import numpy as np
import matplotlib.pyplot as plt
# Chargement de l'image
image_path = '20220916_213716 (1).jpg'
image = cv2.imread(image_path)
```

```
# Prétraitement de l'image pour l'entrée du modèle
input_image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
input_image = cv2.resize(input_image, (300, 300))
input_image = np.expand_dims(input_image, axis=0)
# Définir la fonction d'inférence
@tf.function
def run_inference(image) :
    model_pred = model.signatures['serving_default']
    detections = model_pred(image)
    return detections
# Prédiction de l'image à l'aide du modèle
detections = run_inference(input_image)
# Analyser les détections et dessiner des rectangles autour des humains détectés
for i, detection in enumerate(detections['detection_scores'][0]) :
    score = detection.numpy()
    class_id = int(detections['detection_classes'][0][i].numpy())
    if score > 0.5 and class_id == 1:
        box = detections['detection_boxes'][0][i].numpy()
        (start_y, start_x, end_y, end_x) = box
        (h, w) = image.shape[:2]
        start_x = int(start_x * w)
        start_y = int(start_y * h)
        end_x = int(end_x * w)
        end_y = int(end_y * h)
        cv2.rectangle(image, (start_x, start_y), (end_x, end_y), (0, 255, 0), 2)
        cv2.putText(image, f"Person : {score}", (start_x, start_y - 10),

                    cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)
        roi = image[start_y :end_y, start_x :end_x]
        output_image_path = os.path.join(output_folder, f'person_{i}.jpg')
        cv2.imwrite(output_image_path, roi)
# Afficher l'image avec les rectangles entourant les humains détectés
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
plt.imshow(image)
plt.axis('off')
plt.show()
```

3.5.2 Résultats de détection et reconnaissance faciale

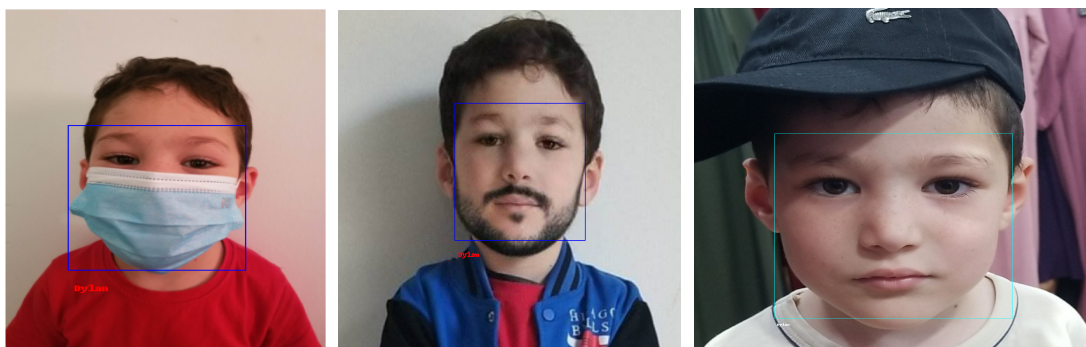
Code sur la reconnaissance faciale

Nous allons effectuer des tests sur des personnes existantes dans la base de données et sur d'autres qui n'existent pas.

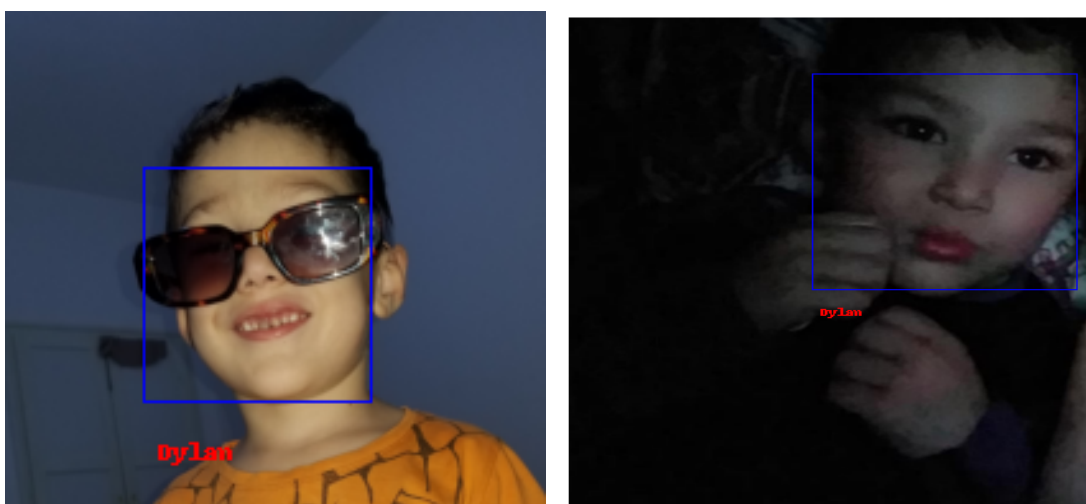
1. Exemple des personnes déjà enregistré dans la base de données

La figure ci-dessous (3.15) illustre la capacité de notre programme (puissance) à recon-

naître le visage d'une personne, indépendamment des filtres appliqués tels que des lunettes, une barbe, un masque, etc

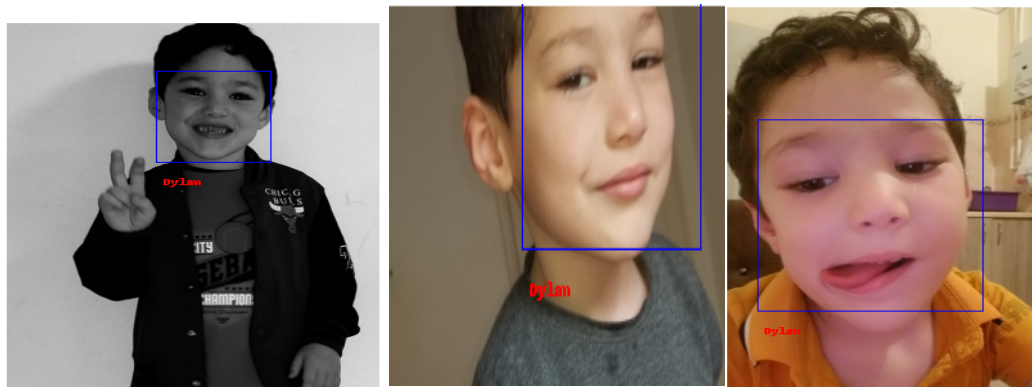


Ex1. Avec bavette Ex2. avec barbe Ex3. Avec chapeau



Ex4. Avec lunette

Ex5. dans le noir



Ex6. mode sombre

Ex7. avec grimace

Fig. 3.15 : Exemple reconnaissance facial

2. Exemple sur des personnes inconnu

Lorsqu'une image est exécutée et que des visages inconnus sont détectés, le programme affiche une alerte et enregistre ces visages inconnus dans un dossier appelé "faces_unknown". L'administrateur peut ensuite récupérer les images de ce dossier et décider s'il faut les ajouter au dossier des personnes connues ou non (figure 3.16).

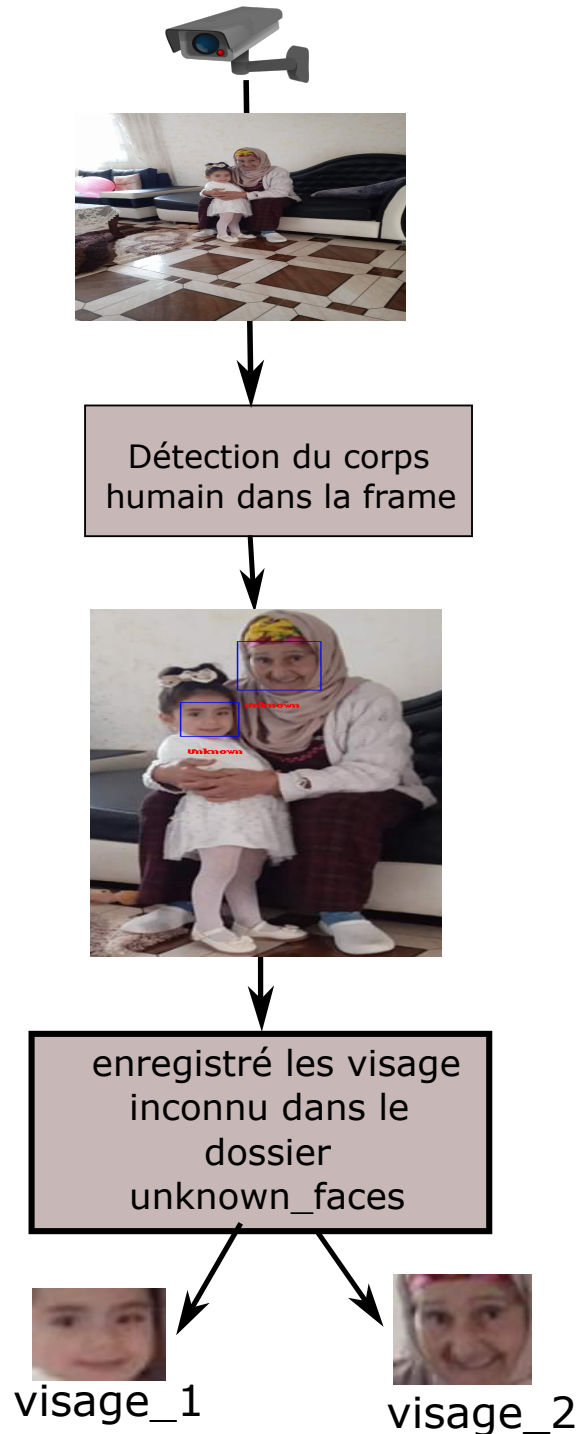


Fig. 3.16 : Détection des personnes inconnu

Les étapes suivantes décrivent la procédure à suivre pour ajouter une personne à la base

de données non

personnes qui ne figurent pas encore dans la base de données, mais que l'on souhaite ajouter

La décision d'ajouter ou non la personne à la base de données sera prise par l'administrateur. voir la figure 3.17

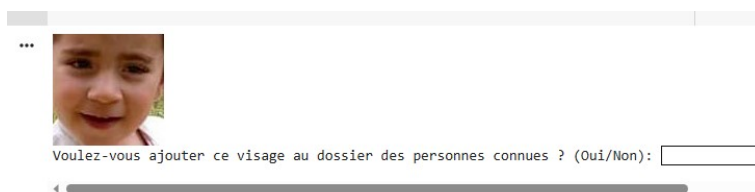


Fig. 3.17 : Décision

Si la réponse est oui et que la personne doit être ajoutée à la base de données, attribuons-lui un nom comme illustré dans la figure ci-dessous (3.18)

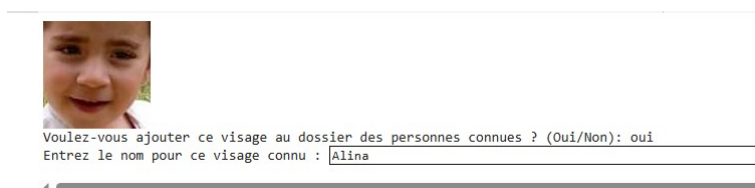


Fig. 3.18 : donner un nom a la personne

Une fois la personne inconnue ajoutée à la base de données, le programme affiche un message de confirmation (voire figure3.19)

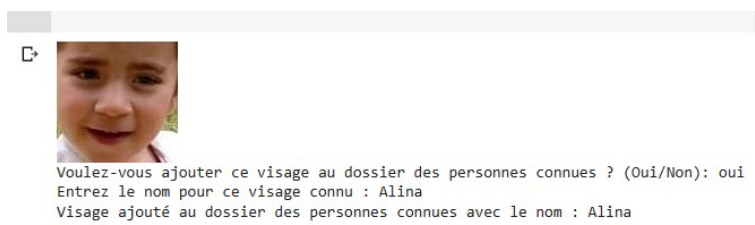


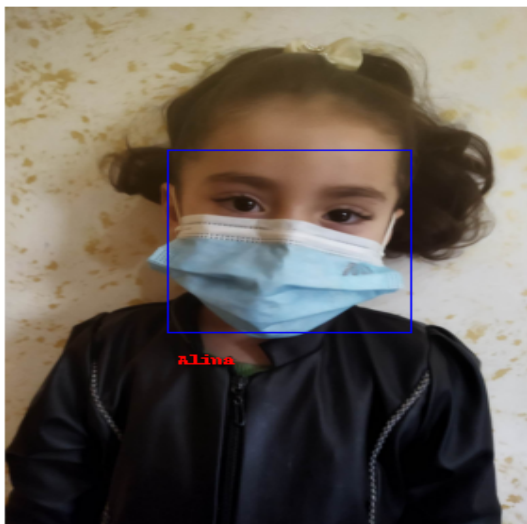
Fig. 3.19 : Visage ajouter a la base de deonnée

La figure (3.20) illustre le résultat final obtenu après l'ajout de la personne inconnue à la base de données.

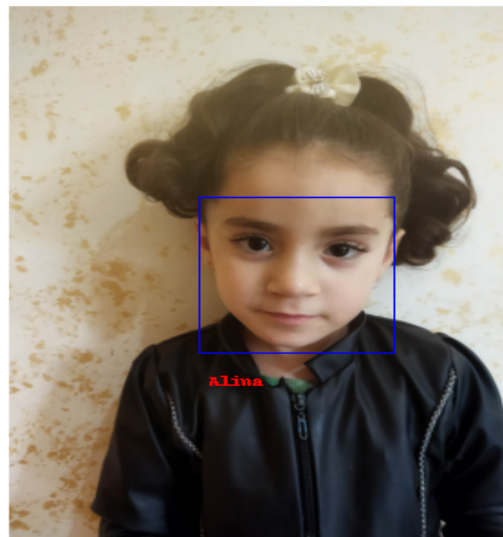


Fig. 3.20 : Resultat Final

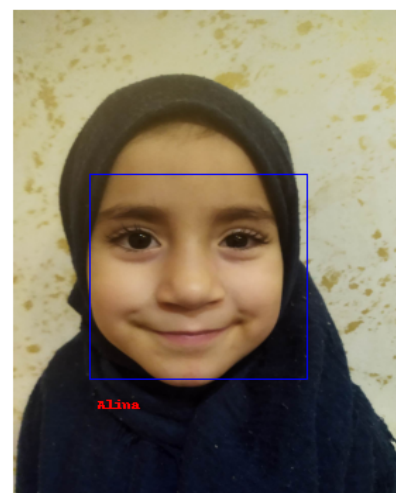
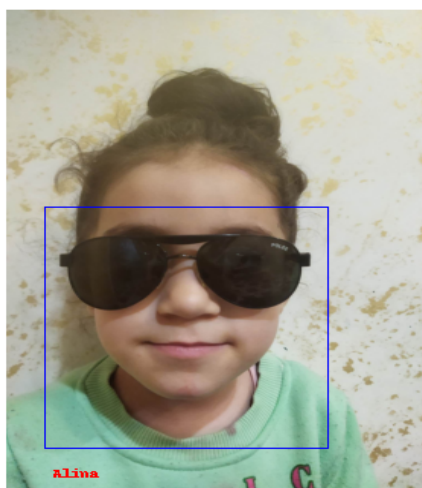
Après avoir ajouté Alyna à la base de données des personnes connues, nous avons testé notre programme en utilisant un ensemble d'images. Voici les résultats obtenus voir figure (3.21)



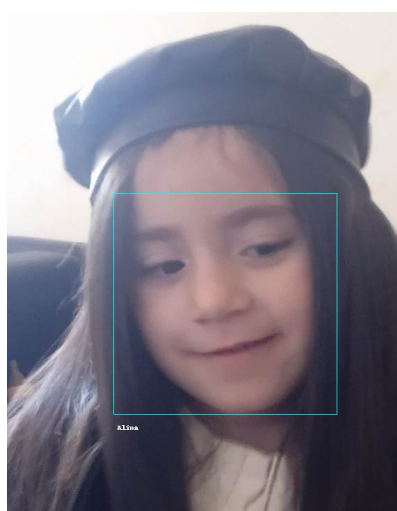
Ex1. alina avec bavette



Ex2. alina reconnais



Ex3. alina avec lunette Ex4. alina avec lunette Ex5. alina avec foulard et foulard



Ex6. alina avec chapeau Ex7. alina avec bruit Ex8. alina avec une expression facial

Fig. 3.21 : teste sur Alina

exemple de personnes qui ne figurent pas encore dans la base de données, et qui sont considérées comme des intrus.



Fig. 3.22 : Détection de visage inconnu

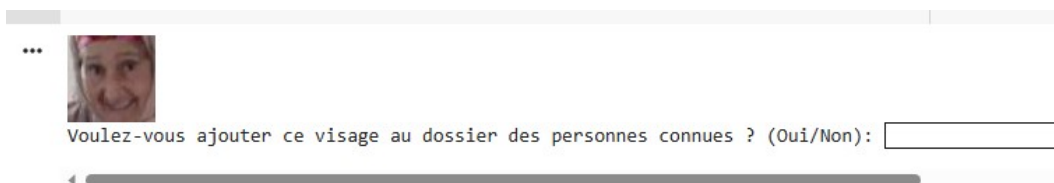


Fig. 3.23 : Décision

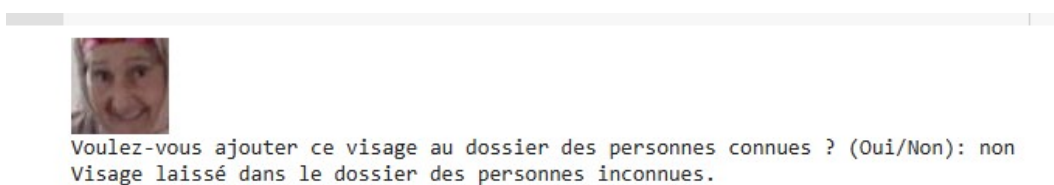


Fig. 3.24 : Resultat final

Code de la détection et la reconnaissance facial

Dans cette section nous mettons notre code python pour la détection et la reconnaissance facial :

```
import face_recognition
from PIL import Image, ImageDraw
from IPython.display import display
```

Chapitre 3. Notre proposition : SSD pour la détection d'intrusion

```
unknown_image = face_recognition.load_image_file("20220916_213716 (1).jpg")
pil_image = Image.fromarray(unknown_image)
draw = ImageDraw.Draw(pil_image)
for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings)
    # Vérifiez si le visage correspond à un visage connu
    matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
    name = "Unknown"
    face_distances = face_recognition.face_distance(known_face_encodings,

    face_encoding)
    best_match_index = np.argmin(face_distances)
    if matches[best_match_index] :
        name = known_face_names[best_match_index]
    else :
        # Le visage est inconnu, alerte l'utilisateur et enregistre la photo dans le

        dossier "unknown_faces"
        name = "Unknown"
        print("Visage inconnu détecté !")
        unknown_face_image = unknown_image[top :bottom, left :right]
        unknown_face_pil_image = Image.fromarray(unknown_face_image)
        unknown_face_pil_image.save(f"unknown_faces/unknown_face_{len(os.listdir

        ('unknown_faces')) + 1}.jpg")

    # Dessinez une boîte autour du visage avec le module Pillow
    draw.rectangle(((left, top), (right, bottom)), outline=(0, 0, 255))
    print(name)
# Dessinez une étiquette avec le nom sous le visage

    text_width, text_height = draw.textsize(name)
    #text_width, text_height = draw.textsize(name)
    draw.text((left + 6, bottom - text_height + 25), name, fill=(255, 0, 0, 255),)

# Remove the drawing library from memory as per the Pillow docs
del draw

# Display the resulting image
display(pil_image)

plt.show()
```

3.6 Conclusion

Nous avons introduit dans cette section une étude complète sur la détection humaine et la reconnaissance faciale utilisant notre méthode appelée SSD-FID. Un schéma global de notre proposition été donné en mentionnant toutes les étapes de notre approche. Les résultats des différents tests effectués ont confirmé le fonctionnement de la détection des personnes parmi les autres objets, l'identification de ces personnes à traver la reconnaissance faciale et la distinction entre un intrus et une personne autorisée d'entrer dans une zone surveillée. Toutes les données images sont enregistrées dans une base de données pour d'éventuelles utilisations.

Conclusion et perspectives

Conclusion Générale

Notre travail consistait à proposer un système de surveillance permettant la détection humaine, la reconnaissance faciale et l'identification des intrus. Nous avons ainsi étudié différents travaux sur les méthodes qui ont traité ces problématiques. Le but était de cerner les méthodes robustes permettant d'analyser en temps réel les flux vidéos et de faire de la reconnaissance. SSD est l'une de ces puissantes méthodes.

Nous avons réalisé un système intelligent de surveillance basé sur la méthode SSD pour l'identification d'intrusion. Pour cela nous avons réalisé un système nommé SSD-FID qui permet

- la détection des humains afin de pouvoir localiser chaque personne dans la zone de couverture de la caméra.
- la détection du visage pour avoir le nombre de personne à reconnaître.
- la reconnaissance faciale qui permet d'identifier chaque personne comme connue ou intrus.

Toutefois notre approche peut être améliorée :

- en intégrant la reconnaissance des personnes par leurs profils.
- prendre en charge le traitement des images vidéos en temps réel.
- embarquer sur des systèmes mobiles comme des voitures ou drones.
- ajouter des services d'alertes en utilisant les smartphones.
- respecter la vie privée des individus.

Bibliographie

- [1] Adlan Hakim Ahmad et al. “Real time face recognition of video surveillance system using haar cascade classifier”. In : *Indonesian Journal of Electrical Engineering and Computer Science* 21.3 (2021), p. 1389-1399.
- [2] Mohd Aquib Ansari et Dushyant Kumar Singh. “Human detection techniques for real time surveillance : A comprehensive survey”. In : *Multimedia Tools and Applications* 80 (2021), p. 8759-8808.
- [3] Zankruti Arya et Vibha Tiwari. “Automatic face recognition and detection using OpenCV, haar cascade and recognizer for frontal face”. In : *Int. J. Eng. Res. Appl. www.ijera.com* 10.6 (2020), p. 13-19.
- [4] Ibrahim Benamiour, Souad Encadreur Biad et al. “Mise au point d’un système de reconnaissance de visage basée Arduino”. Thèse de doct. Université de Jijel, 2019.
- [5] Abderraouf BERRAMDANE. “Détection de la fatigue du conducteur avec Raspberry Pi”. In : (2022).
- [6] Yassine Bouafia, Larbi Guezouli et Hicham Lakhlef. “Human Detection in Surveillance Videos Based on Fine-Tuned MobileNetV2 for Effective Human Classification”. In : *Iranian Journal of Science and Technology, Transactions of Electrical Engineering* 46.4 (2022), p. 971-988.
- [7] Malik Braik, Hussein Al-Zoubi et Heba Al-Hiary. “Pedestrian detection using multiple feature channels and contour cues with census transform histogram and random forest classifier”. In : *Pattern Analysis and Applications* 23.2 (2020), p. 751-769.
- [8] RK Chandana et AC Ramachandra. “Real Time Object Detection System with YOLO and CNN Models : A Review”. In : *arXiv preprint arXiv :2208.00773* (2022).
- [9] Tausif Diwan, G Anirudh et Jitendra V Tembhrne. “Object detection using YOLO : Challenges, architectural successors, datasets and applications”. In : *Multimedia Tools and Applications* 82.6 (2023), p. 9243-9275.
- [10] Ross Girshick. “Fast r-cnn”. In : *Proceedings of the IEEE international conference on computer vision*. 2015, p. 1440-1448.
- [11] Ross Girshick et al. “Rich feature hierarchies for accurate object detection and semantic segmentation”. In : *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, p. 580-587.
- [12] Guangxing Han et al. “Meta faster r-cnn : Towards accurate few-shot object detection with attentive feature alignment”. In : *Proceedings of the AAAI Conference on Artificial Intelligence*. T. 36. 1. 2022, p. 780-789.

- [13] Kaiming He et al. “Mask r-cnn”. In : *Proceedings of the IEEE international conference on computer vision*. 2017, p. 2961-2969.
- [14] Rui-Yang Ju et al. “Efficient convolutional neural networks on Raspberry Pi for image classification”. In : *Journal of Real-Time Image Processing* 20.2 (2023), p. 1-9.
- [15] Ali Farouk Khalifa, Eman Badr et Hesham N Elmahdy. “A survey on human detection surveillance systems for Raspberry Pi”. In : *Image and Vision Computing* 85 (2019), p. 1-13.
- [16] Muhammad Attique Khan et al. “Human action recognition using fusion of multiview and deep features : an application to video surveillance”. In : *Multimedia tools and applications* (2020), p. 1-27.
- [17] Douki Koceila et Cherk Ramdane. “Conception et réalisation d’un système de surveillance d’une serre agricole avec une carte Raspberry PI 2”. Thèse de doct. Université Mouloud Mammeri, 2017.
- [18] Ashu Kumar, Amandeep Kaur et Munish Kumar. “Face detection techniques : a review”. In : *Artificial Intelligence Review* 52 (2019), p. 927-948.
- [19] Arati Kushwaha, Ashish Khare et Manish Khare. “Human activity recognition algorithm in video sequences based on integration of magnitude and orientation information of optical flow”. In : *International Journal of Image and Graphics* 22.01 (2022), p. 2250009.
- [20] Redouan Lahmyed, Mohamed El Ansari et Zakaria Kerkaou. “A novel visible spectrum images-based pedestrian detection and tracking system for surveillance in non-controlled environments”. In : *Multimedia Tools and Applications* 81.27 (2022), p. 39275-39309.
- [21] Alessandro Leone et al. “Human Postures Recognition by Accelerometer Sensor and ML Architecture Integrated in Embedded Platforms : Benchmarking and Performance Evaluation”. In : *Sensors* 23.2 (2023), p. 1039.
- [22] *Les ordinateurs monocartes*. <https://www.onesdr.com/wp-content/uploads/2021/01/Best-SBC-title-pic.png>.
- [23] Kaidong Li et al. “Object detection with convolutional neural networks”. In : *Deep Learning in Computer Vision*. CRC Press, 2020, p. 41-62.
- [24] 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.
- [25] Yuxuan Liu et al. “Active shape model registration of ocular structures in computed tomography images”. In : *Physics in Medicine & Biology* 67.22 (2022), p. 225008.
- [26] Wen-Yao Lu et YANG Ming. “Face detection based on viola-jones algorithm applying composite features”. In : *2019 International Conference on Robots & Intelligent System (ICRIS)*. IEEE. 2019, p. 82-85.
- [27] Rahma Mammeri, Sama Laouadi et Souheila Benkhaira. “La Reconnaissance biométrique de la démarche”. In : (2022).
- [28] Muralindran Mariappan, Lim Kean Thong et Karthigayan Muthukaruppan. “A design methodology of an embedded motion-detecting video surveillance system”. In : *International Journal of Integrated Engineering* 12.2 (2020), p. 55-69.

- [29] Ryo Matsumura et Akitoshi Hanazawa. "Human detection using color contrast-based histograms of oriented gradients". In : *International Journal of Innovative Computing, Information and Control* 15.4 (2019), p. 1211-1222.
- [30] Nafis Mustakim et al. "Face recognition system based on raspberry Pi platform". In : *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*. IEEE. 2019, p. 1-4.
- [31] Vipul Narayan et al. "Deep Learning Approaches for Human Gait Recognition : A Review". In : *2023 International Conference on Artificial Intelligence and Smart Communication (AISC)*. IEEE. 2023, p. 763-768.
- [32] Huy Hoang Nguyen et al. "Yolo based real-time human detection for smart video surveillance at the edge". In : *2020 IEEE Eighth International Conference on Communications and Electronics (ICCE)*. IEEE. 2021, p. 439-444.
- [33] Chen Ning et al. "Survey of pedestrian detection with occlusion". In : *Complex & Intelligent Systems* 7 (2021), p. 577-587.
- [34] *obj detection*. <https://medium.com/swlh/image-object-detection-tensorflow-2-object-detection-api-af7244d4c34e>.
- [35] Wahyu Rahmaniar et Ari Hernawan. "Real-time human detection using deep learning on embedded platforms : A review". In : *Journal of Robotics and Control (JRC)* 2.6 (2021), p. 462-468.
- [36] *raspberrypi*. <https://www.raspberrypi.com/>.
- [37] *raspberrypi4B*. <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>.
- [38] Shaoqing Ren et al. "Faster r-cnn : Towards real-time object detection with region proposal networks". In : *Advances in neural information processing systems* 28 (2015).
- [39] Adrian Rosebrock. "Pedestrian Detection OpenCV". In : *Recuperado de http://www.pyimagesearch.com/2015/11/09/pedestrian-detection-opencv* (2015).
- [40] Saadman Sakib et al. "A framework for pedestrian attribute recognition using deep learning". In : *Applied Sciences* 12.2 (2022), p. 622.
- [41] G Sreenu et Saleem Durai. "Intelligent video surveillance : a review through deep learning techniques for crowd analysis". In : *Journal of Big Data* 6.1 (2019), p. 1-27.
- [42] KH Teoh et al. "Face recognition and identification using deep learning approach". In : *Journal of Physics : Conference Series*. T. 1755. 1. IOP Publishing. 2021, p. 012006.
- [43] Juan Terven et Diana Cordova-Esparza. "A Comprehensive Review of YOLO : From YOLOv1 to YOLOv8 and Beyond". In : *arXiv preprint arXiv :2304.00501* (2023).
- [44] J Usha Rani et P Raviraj. "Real-Time Human Detection for Intelligent Video Surveillance : An Empirical Research and In-depth Review of its Applications". In : *SN Computer Science* 4.3 (2023), p. 258.
- [45] Hengsen Wang et al. "Video Saliency Object Detection with Motion Quality Compensation". In : *Electronics* 12.7 (2023), p. 1618.
- [46] Esra Yılmaz et Sibel Tariyan Özyer. "Remote and autonomous controlled robotic car based on arduino with real time obstacle detection and avoidance". In : (2019).

- [47] Ayesha Younis et al. “Real-time object detection using pre-trained deep learning models MobileNet-SSD”. In : *Proceedings of 2020 the 6th international conference on computing and data engineering*. 2020, p. 44-48.
- [48] Kaipeng Zhang et al. “Joint face detection and alignment using multitask cascaded convolutional networks”. In : *IEEE signal processing letters* 23.10 (2016), p. 1499-1503.
- [49] XueMei Zhao et ChengBing Wei. “A real-time face recognition system based on the improved LBPH algorithm”. In : *2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP)*. IEEE. 2017, p. 72-76.
- [50] Wei Zhou et al. “Histogram of oriented gradients feature extraction from raw bayer pattern images”. In : *IEEE Transactions on Circuits and Systems II : Express Briefs* 67.5 (2020), p. 946-950.
- [51] Zhengxia Zou et al. “Object detection in 20 years : A survey”. In : *Proceedings of the IEEE* (2023).