الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Electronique

Mémoire de Projet de Fin d'Études

Présenté par

AIT ALI YAHIA Rayane

&

MENNAA Maroua

Pour l'obtention du diplôme de Master en Electronique

Spécialité : Electronique des Systèmes Embarqués

Thème

Application du Deep Learning en vidéo surveillance

Promoteur: Mr. KABIR Yacine

Président: Mr. YKHLEF Farid

Examinateur: Mr. NAMANE Abderrahmane

Année Universitaire : 2022-2023

Dédicace

Je dédie ce travail particulièrement à la personne la plus chère à mes yeux, ma mère, pour son amour et les sacrifices qu'elle a faits pour moi, ainsi que pour ses encouragements et tout ce qu'elle a pu m'apporter. Je prie Dieu de la protéger.
Je tiens également à exprimer ma profonde reconnaissance envers mon père pour son soutien inconditionnel, ses conseils précieux et son amour. .

À mon grand-père et à ma grand-mère Lila que j'aime énormément, que Dieu les garde.

A mes chères sœurs Sarah, Manel et Lilia et à mes chers neveux..

À ma binôme Maroua, pour son écoute, son attention et son soutien tout au long de ce projet.

À tous les membres de ma famille.

À tous mes amis qui m'ont soutenue.

Et à toutes les autres personnes qui m'ont apporté leur soutien et que j'ai oublié de mentionner.

Ait Ali Yahia Rayane

Dédicace

Je dédie ce travail :

Aux deux personnes les plus précieuses que j'ai perdues dans ma vie, à ceux qui nous ont quittés et je ne les oublierai jamais.

A l'âme de mon cher père et mon frère Aymen
(Que Dieu leur fasse miséricorde et leur accorde le paradis).

A ma chère mère, Merci Mama d'être toujours la pour nous, Merci pour tes sacrifices et tes encouragements.

A mes deux chères sœurs Imene et Batoul.
Et mon cher frère Oussama

A tous les membres de ma famille.

A tous mes amis, et surtout à tous ceux qui m'ont aidé à accomplir ce travail.

Maroua Mennaa

Remerciments

D'abord et avant tout, Nous remercions Dieu le tout puissant qui nous a donné la volonté et le courage pour réaliser ce travail.

Nous remercions vivement notre encadreur Mr kabir Yacine pour avoir dirigé ce mémoire.

Nous avons eu le plaisir de travailler sous sa direction. Nous tenons à le remercier pour sa gentillesse et sa spontanéité avec lesquelles il a dirigé ce travail, ainsi que pour sa disponibilité et ses précieux conseils qui nous ont permis d'atteindre notre objectif. Nous espérons que sa confiance en nous et notre mémoire est pleinement satisfaite.

Nous espérons que votre confiance que vous nous accordez et que ce mémoire est à la hauteur de vos espérances.

Nous exprimons notre sincère gratitude aux membres du jury, Mr. Namane Abderrahmane et Mr. YKhlef Farid, pour avoir accepté de juger notre travail.

Nous souhaitons également exprimer notre reconnaissance à notre chef de spécialité, Mme Naceur Djamila, pour nous avoir accueillis dans la spécialité électronique des systèmes embarqués. Sa confiance en notre potentiel nous a motivés à donner le meilleur de nous-mêmes.

Nous remercions tous les enseignants du cycle Licence et Master ainsi que toutes les personnes qui nous ont soutenus de près ou de loin dans la réalisation de ce travail. Leurs connaissances partagées.

Nous tenons à exprimer notre plus profonde gratitude envers nos parents et nos sœurs qui ont été une source constante de soutien et d'encouragement tout au long de notre parcours académique. Leur amour inconditionnel, leurs sacrifices et leur confiance en nous ont été des piliers solides dans notre réussite. Leurs encouragements et leur présence ont été une source inestimable de motivation et de réconfort.

Nous espérons sincèrement que ce mémoire est à la hauteur des attentes de notre encadreur et de l'ensemble de notre jury. C'est grâce à tous ces soutiens que nous avons pu mener ce projet à terme. Nous en sommes profondément reconnaissants.

Résumé :

L'objectif de ce projet de fin d'étude est de réaliser un système capable de compter les foules et de vérifier les visages des personnes suspects présents dans une vidéo surveillance en temps réel en utilisant l'apprentissage profond. Ce projet est divisé en deux parties, dans la première partie le modèle yolov3 est utilisé pour détecter et compter les personnes, et dans la deuxième partie le modèle FaceNet est utilisé afin de détecter les visages des personnes suspects et nous l'entraînons également sur des images en supposant qu'il s'agit d'images des personnes suspectes, nous avons obtenu des résultats satisfaisants et enfin, nous combinons ces deux parties dans une interface graphique.

Les mots clés : apprentissage profond, vidéo surveillance, yolov3, vérification des visages.

ملخص :

الهدف من مشروع نهاية الدراسة هذا هو إنشاء نظام قادر على عد الحشود والتحقق من وجوه الأشخاص المشبوهين الموجودين في فيديو المراقبة في الوقت الفعلي باستخدام التعلم العميق. ينقسم هذا المشروع إلى جزأين, في الجزء الأول يستخدم نموذج Yolov3 لاكتشاف الأشخاص و عدهم, و في الجزء الثاني يستخدم نموذج Facent لاكتشاف وجوه المشبوهين ونقوم بتدريبه على صور بافتراض أنها صور لأشخاص مشبوهة ,تحصلنا على نتائج مرضية ،و في الأخير نقوم بجمع هاذين الجزأين في واجهة المستخدم الرسومي.

كلمات المفتاحية: التعلم العميق, فيديو المراقبة,yolov3,التحقق من الوجوه.

Abstract:

The objective of this end-of-study project is to create a system capable of counting crowds and and verify the faces of suspect people present in real-time video surveillance using deep learning. This project is divided into two parts, in the first part the yolov3 model is used to detect and count people, and the second part, it uses the FaceNet model in order to detect the faces of suspect people and we also train it on images, assuming they are images of suspect people, we obtained satisfactory results and finally, we combine these two parts in the graphic interface.

Keywords: deep learning, video surveillance, yolov3, face verification.

Liste des abréviations

AP : Average Precision

Bbox : Boîte englobante (bounding box)

CNN : convolutional Neural Network

DL : deep learning

FCL : Fully Connected Layer

FPS : Frame Per Second

GPU : Graphics processing unit

IA : Intelligence Artificielle

IoU : Intersection over Union

MTCNN : Multi-Task Cascaded Convolutional Networks

mAP : mean average precision

ML : machine learning

NMS : suppression non maximale

OpenCv : Open Source Computer Vision Library

ReLU : the rectified linear unit

R-CNN : Region-based Convolutional Neural Network

SSD : Single Shot MultiBox Detector

YOLO : You Only Look once

Yolov3 : YOLO version 3

Table des matières

Table des figures

Liste des tableaux

INTRODUCTION GENERALE

Aujourd'hui, nous avons remarqué que les systèmes de sécurité tels que les caméras de surveillance sont devenues abondantes dans les rues, les hypermarchés et les aéroports, les stades, etc. C'est-à-dire les zones très fréquentées, et des zones avec équipements sociaux, pour surveiller les personnes et leurs déplacements afin de contrôler le système de sécurité dans ces lieux.

Le but de ce projet se concentre sur l'amélioration de ce système de sécurité dans les zones surpeuplées en utilisant le deep learning. Nous avons également remarqué le développement de l'intelligence artificielle dans plusieurs domaines comme la vision par ordinateur, et ce développement est très rapide et accrocheur, et parmi les technologies de vision par ordinateur figurent la détection d'objets, et la reconnaissance faciale. La détection d'objets peut être appliquée pour réaliser le décompte des objets présents dans une scène, ainsi que pour déterminer et suivre précisément leurs emplacements, tout en leur attribuant des étiquettes précises. La reconnaissance faciale est un processus qui vise à identifier et à vérifier automatiquement l'identité des individus en se basant sur leurs caractéristiques faciales.

Dans ce projet, nous proposons d'appliquer l'apprentissage en profondeur (deep learning) à la vidéosurveillance. Nous nous intéressons à deux techniques principales. La première technique concerne le comptage de foule, qui consiste à surveiller le nombre de personnes dans une zone fréquentée afin d'assurer la sécurité et d'éviter la surpopulation. La deuxième technique concerne l'identification des visages des personnes suspectes. Nous nous concentrons spécifiquement sur la résolution du problème du comptage des foules en utilisant le modèle YOLOv3 pour détecter les personnes présentes et les compter en temps réel. De plus, nous utilisons le modèle FaceNet pour la reconnaissance faciale des personnes suspectes, ce qui nous permet d'obtenir une précision élevée dans cette tâche.

Ce mémoire est structuré en quatre chapitres.

- Le premier chapitre :

Dans ce chapitre nous allons introduire la vidéosurveillance dans son ensemble, en abordant les concepts généraux ainsi que la problématique liée à ce domaine. nous allons examiner ensuite plus spécifiquement le comptage de foule et la reconnaissance faciale.

- Le deuxième chapitre :

Ce chapitre se concentrera sur l'application de l'intelligence artificielle sur la vidéosurveillance, et plus précisément du machine learning et du Deep Learning, dans la vidéosurveillance.

- Le troisième chapitre :

Dans le troisième chapitre, nous aborderons la détection des objets par vision par ordinateur, ainsi que les modèles de reconnaissance faciale tout en expliquant le modèle yolov3 et FaceNet d'une manière plus détaillée.

- Le quatrième chapitre :

Dans ce chapitre, nous mettrons l'accent sur l'implémentation et la conception de notre système proposé. Nous détaillerons les différentes étapes du système que nous avons développé et discuterons des résultats obtenus.

Chapitre 1

LA VIDEO DE SURVEILLANCE

1.1 Introduction

La vidéosurveillance est aujourd'hui un élément incontournable de notre société moderne. Que ce soit dans les lieux publics, les entreprises, les résidences ou les institutions gouvernementales, les caméras de vidéosurveillance sont omniprésentes. Leur rôle est de capturer et d'enregistrer des images et des vidéos afin de garantir la sécurité des lieux et des individus, de prévenir les actes criminels et de faciliter les enquêtes en cas d'incident.
Dans ce chapitre, nous présenterons une vue d'ensemble de la vidéosurveillance, ainsi que le problème qui nous a conduit à entreprendre ce projet. Nous aborderons également les travaux de recherche existants sur le comptage de foule et la reconnaissance faciale.

1.2 Vidéo Surveillance

La vidéosurveillance fait référence à l'utilisation de caméras et de systèmes de transmission d'images pour surveiller à distance un espace, qu'il soit public ou privé.

Le système comprend l'installation de caméras à l'intérieur et/ou à l'extérieur de la zone surveillée. Les images capturées par ces caméras peuvent être traitées automatiquement, consultées en temps réel et archivées. En cas de détection d'une intrusion ou d'un mouvement suspect, le système envoie immédiatement une alerte par téléphone, SMS ou e-mail au propriétaire ou à une société de sécurité chargée de la surveillance [1].

FIGURE 1.1 – video surveillance [2]

1.3 Historique

Le système de vidéosurveillance a été inventé en Allemagne en 1942 par l'ingénieur Walter Bruch. Il a été utilisé initialement par l'armée allemande pour surveiller les lancements de missiles pendant la Seconde Guerre mondiale. Après la guerre, la technologie a été exportée aux États-Unis et commercialisée en 1949. En 1969, le premier système de vidéosurveillance domestique a été commercialisé par Marie Van Brittan Brown. La ville d'Olean, dans l'État de New York, a été la première aux États-Unis à surveiller ses rues en 1968, suivi de Times Square en 1973. Le Royaume-Uni a généralisé la vidéosurveillance dans les années 1980 en raison des attentats de l'IRA (Armée Républicaine Irlandaise). Londres est actuellement la ville d'Europe la plus télé surveillée. Au fil des décennies, la vidéosurveillance a évolué avec l'adoption de technologies numériques telles que la reconnaissance faciale, l'analyse vidéo avancée et la connectivité réseau, permettant une surveillance plus précise et une réponse plus rapide aux incidents de sécurité [3].

1.4 Évolution de la vidéo surveillance

Les premières caméras avaient des images en noir et blanc de mauvaise qualité et ne pouvaient pas zoomer ou changer de perspective. Les caméras modernes les plus puissantes ont des caméras couleur qui permettent un zoom et une mise au point très nets. Les dispositifs d'enregistrement et d'analyse sont plus précis et efficaces. En contrôlant ces caméras avec un

ordinateur, vous pouvez suivre des mouvements tels que B. Détecter le mouvement là où il ne devrait pas y avoir de mouvement, ou vice versa, en se concentrant sur une personne et en la suivant à travers une scène. L'ordinateur peut coordonner plusieurs caméras pour le suivre dans l'espace urbain. L'une des évolutions les plus probables de la vidéosurveillance est la coordination des enregistrements avec les données biométriques[4].

FIGURE 1.2 – Caméras de surveillance [5]

1.5 Domaines d'utilisation de la vidéosurveillance

La vidéosurveillance est un système de caméras placées dans des espaces publics ou privés pour les surveiller. Les images obtenues à l'aide du système de vidéosurveillance sont ensuite visualisées et/ou archivées.

Les entreprises utilisent la vidéosurveillance à des fins différentes. Voici une liste non exhaustive de ces utilisations potentielles :

• **Blocage des comportements malveillants**

Placer des caméras de vidéosurveillance, même fausses, dans des endroits bien visibles peut souvent dissuader les criminels de vol, d'agression, d'insultes, etc.

• **Fournir des preuves**

Les images fournies par la vidéosurveillance peuvent vous permettre d'identifier l'auteur et de fournir la preuve de sa culpabilité à un tribunal ou à une compagnie d'assurance.

• **Flux de contrôle**

Qu'il s'agisse de compter le nombre d'entrées et de sorties ou de voir où vont les gens dans un espace défini, la vidéosurveillance peut recueillir des informations stratégiques vitales pour l'aménagement des magasins, des parcs, des réseaux routiers et plus encore.

• **Contrôler les transactions en espèces dans les magasins**

Associées aux caisses, les caméras de vidéosurveillance permettent de suivre le bon déroulement des opérations de caisse : identification des saisies, annulations, remises, montant des encaissements par rapport aux encaissements, etc.

• **Lutte contre le terrorisme**

Grâce à la vidéosurveillance, les objets perdus ou abandonnés peuvent être repérés et inspectés immédiatement. Vous pouvez également remonter à l'époque où l'objet a été abandonné pour retrouver le propriétaire[6].

1.6 Architecture d'un système de vidéosurveillance

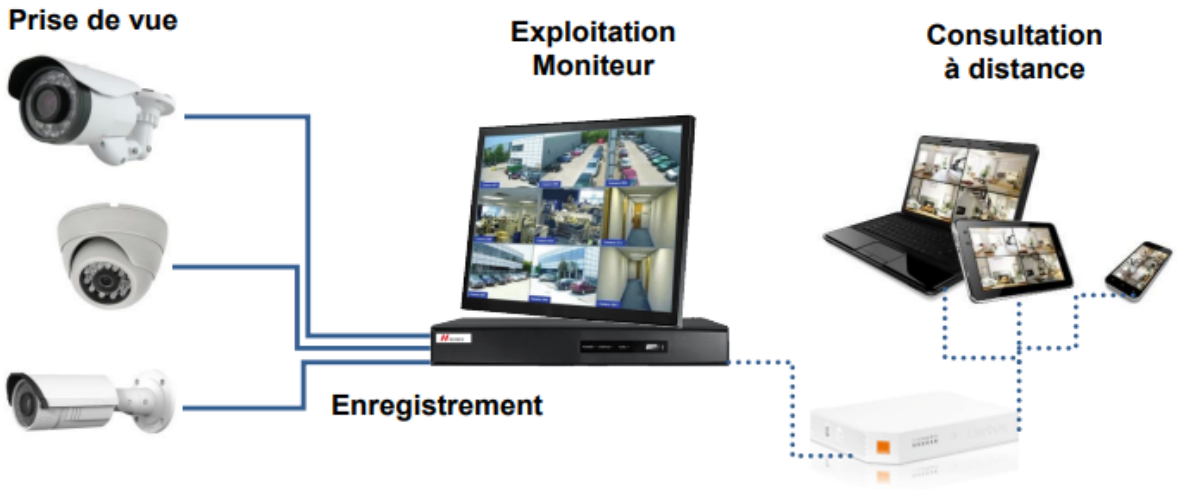

FIGURE 1.3 – Architecture d'un système de vidéosurveillance [7]

l'architecture d'un vidéo surveillance se compose par :

— **Prise de vue :** Caméra HD, caméra IP, caméra Wifi, caméra mobile, Web cam.

— **Enregistrement :** Disque dur, carte SD, serveur distant, Cloud.

— **Exploitation :** Enregistreur, logiciel CMS, clavier, souris, moniteur.

— **Consultation à distance :** Une solution pratique pour la surveillance à distance par un PC portable ou un téléphone etc.

1.7 Éléments essentiels de la vidéo surveillance

• **L'enregistreur**

L'enregistreur stocke les images de caméras analogiques et réseau. Les images peuvent être visualisées en direct, enregistrées, lues et transmises via le réseau. La durée d'enregistrement

dépend de la qualité de l'image, du nombre de caméras et de la capacité du disque dur[7].

• La compression

La compression de données vidéo réduit la taille des fichiers vidéo pour accélérer la transmission de données par rapport au format non compressé. Elle permet également de diminuer l'espace mémoire requis pour l'enregistrement.

• Le moniteur

Les écrans TFT utilisent des transistors à couche mince, également connus sous le nom de Thin Film Transistors (TFT) en anglais. Ces transistors sont des composants spéciaux à effet de champ qui permettent de créer des circuits électroniques de grande taille.

• La caméra

Le choix d'une caméra de vidéosurveillance dépend principalement de vos besoins et exigences. Les facteurs clés à considérer sont la luminosité, la qualité de l'image et la taille de l'objectif [7].

1.8 Problématique

Nous savons que les systèmes de sécurité tels que les caméras de surveillance sont devenus abondants dans les rues, les hypermarchés et les aéroports, les stades..., c'est-à-dire les zones très fréquentées, et zones avec des services sociaux.

Quelles mesures peuvent être prises, en intégrant des systèmes de vidéosurveillance, pour renforcer la sécurité dans les zones surpeuplées en permettant l'organisation, le comptage précis des individus présents et la détection des personnes suspectes parmi eux ?

Pour répondre à ce problème, nous proposons la technique de comptage de foule pour améliorer la surveillance vidéo à des fins de sécurité est en effet pertinente, comme la surveillance du nombre de personnes dans une zone surpeuplée telle qu'une gare ou lors de manifestations, etc., pour assurer la sécurité et éviter la surpopulation.

Ce processus est généralement réalisé à l'aide de techniques de vision par ordinateur et de traitement d'images. Et de notre part nous proposons d'appliquer l'apprentissage profond (DL) Sur vidéo surveillance pour détecter et compter les individus dans une vidéo en temps réel.

Et aussi d'utiliser un des modèles de DL pour la reconnaissance faciale pour la détection des personnes suspects présentes dans une zone donnée et pour renforcer la surveillance sur eux.

1.9 Comptage de foule

Le comptage de foule fait référence au processus d'estimation du nombre de personnes présentes dans une zone, une photo ou une vidéo donnée.
C'est une tâche difficile qui a des applications dans divers domaines tels que la surveillance, la gestion des foules, le contrôle du trafic et l'organisation d'événements. Les techniques de comptage des foules bénéficient de la vision par ordinateur et des algorithmes d'apprentissage automatique pour analyser les données visuelles et estimer avec précision la densité de la foule.

FIGURE 1.4 – Compter les personnes dans les foules avec l'IA[8]

Ces dernières années Le comptage des foules à l'aide de l'apprentissage profond a donné des résultats prometteurs et est devenu une approche populaire. Les modèles de DL, ont démontré leur capacité à apprendre automatiquement des caractéristiques et des modèles complexes à partir d'images de foule, ce qui permet un comptage plus précis des foules.

1.9.1 Applications du comptage de foule

Le comptage de foule a diverses applications dans différents domaines. Voici quelques applications courantes du comptage de foule :

• **Sécurité publique :** il aide les autorités à gérer les foules lors d'événements et de manifestations etc. et à prévoir les problèmes qui pourraient survenir, et ceci afin d'assurer la sécurité publique.

• **Aménagement urbain :** il peut aider les urbanistes à comprendre la répartition de la population et les schémas de déplacement dans les villes.

• **Publicité et marketing :** il peut être utile pour les campagnes de publicité et de marketing extérieures.

• **Planification des transports :** il est utile dans la planification et la gestion des transports. Il permet d'estimer le nombre de personnes utilisant différents modes de transport tels que les bus, les trains etc.

• **Contrôle des foules et intervention d'urgence :** lors de situations d'urgence ou de catastrophes naturelles, le comptage des foules peut aider les intervenants d'urgence à estimer le nombre de personnes présentes dans les zones touchées.

• **Gestion d'événements :** il joue un rôle crucial dans la gestion d'événements pour les concerts, les festivals, et les conférences.

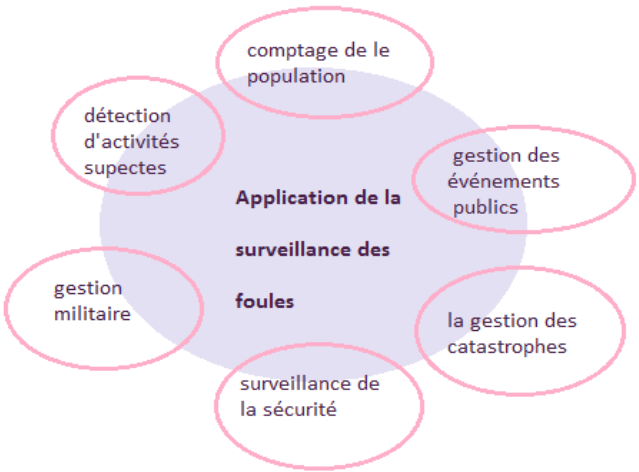

FIGURE 1.5 – applications de surveillance des foules

1.9.2 Les travaux réalisés sur le comptage de foule

Il existe plusieurs travaux dans la littérature sur le comptage de foule.

1.9.2.1 Comptage des manifestants en France avec Eurecam

Eurecam a été créée en 2005 par deux ingénieurs spécialisés en électronique et en algorithmes de traitement vidéo qui ont réuni leurs savoir-faire au sein d'une société innovante, une société spécialisée dans la vidéosurveillance intelligente et proposant notamment des solutions de comptage de foule à l'aide de caméras de surveillance équipées de logiciels d'analyse d'images.

Cette technologie a fait ses preuves et est utilisée pour calculer les flux de personnes dans les aéroports, les centres commerciaux ou les événements[9].

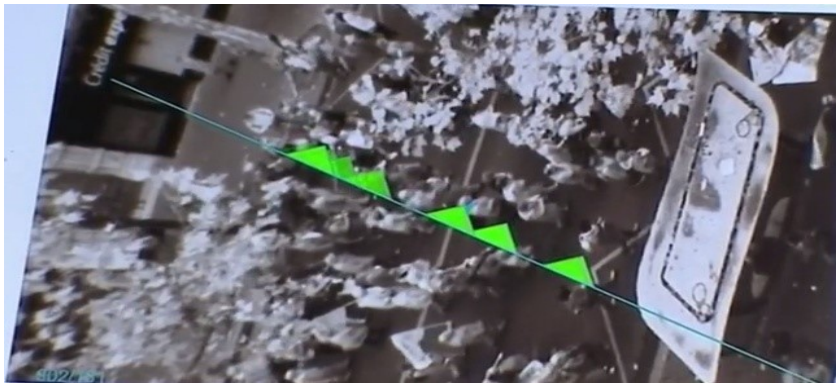

FIGURE 1.6 – l'affichage de l'image captée et la ligne qui est tracée[10]

En même temps des petits calculs humains sont effectués pendant une durée de 20 à 30 secondes à des intervalles différents. Les manifestants sont recomptés humainement sur les extraits vidéo pour ajuster le comptage effectué par le capteur et déterminer la marge d'erreur.

1.9.2.2 Comptage des foules à l'aide de la segmentation d'images sémantiques de bout en bout

— **Technique :** apprentissage profond.

— **Méthode utilisée : SSS (Segmentation de scène sémantique)** de bout en bout a une image bondée, cette méthode proposée a mis en évidence la région de la tête en supprimant la partie non-tête grâce à une nouvelle fonction de perte (loss) optimisée, le cadre proposé était basé sur la segmentation sémantique des scènes à l'aide d'un réseau de neurones convolutifs optimisés, et aussi sont basés sur la fonction de classification, ils ont obtenu un comptage de foule en intégrant des cartes de densité. L'algorithme

proposé a classé la foule comptant dans chaque image en groupes pour s'adapter à la variation survenant dans le décompte des foules[11].

FIGURE 1.7 – les échantillons de l'ensemble de données NWPU-Crowd et les cartes de densité de vérité terrain[12]

Base de données : dans cette méthode il ya quatre bases de données(NWPU-crowd ,UCF-QNRF ,Shanghai Tech ,World Expo10), qui rapportent de meilleurs résultats par rapport aux résultats précédents.

1.9.2.3 Un modèle de classification de la densité de foule profonde

- **Technique :** apprentissage profond.
- **Méthode utilisée :** Fully-convolutional-neural-network (FCNN), a été introduit pour l'analyse des foules et estimation de la densité de la foule et notamment pour le classement de la densité de foule .

  La méthode proposée se composait de trois parties, Deep-CNN features(caractéristiques), Processus de densité de foule (étiquetage d'image), Processus de classification de la densité de la foule.

  Une fois le FCNN formé, il peut être utilisé pour classer la densité de foule en temps réel, et le FCNN peut être utilisé pour analyser rapidement les images et fournir une classification de la densité de la foule.[13]
- **Processus de densité de foule (TESTING) :** Illustrent le processus de test de densité d'analyse de foule à l'aide de FCNN Premièrement, préparation d'un nouvel ensemble de données d'image de test, puis l'exécution de l'ensemble complet d'images pour les tests. Deuxième, teste de cinq classes à l'aide de FCNN. Enfin, obtenir des résultats de classification pour les cinq classes[13].

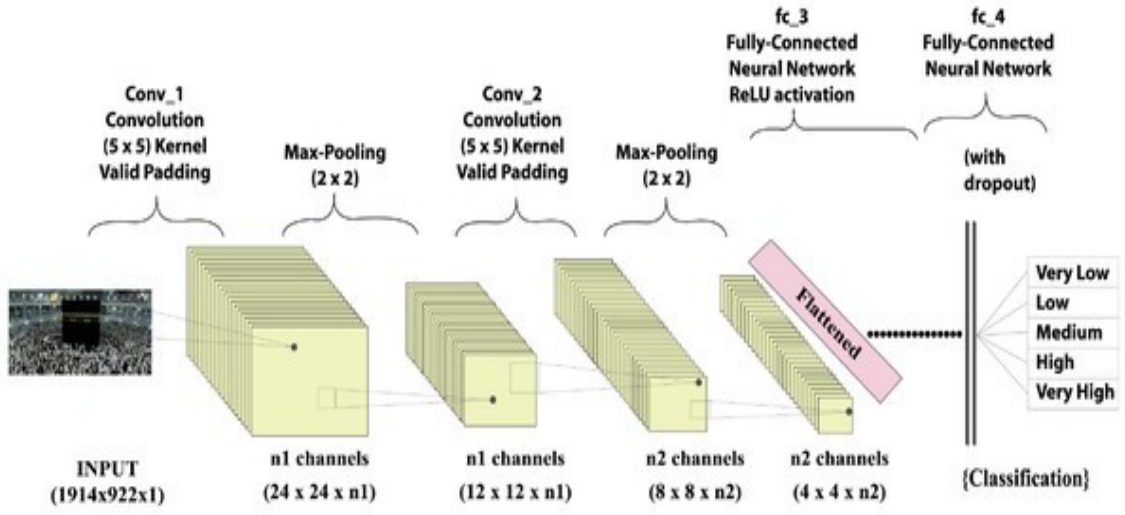

FIGURE 1.8 – les échantillons de l'ensemble de données NWPU-Crowd et les cartes de densité de vérité terrain[13]

- **Base de données :**

hajj-crowd : cette base de données a été collecté à partir de la diffusion en direct de YouTube à La Mecque Hajj à partir de 2015 à 2019, qui contient 27000 images basées sur 5 classes (very low , low, medium, high, very high) chaque classes étant composée de 5400 images .

1.9.2.4 Estimation rapide de la densité de foule avec des réseaux de neurones convolutifs

— **Technique :** apprentissage profond.

— **Méthode utilisée :** l'estimation la densité de la foule par un réseau neuronal convolutif optimisé (ConvNet), cette méthode est basée sur la multi-étape (stage) ConvNet qui se compose de deux étapes de cartes de caractéristiques.

— **Base de données :** PETS-2009 et Subway sont tirés d'ensembles de données d'autres travaux et Chunxi-Road est une vraie vidéo tirée de la vraie vie. Comme l'environnement de Chunxi-Road est un peu compliqué avec des panneaux d'affichage et des bâtiments, et les zones intéressées sont découpées et façonnées Base de données.

1.10 La reconnaissance faciale

La reconnaissance faciale est une technique permettant d'authentifier ou d'identifier une personne à partir de ses traits faciaux. Elle se déroule en deux phases : la création d'un mo-

dèle représentant les caractéristiques du visage à partir de l'image, et la comparaison de ce modèle avec ceux enregistrés préalablement. Dans le cas de l'authentification, le système vérifie si l'identité prétendue correspond au modèle enregistré, tandis que dans l'identification, il recherche une correspondance parmi les modèles de la base de données.

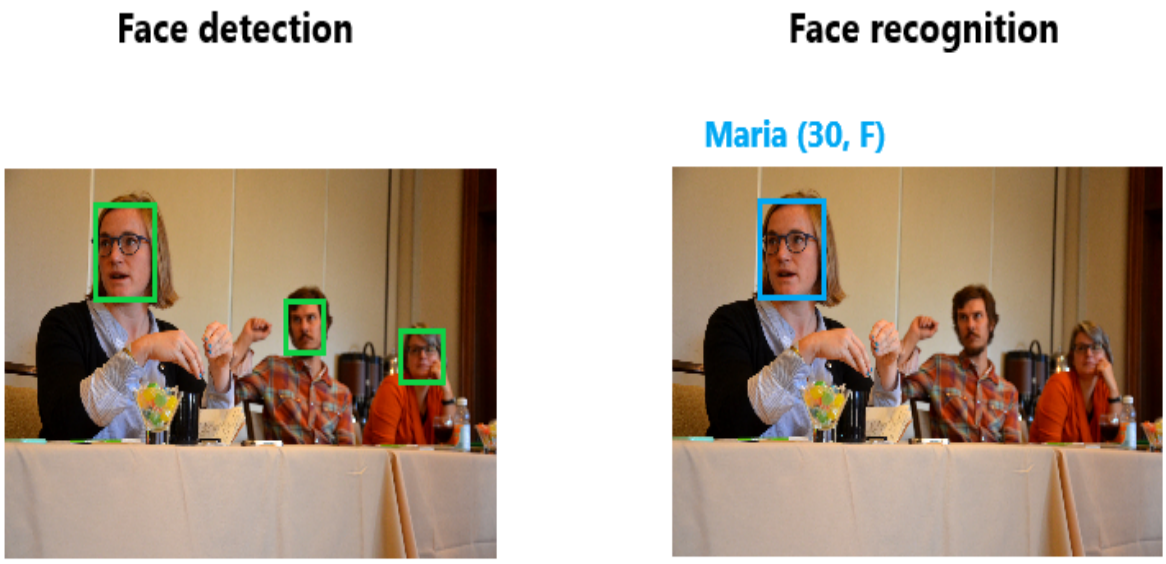

FIGURE 1.9 – Détection et reconnaissance facial

Il existe plusieurs méthodes pour face-id (classique et moderne) qui passent par différentes phases comme on le voit sur la (figure 1.10)

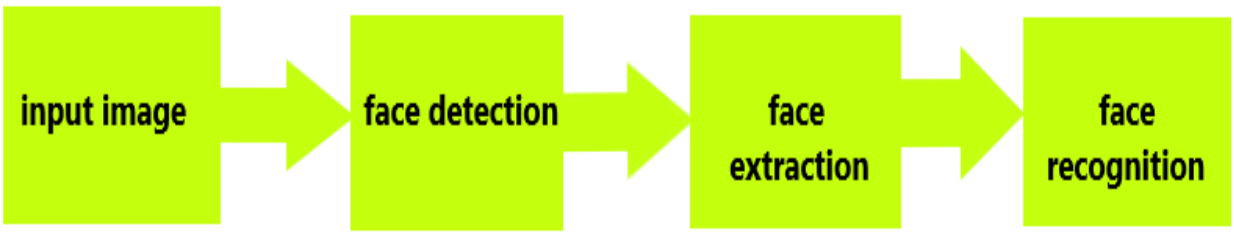

FIGURE 1.10 – Les étapes de reconnaissance faciale

1.10.1 Domaines d'application de la reconnaissance faciale

Sécurité publique : La reconnaissance faciale est utilisée pour la détection et l'identification d'individus suspects ou recherchés dans les lieux publics tels que les aéroports, les gares, les centres commerciaux, etc. Elle permet aux autorités de repérer rapidement les individus présentant un risque pour la sécurité publique.

Gestion des foules : La reconnaissance faciale est utilisée pour gérer les foules lors d'événements publics ou dans des lieux à forte affluence tels que les stades, les concerts ou les festivals. Elle permet de compter et de suivre les individus présents, facilitant ainsi la planification des mesures de sécurité, la gestion du flux de personnes et la prévention des situations de surpopulation.

Contrôle d'accès : La reconnaissance faciale est utilisée pour contrôler l'accès à des zones restreintes ou sécurisées. Les systèmes de reconnaissance faciale permettent de comparer les visages des individus avec une base de données d'utilisateurs autorisés, permettant ainsi de faciliter l'entrée des personnes autorisées et de détecter les tentatives d'intrusion ou d'accès non autorisées.

Prévention de la criminalité : La reconnaissance faciale est un outil précieux pour prévenir la criminalité. Elle peut être utilisée pour identifier et suivre les individus suspects dans les zones de haute criminalité. Les systèmes de reconnaissance faciale peuvent alerter les forces de l'ordre lorsqu'une correspondance est trouvée avec des individus recherchés, contribuant ainsi à la prévention des délits et à l'amélioration de la sécurité générale.

Identification des personnes disparues : La reconnaissance faciale peut être utilisée pour aider à identifier les personnes disparues.

1.10.2 Les travaux déjà réalisés sur la reconnaissance faciale

Il y a eu de nombreux travaux réalisés dans le domaine de la reconnaissance faciale. Voici une liste des principales avancées et techniques utilisées :

1.10.2.1 Propres visages (Eigenfaces)

Les "Eigenfaces" sont des composants utilisés dans la reconnaissance faciale. Ils sont obtenus en convertissant un lot d'images en vecteurs de caractéristiques. Lors de la reconnaissance faciale, les images sont projetées sur les "Eigenfaces" pour déterminer leur identité. Cette méthode utilise l'analyse en composantes principales (PCA) pour la transformation spatiale. Elle regroupe les images similaires et les sépare des images différentes de manière efficace.[14]

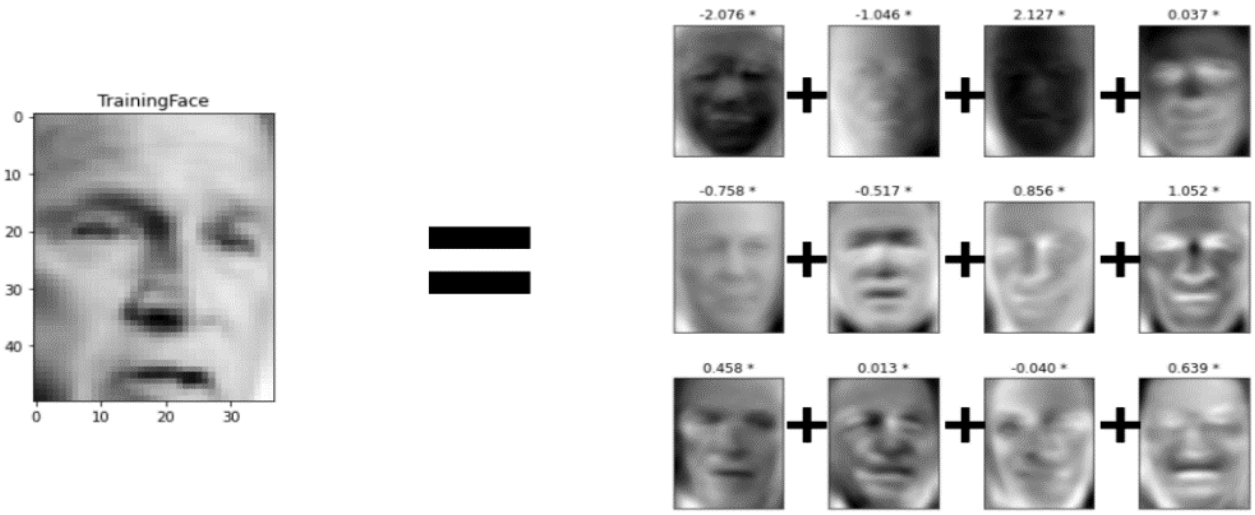

FIGURE 1.11 – reconnaissance facial utilisée Eigenfaces[15]

1.10.2.2 Les motifs binaires locaux (Local Binary Patterns)

Le modèle binaire local (LBP) est un opérateur de texture simple et efficace utilisé pour étiqueter les pixels d'une image en évaluant leur voisinage à l'aide d'un seuil. Il a été introduit en 1994 et est largement utilisé dans la classification de textures. Lorsqu'il est combiné avec des histogrammes de descripteurs de gradient (HOG), il améliore significativement les performances de reconnaissance sur certains jeux de données. L'utilisation de LBP avec un histogramme permet de représenter une image de visage sous la forme d'un vecteur de données. Le LBP peut également être utilisé pour la reconnaissance faciale en tant que descripteur visuel[16].

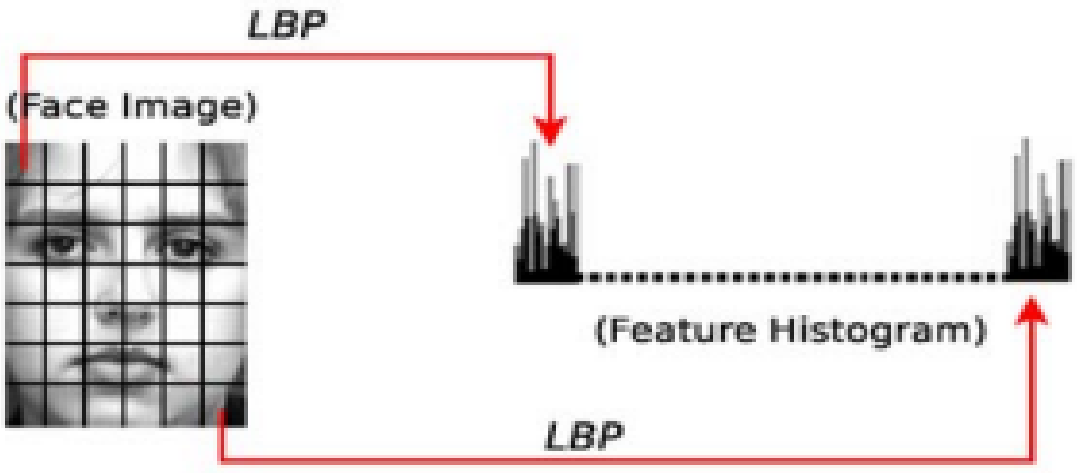

FIGURE 1.12 – Local Binary Patterns (LBP)[17]

1.10.2.3 Viola-Jones (Algorithme Haar cascade)

L'algorithme Viola-Jones, nommé d'après ses créateurs Paul Viola et Michael Jones, est utilisé pour la détection d'objets en temps réel, notamment la détection de visages. Malgré sa relative ancienneté, il reste puissant et rapide. L'algorithme fonctionne en analysant de petites sections de l'image à la recherche d'attributs spécifiques des visages. Il utilise des caractéristiques de type Haar et suit quatre principales étapes : sélection des caractéristiques, création d'une image intégrale, entraînement avec AdaBoost et création de cascades de classifieurs. Ces étapes permettent une détection efficace des visages dans une image donnée.

1.11 Conclusion

En conclusion, ce chapitre a introduit la vidéo surveillance en définissant ses principes de base et en explorant son évolution historique. Nous avons examiné les différents domaines d'application de la vidéo surveillance et discuté de son architecture. Ensuite, nous avons abordé la problématique liée à ce domaine, suivi d'une analyse du comptage de foule et de la reconnaissance faciale des personnes suspects, mettant en évidence les travaux de recherche existants pour donner une idée générale.
Dans le prochain chapitre nous nous concentrerons sur le domaine de l'IA ,ML ,DL, en mettant en évidence les applications du ML et du deep learning en la vidéo surveillance.

Chapitre 2

APPLICATION DE L'INTELLIGENCE ARTIFICIELLE POUR LA VIDEOSURVEILLANCE

2.1 Introduction

L'IA et la vidéosurveillance sont deux domaines interconnectés qui ont suscité une attention et des progrès considérables ces dernières années.

L'IA fait référence à la simulation de l'intelligence humaine dans des machines programmées, DL et ML sont deux sous-domaines de l'IA qui se concentrent sur la formation d'algorithmes pour apprendre à partir de données et faire des prédictions ou des décisions, et la vidéosurveillance implique l'utilisation des caméras et des systèmes de surveillance pour capturer et analyser des informations visuelles.

Le DL et le ML sont largement utilisés dans la vidéosurveillance pour automatiser diverses tâches, telles que la détection d'objets, le suivi et la reconnaissance de comportement.

Dans ce chapitre, nous donnerons un aperçu de l'IA ensuit nous parlerons de ML et DL et quelle est la différence entre eux,et à la fin nous allons voire les applications du ML et DL dans la vidéosurveillance.

2.2 C'est quoi l'intelligence artificielle

L'intelligence artificielle est la simulation des processus de l'intelligence humaine par des machines et pour simuler l'intelligence humaine l'idée était de reproduire les fonctions cognitives. En même temps que l'IA sont nées les sciences cognitives qui étudient l'intelligence humaine, elles ont permis d'identifier les différents mécanismes qui sont à l'œuvre dans le fonctionnement de la pensée.

Cette pensée qui nous permet de percevoir, de mémoriser, de parler, de raisonner, de planifier... L'IA est composée de données d'algorithmes, de matériel et de connectivité et d'ensemble des techniques et théories mise en œuvre en vue de réaliser des machines capables de simuler l'intelligence[18].

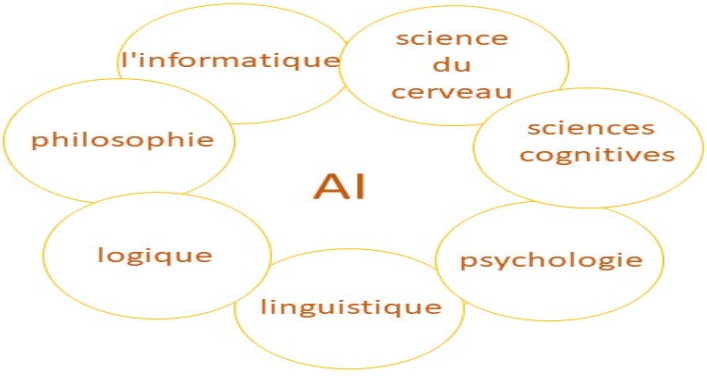

FIGURE 2.1 – l'intelligence Artificielle

2.3 Historique de l'IA

En 1956, lors d'une conférence, les scientifiques américains ont utilisé le terme "IA" pour la première fois, décrivant les mécanismes d'apprentissage et d'intelligence humaine à reproduire dans les machines.

Dans les années 1960, l'IA a connu un déclin. Les années 1970-1990 ont vu le développement des systèmes experts et des recherches sur le traitement du langage naturel et la vision par ordinateur.

Les années 2000 ont été marquées par le regain d'intérêt grâce au big data et au deep learning. Depuis 2010, l'IA est omniprésente, utilisée dans des domaines variés tels que la reconnaissance vocale, l'analyse d'images, les véhicules autonomes et le diagnostic médical.

2.4 Les domaines de l'IA

- **Santé :** L'IA dans les soins de santé se manifeste de plusieurs manières, telles que la recherche de nouveaux liens entre les codes génétiques, l'alimentation de robots d'assistance chirurgicale, l'automatisation des tâches administratives, la personnalisation des options de traitement et bien plus encore.
- **Automobile :** L'IA transforme rapidement l'industrie automobile, rendant la conduite plus sûre, plus efficace et plus agréable pour les consommateurs.
- **Commerce :** finance (trading algorithmique) : l'utilisation de système d'IA complexe pour prendre des décisions commerciales, et détection de tendance...
- **Education :** L'IA a le potentiel de transformer le secteur de l'éducation en offrant aux étudiants des expériences d'apprentissage plus personnalisées, efficientes et efficaces comme les systèmes de tutorat intelligents, la notation et l'évaluation automatisées...
- **Industrie :** Maintenance prédictive.
- **Environnement :** L'eau, l'agriculture, la biodiversité et le changement climatique
- **Informatique :** L'IA est un domaine en croissance rapide de l'informatique qui a de nombreuses applications dans divers domaines tels que la vision par ordinateur, l'informatique cognitive, la robotique etc. L'IA permet aux machines d'apprendre à partir des données, de comprendre le langage naturel, d'interpréter les informations visuelles, d'interagir avec les humains et de raisonner comme les humains.

2.5 Machine Learning (Apprentissage automatique)

L'apprentissage automatique (ML) est un sous-ensemble de l'IA qui se concentre sur le développement d'algorithmes et de modèles statistiques permettant aux ordinateurs d'apprendre à partir de données sans être explicitement programmés, d'une autre façon , les algorithmes de ML peuvent automatiquement améliorer leurs performances sur une tâche en apprenant à partir d'exemples ou d'expériences passées, Le ML est une méthode de modélisation des phénomènes pour prendre des décisions stratégiques[19].

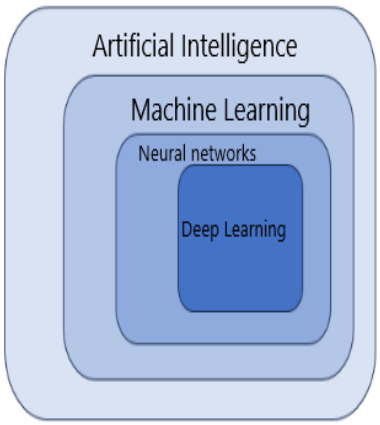

FIGURE 2.2 – la relation entre AI et ML et ANN et DL

— **Les types de ML**

Il existe trois types de ML :

• Apprentissage supervisé

• Apprentissage non supervisé

• Apprentissage forcé (reinforcement learning)

2.6 Deep learning (Apprentissage profond)

L apprentissage profond (DL) est un sous-domaine de ML qui implique la construction et la formation de réseaux de neurones artificiels à plusieurs couches. Dans le ML traditionnel, l'algorithme est fourni avec des données d'entrée et un ensemble de fonctionnalités qu'il utilise pour effectuer des prédictions ou des classifications. Cependant, dans le DL, l'algorithme apprend ces fonctionnalités par lui-même en faisant passer les données d'entrée à travers plusieurs couches de transformations non linéaires.

Au début des années 2000, une percée a été faite lorsque les chercheurs ont développé une technique appelée Convolutional Neural Networks (CNN), qui a révolutionné la reconnaissance d'images. Cette technique a permis le traitement efficace de grandes quantités de données d'image et a contribué à stimuler le développement de DL[20].

L'essor des méga données (big data) et le développement de technologies informatiques puissantes telles que les unités de traitement graphique (GPU) et le cloud computing ont également contribué aux progrès rapides de DL ces dernières années. Depuis lors, DL a été appliqué à un large éventail d'applications, notamment la vision par ordinateur, le traitement du langage naturel et la reconnaissance vocale. Aujourd'hui, le DL est l'un des domaines de recherche les plus actifs en intelligence artificielle et a le potentiel de transformer de nombreux secteurs et aspects de nos vies.

— **Pourquoi le Deep Learning ?**

• **Capacité à apprendre des représentations complexes :** les algorithmes de DL sont capables d'apprendre des représentations complexes de données, qui peuvent être utilisées pour une variété de tâches.

• **Extraction automatique des caractéristiques :** les algorithmes de DL sont capables d'extraire automatiquement les caractéristiques pertinentes des données sans avoir besoin d'une ingénierie manuelle des caractéristiques.

• **Meilleures performances :** il a été démontré que les algorithmes de DL surpassent les techniques traditionnelles de ML sur de nombreux ensembles de données de référence. Par exemple, des tâches de reconnaissance d'images telles que le défi ImageNet.

• **Évolutivité :** les algorithmes de DL peuvent être adaptés à de grands ensembles de données et à des systèmes distribués, ce qui permet des temps de formation plus rapides et de meilleures performances. Ceci est particulièrement important pour des applications telles que l'analyse de méga données et le traitement de données en temps réel.

• **Flexibilité :** les algorithmes d'apprentissage en profondeur peuvent être appliqués à un large éventail d'applications.

2.7 Mchine learning VS Deep learning

— **Deep learning :**

Un sous-ensemble de ML.

Il nécessite de grandes quantités de données.

A une grande précision et il a besoin d'un GPU spécialisé pour l'entrainement.

Il apprend par lui-même.

— **Machine learning :**

Un sous-ensemble d'IA.

Peut s'entraîner sur des ensembles de données plus petits.

Entraîneur plus court et moins de précision.

Nécessite plus d'intervention humaine pour corriger et apprendre.

Peut s'entraîner sur un CPU.

Les algorithmes de DL peuvent découvrir automatiquement des caractéristiques à partir de données brutes mais en ML, l'extraction de caractéristiques est généralement effectuée manuellement comme illustré dans la (Figure 2.3).

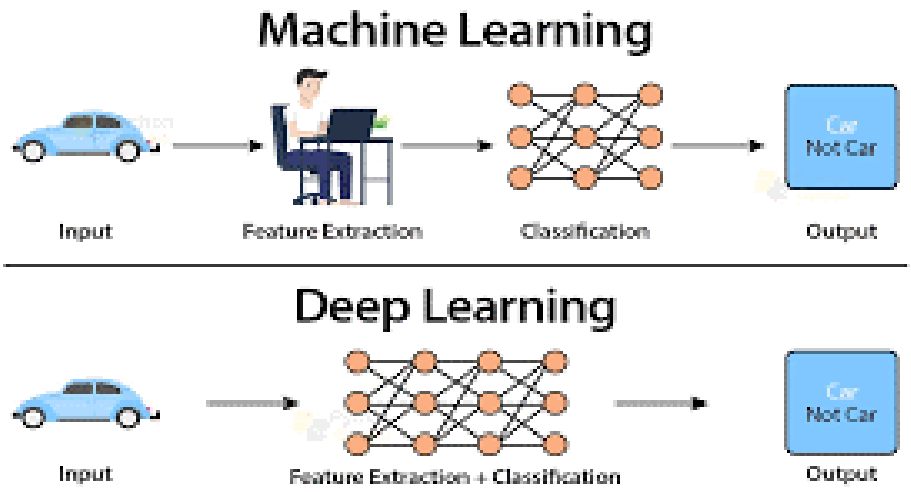

FIGURE 2.3 – ML vs DL [21]

2.8 Topologies des réseaux de neurones

La topologie d'un réseau de neurones fait référence à la structure ou à l'architecture du réseau, y compris le nombre de couches, le nombre de neurones dans chaque couche et les connexions entre les neurones.

Les réseaux de neurones peuvent avoir une variété de topologies en fonction de leur objectif et du problème qu'ils tentent de résoudre.

2.8.1 Perceptron

Le perceptron est un type de réseau neuronal artificiel développé dans les années 1950 et 1960. Il s'agit d'un simple réseau neuronal linéaire (monocouche) prédictif composé d'une seule couche de nœuds d'entrée et d'un seul nœud de sortie. Les nœuds d'entrée reçoivent des entrées, et chaque entrée est multipliée par un poids, puis additionnée. La somme est ensuite passée

à travers une fonction d'activation pour produire la sortie. Les poids sont ajustés pendant la formation pour minimiser l'erreur entre la sortie et la sortie souhaitée.

L'algorithme perceptron est un algorithme d'apprentissage supervisé utilisé pour les problèmes de classification binaire. Il peut être utilisé pour apprendre une frontière de décision qui sépare deux classes de données[22].

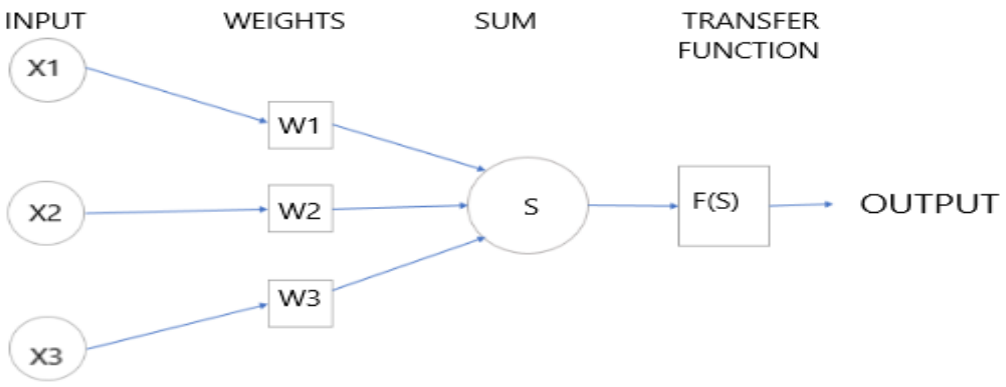

FIGURE 2.4 – Réseau de neurone simple

Une itération de l'apprentissage du perceptron est réalisée en deux phases :

La phase de sommation : consiste à calculer la somme des produits des entrées avec leur poids correspondant plus un certain biais qui représente le décalage sur l'axe désordonnée de la droite de classification pour avoir une meilleure classification :

$$S = \sum_{0}^{n} wi * xi \tag{2.1}$$

La phase d'activation : ou fonction de transfert est une fonction seuil :

$$Y = \begin{cases} 1 \quad si \quad \sum_0^n wi * xi > 0 \\ 0 \quad si \quad \sum_0^n wi * xi \leq 0 \end{cases} \tag{2.2}$$

2.8.2 Perceptron multicouche

Un perceptron multicouche (MLP) est un type de réseau neuronal artificiel qui contient plusieurs couches de perceptrons. Les MLP sont parfois appelés réseaux de neurones à anticipation car les signaux d'entrée sont traités à travers le réseau dans le sens direct, de la couche d'entrée à travers une ou plusieurs couches cachées jusqu'à la couche de sortie. Chaque couche

de perceptrons dans un MLP est connectée à la couche suivante, et les connexions entre les couches sont pondérées. Les MLP sont capables d'effectuer des mappages non linéaires complexes entre les données d'entrée et de sortie. Ils sont souvent utilisés pour des tâches telles que la reconnaissance de formes, la classification et la prédiction[23].

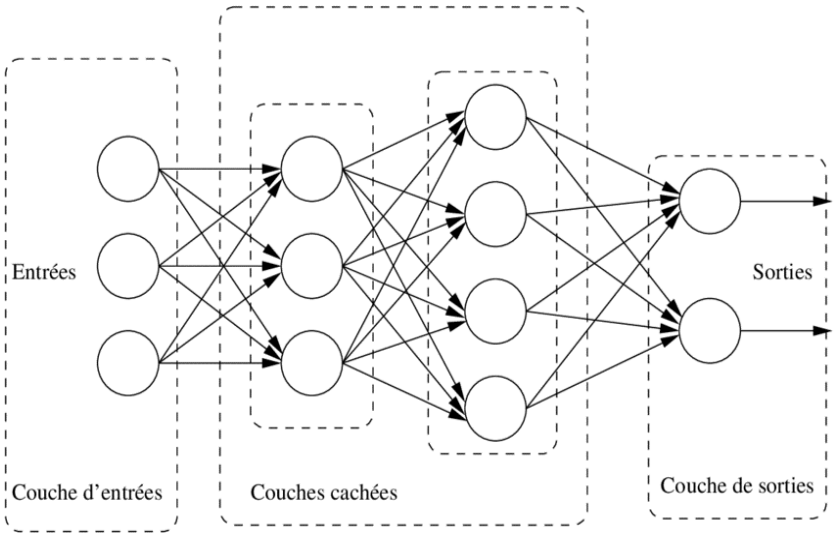

FIGURE 2.5 – perceptron multicouche[24]

Les MLP peuvent être formés à l'aide de divers algorithmes d'apprentissage supervisé, tels que la rétro-propagation (Backpropagation), qui ajuste les poids et les biais dans le réseau pour minimiser la différence entre la sortie prévue et la sortie réelle[24].

2.9 Les couches d'un réseau de neurone

il existe plusieurs couches d'un réseau neuronal :

• **Couche d'entrée :** est la première couche d'un ANN qui reçoit les données d'entrée. Chaque nœud de cette couche représente une caractéristique ou une variable d'entrée.

• **Couche cachée :** une ou plusieurs couches cachées qui traitent les données d'entrée lors de leur transmission sur le réseau, les couches masquées sont les couches intermédiaires entre les couches d'entrée et de sortie. Ces couches effectuent des calculs sur les données d'entrée à l'aide d'un ensemble de pondérations et de biais pour produire une sortie.

• **Couche de sortie :** La couche de sortie est la dernière couche d'un ANN qui produit la sortie finale du réseau et qui fournit les prédictions du réseau sur l'entrée donnée, Le nombre de nœuds dans la couche de sortie dépend du type de problème à résoudre[25].

2.10 Les fonctions d'activation

Une fonction d'activation dans un réseau neuronal artificiel détermine la sortie d'un neurone pour une entrée donnée. Il introduit la non-linéarité dans le réseau, lui permettant d'apprendre des relations complexes entre les entrées et les sorties.

- **Fonction ReLU (the rectified linear unit)**

Le choix le plus populaire, en raison à la fois de la simplicité de mise en œuvre et de ses bonnes performances sur une variété de tâches prédictives :

$$ReLu(x) = \max(x, 0) \tag{2.3}$$

La fonction ReLU ne retient que les éléments positifs et supprime tous les éléments négatifs en mettant les activations correspondantes à 0, Lorsque l'entrée est négative, la dérivée de la fonction ReLU est 0, et lorsque l'entrée est positif, la dérivée de la fonction ReLU est 1.

- **Fonction Sigmoïde**

La fonction sigmoïde transforme ses entrées, dont les valeurs se situent dans le domaine R, en sorties qui se trouvent sur l'intervalle (0, 1). Pour cette raison, le sigmoïde est souvent appelé une fonction d'écrasement .

Il écrase toute entrée dans la plage (-inf, inf) à une valeur dans la plage (0, 1) :

$$Sigmoid(x) = \frac{1}{1 + \exp(-x)} \tag{2.4}$$

- **Fonction Softmax**

La fonction Softmax est utilisée dans la couche de sortie d'un réseau neuronal utilisé pour la classification multiclasse. Il prend en entrée un vecteur de valeurs et renvoie une distribution de probabilité sur les classes. La formule de Softmax est :

$$Softmax(x) = \frac{\exp(xi)}{\sum \exp(x)} \tag{2.5}$$

- **Fonction Tanh**

Comme la fonction sigmoïde, la fonction tanh (tangente hyperbolique) écrase également ses entrées, en les transformant en éléments sur l'intervalle entre -1 et 1 [26] :

$$tanh(x) = \frac{1 - \exp(-2x)}{1 + \exp(-2x)} \tag{2.6}$$

2.11 CNN (convolutional Neural Network)

Réseau Convolutif, également appelé Réseau de Neurone Convolutif ou CNN, est un type de réseau neuronal profond spécialisé pour les tâches de vision par ordinateur et la reconnaissance d'images(pour traiter les données de pixels), il peut également être utilisé pour d'autres problèmes d'analyse de données ou de classification.

Le nom « neural convolutif réseau » indique que le réseau emploie une opération mathématique appelée convolution qu'elle est un type spécialisé d'opération linéaire. Il est particulièrement efficace dans le traitement des images car il est conçu pour détecter et apprendre automatiquement les caractéristiques visuelles à partir des données de pixels brutes[27].

2.11.1 Les couches de CNN

Le CNN se compose de plusieurs couches, chacune avec sa propre fonction spécifique :

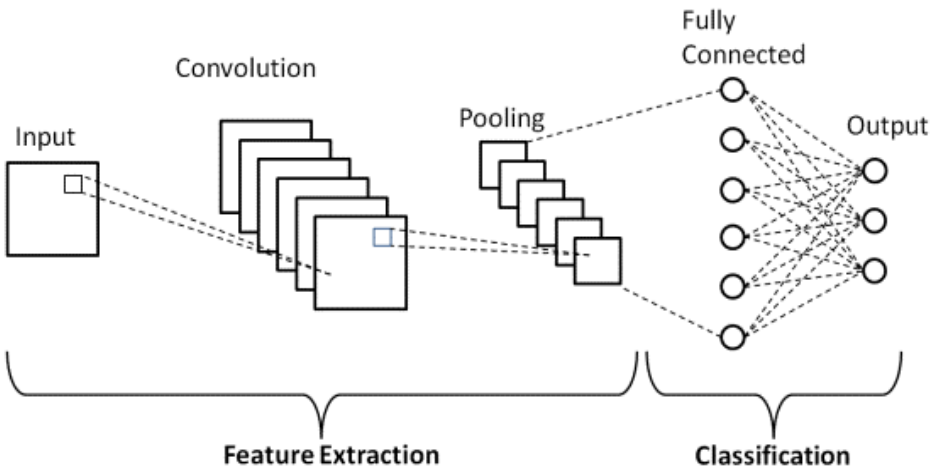

FIGURE 2.6 – Représentation des couches de CNN

• **Couche Convolutive :**

un CNN a des couches cachées appelées couches convolutives et ces couches sont précisément ce qui fait qu'un CNN est bien un CNN, tout comme n'importe quelle autre couche, une couche convolutives reçoit une entrée puis transforme l'entrée d'une manière ou d'une autre, et cette transformation est une opération de convolution.

Les couches convolutives effectuent des opérations de convolution pour extraire des caractéristiques de l'entrée. Plus précisément, avec chaque couche convolutive, nous devons spécifier le nombre de filtres (kernel) que les couches doivent avoir et ces filtres sont en fait ce qui détecte les caractéristiques[28].

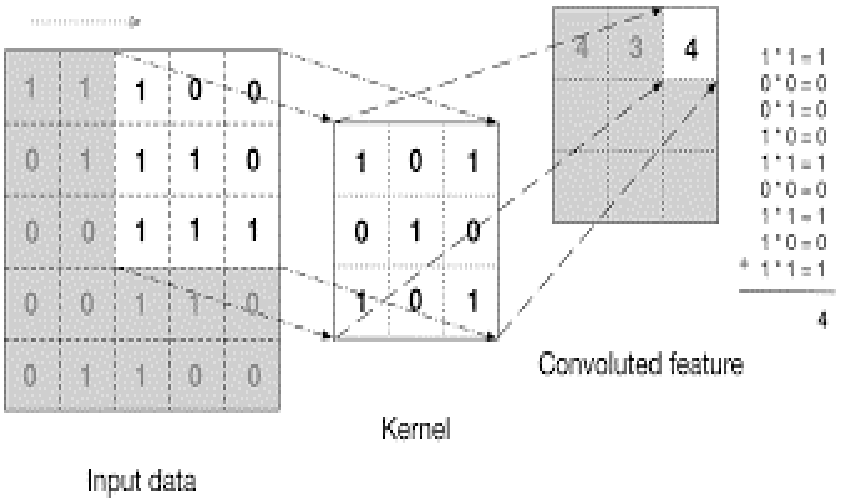

FIGURE 2.7 – Exemple de la couche de convolution dans CNN [26]

Un filtre (kernel) peut techniquement être simplement considéré comme une matrice relativement petite, les nombres de lignes et de colonnes (la taille) de cette matrice doivent être choisis.

Le filtre glissera sur la largeur et la hauteur du volume d'entrée, et les produits scalaires entre les entrées du filtre et l'entrée à n'importe quelle position sont calculés, pour produire des valeurs uniques dans la carte des caractéristiques de sortie (feature map) voir la figure précédente (Figure 2.7). La couche convolutive est également connectée à la couche relu (fonction d'activation) qui introduit la non-linéarité dans le réseau, lui permettant d'apprendre des fonctionnalités et des modèles plus complexes.

• Couche de Pooling (Regroupement) :

Pooling est un type d'opération qui est généralement ajouté aux couches convolutives individuelles suivantes de CNN, pooling est ajouté après une couche convolutive il réduit la dimensionnalité des images en réduisant le nombre de pixels dans la sortie de la couche convolutive précédente comme illustré dans la figure suivante(Figure 2.8).

Avant l'opération de pooling, il doit définir une région N×N comme filtre correspondant pour l'opération de pooling et puis il devrait définir une Stride signifiant par combien de pixels voulons-nous que notre filtre se déplace lorsqu'il glisse sur l'image[29].

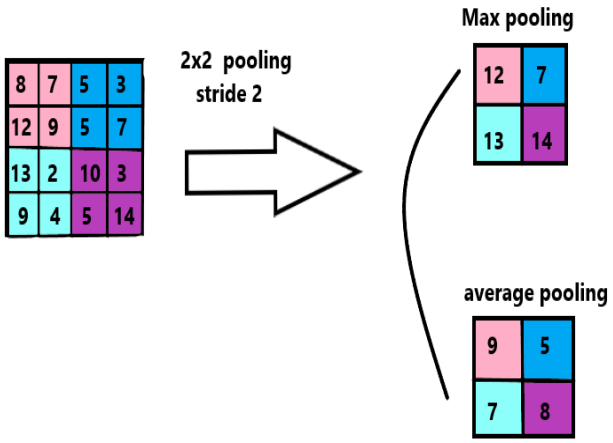

FIGURE 2.8 – Exemple sur Max Pooling et Average pooling

L'opération de pooling la plus courante est le Max Pooling :
Pour une image particulière notre réseau cherchera à extraire certaines caractéristiques particulières, des bords... à partir de la sortie de la couche convolutive, nous pouvons considérer les pixels de valeur supérieure comme étant ceux qui sont les plus activés, tandis que dans Average pooling, la valeur moyenne de chaque région est utilisée.
La couche de Pooling contribue à améliorer l'efficacité des CNN pour les tâches de reconnaissance d'images en réduisant les dimensions spatiales de la carte des caractéristiques d'entrée et en conservant les informations les plus importantes.

• **Fully Connected Layer (Couche entièrement connectée) :**
FCLs Sont les couches où chaque nœud est connecté au nœud suivant.
Après l'extraction des caractéristiques de l'image avec des couches convolutives et des couches de max pooling , ces caractéristiques seront transmises à FCL en aplatissant (flatten) ces caractéristiques à 1 vecteur et elles seront utilisées comme entrée pour FCL et puis chaque neurone sera connecté avec le neurone suivant[30].

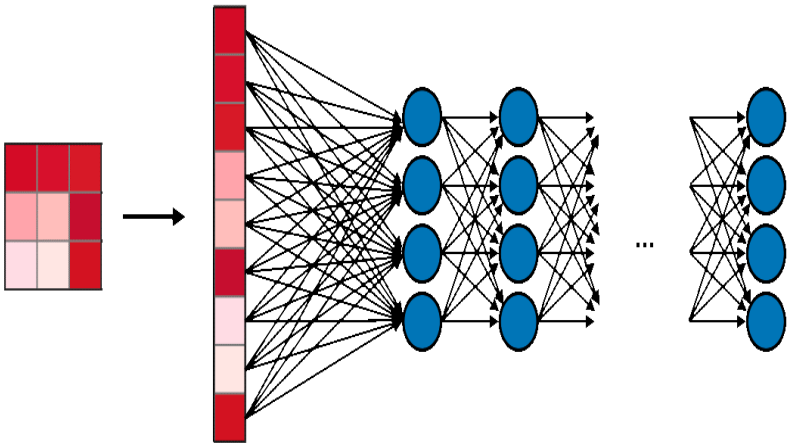

FIGURE 2.9 – Représentation de Fully Connected Layer (FCL)[31]

Le but de FCL est de classer l'image dans des catégories particulières et d'associer des caractéristiques à des étiquettes particulières.

2.12 la propagation avant et la rétropropagation

Feedforward et rétropropagation sont des concepts fondamentaux dans l'apprentissage profond, en particulier dans la formation et l'utilisation de réseaux de neurones profonds.

• **propagation avant**

propage les données d'entrée à travers les différentes couches du réseau de neurones, en appliquant des calculs pondérés et des fonctions d'activation pour générer des sorties. Ce processus se fait de manière unidirectionnelle, de l'entrée vers la sortie. Il est utilisé pour effectuer des prédictions ou des classifications en exploitant les informations apprises par le réseau pendant la phase d'apprentissage[26].

• **Rétropropagation**

est un algorithme utilisé pour former des réseaux de neurones profonds en ajustant les poids et les biais pour minimiser l'erreur ou la perte entre les prédictions du réseau et la sortie réelle souhaitée. Cela fonctionne en propageant l'erreur vers l'arrière à travers le réseau, de la couche de sortie à la couche d'entrée. L'erreur est utilisée pour mettre à jour les poids et les biais dans chaque couche à l'aide de l'optimisation de descente de gradient[26].

• **La descente de gradient**

un algorithme d'optimisation utilisé dans le processus d'apprentissage des réseaux de neurones et dans d'autres modèles de ML et DL. Son objectif est de trouver les valeurs optimales des paramètres (poids et biais) du modèle afin de minimiser une fonction de coût ou une erreur[32].

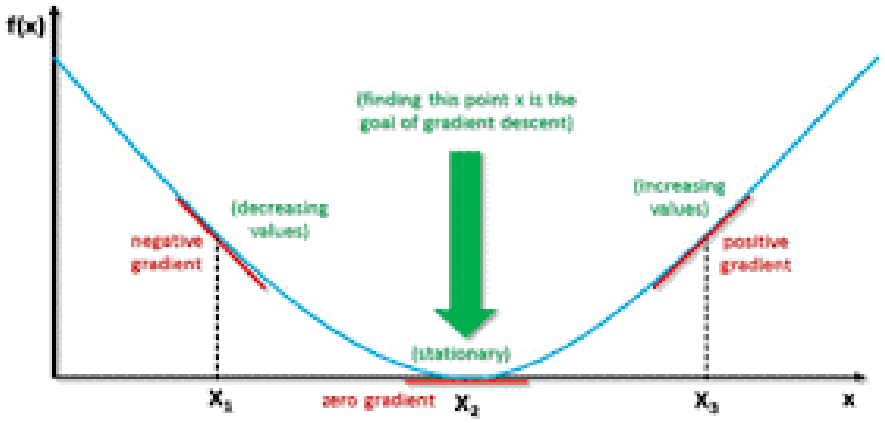

FIGURE 2.10 – La descente de gradient[32]

2.13 Fonction de perte (Loss)

Dans un CNN, une fonction de perte (Loss) est généralement utilisée pour mesurer la différence entre la sortie prédite et la sortie réelle. L'objectif du réseau est de minimiser cette différence[33].

• **Mean Squared Error (MSE)**

cette fonction est utilisée dans les problèmes de régression, elle calcule la différence quadratique moyenne entre la sortie prévue et la sortie actuelle.

• **Binary Cross Entropy**

cette fonction de perte est utilisée dans les problèmes de classification binaire, elle mesure la différence entre la probabilité prédite et la probabilité actuelle.

• **Categorical Cross Entropy**

cette fonction de perte est utilisée dans les problèmes de classification multi-classes, elle mesure la différence entre la distribution de probabilité prédite et la distribution de probabilité actuelle.

• **Hinge Loss**

Cette fonction de perte est généralement utilisée pour les problèmes de classification avec un grand nombre de classes possibles. elle mesure la différence entre le score prédit et le score réel, où le score est la sortie du réseau avant de passer par une fonction softmax.

Le choix de la fonction de perte dépend du problème spécifique et du type de sortie qui est prédit.

2.14 Les applications du ML et DL dans la vidéosurveillance

Le DL et le ML ont apporté de nombreuses avancées dans le domaine de la vidéosurveillance Voici quelques-unes des applications les plus courantes :

2.14.1 Détection d'objets

Les algorithmes de DL peuvent être utilisés pour détecter et suivre automatiquement des objets dans des vidéos de surveillance en temps réel. Cela aide, par exemple, à détecter et même à comptabiliser les individus qui composent une foule et peut même surveiller leur comportement ou même identifier ou détecter des objets non autorisés, par exemple, la détection d'armes.

FIGURE 2.11 – Détection d'armes pour la sécurité et la vidéosurveillance[34]

2.14.2 Reconnaissance faciale

Le DL permet d'effectuer une reconnaissance faciale précise et rapide, même dans des conditions de faible luminosité ou avec des angles de caméra différents, par des techniques de CNN, Réseaux de neurones siamois, ou réseaux de neurones LSTM (long short term Memory) ou bien avec des réseaux de neurones pré-entraînés etc.
Cela peut être utilisé pour identifier les individus, rechercher des personnes disparues ou suspectées.

2.14.3 Reconnaissance de plaques d'immatriculation

Les techniques de DL peuvent être appliquées pour reconnaître automatiquement les plaques d'immatriculation des véhicules dans les vidéos de surveillance. Cela permet de surveiller les mouvements des véhicules, de contrôler les accès ou de rechercher des véhicules impliqués dans

des activités criminelles. Les principales étapes du processus de reconnaissance de plaques d'immatriculation utilisant le DL sont : la détection de plaques d'immatriculation les (CNN) sont couramment utilisés pour cette tâche, notamment des architectures telles que YOLO ou Faster R-CNN, et puis segmentation de caractères et la reconnaissance de caractères.

FIGURE 2.12 – Exemple de détection plaques d'immatriculation

2.14.4 Analyse du comportement

Les techniques de DL peuvent être utilisées pour analyser les schémas de comportement dans les vidéos de surveillance. Par exemple, la détection d'anomalies permet d'identifier des comportements inhabituels, tels que des mouvements erratiques ou des interactions agressives. Une des principales forces du DL est sa capacité à extraire des caractéristiques de haut niveau à partir de données brutes, les réseaux de neurones profonds les CNN, RNN (Réseau de Neurones Récurrent), les réseaux de neurones à attention sont entraînés sur de grandes quantités de données étiquetées, ce qui leur permet d'apprendre des modèles complexes et des représentations abstraites du comportement humain. Nous prenons comme un exemple l'analyse du comportement en classe basée sur DL [35].

2.15 Conclusion

Ce chapitre a introduit les bases de l'IA appliquée à la vidéosurveillance Nous avons également examiné le ML et le DL, en soulignant leurs différences. De plus, nous avons étudié les réseaux de neurones artificiels, en mettant l'accent sur les couches et les architectures, notamment les réseaux convolutionnels (CNN). Enfin, nous avons abordé les applications spécifiques du ML et du DL dans le contexte de la vidéosurveillance Dans le prochain chapitre, nous

aborderons les modèles de la détection des objets et la reconnaissance faciale par vision par ordinateur et surtout nous nous concentrons sur le modèle yolov3 et FaceNet que nous avons utilisés dans notre projet.

Chapitre 3

LA DETECTION D'OBJET ET LA RECONNAISSANCE FACIALE PAR VISION ORDINATEUR

3.1 Introduction

La vision par ordinateur est un domaine de l'IA qui forme les ordinateurs à interpréter et à comprendre le monde visuel. À l'aide d'images numériques provenant de caméras et de vidéos et de modèles de DL, les machines peuvent identifier et classer avec précision les objets, ce qu'il signifie qu'ill existe de nombreux types de vision par ordinateur qui sont utilisés de différentes manières, nous sommes intéressés par la détection d'objet qui identifie les individus dans une vidéo et puis on les compte. Et La reconnaissance faciale pour identifier les personnes suspectes. Dans ce chapitre nous allons voire c'est quoi la détection d'objets en générale ensuite quelques modèles de détection d'objets par DL comme SSD le RCNN et surtout le modèle Yolov3, et à la fin nous allons voir aussi quelques modèles de reconnaissance faciale comme le modèle FaceNet et puis une conclusion.

3.2 La détection d'objets

La détection d'objets est une tâche de vision par ordinateur qui consiste à identifier et à localiser des objets dans une image ou une vidéo. L'objectif est de détecter et de classer plusieurs objets d'intérêt en temps réel ou quasi réel.
Elle a été déterminée par des nombreuses applications de la vision par ordinateur telles que le suivi d'objets, la vidéosurveillance, le sous-titrage d'images, la segmentation d'images, imagerie médicale et plusieurs autres applications. Les algorithmes de détection d'objets suivent généralement un processus en deux étapes :

• **Génération de proposition de région :** également appelées Bbox, sont générées.

• **Classification d'objet :** les propositions de région générées sont introduites dans un classifieur pour déterminer les étiquettes de classe des objets dans les régions proposées. La sortie finale de la détection d'objet se compose des objets détectés avec leurs étiquettes de classe correspondantes et les coordonnées de Bbox.[36]

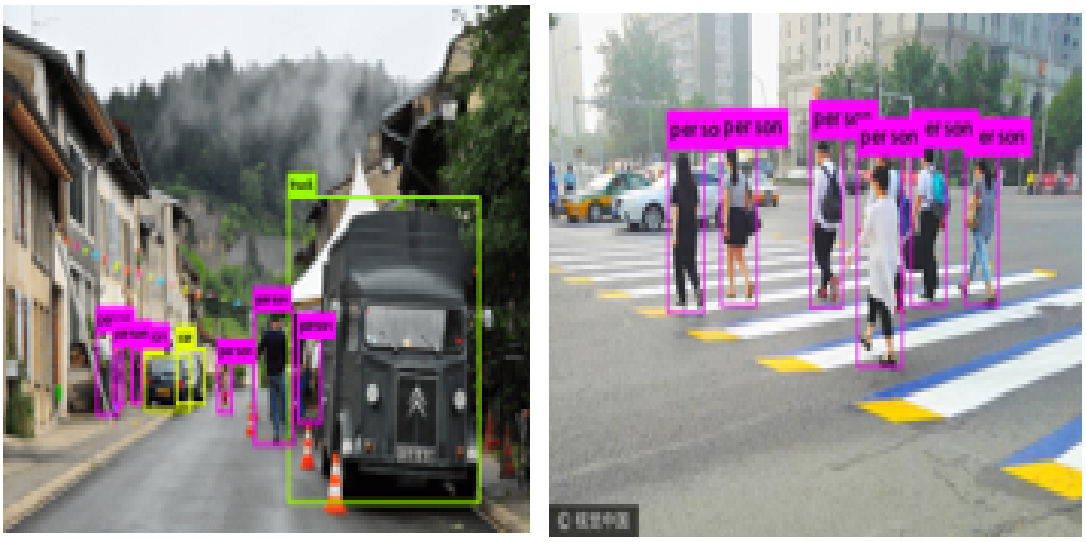

FIGURE 3.1 – une sortie finale avec un des modèles de détection d'objet[37]

3.3 Les modèles de détection

3.3.1 Le modèle R-CNN (Region-basedConvolutional Neural Network)

Introduit en 2014, R-CNN était la première itération du framework. Il effectue la détection d'objets en proposant des régions dans l'image à l'aide d'une recherche sélective, puis en appliquant un CNN à chaque région proposée pour l'extraction de caractéristiques. Ces caractéristiques sont introduites dans des classifieurs distincts pour la classification des objets et la régression de la Bbox[38].

3.3.1.1 Fast R-CNN

une extension de R-CNN, Fast R-CNN répond à certaines des limites du modèle original. Au lieu de traiter chaque région proposée séparément, Fast R-CNN introduit la couche de regroupement RoI, qui permet d'effectuer l'extraction de caractéristiques sur l'ensemble de l'image en une seule fois. Cela améliore considérablement la vitesse et l'efficacité du modèle[39].

3.3.1.2 Faster R-CNN

Introduit la même année que Fast R-CNN, Faster R-CNN améliore encore le pipeline de détection d'objets. Il remplace l'algorithme de recherche sélective pour la proposition de région par le réseau de proposition de région (RPN). Le RPN est un réseau entièrement convolutif qui génère des propositions de région directement à partir de la carte des caractéristiques produite

par la dorsale CNN. Cette intégration élimine le besoin d'une méthode de proposition de région externe et rend l'ensemble du système entraînable de bout en bout[40].

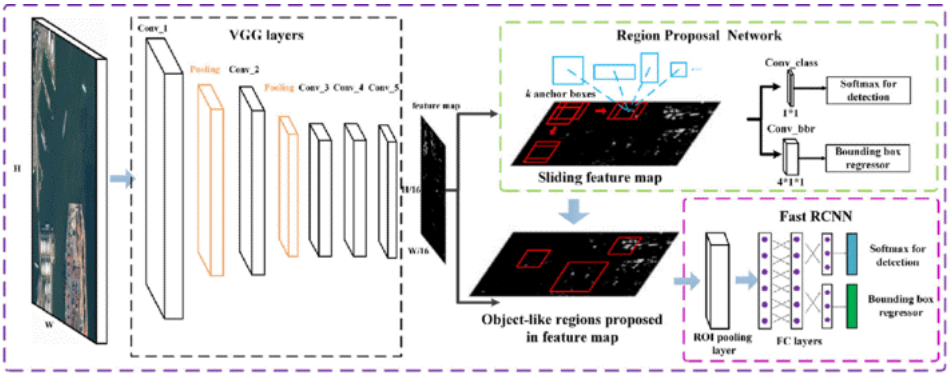

FIGURE 3.2 – architecture de Faster R-CNN[40]

3.3.1.3 Mask R-CNN

S'appuyant sur Faster R-CNN, Mask R-CNN étend le cadre pour effectuer également une segmentation d'instance en plus de la détection d'objets. Il introduit une branche supplémentaire qui prédit les masques au niveau des pixels pour chaque région d'objet proposée[41].

3.3.2 Le modèle SSD (Single Shot MultiBox Detector)

L'approche SSD Détecteur MultiBox à un coup en français est basée sur un réseau convolutif à anticipation (feed-forward) qui produit une collection de taille fixe de Bbox et des scores pour la présence d'instances de classe d'objets dans ces boîtes, suivie d'une étape de suppression non maximale pour produire les détections finales.

Les premières couches de réseau sont basées sur une architecture standard utilisée pour la classification d'images de haute qualité (tronquée avant toute couche de classification), c'est le réseau de base de ce modèle. Et puis une structure auxiliaire au réseau pour produire des détections avec les caractéristiques clés suivantes :

- Cartes d'entités multi-échelles pour la détection.
- Prédicteurs convolutifs pour la détection.
- Boîtes et aspect par défaut [42].

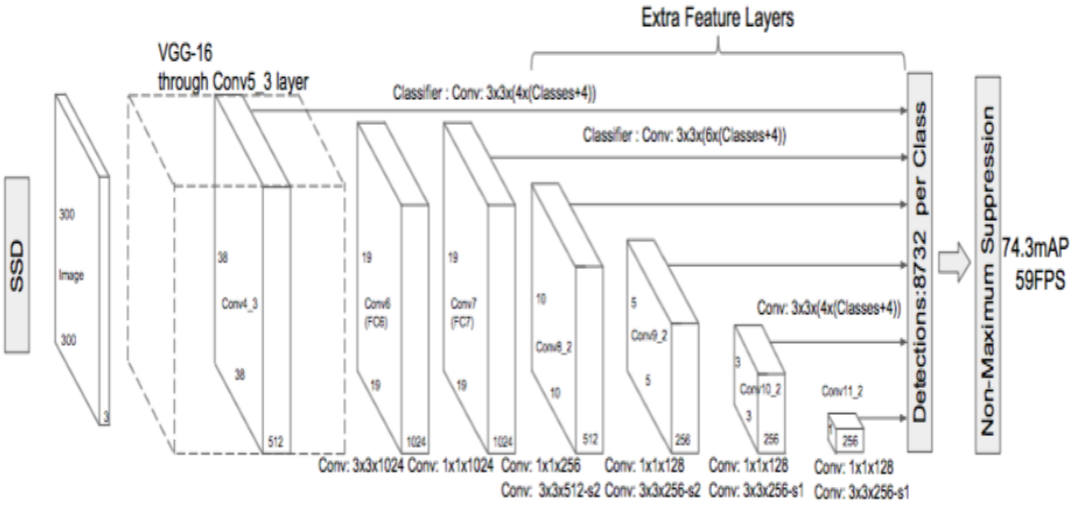

FIGURE 3.3 – L'architecture de modèle SSD[42]

3.3.3 Le modèle YOLO

Il y a eu plusieurs versions de YOLO jusqu'à présent, chacune introduisant des améliorations pour améliorer la précision et la vitesse, mais dans ce projet nous avons utilisé la version 3 pour sa simplicité et son efficacité et dont nous parlerons par la suite.
YOLO est une forme abrégée de « You Only Look once », et il utilise des CNN pour la détection d'objets, Il a été décrit pour la première fois en 2015 dans l'article de Joseph Redmon. YOLO peut détecter plusieurs objets sur une seule image cela signifie qu'en plus de prédire les classes d'objets, et il détecte également les emplacements de ces objets sur l'image ou vidéo.

L'algorithme YOLO fonctionne en divisant une image d'entrée en une grille et en associant chaque cellule de la grille à plusieurs prédictions de Bbox et aux probabilités de classe correspondantes. Le modèle prédit ensuite les probabilités de classe et les Bbox des objets dans chaque cellule de la grille. Cela signifie que ce réseau raisonne globalement sur l'ensemble de l'image et sur tous les objets qu'elle contient. Les prédictions de la Bbox sont affinées à l'aide de boîtes d'ancrage, qui représentent différents rapports d'aspect et tailles d'objets.

3.3.3.1 L'architecture de YOLO

Ce réseau de détection comporte 24 couches convolutionnelles suivies de 2 couches entièrement connectées. Alternance 1 × 1les couches convolutionnelles réduisent l'espace des caractéristiques des couches précédentes [43].

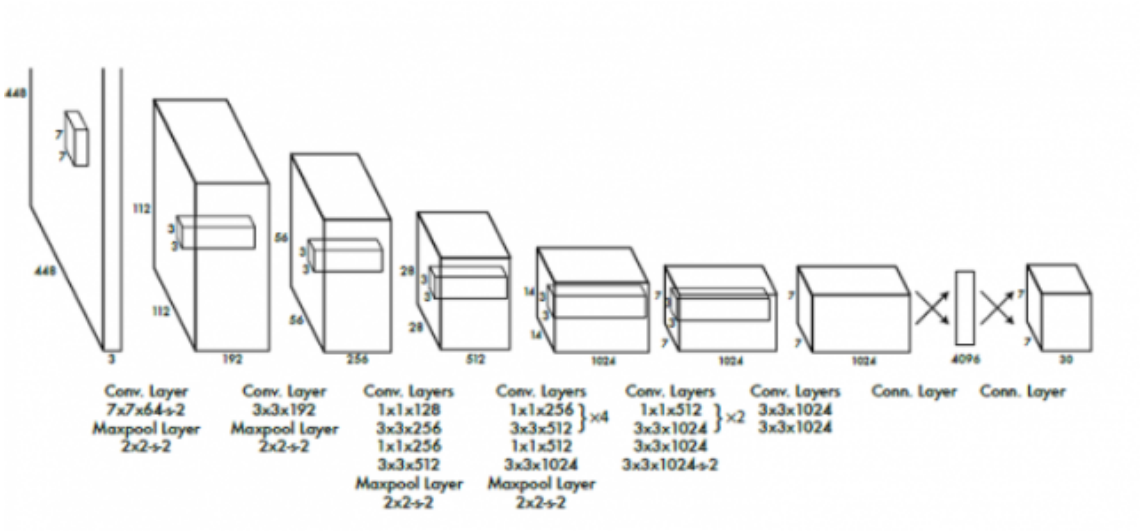

FIGURE 3.4 – L'architecture de modèle YOLO

Ce système modélise la détection comme un problème de régression. Il divise l'image en une grille S × S et pour chaque cellule de la grille prédit les Bbox B, la confiance pour ces boîtes, et les probabilités de classe C. Ces prédictions sont encodées sous la forme d'un tenseur S × S × (B 5 + C).

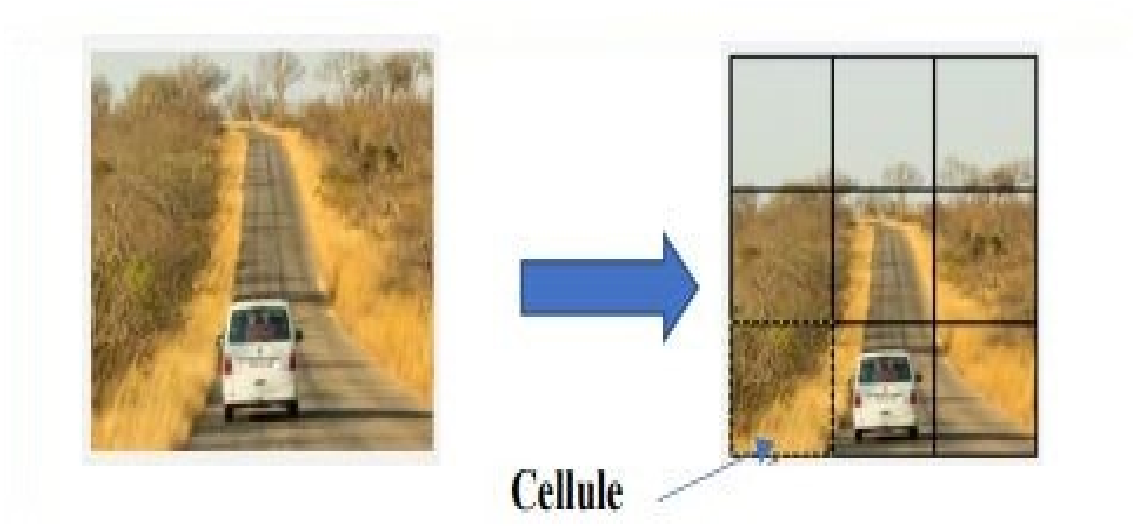

FIGURE 3.5 – Une image divisée en grille (S × S)[44]

Chaque Bbox est constituée de 5 prédictions : x, y, w, h, et confiance. Les coordonnées (x, y) représentent le centre de la boîte par rapport aux limites de la cellule de la grille de la boîte par rapport aux limites de la cellule de la grille.
(w,h) représentent la largeur et la hauteur et sont prédites par rapport à l'image entière et finalement la prédiction de confiance représente le IoU.

3.3.3.2 L'évaluation de modèle YOLO

— **Intersection sur Union (IoU)**

IoU signifie Intersection over Union. IOU est une métrique utilisée pour évaluer la précision des algorithmes de détection d'objets. L'intersection sur l'Union mesure le chevauchement entre la Bbox prédite et la Bbox de vérité terrain pour un objet détecté. Il est calculé en divisant l'aire d'intersection entre les deux boîtes par l'aire de leur union.

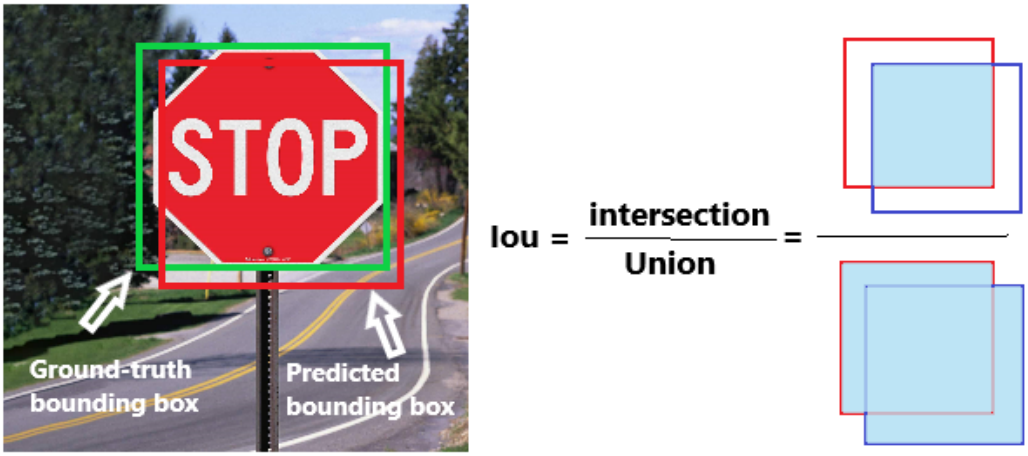

FIGURE 3.6 – Intersection sur Union sur une image et sa formule

Cette métrique aide à déterminer dans quelle mesure la zone de délimitation prédite s'aligne sur la zone de délimitation de vérité terrain. Des valeurs IoU plus élevées indiquent une meilleure précision, car cela signifie que la boîte prédite correspond étroitement à la boîte de vérité terrain[45].

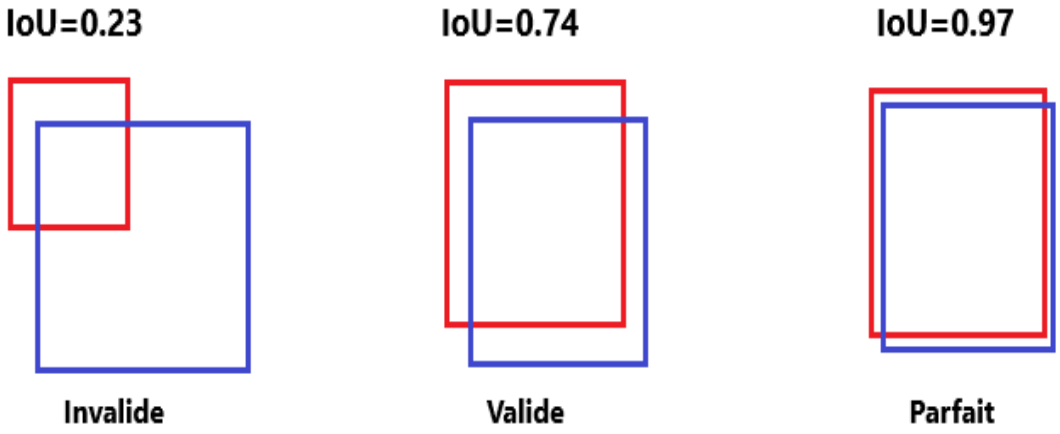

FIGURE 3.7 – exemple sur l'IoU.

— **Boîte d'ancrage (Anchor Box)**

est une forme et une taille de Bbox prédéfinies qui sont utilisées comme référence pendant le processus de détection. YOLO divise l'image d'entrée en une grille et associe chaque cellule de la grille à un ensemble de boîtes d'ancrage, mais si on a plus qu'une boite dans la même cellule, donc Les boîtes d'ancrage sont l'algorithme de YOLO qui sépare les objets si prédire plusieurs Bbox se trouvent dans la même cellule de grille.

Chaque cellule représenter par un vecteur, mais si plusieurs boîtes se trouvent dans la même cellule, ce modèle augmentera le vecteur correspondant pour représenter ces boîtes :

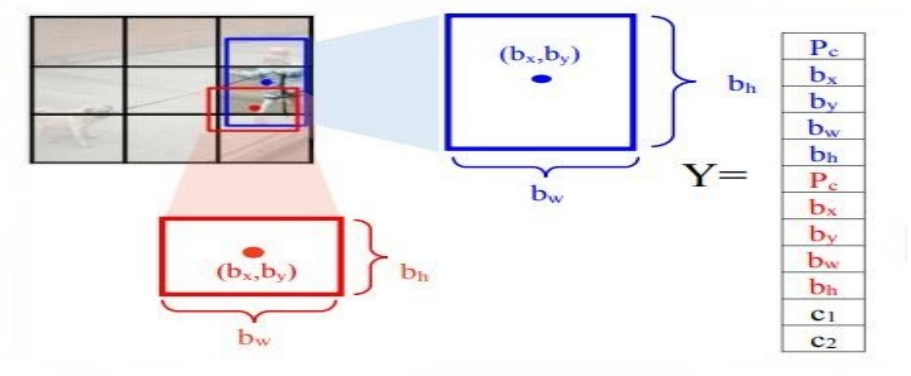

FIGURE 3.8 – la prédiction d'un vecteur dans le cas où plusieurs boîtes se trouvent dans une cellule[44]

— **La suppression non maximale**

Est une étape importante dans le modèle YOLO pour éliminer les détections redondantes et générer des prédictions finales plus précises.

La suppression non maximale est appliquée pour filtrer les détections redondantes et ne conserver que les boîtes les plus pertinentes pour éviter les détections multiples pour un même objet et pour améliorer la précision des résultats de détection du modèle YOLO.

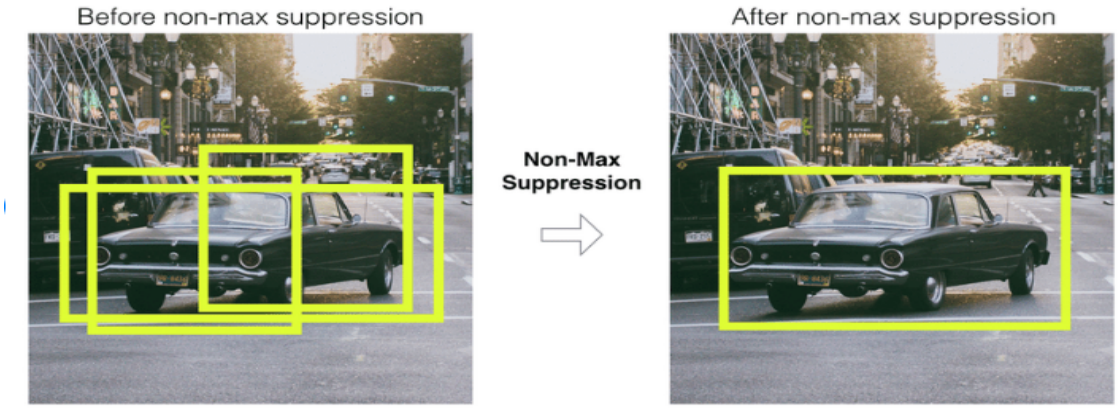

FIGURE 3.9 – l'application de la suppression non maximale[46]

3.4 Le modèle YOLO version 3

YOLO est un algorithme de détection d'objets en temps réel, Comme nous l'avons mentionnée précédemment. Ils ont développé plusieurs versions jusqu'à présent, mais nous avons utilisé dans notre travail la version 3 pour sa simplicité et son efficacité.
Et dans cette partie nous allons parlons plus précisément sur l'architecture et le fonctionnement de yolov3 et leurs avantages.

3.4.1 L'architecture de yolov3

YOLO utilise des couches convolutives et yolov3 se compose à l'origine de 53 couches convolutives également appelées Darknet-53, donc le Backbone de cette version est le Darknet-53 pour l'extraction de caractéristique.
Mais pour les tâches de détection architecture originale empilée avec 53 couches supplémentaires qui nous donnent 106 couches d'architecture pour yolov3, les détections sont faites au niveau des trois couches 82,94 et 106.

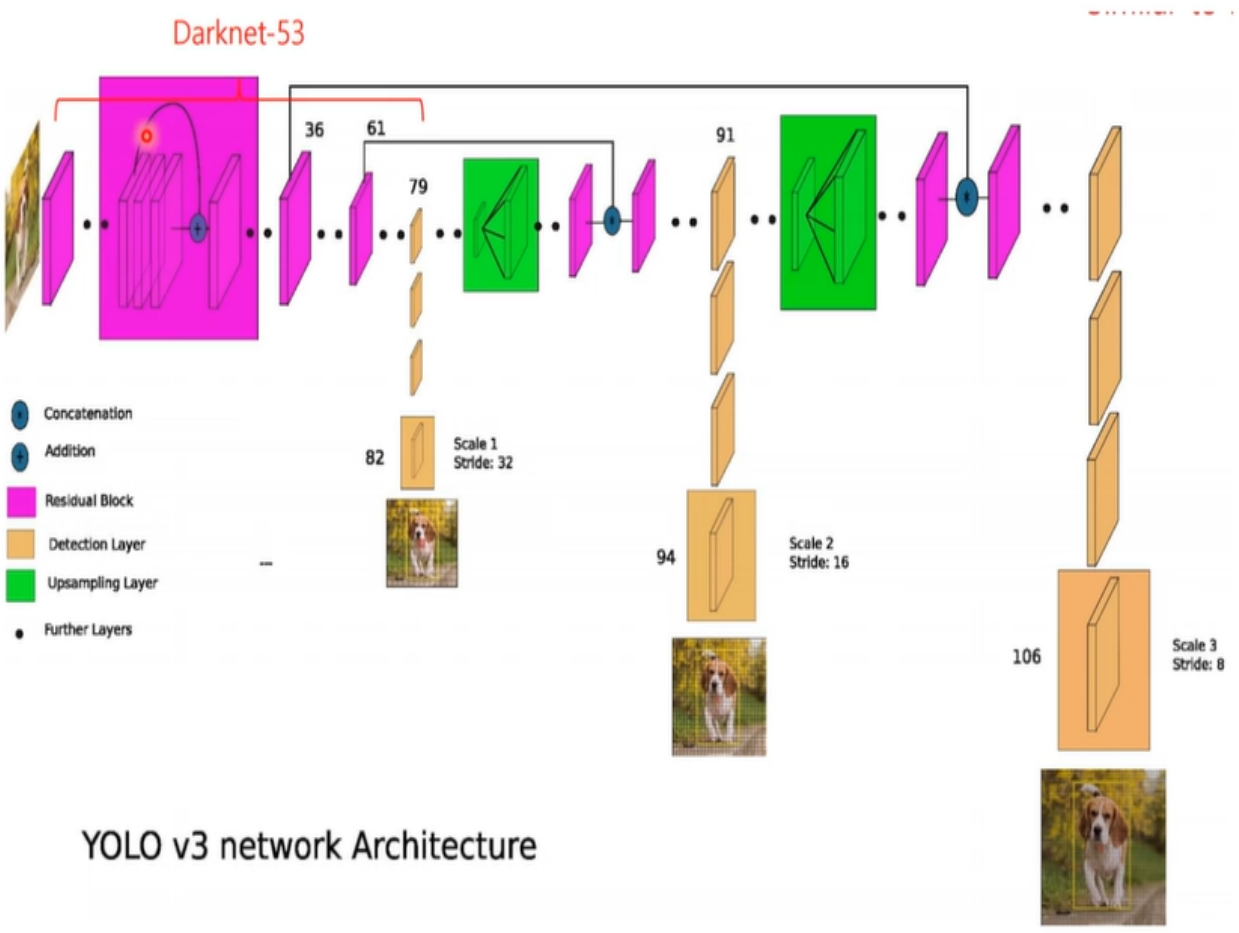

FIGURE 3.10 – architecture de yolov3[47]

Cette version intègre certains des éléments les plus essentiels :
Blocs résiduels, Sauter les connexions (skip connections) et Sur-échantillonnage (Up–Sampling), chaque couche convolutive est suivie d'une batch normalization et d'une fonction d'activation LeakyReLU.
Il n'y a pas de couches de Pooling, mais à la place, des couches convolutives supplémentaires avec Stride 2 sont utilisées pour sous-échantillonner (down-sample) les cartes d'entités (feature-maps), car l'utilisation de couches convolutives supplémentaires pour sous-échantillonner les cartes d'entités empêche la perte d'entités de bas niveau que la couche de regroupement ne fait qu'exclure, puis cela a aidé à améliorer la capacité de détection de petits objets.

3.4.2 Images d'entrée

L'entrée est un lot d'images en forme de (416,416) « largeur et hauteur » et ce nombre peut être modifié et défini comme 608, ou tout autre nombre divisible par 32 sans vivre de reste, les images d'entrée elles-mêmes :

- Peut-être de n'importe quelle taille.
- Sera redimensionné à la taille du réseau.
- Peut-être gardé ou non.

3.4.3 Détections à trois échelles (Scales)

Yolov3 effectue des détections à trois échelles différentes et à trois endroits distincts pour les détections se trouvent les couches 82, 94 et 106.
Le réseau sous-échantillonne l'image d'entrée par les facteurs suivants : 32, 16 et 8 à ces endroits distincts du réseau en conséquence, ces trois nombres sont appelés foulée du réseau et ils montrent comment la sortie à trois endroits distincts du réseau est plus petite que l'entrée au réseau.

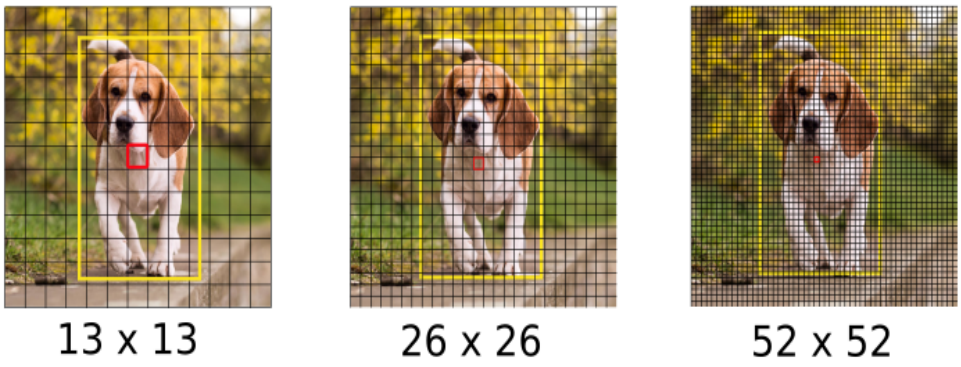

FIGURE 3.11 – Détection à différentes échelles[49]

3.4.4 Extracteur de caractéristique

Darknet-53 à également des connexions de raccourci et est beaucoup plus grand, il a 53 couches convolutives :

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Table 1. **Darknet-53.**

FIGURE 3.12 – L'architecture de Darkent-53[48]

Ce nouveau réseau est beaucoup plus puissant que Darknet19 mais toujours plus efficace que ResNet-101 ou ResNet-152.

3.4.5 Les noyaux de détection

Dans YOLOv3, les noyaux de détection sont utilisés pour prédire les informations relatives aux objets détectés. La forme des noyaux de détection est déterminée par l'équation :
1×1× (b× (5+c)) où b=3, nombre de Bbox, et c est le nombre de classes.
Parce que yolov3 prédit 3 Bbox pour chaque cellule de ces cartes d'entités :
Coordonnées centrales de la Bbox, largeur et hauteur qui sont les dimensions de la Bbox, le score d'objectivité et la liste des confiances pour chaque classe à laquelle cette Bbox peut appartenir.

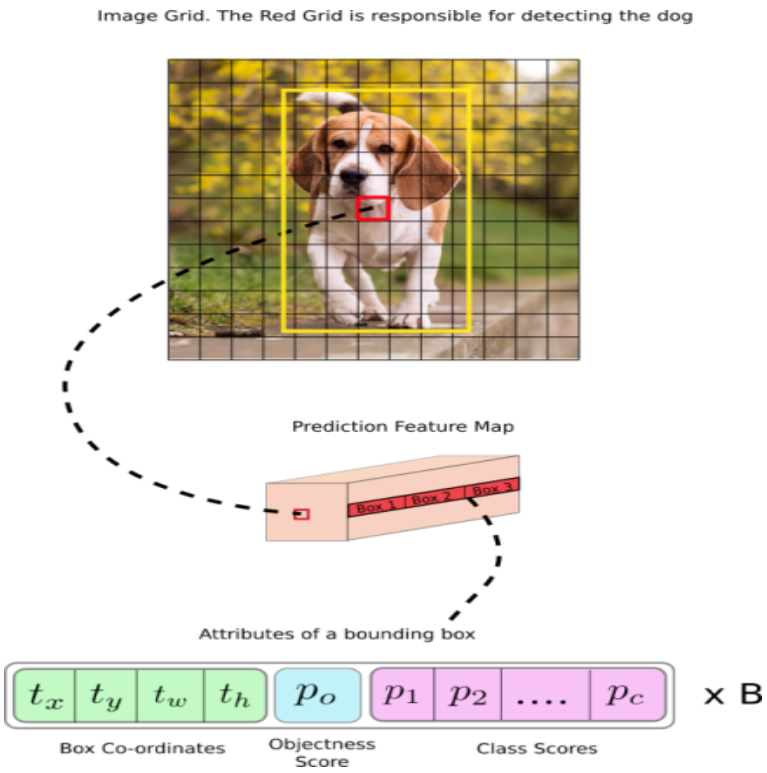

FIGURE 3.13 – Les attributs de la boîte englobante[49]

3.4.5.1 Les avantages de yolov3

— **Détection d'objets en temps réel :**
YOLOv3 est conçu pour réaliser une détection d'objets en temps réel, ce qui signifie qu'il peut traiter des images ou des images vidéo à une fréquence d'images élevée, comme des véhicules autonomes ou des systèmes de vidéosurveillance.

— **Yolov3 est meilleur pour détecter les petits objets :**
Les détections à différentes couches aident à résoudre le problème de la détection de petits objets. Les couches sur-échantillonnées concaténées avec les couches précédentes aident à préserver les caractéristiques à grain fin qui aident à détecter les petits objets. La couche 13 x 13 est responsable de la détection des gros objets, tandis que la couche 52 x 52 détecte les objets plus petits, la couche 26 x 26 détectant les objets moyens.

3.5 Les modèles de reconnaissance faciale par DL

Les modèles de reconnaissance faciale basés sur DL ont fait des progrès significatifs ces dernières années. Ces modèles tirent parti de la puissance des réseaux de neurones profonds pour extraire des caractéristiques de haut niveau à partir d'images faciales et identifier avec précision les individus. Nous allons parler dans cette partie de certains modèles de reconnaissance faciale

populaires basés sur DL :

3.5.1 Le modèle MTCNN (Multi-Task Cascaded Convolutional Networks)

MTCNN est un algorithme populaire principalement conçu pour la détection de visage. Il utilise des techniques DL pour localiser et extraire avec précision les visages des images ou des images vidéo.

MTCNN exploite une architecture en cascade à trois étapes de réseaux convolutifs profonds soigneusement conçus pour prédire l'emplacement des visages et des points de repère de manière grossière à fine.

Étape 1 : Le réseau Proposal Network (P-Net) est utilisé pour proposer des fenêtres faciales candidates et leurs vecteurs de régression de Bbox. Ensuite, les candidats sont ajustés en fonction des vecteurs de régression de boîte estimés.

Étape 2 : Tous les candidats sont dirigés vers un autre CNN, appelé Refine Network (R-Net), qui rejette en outre un grand nombre de faux candidats, effectue un étalonnage avec limitation box régression, et conduit NMS.

Étape 3 : Cette étape est similaire à la deuxième étape, mais en cette étape, il ajoute le point de repère à 5 points des yeux, du nez et de la bouche dans la Bbox finale contenant le visage détecté[50].

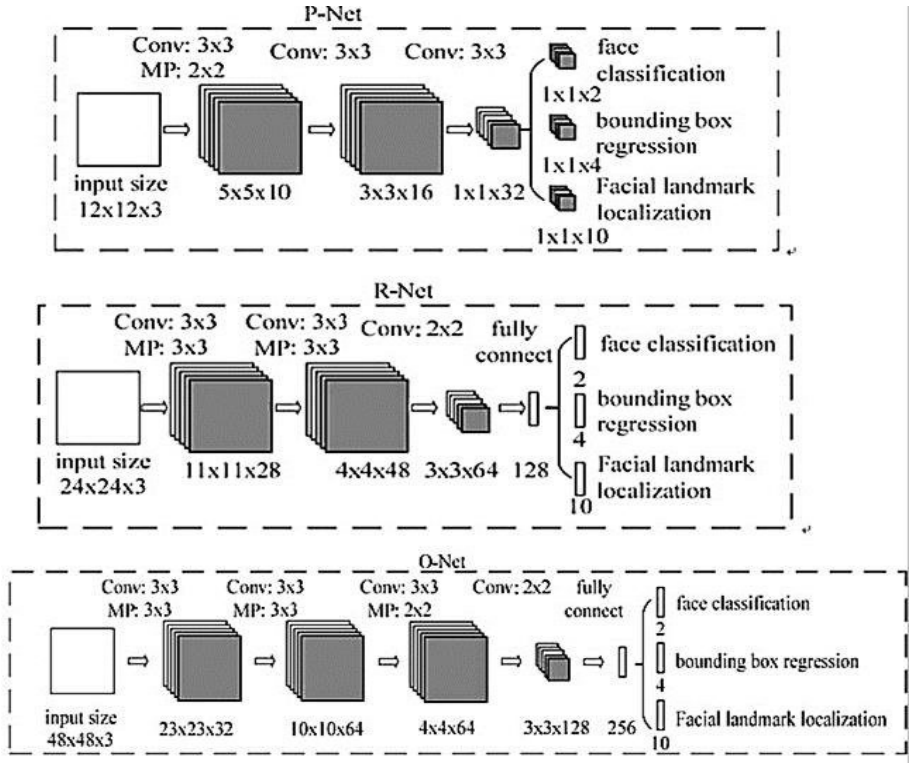

FIGURE 3.14 – Les trois étapes de MTCNN[50].

3.5.2 Le modèle VGG-Face

VGG-Face est un modèle de réseau neuronal à convolution profonde (CNN) spécialement conçu pour les tâches de reconnaissance faciale. Il a été développé sur la base de l'architecture VGG-Net, qui signifie Visual Geometry Group Network.

Il est conçu pour extraire les traits du visage et effectuer la reconnaissance faciale sur un ensemble de données faciales à grande échelle.

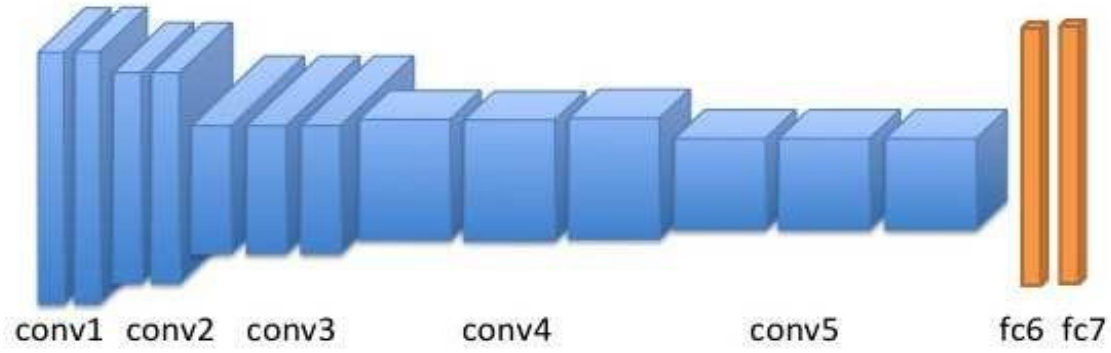

FIGURE 3.15 – Architecture de modèle VGG-Face[51].

L'utilisation de VGG-Face dans la reconnaissance faciale implique le prétraitement des images de visage, l'extraction des caractéristiques à l'aide du réseau VGG-Face et la comparaison des caractéristiques pour reconnaître les visages. Ces étapes permettent d'identifier

les individus en se basant sur les similitudes entre leurs caractéristiques faciales extraites[51]. Il y a beaucoup de modèles pour la tâche de reconnaissance faciale, nous avons utilisé l'un d'eux pour notre projet qui est FaceNet En raison de sa précision, de sa robustesse, de sa facilité d'utilisation et de sa popularité dans la partie suivante nous allons parlons sur l'architecture et fonctionnement de modèle FaceNet.

3.5.3 Le modèle FaceNet pour la reconnaissance faciale

FaceNet est un modèle de DL développé pour les tâches de reconnaissance faciale en 2015. Il utilise un réseau neuronal convolutif (CNN) pour extraire des intégrations de caractéristiques de haute dimension à partir d'images faciales. Ces incorporations représentent des caractéristiques uniques du visage et peuvent être utilisées pour diverses tâches liées au visage, telles que la vérification du visage, l'identification du visage et le regroupement des visages (clustering)[51].

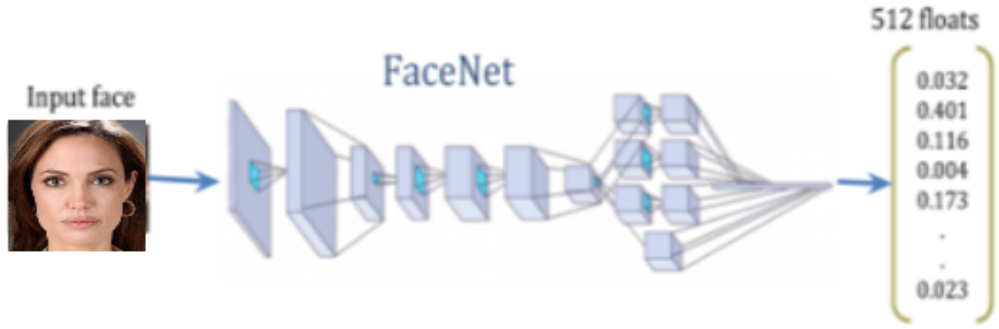

FIGURE 3.16 – Le modèle FaceNet

L'idée clé derrière FaceNet est de créer une fonction de cartographie qui convertit directement les images faciales en un espace euclidien compact. Cela permet de mesurer la similarité entre les visages en utilisant des métriques de distance simples comme la distance euclidienne ou la similarité cosinus. Cette fonction de cartographie est entraînée à l'aide d'une fonction de perte de triplet (triplet loss fucntion en Anglais), Cette fonction encourage les intégrations des visages de la même personne à être proches les unes des autres dans l'espace d'intégration et celles de personnes différentes à être éloignées[52].

3.5.3.1 La fonction de perte de triplet

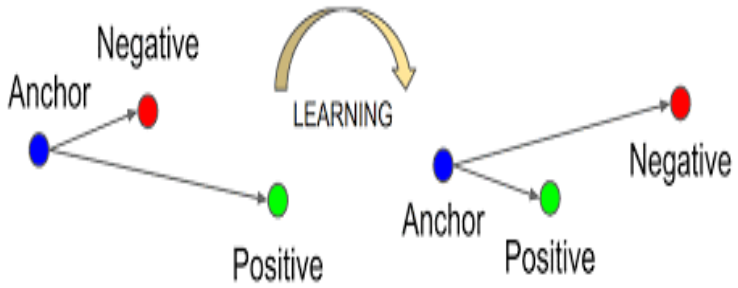

FIGURE 3.17 – La fonction de perte de triplet[52]

Pour chaque itération d'entraînement, le modèle FaceNet prend trois images d'entrée pour chaque exemple d'entraînement : une image d'ancrage, une image positive (une image de la même personne que l'ancre), et une image négative (une image d'une autre personne). Ces images sont choisies parmi d'un ensemble de données de visages.

Les embeddings « une représentation numérique qui capture les traits et caractéristiques uniques d'un visage» de l'image d'ancrage, des images positives et négatives sont comparés en utilisant la distance euclidienne ou la similarité cosinus. Cette métrique de distance mesure la similarité ou la différence entre les représentations de visage.

Le but de la fonction de perte de triplet est d'encourager les incorporations d'images d'ancrage à être plus proches des images positives et plus éloignées des images négatives. En calculant les distances entre l'ancre et les plongements positifs (d-ap) ainsi qu'entre l'ancre et les plongements négatifs (d-an), la fonction de perte vise à minimiser d-ap et maximiser d-an tout en maintenant une marge entre eux (voir figure 3.17)[52].

3.5.3.2 L'architecture de FaceNet

L'architecture faceNet utilise des réseaux de neurones convolutifs (CNN) qui se compose de plusieurs couches convolutionnelles suivies de fonctions d'activation non linéaires « ReLU » pour extraire les caractéristiques du visage et cartographier les visages dans un espace d'intégration de grande dimension où des visages similaires sont regroupés[52].

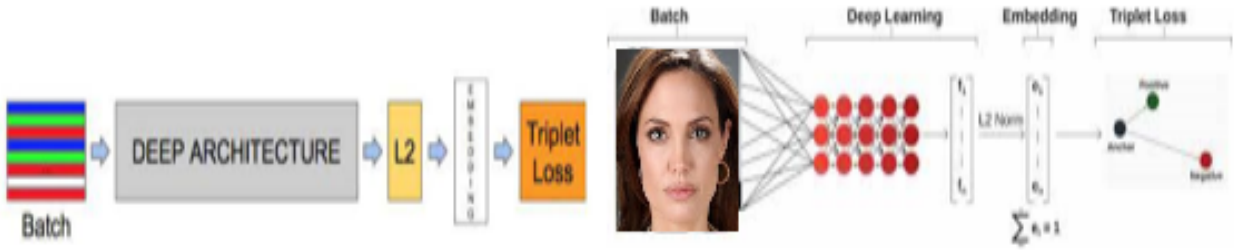

FIGURE 3.18 – L'architecture de FaceNet[52]

"L2" fait référence à la normalisation L2 appliquée aux incrustations (Embeddings) de visage. La normalisation L2, également connue sous le nom de normalisation euclidienne, est une technique utilisée pour mettre à l'échelle les vecteurs de caractéristiques afin qu'ils aient une norme ou une longueur unitaire.
L'objectif de la fonction de perte de triplet est de minimiser la distance entre l'ancre et les images positives tout en maximisant la distance entre l'ancre et les images négatives dans l'espace d'intégration comme nous l'avons dit dans la partie précédente.

3.6 Conclusion

Dans de ce chapitre nous avons présenté quelque modèle de la détection d'objet et la reconnaissance faciale par vision ordinateur, et nous avons basé beaucoup plus sur le modèle yolov3 et FaceNet qui sont-ils utilisés dans notre conception. Dans le chapitre suivant, nous aborderons la conception de notre projet et les résultats que nous avons obtenus.

Chapitre 4

CONCEPTION ET IMPLEMENTATION

4.1 Introduction

Ce chapitre couvrira tous les éléments liés à la mise en œuvre de notre système, en commençant par la configuration de l'environnement de travail. Nous aborderons ensuite l'apprentissage du modèle yolov3 pour le comptage de la foule, ainsi que l'utilisation du modèle FaceNet pour la reconnaissance des visages des personnes suspectes. Nous présenterons les résultats de l'analyse réalisée par ce système. Enfin, nous explorerons la mise en œuvre de notre système en utilisant une interface graphique conviviale.

4.2 Environnement de Travail

4.2.1 Langage python

Python est un langage de programmation interprété de haut niveau connu pour sa simplicité et sa lisibilité.
La simplicité, les bibliothèques puissantes, la communauté active et la flexibilité de Python en font un choix privilégié pour le développement de l'IA, permettant aux chercheurs et aux développeurs de se concentrer sur la résolution de problèmes d'IA complexes.

4.2.2 Le pc portable utilisé

- Processeur : Intel(R) Pentium(R) CPU 2117U
- Mémoire RAM : 4,00 Go

4.2.3 Les logiciels utilisés :

- **Labelling :**
  Est un outil d'annotation d'images graphiques open source utilisé pour étiqueter manuellement des objets dans des images pour former des modèles de détection d'objets.
- **Google Colab :**
  Abréviation de Google Colaboratory, est une plate-forme basée sur le cloud fournie par Google pour exécuter du code Python et exécuter des fichiers Jupyter Notebook. Il offre un environnement gratuit et pratique pour l'analyse de données, l'apprentissage automatique.
  Google Colab fournit un accès gratuit aux ressources GPU et TPU et nous l'avons utilisé pour la formation.

— **Visual studio :**

Est un environnement de développement intégré (IDE) développé par Microsoft. Il fournit un ensemble complet d'outils et de fonctionnalités pour le développement de logiciels sur diverses plates-formes et langues. Visual Studio prend en charge plusieurs langages de programmation, notamment C++, C, Python, JavaScript, etc.

4.2.4 Les bibliothèques utilisées :

Dans notre projet nous avons utilisé plusieurs librairies dont les plus importantes sont :

— **OPENCV :**

(Open Source Computer Vision Library) est une bibliothèque de logiciels open source de vision par ordinateur et d'apprentissage automatique. Il fournit un large éventail de fonctions et d'algorithmes qui permettent aux développeurs de traiter, d'analyser et de comprendre des données visuelles, telles que des images et des vidéos.

— **NUMPY :**

NumPy (Numerical Python) est une bibliothèque Python puissante dédiée au calcul numérique. Elle offre un support pour les tableaux et matrices multidimensionnels de grande taille, ainsi qu'une collection de fonctions mathématiques permettant d'effectuer des opérations efficaces sur ces tableaux.

— **TENSORFLOW :**

Est un framework d'apprentissage automatique open source développé par Google. Il fournit un ensemble complet d'outils, de bibliothèques et de ressources pour créer et déployer des modèles d'apprentissage automatique. TensorFlow est largement utilisé pour diverses tâches, notamment l'apprentissage en profondeur, le traitement du langage naturel, la vision par ordinateur, etc.

— **Tkinter :**

Est une bibliothèque Python intégrée utilisée pour créer des interfaces utilisateur graphiques (GUI). Il fournit un ensemble d'outils et de widgets pour concevoir et développer des applications GUI.

4.3 Implémentation du système

Dans cette partie, nous allons diviser notre travail en deux étapes :
La première étape concerne la détection et le comptage de la foule, c'est-à-dire le nombre de personnes présentes dans une zone donnée, par le modèle YOLOv3.

La deuxième étape consiste à effectuer la reconnaissance faciale des personnes suspectes à l'aide du modèle FaceNet. Enfin, la partie de l'interface graphique pour l'ensemble du système.

4.3.1 La détection et le comptage de la foule à l'aide yolov3

Avant de commencer l'apprentissage par le modèle yolov3, il est juste utile de préciser que le choix ou la création de la base de données est une étape très essentielle. Il est essentiel que la base de données soit correctement étiquetée et que les fichiers d'annotations (ground truth) soient au format YOLO.
Sur internet on peut trouver des bases de données prêt mais nous avons préféré de créer notre propre base de données.

4.3.2 La création de la base de données

A- Collecte d'images pour l'ensemble de données :
Dans le cadre de la création de notre base de données, nous avons effectué une collecte d'images de foules à partir de sources en ligne, notamment Google Images. Nous avons rassemblé un total de 78 images mais chaque image comportant un certain nombre de personnes voir (Figure 4.1).

FIGURE 4.1 – Exemple d'images de la base de données

B- Etiquetage d'images (annotations) :
L'annotation est un processus essentiel pour localiser les personnes dans une image. Cela implique de placer un rectangle (Bbox) autour de chaque personne et d'enregistrer leurs coordonnées, ce qui fournit les informations nécessaires pour chaque image, telles que la classe de l'objet et les coordonnées de la Bbox., car ce sont précisément les données dont yolov3 a besoin

pour entraîner la base de données. Nous avons utilisé l'application LabelImg pour l'étiquetage des images, et nous avons concentrés sur la partie de la tête des personnes et nous avons choisi YOLO format voir la(Figure 4.2).

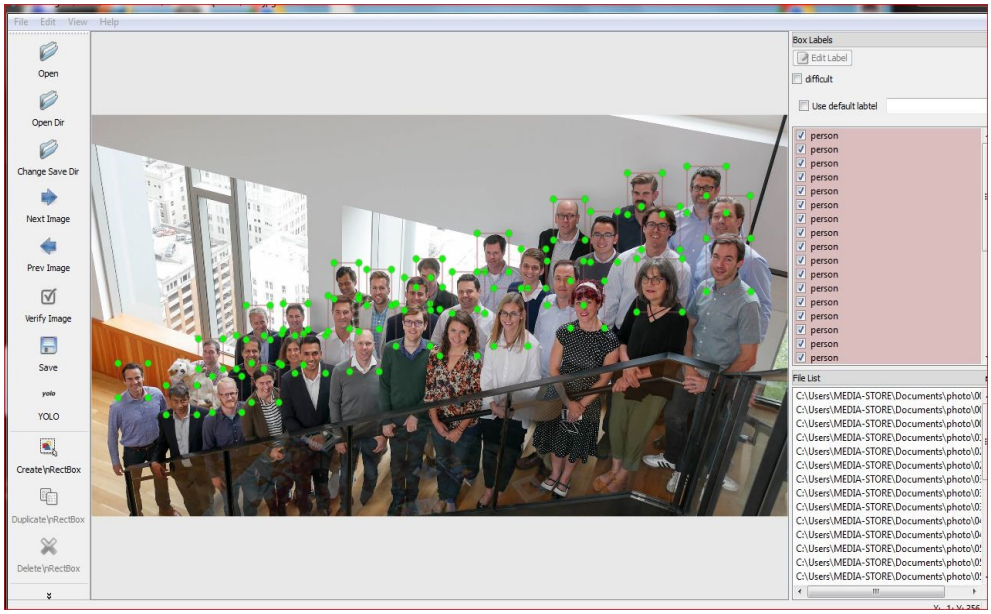

FIGURE 4.2 – Etiquetage d'images a l'aide LabelImg

Après l'étiquetage de toutes les images, nous avons obtenu des fichiers texte (.txt). Chaque fichier contient les coordonnées normalisées (x, y, w, h) de la Bbox correspondant à chaque tête de personne dans l'image. La classe attribuée à chaque Bbox est 0, conformément au format YOLO. Toutes ces coordonnées sont normalisées entre 0 et 1, comme illustré dans(Figure 4.3).

<La classe d'objet ><centre-x><centre-y><largeur><hauteur>

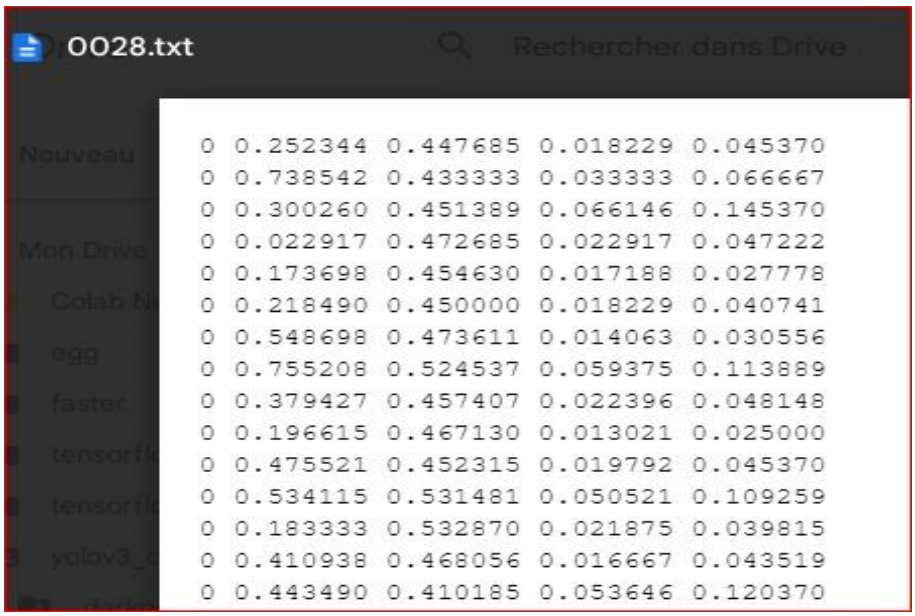

FIGURE 4.3 – Exemple d'annotation de la base de données

A la fin nous avons obtenus 78 images avec 1377 annotations parce que nous étions limités dans le temps.

4.3.3 Entrainement avec le modèle yolov3

Dans cette partie, nous avons utilisé Google Colab en raison de sa capacité à fournir des GPU, ce qui accélère le processus d'entraînement.
Ensuite nous avons cloné le référentiel "darknet" qui contient le model yolov3 voir (Figure 4.4) :

FIGURE 4.4 – Comment cloner le référentiel darknet

L'étape suivante consiste à quelque modification pour activer le GPU, OPENCV, CUDNN dans le Makefile. Puis compile le darknet en utilisant les paramètres modifiés comme la(Figure 4.5).

```
%cd darknet
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile
!sed -i 's/GPU=0/GPU=1/' Makefile
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile
!make
```

FIGURE 4.5 – modification pour activer le GPU, OPENCV, CUDNN dans le Makefile

L'étape suivante est l'importation de fichier yolov3 configuration :

```
[ ] !cp cfg/yolov3.cfg cfg/yolov3_training.cfg
```

FIGURE 4.6 – Importation de fichier yolov3 cfg

Et puis l'étape suivante est de modifier le fichier de configuration « yolov3-training.cfg » pour préparer le modèle pour l'entrainement et pour adapter notre base de données :

Les paramètres qui sont modifiés étaient calculésde la manière suivante :

Le nombre d'itérations (max-batches) :

2000*nombre des classes, dans notre cas 2000*1=2000 car nous avons 1 seul classe (person), mais nous avons changés les d'itérations a 5000 pour laisser le modèle de compléter sa formation.

Le nombre de la classe=80 par classe=1.

Les nombre de filtres=255 par (nombre de la classe +5)*3 donc filtres= (1+5)*3=18.

Et le nombre de batch et subdivision (batch = 64, subdivision = 16).

yolov3_training.cfg

```
[net]
# Testing
#batch=64
#subdivisions=16
# Training
 batch=64
subdivisions=16
width=416
height=416
channels=3
momentum=0.9
decay=0.0005
angle=0
saturation = 1.5
exposure = 1.5
hue=.1

learning_rate=0.001
burn_in=1000
max_batches = 5000
policy=steps
steps=400000,450000
scales=.1,.1
```

FIGURE 4.7 – Fichier de training cfgde yolov3

Après avoir décompressé notre base de données dans un dossier appelé "data/obj", nous avons également crée un dossier appelé "data/obj.names" qui contient le nom des classes, ainsi

qu'un répertoire de sauvegarde appelé "backup" pour stocker les poids qui seront mis à jour au cours de l'entraînement.
Ensuite, nous avons téléchargé les poids pré-entraînés "darknet53.conv.74" pour l'apprentissage de ce modèle avec notre base de données pour la première fois. Par la suite, nous utilisons toujours les poids finaux "last-weights" pour continuer l'entraînement, afin de ne pas repartir de zéro.

```
# Start the training

!./darknet detector train data/obj.data cfg/yolov3_training.cfg darknet53.conv.74 -dont_show
```

FIGURE 4.8 – L'entrainement de modèle avec la base de données

En raison de la limitation de la GPU dans Google Colab, l'entraînement du modèle peut s'arrêter et vous devez utiliser les poids à chaque fois pour reprendre l'apprentissage là où il s'était arrêté.

Chaque fois que l'entraînement est interrompu, un graphe montrant l'évolution de la valeur de perte (loss) en fonction des itérations est automatiquement affiché, nous avons obtenus plusieurs graphes comme illustré dans la (Figure 4.9) :

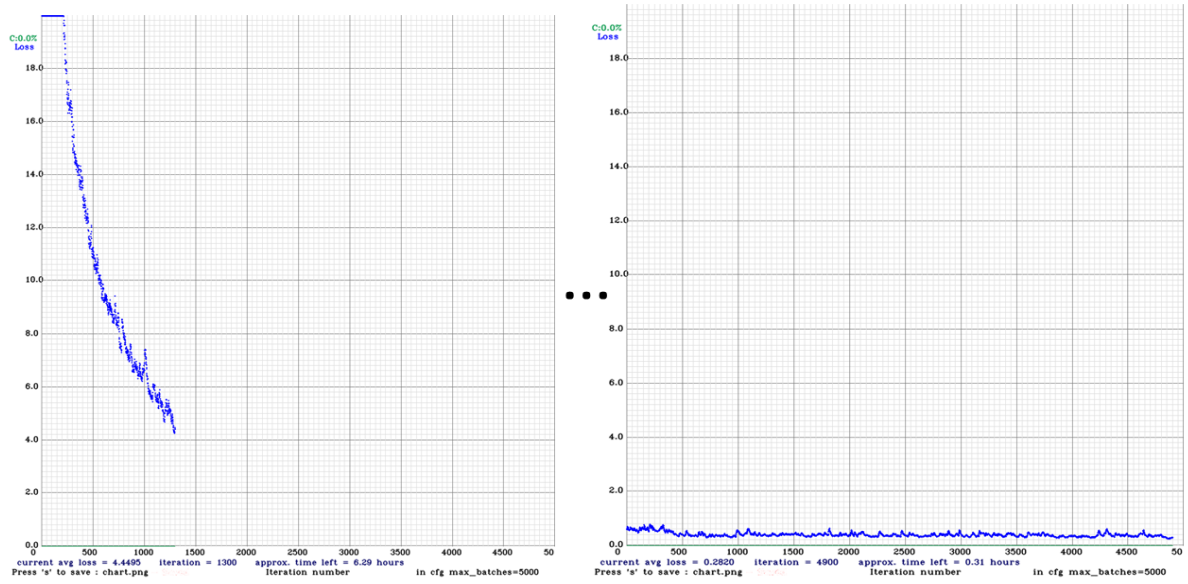

FIGURE 4.9 – Le graphique illustre les résultats de notre entraînement.

Nous allons voir que la perte diminue chaque fois que nous ré-entrainons le modèle.
Nous avons obtenus la valeur de loss= 0.28.

4.3.4 Evaluation de modèle

mAP(mean-average-precision) et IoU sont des mesures importantes pour évaluer les performances des modèles de détection d'objets tels que YOLOv3.
mAP fournit une mesure complète de la précision du modèle dans différentes catégories d'objets, tandis que IoU évalue la capacité du modèle à localiser avec précision les objets.
Nous avons cité dans le tableau suivant les variations de mAP mesuré dans deux différentes seuils de IoU a IoU=0.6, et a IoU=0.8(la comparaison entre la variation de résultats de détections de chaque poids âpres chaque nombre d'itérations et ground-truth "Vérité terrain") voir la table (TABLE 4.1) :

TABLE 4.1 – Les variations de mAP

Nombre d'itérations	1300	1600	2400	2500	6500
mAP(IoU=0.60)	0.59	0.61	0.88	0.89	0.9072
mAP(IoU=0.80)	0.34	0.43	0.63	0.69	0.72

Nous remarquons que plus le modèle est entraîné, plus le mAP est élevé.
Alors les best weights= mAP= 90.72% .

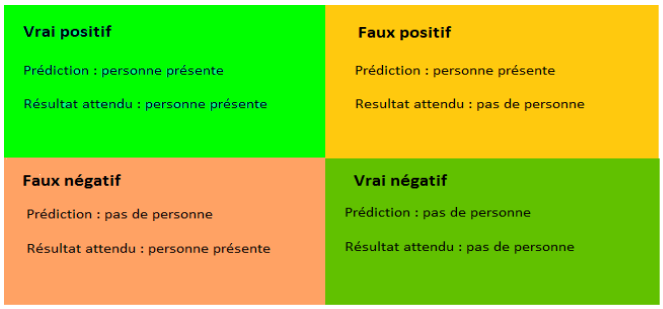

FIGURE 4.10 – Les indicateurs de performance

Les indicateurs de performance utilisés pour obtenir une évaluation globale de l'efficacité de notre modèle, ces indicateurs calculer par la matrice de confidence qui contient des nombres comme Tp(vrai positif) Fp (faux positif) Fn(faux négatif) Tn(vrai négatif) voir la figure précédente(FIGURE 4.10).

```
# mAP of all classes
mAP = 90.72%

# Mean average IOU rate  of all classes:
mIOU =82.65%


# Number of ground-truth objects per class
person: 1377

# Number of detected objects per class
person: 1367 (tp:1283, fp:84, fn:77)
```

FIGURE 4.11 – Les résultats d'évaluation

FIGURE 4.12 – Exemple de résultats d'IoU

La précision :

elle correspond à la proportion de détections exactes exprimée en pourcentage :

$$Prcision = \frac{VraiPositif}{VraiPositif + FauxPositif} \tag{4.1}$$

Rappel :

il mesure la capacité d'un modèle à prédire l'ensemble des résultats attendus :

$$Rappel = \frac{VraiPositif}{VraiPositif + Fauxngatif} \tag{4.2}$$

TABLE 4.2 – Les résultats d'apprentissage

Classe	Précision	AverageIoU	Rappel
personne	0.93	0.82	0.94

Cette table représente la performance de notre modèle, nous avons obtenus 94% de Rappel, et 93% de précision.

4.4 Comptage de foule

- **Partie test**

Au début, nous avons utilisé la fonction cv2.dnn.readNet qui fait partie de la bibliothèque OpenCV, en particulier dans le module Deep Neural Network (DNN). Il est utilisé pour charger un modèle de réseau neuronal pré-formé à partir du disque dans la mémoire.
Et aussi nous avons utilisé aussi la fonction cv2.dnn.blobFromImage Il est utilisé pour prétraiter les images d'entrée avant de les faire passer par un réseau de neurones.
Et cv2.dnn.NMSBoxes pour effectuer la Suppression Non-Maximale. Et à la fin nous avons ajouté un compteur qui compte les personnes détectés.

FIGURE 4.13 – Résultats de détection et le nombre total de personnes détectés

Dans la première image Résultat de la détection =27 personnes / le nombre réel= 30 personnes.

Dans la deuxième image Résultat de la détection =20 personnes / le nombre réel= 24 personnes.

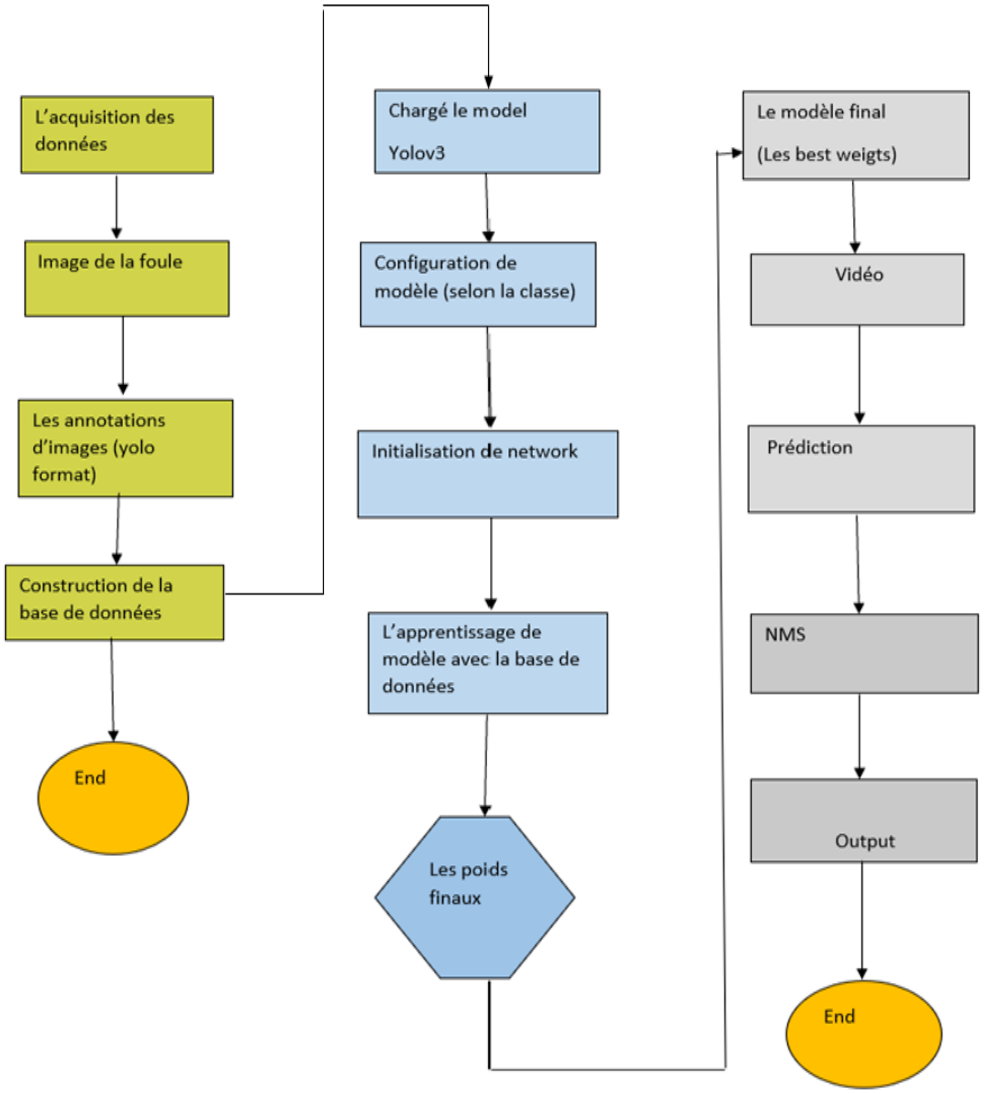

FIGURE 4.14 – Architecture de la partie de comptage de la foule

4.5 La reconnaissance faciale de personnes suspectes par le modèle

4.5.1 FaceNet

Pour cette partie nous avons utilisé le modèle pré-entrainer FaceNet, Il a été formé sur la base de données MS-Celeb-1M, l'ensemble de données contient 1 million d'images de célébrités.

• **La partie 1 (Prétraitement de données) :**

La Collecte d'images :

Nous avons collecté un ensemble d'images pour chaque personne que nous souhaitons identifier comme suspects. Ces images sont stockées dans des fichiers portant le nom de chaque personne respective.

Et nous avons mis tous ces fichiers dans un fichier train-img :

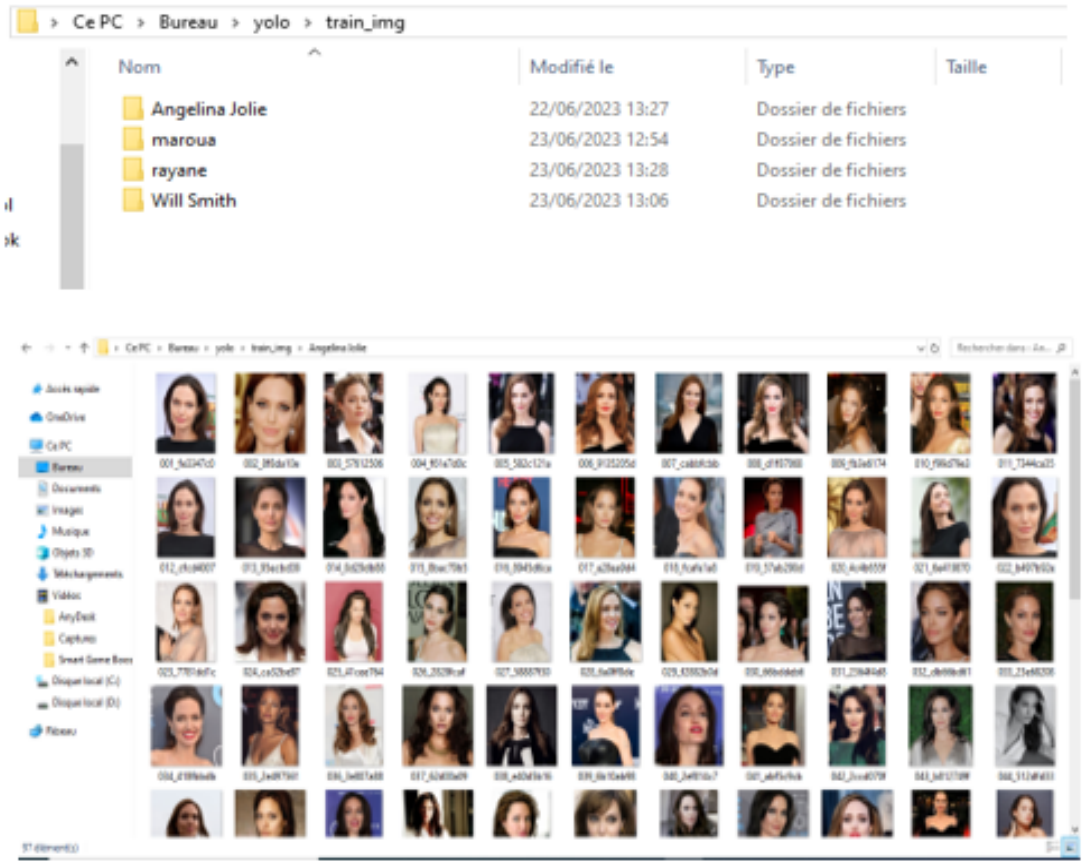

FIGURE 4.15 – les fichiers d'images de données

Prétraitement de données :cette partie du code effectue le prétraitement des images en les alignant pour une utilisation ultérieure dans le processus de reconnaissance faciale.

```python
from preprocess import preprocesses

input_datadir = './train_img'
output_datadir = './aligned_img'

obj=preprocesses(input_datadir,output_datadir)
nrof_images_total,nrof_successfully_aligned=obj.collect_data()

print('Total number of images: %d' % nrof_images_total)
print('Number of successfully aligned images: %d' % nrof_successfully_aligned)
```

FIGURE 4.16 – Prétraitement des données

preprocesses effectue des tâches de prétraitement d'image telles que la détection de visage, l'alignement et l'enregistrement des images alignées. Ce code utilise l'algorithme de détection de visage MTCNN pour détecter et aligner les visages dans un ensemble d'images d'entrée, en enregistrant les images alignées et leurs coordonnées de Bbox correspondantes.

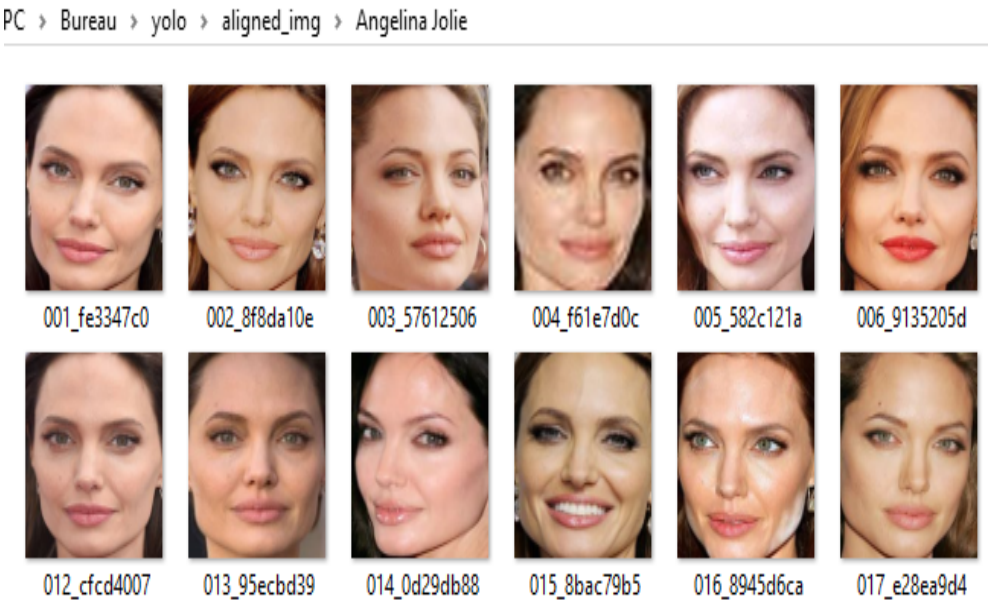

FIGURE 4.17 – la sortie d'aligned-img

L'objectif de l'alignement des images dans les tâches d'analyse de visage est de garantir que les visages dans les images sont positionnés et mis à l'échelle de manière cohérente, permettant une comparaison et une analyse plus précises des traits du visage. L'étape suivante procède a l'extraction de caractéristique les embeddings a l'aide de modèle pré-entrainé FaceNet d'images alignées et puis de les classifier par rapport la différence entre la distance calculée entre eux.

Pendant le prétraitement de donnée, la perte de triplet minimise la distance entre une ancre

et un positif, les deuxavoir la même identité et maximise la distance entre ancre et un négatif.

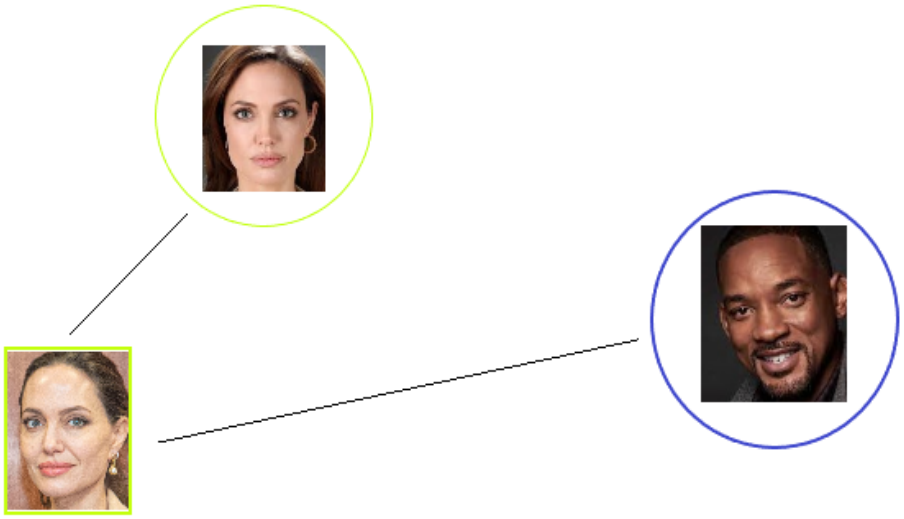

FIGURE 4.18 – Example sur la distance entre les embeddings

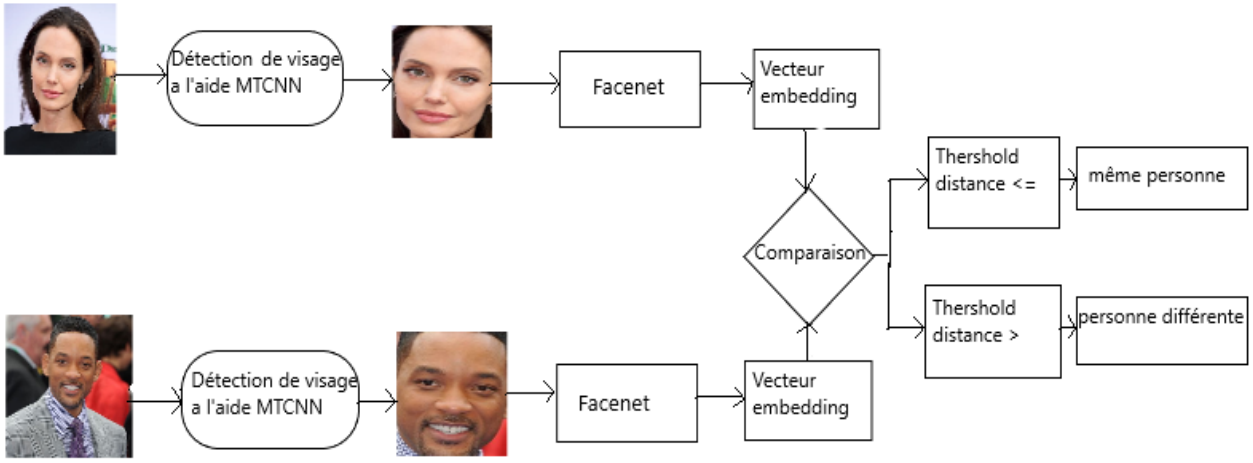

FIGURE 4.19 – classification avec MTCNN et FaceNet

La différence minimale doit être inférieure à celle du seuil à considérer lui comme une personne similaire. Lorsque la différence est plus grande, la personne est considérée comme inconnue.

•La partie 2(reconnaissance facial)

Voici l'organigramme qui se présente la partie de la reconnaissance faciale en utilisant les modèles MTCNN et FaceNet entrainé et le classifieur.

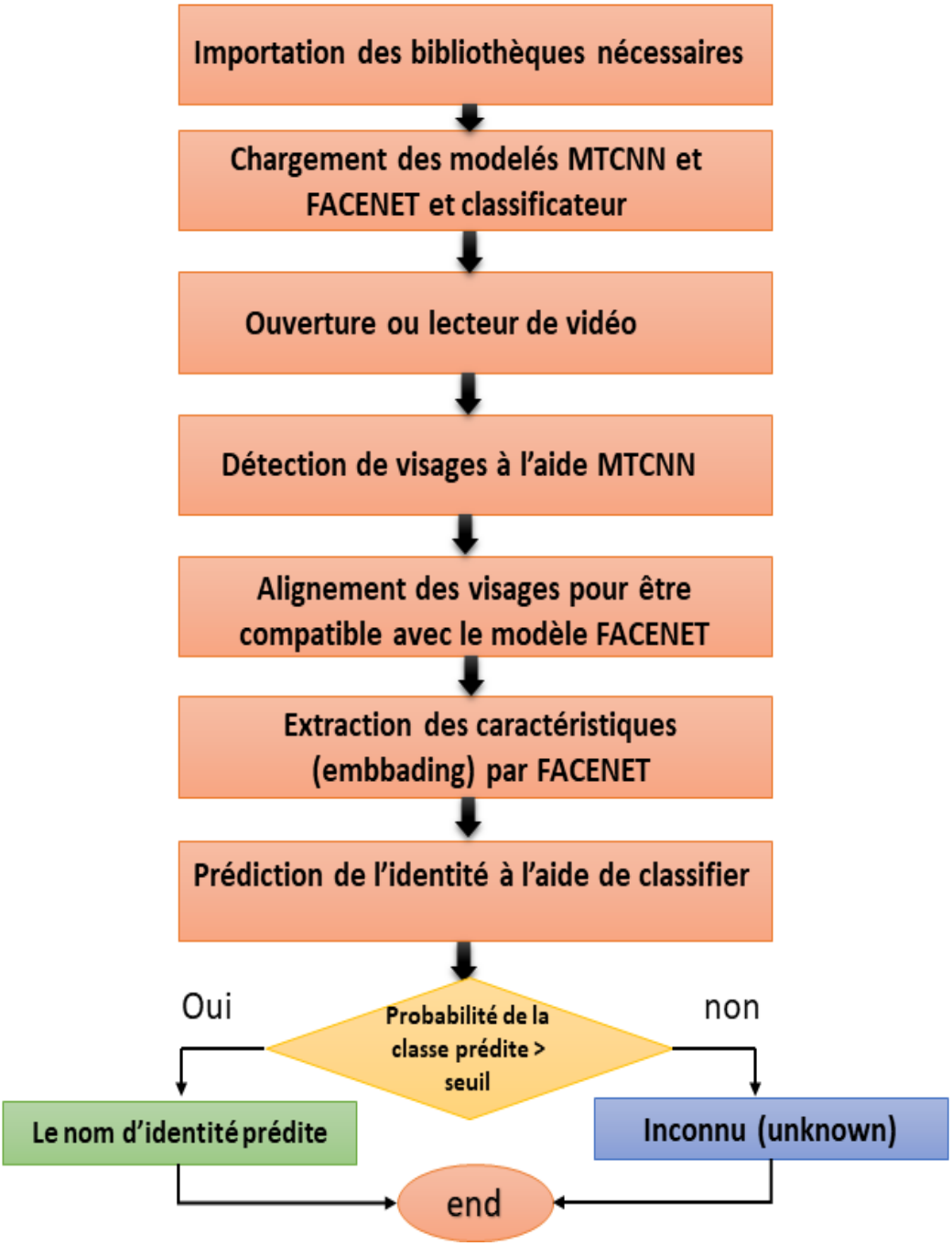

FIGURE 4.20 – Organigramme de la partie teste de reconnaissance facial

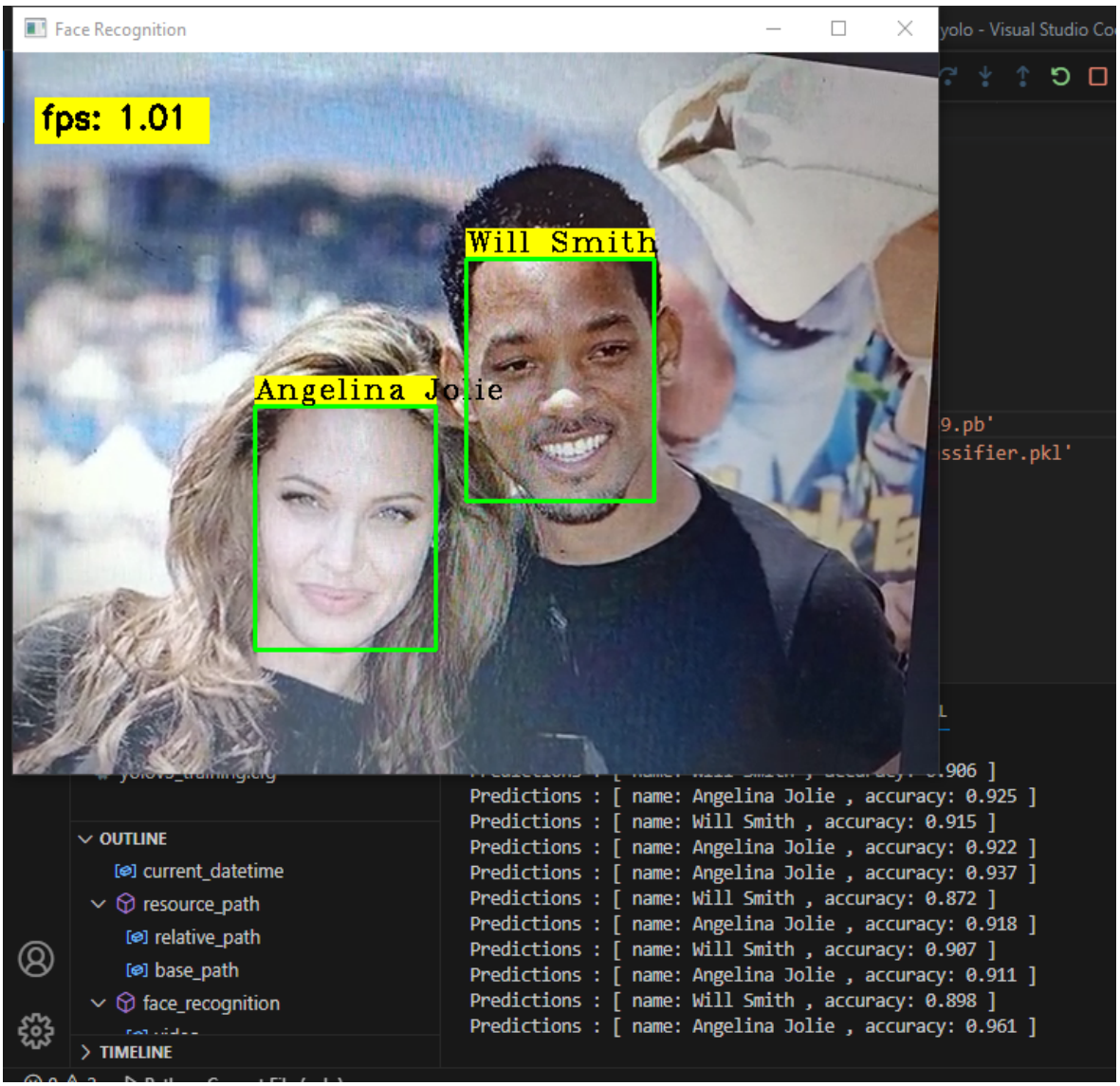

FIGURE 4.21 – test du modèle FaceNet

- **Accuracy :** indique le pourcentage de bonnes prédictions.

Dans la partie de test du modèle, les photos d'Angelina Jolie et de Will Smith ont été identifiées comme des personnes suspectes. Le modèle a effectué plusieurs prédictions avec des noms et des accuracy associées à chaque prédiction. Les accuracy allant de 0.874 à 0.943. Ces résultats démontrent que le modèle a réussi à détecter et à reconnaître les visages des personnes suspectes avec une certaine accuracy.

Un FPS de 1.01 signifie que le modèle est capable de traiter environ 1 image par seconde.

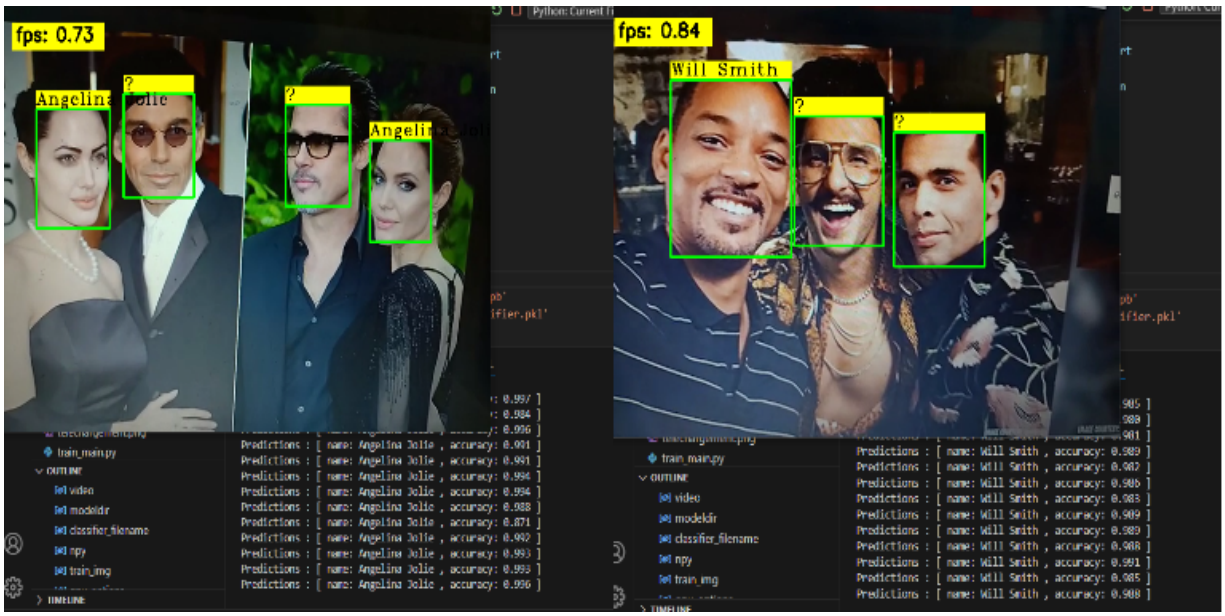

FIGURE 4.22 – teste le modele FaceNet sur plusieures personnes

Lorsque le modèle affiche "unknown" (?) pour une personne, cela signifie que le modèle n'a pas pu reconnaître ou identifier cette personne spécifique. Cela peut se produire si la personne n'est pas présente dans la base de données d'entraînement du modèle. Si le modèle n'a été entraîné qu'avec des images d'Angelina Jolie et de Will Smith, il ne pourra pas reconnaître d'autres personnes, d'où l'affichage de "unknown" pour celles-ci.

4.6 Interface graphique

Une interface graphique (GUI) est un moyen visuel d'interagir avec un programme informatique.

Le projet auquel nous avons travaillé consiste à développer une application qui combine deux fonctionnalités principales : le comptage de foule à l'aide de modèle yolov3 et la vérification des visages des perssones suspects avec le modèle FaceNet. Afin de rendre cette application conviviale et accessible, nous avons créé une interface graphique utilisant la bibliothèque Tkinter.

4.6.1 Les bibliothèques de l'interface graphique

```
import tkinter as tk
from tkinter import ttk,messagebox
import base64
from tkinter import *
import tkinter.font as font
from PIL import ImageTk, Image
from tkinter import filedialog
import sys


import cv2
import numpy as np
import os

import cv2
import numpy as np
import facenet
import detect_face
import os
import time
import pickle
from PIL import Image
import PIL.ImageColor as ImageColor
import PIL.ImageDraw as ImageDraw
import PIL.ImageFont as ImageFont
```

FIGURE 4.23 – les bibliothèques utilisées dans l'interface graphique

Il semble que nous ayons importé différents modules et bibliothèques dans notre code pour créer notre propre interface graphique. Nous avons utilisé Tkinter pour créer l'interface, ainsi que des bibliothèques telles que PIL (pad), OpenCV, numpy et facenet.

4.6.2 Interface graphique et le comptage de foule

Le comptage de foule est une partie de notre projet qui utilise YOLOv3 pour détecter et suivre les personnes dans une scène.

```
def crowd_couting():

    # Load YOLOv3 weights and configuration
    net = cv2.dnn.readNet("yolov3_training.cfg", "yolov3_training_last8_06.weights")
    classes = ["person"]
```

FIGURE 4.24 – partie de comptage de foule

La fonction crowd-couting() charge le modèle YOLOv3 pré-entraîné pour la détection de personnes dans les vidéos. Cela permet à notre interface graphique de détecter et de compter les personnes dans une foule. Les poids du modèle et sa configuration sont chargés. La classe "person" est utilisée pour la détection des personnes. Cette fonction joue un rôle clé dans notre interface graphique en fournissant la fonctionnalité de comptage de foule.

4.6.3 L'interface graphique et reconnaissance faciale

Notre interface graphique intègre une fonctionnalité de reconnaissance faciale basée sur FaceNet.

```
def face_recognition():


  video= 0
  modeldir = './model/20180402-114759.pb'
  classifier_filename = './class/classifier.pkl'
  npy='./npy'
  train_img="./train_img"
```

FIGURE 4.25 – partie de la reconnaissance faciale avec FaceNet

La fonction face-recognition() réalise la reconnaissance faciale en utilisant le modèle pré-entraîné FaceNet. Nous avons chargé le modèle sur lequel nous avons entraîné,et le classifieur, pour lui donner le modèle.

4.6.4 Intégration d'Excel dans une interface graphique de reconnaissance faciale

Dans notre code, nous avons ajouté les bibliothèques suivantes pour intégrer Excel à notre interface :

```
#add excel
import openpyxl
from openpyxl import Workbook
from openpyxl.workbook import Workbook
from openpyxl import load_workbook
from openpyxl.styles import Font
from openpyxl.styles import Alignment

import collections
import six

import tensorflow.compat.v1 as tf

import datetime
current_datetime = datetime.datetime.now()
```

FIGURE 4.26 – bibliothèque utilisé pour intégrer Excel

L'idée générale est d'utiliser les bibliothèques openpyxl, load-workbook, et Workbook pour créer, modifier et enregistrer des fichiers Excel. Les bibliothèques Font et Alignment permettent

de formater le texte et l'alignement des cellules dans le classeur Excel. Cela vous permet d'analyser et de gérer les données de notre interface graphique en utilisant Excel.

```
k = 1
  j = 1
  wb = openpyxl.Workbook()
  ws = wb.active
  ws.column_dimensions['A'].width = 15
  ws.column_dimensions['b'].width = 15
  ws.column_dimensions['c'].width = 15
  ws.column_dimensions['D'].width = 15
  ws.cell(row=1, column=1).value = "DATE"
  ws['A1'].alignment = Alignment(horizontal='center', vertical='center')
  ws.cell(row=1, column=2).value = "TIME"
  ws['B1'].alignment = Alignment(horizontal='center', vertical='center')
  ws.cell(row=1, column=3).value = "NAME"
  ws['C1'].alignment = Alignment(horizontal='center', vertical='center')
  ws.cell(row=1, column=4).value = "ACCURACY"
  ws['D1'].alignment = Alignment(horizontal='center', vertical='center')
```

FIGURE 4.27 – Utilisation d'Excel dans l'interface graphique pour la gestion des données.

4.6.5 Interface utilisateur graphique (GUI)

L'interface graphique est construite à l'aide de la bibliothèque Tkinter. L'application comprend une page d'accueil et deux boutons :

1- **Comptage des foules :** Pour compter les foules en temps réel.

2- **Reconnaissance faciale :** pour identifier les visages des personnes suspectes en temps réel.

FIGURE 4.28 – la page d'accueil d'interface graphique

A1 | fx | DATE

	A	B	C	D
1	DATE	TIME	NAME	ACCURACY
2	2023-06-24	21:59:01	Angelina Jolie	99.13%
3	2023-06-24	21:59:01	Will Smith	97.83%

FIGURE 4.29 – la liste des personnes suspectes dans Execl

4.7 Conclusion

D'après les résultats obtenus par notre système, nous pouvons résoudre le problème du comptage de la foule et identifier les visages des personnes suspectes dans la vidéo de surveillance, nous avons obtenus des bons résultats avec le modèle yolov3 pour le comptage de la foule et aussi une bonne accuracy par le modèle FaceNet pour la reconnaissance de visage des personnes suspectes.

CONCLUSION GENERALE

Dans ce projet, nous avons développé un système de sécurité basé sur la vidéo surveillance utilisant le Deep Learning, nous avons ajouté deux techniques dans ce système, la première était le comptage des foules et la seconde était la reconnaissance faciale pour détecter les personnes suspectes.

Nous avons entrainé le modèle yolov3 pour la détection et le comptage de personnes dans une zone surpeuplée, et puis nous avons aussi ré-entrainé le modèle pour FaceNet pour la reconnaissance faciale des personnes que nous avons supposé qu'il s'agissait de personnes suspecteset avec l'utilisation de modèle MTCNN pour la détection des visages. Après avoir obtenu les résultats, nous avons remarqué que les deux techniques fonctionnent bien, et les résultats étaient satisfaisants, et ils peuvent être pris en considération pour renforcer le système de sécurité dans la vidéo surveillance.

Nous avons également créé une interface utilisateur graphique pour notre application qui comprend deux fonctionnalités : le comptage de la foule et la reconnaissance faciale de personnes suspectes et la vérification la liste de présence.

Ce projet de fin d'étude nous a permis de découvrir un nouveau domaine, Le nouveau monde de la connaissance, Nous avons acquis de précieuses connaissances sur l'intelligence artificielle et le Deep Learning, et nous avons découvert de nombreuses technologies qui peuvent être appliquées dans le domaine de la vision par ordinateur. Et la force de ce domaine vaut certainement la peine d'être explorée. Bien sûr, comme pour tout projet, nous avons rencontré plusieurs obstacles tout au long du processus :
Au début nous avons entrainé le modèle yolov3 avec une base de données très large et elle n'était pas bien étiquetée, par conséquent, nous n'avons obtenu aucun résultat, c'est pour cela vous

avons préféré de créer notre base de données. La limitation d'utilisation de GPU dans google colab, c'était vraiment un problème car pour entrainer le modèle YOLO, vous avez besoin GPU c'est pour cal'apprentissage prend du temps.

En outre, un problème supplémentaire que nous avons rencontré est que nous n'avons pas inclus suffisamment d'images dans notre base de données. Par conséquent, il peut y avoir des cas où le système ne parvient pas à reconnaître les personnes dont les visages ne sont pas visibles par la caméra. Au final, nous espérons que notre travail servira de base solide pour les améliorations et développements futurs dans ce domaine de sécurit.

Nous souhaitons aussi que notre projet soit pris en compte et nous espérons qu'il sera amélioré dans l'avenir, par exemple l'amélioration de la précision du comptage de foule en entraînant le modèle sur des ensembles de données plus large, ou ajouter une partie mécanique à ce système qui active une alarme lors de la détection d'une personne suspecte, ou bien Intégrer d'autres modules de sécurité tels que la détection d'intrusion et le suivi d'objets pour créer un système complet.

Bibliographie

[1] Robert T., LIPTON, Alan J., KANADE, Takeo, et al. A system for video surveillance and monitoring. VSAM final report, 2000, vol. 2000, no 1-68, p. 1.

[2] BFM RMS ,Supermarchés : la surveillance des caissières de plus en plus répandue (bfmtv.com) ,Consulté le 29/06/2023.

[3] YESIL, Bilge. Watching ourselves : Video surveillance, urban space and self-responsibilization. Cultural Studies, 2006, vol. 20, no 4-5, p. 400-416.

[4] HAERING, Niels, VENETIANER, Péter L., et LIPTON, Alan. The evolution of video surveillance : an overview. Machine Vision and Applications, 2008, vol. 19, no 5-6, p. 279-290.

[5] Rennes Magazines,https ://rennes-magazines.fr/criteres-choix-camera-surveillance/ Consulté le 29/06/2023.

[6] FLÜCKIGER, Alexandre et AUER, Andreas. La vidéosurveillance dans l'oeil de la Constitution. AJP : Aktuelle juristische Praxis, 2006, vol. 8, p. 924-942.

[7] Le guide de la video surveillance,2014.

[8] Canon global,https ://global.canon/en/technology/count2019.html,Consulté le 29/06/2023.

[9] ADARY, Assael, DIDIER, Emmanuel, et PREVIEUX, Julien. Compter les manifestants Interview d'Assael Adary, président et fondateur du cabinet Occurrence. Statistique et Société, 2021, vol. 9, no 3, p. 95-100.

[10] LOGITHEQUE,https ://www.logitheque.com/articles/qui-a-la-plus-grosse-ces-methodes-pour-compter-les-manifestants-qui-font-polemique-16928.Consulté le 29/06/2023.

[11] KHAN, Khalil, KHAN, Rehan Ullah, ALBATTAH, Waleed, et al. Crowd counting using end-to-end semantic image segmentation. Electronics, 2021, vol. 10, no 11, p. 1293.

[12] LEI, Tao et Zhang, Dong et Wang, Risheng et Li, Shuying et Zhang, Weijiang et Nandi, Asoke. (2021). IET Image Processing MFP-Net : Multi-scale feature pyramid network for crowd counting. IET Image Processing. 15. 10.1049/ipr2.12230.

[13] BHUIYAN, Md Roman, ABDULLAH, Junaidi, HASHIM, Noramiza, et al. A deep crowd density classification model for Hajj pilgrimage using fully convolutional neural network. PeerJ Computer Science, 2022, vol. 8, p. e895.

[14] Eigenfaces : Recovering Humans from Ghosts,https ://towardsdatascience.com/eigenfaces-recovering-humans-from-ghosts17606c328184,Consulté le 16/06/2023.

[15] TURK, Matthew et PENTLAND, Alex. Eigenfaces for recognition. Journal of cognitive neuroscience, 1991, vol. 3, no 1, p. 71-86.

[16] AHONEN, Timo, HADID, Abdenour, et PIETIKÄINEN, Matti. Face recognition with local binary patterns. In : Computer Vision-ECCV 2004 : 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I 8. Springer Berlin Heidelberg, 2004. p. 469-481.

[17] HUANG, Di, SHAN, Caifeng, ARDABILIAN, Mohsen, et al. Local binary patterns and its application to facial image analysis : a survey. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2011, vol. 41, no 6, p. 765-781.

[18] TARIK, A. L. QU'EST-CE QUE L'INTELLIGENCE ARTIFICIELLE ?.

[19] FRÉDÉRIC, S. U. R. Introduction à l'apprentissage automatique. École desMines de Nancy, 2020, vol. 2021.

[20] GOODFELLOW, Ian, BENGIO, Yoshua, et COURVILLE, Aaron. Deep learning. MIT press, 2016. The MIT Press ,Cambridge, Massachusetts London, England.

[21] MACHINE Learning VS Deep Learning, https ://pythongeeks.org/deep-learning-vs-machine-learning/ ,Consulté le 29/06/2023.

[22] TOUZET, Claude. les réseaux de neurones artificiels, introduction au connexionnisme. Ec2, 1992.

[23] SAUGET, Marc. (2007). Parallélisation de problèmes d'apprentissage par des réseaux neuronaux artificiels. Application en radiothérapie externe.

[24] ZHANG, Aston, LIPTON, Zachary C., LI, Mu, et al. Dive into deep learning. arXiv preprint arXiv :2106.11342, 2021.

[25] TOUZET, Claude. les réseaux de neurones artificiels, introduction au connexionnisme. Ec2, 1992.

[26] Fronzetti, Nicola. (2019). Predictive Neural Network Applications for Insurance Processes.

[27] Convolutional Neural Network,https ://datascientest.com/convolutional-neural-network,Consulté le 29/06/2023.

[28] Real-World Applications of Convolutional Neural Networks,https ://vitalflux.com/real-world-applications-of-convolutional-neural-networks/,Consulté le 29/06/2023.

[29] SINGH, Pravendra, RAJ, Prem, et NAMBOODIRI, Vinay P. EDS pooling layer. Image and Vision Computing, 2020, vol. 98, p. 103923.

[30] BASHA, SH Shabbeer, DUBEY, Shiv Ram, PULABAIGARI, Viswanath, et al. Impact of fully connected layers on performance of convolutional neural networks for image classification. Neurocomputing, 2020, vol. 378, p. 112-119.

[31] Fully Connected (FC),https ://stanford.edu/ shervine/l/fr/teaching/cs-230/pense-bete-reseaux-neurones-convolutionnels,Consulté le 29/06/2023.

[32] ANDRYCHOWICZ, Marcin, DENIL, Misha, GOMEZ, Sergio, et al. Learning to learn by gradient descent by gradient descent. Advances in neural information processing systems, 2016, vol. 29.

[33] CHRISTOFFERSEN, Peter et JACOBS, Kris. The importance of the loss function in option valuation. Journal of Financial Economics, 2004, vol. 72, no 2, p. 291-318.

[34] Soft Computing and Intelligent Information Systems, https ://sci2s.ugr.es/es/weapons-detection,Consulté le 29/06/2023.

[35] Convolutional Neural Networks for Visual Recognition. (n.d.). Retrieved November 27, 2016, from http ://cs231n.github.io/neural-networks-1/ .

[36] CHAPEL, Marie-Neige. Détection d'objets en mouvement à l'aide d'une caméra mobile. 2017. Thèse de doctorat. Université de Lyon..

[37] ZHAO, Zhong-Qiu, ZHENG, Peng, XU, Shou-tao, et al. Object detection with deep learning : A review. IEEE transactions on neural networks and learning systems, 2019, vol. 30, no 11, p. 3212-3232.

[38] GIRSHICK, Ross, DONAHUE, Jeff, DARRELL, Trevor, 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.

[39] ZHANG, Bin, ZHANG, Yubo, et PAN, Qinghui. Irregular Target Object Detection Based on Faster R-CNN. In : IOP Conference Series : Earth and Environmental Science. IOP Publishing, 2019. p. 042111.

[40] REN, Shaoqing, HE, Kaiming, GIRSHICK, Ross, et al. Faster r-cnn : Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 2015, vol. 28.

[41] MESBAH, FETHIA. Détection d'objets par Deep Neural Network à l'aide du modèle YOLO en temps réel. 2021.

[42] LIU, Wei, ANGUELOV, Dragomir, ERHAN, Dumitru, 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 International Publishing, 2016. p. 21-37.

[43] REDMON, Joseph, DIVVALA, Santosh, GIRSHICK, Ross, et al. You only look once : Unified, real-time object detection. In : Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 779-788.

[44] MESBAH, FETHIA. Détection d'objets par Deep Neural Network à l'aide du modèle YOLO en temps réel. 2021.

[45] REZATOFIGHI, Hamid, TSOI, Nathan, GWAK, JunYoung, et al. Generalized intersection over union : A metric and a loss for bounding box regression. In : Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019. p. 658-666.

[46] Jain, Harshil et Nandy, S. (2019). Incremental Training for Image Classification of Unseen Objects. 10.13140/RG.2.2.10266.47046.

[47] What's new in YOLO v3 ?,https ://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b,Consulté le 29/06/2023.

[48] REDMON, Joseph et FARHADI, Ali. Yolov3 : An incremental improvement. arXiv preprint arXiv :1804.02767, 2018.

[49] Understanding Anchors(backbone of object detection) using YOLO,https ://becominghuman.ai/understanding-anchors-backbone-of-object-detection-using-yolo-54962f00fbbb,Consulté le 29/06/2023.

[50] Identifying Faces with MTCNN and VggFace,https ://medium.com/swlh/identifying-faces-with-mtcnn-and-vggface-9d0d4927cccf,Consulté le 29/06/2023.

[51] DeepFace vs Facenet for face recognition,https ://projectsflix.com/machine-learning/deepface-and-facenet-for-face-recognition/,Consulté le 29/06/2023.

[52] NAKADA, Masaki, WANG, Han, et TERZOPOULOS, Demetri. AcFR : active face recognition using convolutional neural networks. In : Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2017. p. 35-40.