République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Saad Dahlab Blida

Mémoire de Master

Filière : Informatique
Spécialité : Sécurité des systèmes d'information

Sécurisation de réseau et prévention de pertes des données en exploitant la reconnaissance faciale via le Deep Learning

Sujet proposé par :
Promoteur : Mr.BENYAHIA Mohammed
Encadrant : Mr. MELLAS Salim

Présenté par :
HAROUZ Randa
GUESSOUM Lynda

Soutenu le : 29/09/2022

Devant le jury composé de :
Président : Mr. OULD KHAOUA Mohamed
Examinatrice: Mme GHEBGHOUB Yasmina
Promoteur : Mr.BENYAHIA Mohammed

Année Universitaire
2021/2022

Remerciements

Dieu merci de nous avoir donné la santé et le courage pour finir ce modeste travail.

En premier, je tiens à remercier mes encadrants Mr. MELLAS Salim et Mr. BENYAHIA Mohamed pour leur orientation, confiance et patience qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené à bon port.

Je tiens à exprimer mes sincères remerciements à tous les professeurs qui m'ont enseigné et qui par leurs compétences m'ont soutenu dans la poursuite de mes études.

Un salut particulier à tous les membres de jury qui m'ont honoré par leur présence et leur acceptation d'évaluer ce travail.

Enfin, je remercie tous ceux qui, de près ou de loin, ont contribué à la réalisation de ce travail.

Merci à tous et à toutes

DEDICACE

A ceux qui me sont les plus chers au monde,

qui n'ont jamais cessé de m'encourager, de prier pour moi,

et qui m'ont toujours entouré de toute leur affection et leur amour.

Que Dieu vous protège, chers parents.

A tous mes frères et sœurs.

A mon binôme Randa.

A mes amis et collègues.

A Tous ceux qui me sont chers,

Je dédie ce modeste travail.

Lynda

DEDICACE

Je dédie ce travail

A ma famille, elle qui m'a doté d'une éducation digne, son amour a fait de moi ce que je suis aujourd'hui :

Particulièrement à mon père HOUCINE, pour le gout à l'effort qu'il a suscité en moi, de par sa rigueur.

A ma mère HOUDA, ceci ma profonde gratitude pour ton éternel amour, que ce rapport soit le meilleur cadeau que je puisse t'offrir.

A vous ma sœur LINA et mon frère ZAKARIA qui m'avez toujours soutenu et encouragé durant ces années d'études. Puisse Dieu vous donne santé, bonheur, courage et surtout réussite

A mon mari ZERROUKI MOHAMED, qui a été à mes côtés pour me soutenir et m'encourager.

A mon collègue MOULOUD IBRAHIM qui m'aider durant toute la période du travail.

Sans oublier mon binôme LYNDA pour son soutien moral, sa patience sa compréhension tout au long de ce projet.

Randa

Résumé

La tâche de reconnaissance faciale consiste à reconnaître les visages dans les images, tandis que la détection d'objets consiste à déterminer l'emplacement des objets dans les images. Pour atteindre cet objectif, nous avons développé un modèle capable de détecter des visages ainsi que les reconnaître. Le modèle YOLO (You Only Look Once) a été utilisé pour détecter les visages dans l'image. Si une personne est détectée par le modèle, une image recadrée du visage de la personne est transmise au modèle CNN. Le réseau siamois identifie la personne en se référant à la base de données des personnes connues. En ajoutant le réseau siamois, le cadre devient plus évolutif et adaptable.

ملخص

تتمثل مهمة التعرف على الوجه في التعرف على الوجوه في الصور، بينما يتمثل اكتشاف الكائنات في تحديد موقع الكائنات في الصور. ولتحقيق هذا الهدف، قمنا بتطوير نموذج يمكنه اكتشاف الوجوه وكذلك التعرف عليها. تم استخدام نموذج YOLO (أنت تنظر مرة واحدة فقط) للكشف عن الوجوه في الصورة. إذا تم اكتشاف شخص بواسطة النموذج، يتم نقل صورة مقتصرة لوجه الشخص إلى نموذج .CNN وتحدد الشبكة السيامية هوية الشخص بالرجوع إلى قاعدة بيانات الأشخاص المعروفين. من خلال إضافة شبكة سيامي، يصبح الإطار أكثر قابلية للتطوير والتكيف.

Abstract

The task of facial recognition is to recognize faces in images, while object detection is to determine the location of objects in images. To achieve this goal, we have developed a model that can detect faces as well as recognize them. The YOLO (You Only Look Once) model was used to detect faces in the image. If a person is detected by the model, a cropped image of the person's face is transmitted to the CNN model. The Siamese network identifies the person by referring to the database of known persons. By adding the Siamese network, the framework becomes more scalable and adaptable

SOMMAIRE

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABRÉVIATIONS

IA : Intelligence Artificielle

ML: Machine Learning

CNN: Convolutional Neural Network

IP: Internet Protocole

URL : Uniform Resource Locator

DOS : Denial of Service

DDOS : Distributed Denial of Service

SYN : Synchronize

SYN ACK : Synchronize Acquittement

ICMP : Internet Control Message Protocol

ACP : Analyse en Composantes principales

MLP : Multi layer Perceptron

FC : Fully Connected

ReLU : Rectified Linear Unit

R CNN : Regions with Convolutional Neural Networks

SVM : Support Vector machines

SSD : Single-Shot Detector

YOLO : You Only Look Once

GPU : Graphics Processing Unit

DPM : Deformable Parts Model

CSPNet : Cross Stage Partial Network

PANet : Path Aggregation Network

SiLU : Sigmoid Linear Unit

UML: Unified Modeling Language

MVC : Model View Controller

URL: Uniform Resource Locator

JS: JavaScript

API : Application Programming Interface

DOM : Document Object Model

IDE : Integrated Development Environment

CPU: Central Processing Unit

INTRODUCTION GÉNÉRALE

INTRODUCTION GÉNÉRALE

Les violations de données se produisent depuis aussi longtemps que les entreprises tiennent des registres et stockent des données privées. Mais la prolifération des données, les progrès technologiques et la numérisation du stockage des données au cours de la dernière décennie ont entraîné une forte augmentation du nombre de violations de données. Tirant parti de ces développements, la cybercriminalité évolue. Les criminels ont tourné leur attention vers le ventre mou des entreprises : les données stockées.

Aujourd'hui, le risque de perdre ou d'exposer des informations sensibles est plus grand que jamais. Et les entreprises en ressentent les effets : toutes les plus grandes violations de données de l'histoire se sont produites depuis 2005 et la taille et la portée de ces violations ne font que s'agrandir de jour en jour.

En fait, la plupart des violations de données réussies se produisent en moins d'une minute. Pourtant, la majorité des entreprises mettent des semaines à se rendre compte qu'une violation s'est produite. Les données compromises ont de nombreuses conséquences coûteuses. C'est pourquoi les dirigeants d'entreprise pensent que les menaces à la cybersécurité, telles que la faible sécurité des données, sont préoccupantes. Une perte de revenus importante à la suite d'une violation de la sécurité est courante. Des études montrent que les entreprises confrontées à une violation de données finissent par perdre des revenus.

Les pertes financières ne sont pas la seule préoccupation des entreprises du secteur financier ; les violations de données diminuent la confiance des utilisateurs et peuvent ternir la réputation d'une entreprise. Cependant, les clients attachent également de l'importance à leur vie privée, et les violations impliquent souvent des informations de paiement des clients. Les prospects potentiels hésiteront à faire confiance à une entreprise ayant des antécédents de sécurité des données de mauvaise qualité.

Les informations d'identification compromises étant le point d'entrée le plus courant pour les violations de données. Les pirates ont de nombreuses façons d'obtenir des informations d'identification compromises, y compris des attaques de phishing qui incitent les utilisateurs à partager les informations de leur compte.

L'authentification joue un rôle central dans le renforcement de la posture de sécurité d'une organisation. Elle permet à une organisation de sécuriser ses systèmes en autorisant

uniquement les utilisateurs (ou processus) authentifiés à accéder aux ressources protégées, telles que les systèmes informatiques, les réseaux, les bases de données, les sites Web et d'autres applications ou services basés sur le réseau.

Les mots de passe sont la forme d'authentification la plus traditionnelle, mais ils constituent une forme de protection faible ; les utilisateurs sont sujets à de mauvaises pratiques de mot de passe, telles que la réutilisation de mots de passe, l'utilisation de mots de passe prévisibles ou même le partage de mots de passe avec d'autres. Pour lutter contre ce problème, de plus en plus d'entreprises tirent parti des technologies d'intelligence artificielle (IA) et d'apprentissage automatique (ML), telles que les techniques basées sur l'apprentissage profond, pour développer des approches d'authentification meilleures et plus sécurisées, il a été démontré que les algorithmes IA/ML renforcent la cybersécurité en protégeant les appareils contre les cyberattaques et en empêchant les activités frauduleuses. Dans ce mémoire, nous présentons une solution d'IA basée sur l'apprentissage profond et la vision par ordinateur pour effectuer une authentification biométrique basée sur la reconnaissance faciale.

La technologie de reconnaissance faciale n'est pas nouvelle. La plupart d'entre nous utilisent de nos jours des téléphones intelligents, qui utilisent souvent la technologie de reconnaissance faciale pour déverrouiller l'appareil. Cette technologie offre un moyen puissant de protéger les données personnelles et de garantir que même si le téléphone est volé, les données sensibles restent inaccessibles par l'auteur. L'utilisation de la technologie de reconnaissance faciale est appliquée à un ensemble de domaines en constante expansion, notamment la sûreté, la sécurité et les paiements.

La reconnaissance faciale est un vaste problème d'identification ou de vérification d'un visage dans des images numériques ou des images vidéo grâce au modèle et aux données biométriques du visage. La technologie collecte un ensemble de données biométriques uniques de chaque personne associée à son visage et à son expression faciale pour authentifier une personne. La technologie de reconnaissance faciale est principalement utilisée pour deux types de tâches :

Vérification faciale : étant donné une image de visage, faites-la correspondre à des images connues dans une base de données sécurisée, donnez une décision oui/non. La personne existe-t-elle dans la base de données ?

Identification du visage : à partir d'une image de visage, associez-la à des images connues dans une base de données sécurisée, détectez à qui appartient cette image (par exemple, qui est cette personne ?).

L'objectif de ce projet consiste à développer un système de sécurité basé sur la reconnaissance faciale qui répond à des besoins de sécurité et de contrôle d'accès des employés d'une société quelconque. Dans notre cas la Société nationale pour la recherche, la production, le transport, la transformation, et la commercialisation des hydrocarbures SONATRACH où nous avons effectué notre stage de fin d'études.

Nous nous sommes intéressés par l'étude d'un type d'algorithmes de Deep Learning, les réseaux de neurones convolutionnels. Les CNNs sont très utiles pour les tâches de détection et classification d'objets. Nous présenterons les bases des CNNs, les différentes architectures disponibles et leurs utilisations pour des tâches de classification d'images.

Le présent manuscrit est scindé en cinq chapitres. Le premier chapitre est composé de deux sections ; nous exposons dans la première les notions basiques des réseaux informatiques en général et pour la deuxième nous parlerons de la sécurité des réseaux. Dans le deuxième chapitre, nous introduisons la vision par ordinateur et ses défis avec des fondements en imagerie numérique. Le troisième chapitre sera consacré à la présentation du deep learning et les réseaux de neurones convolutionnels ainsi que des architectures de détections d'objets. Dans le quatrième chapitre nous proposerons la conception de notre système. Enfin, le cinquième chapitre fera l'objet d'une implémentation suivie par des discussions des résultats obtenus.

CHAPITRE I : LA SÉCURITÉ DES RÉSEAUX INFORMATIQUES

CHAPITRE I : LA SÉCURITÉ DES RÉSEAUX INFORMATIQUES

I.1 Introduction

Aujourd'hui, la plupart des organisations s'appuient fortement sur les réseaux informatiques pour partager les informations de façon efficace et efficiente au sein d'un réseau défini. Généralement, le réseau informatique d'une organisation est très grand, et si chaque employé dispose un poste de travail, alors une grande entreprise dispose des milliers de postes de travail.

Imaginez sur un réseau d'entreprise, des milliers de postes de travail connectées directement à internet. Cela expose l'entreprise à un environnement non sécurisé où les cyberattaques sont en constante évolution. Pour protéger leur intégrité et préserver leur fonctionnalité, la sécurité réseau comprend toutes les activités qui aident à résoudre ce problème.

Dans ce premier chapitre, nous aborderons la sécurité des réseaux informatique et les différents aspects liés à la sécurité des réseaux informatiques : définitions, risques, attaques et quelques mécanismes de défense contre les cyberattaques.

I.2 Définition de la sécurité informatique

La sécurité informatique est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires qui assurent que les ressources du système d'information (matérielles ou logicielles) d'une organisation sont uniquement utilisées dans le cadre prévu.

I.3 Les principes de sécurité

La sécurité informatique se repose sur cinq principaux objectifs :

- **La confidentialité**

Le principe de confidentialité assure que seuls les utilisateurs autorisés ont accès à l'information.

- **L'intégrité**

Le principe d'intégration garantit que le système et l'information ne soient modifiés que par les utilisateurs autorisés.

•La disponibilité

Le principe de disponibilité assure le bon fonctionnement du système et la disponibilité des ressources à tout moment pour les personnes autorisées.

•L'authentification

Le principe d'authentification exige une preuve d'identité, il garantit que l'origine de l'information ainsi que les différents utilisateurs sont correctement identifiées.

•La non répudiation

Le principe de non répudiation garantit que aucun propriétaire d'une information ne peut contester les opérations qu'il a réalisé dans le cadre de ses actions autorisées.

I.4 Définition de la sécurité des réseaux

La sécurité du réseau comprend les politiques et les pratiques adoptées pour empêcher et surveiller l'accès non autorisé, L'utilisation abusive, la modification ou le refus d'un réseau informatique et des ressources accessibles sur le réseau. [1]

I.5 Risques sur la sécurité des réseaux

I.5.1 Terminologies

• **Vulnérabilité :** est une faiblesse de sécurité, le plus souvent cachée. Elle permette à l'attaquant de porter atteinte à l'intégrité d'un système. Elle découle généralement d'une erreur dans la conception d'un système d'information, d'un logiciel ou d'une erreur dans l'utilisation d'un composant matériel.

• **Menace :** une menace est une cause potentielle ou un évènement qui trouble le bon fonctionnement d'un système d'information. Elle désigne l'exploitation d'une faille de sécurité par un hacker. Elle peut causer des graves dommages à un système informatique. Cette menace peut être accidentelle, intentionnelle comme elle peut être active ou passive.

• **Risque :** C'est la probabilité qu'une vulnérabilité ait un impact sur les actifs et les ressources d'un système.

• **Attaque :** une attaque informatique est un acte offensif mené au moyen d'un réseau informatique dans le but de causer des dommages. C'est l'attaquant qui exploite les failles d'un système à des fins non connues. Les cyberattaques peuvent cibler plusieurs équipements informatiques : des serveurs ou des ordinateurs et des équipements périphériques comme les imprimantes.

I.5.2 Logiciels malveillants

• **Virus :** est un programme malveillant de petite taille qui s'attache à un logiciel pour perturber et endommager un appareil, il est capable de se propager automatiquement. Les virus informatiques ont un large éventail d'effets : il peut ralentir un ordinateur ou détruire carrément les données ciblées.

• **Cheval de Troie (trojan horse) :** est un logiciel malveillant avec une apparence légitime, il exécute des instructions nocives sans l'autorisation de l'utilisateur. En revanche, il n'est pas capable de se propager ou de se répliquer, il est classé dans la catégorie des malwares.

• **Vers (Worms) :** C'est un type de virus qui est capable de se propager en utilisant un réseau informatique. Mais contrairement au virus, le ver est un logiciel malveillant qui s'auto-reproduit d'un ordinateur à un autre sans intervention de l'utilisateur, pour objectif de détruire les données et espionner les appareils dans lesquels il réside. Il peut également réaliser une attaque de déni de service.

• **Spyware :** est un logiciel d'espion invisible, il se comporte sur des appareils (ordinateur, smartphone…) dans le but de collecter des données personnelles (mot de passe, données de formulaire) et les transférer à son concepteur.

• **Spam :** désigne les emails anonymes et indésirables envoyés en grande quantité provenant des utilisateurs inconnus

• **Porte dérobée (backdoor) :** C'est un moyen de contourner les mécanismes de contrôle d'accès. Elle s'agit d'une faille du système de sécurité due à une faute de conception accidentelle ou intentionnelle. [2]

I.5.3 Les différents types d'attaque réseau

I.5.3.1 Les attaques de reconnaissance

Les attaques de reconnaissance visent à collecter une grande quantité d'informations possible pour des futures attaques. Ces attaques recherchent des informations pour identifier les vulnérabilités et les faiblesses dans le réseau ciblé. Dans ce type d'attaque, les acteurs malveillants peuvent utiliser des outils internet comme les commandes « nslookup » et « whois » pour découvrir certaines informations : les adresses IP utilisées, les propriétaires d'un domaine…Une fois ces informations connues, l'attaquant peut lancer des pings à tous les adresses IP de réseau ciblé.

I.5.3.2 Les attaques du mot de passe

Dans ce type d'attaques, l'utilisateur malveillant essaie tous les moyens pour deviner le mot de passe da sa victime. Parmi ces moyens, on distingue les plus utilisés :

- **L'attaque par dictionnaire :** L'attaquant teste une série de mots et de phrases souvent utilisés comme un mot de passe.
- **L'attaque par force brute :** C'est une ancienne méthode consiste à tester toutes les combinaisons possibles (lettres, chiffres, symboles) à l'aide d'un logiciel automatisé pour trouver un mot de passe.

I.5.3.3 Les attaques d'accès

Le but de ces attaques est de voler les informations sensibles d'un réseau informatique. On distingue les méthodes les plus utilisées dans ce type.

- **Le phishing :** C'est une technique d'ingénierie sociale, elle consiste à envoyer des emails non sollicités avec des fausses URL dans le but d'inciter la victime à envoyer des informations personnelles.
- **Le pharming :** est souvent appelé « phishing sans leurre », il s'agit d'un code malveillant qui détourne l'accès d'un site légitime vers un site falsifiés.
- **Man-in-the-middle:** est une cyberattaque ou l'attaquant intercepte toute communication ou un transfert de données entre deux entités. L'objectif de cette attaque est de voler des données précieuses qui peuvent être utilisées pour des activités illégales

I.5.3.4 Les attaques du réseau contre la disponibilité

- **Les attaques DOS :** Denial Of Service ou attaques par déni de service, ayant pour but de perturber le fonctionnement d'un serveur web et le rendre

indisponibles pour les utilisateurs. Nous pouvons diviser ces attaques en deux catégories :

- Les dénis de services par saturation : il s'agit d'inonder des machines par des fakes requêtes pour qu'elles ne puissent plus répondre aux vraies requêtes.
- Les dénis de services par exploitation des vulnérabilités : ça inclut l'exploitation de failles d'un système cible pour le rendre inutilisable.

- **Les attaques DDOS :** est une version distribuée d'une attaque par déni de service DOS, provenant de plusieurs machines en même temps contrôlés par le pirate, le principe de ces attaques repose sur les méthodes suivantes :
- L'attaque SYN flood : Le pirate envoie plusieurs paquets SYN au système cible sans rependre aux accusés de réception SYN-ACK.
- L'attaque ICMP flood : Un hacker envoie de nombreux faux paquets ICMP via plusieurs appareils pour surcharger la bande passante de la cible.

I.5.3.5 Les attaques rapprochées

Dans ce type d'attaque, le pirate profite de l'avantage d'être proche physiquement de la victime pour réaliser ses actions malveillantes.

I.5.3.6 Les attaques de la relation d'approbation

Pour avoir plus de contrôle, l'attaquant exploite la relation d'approbation entre une machine cible et les différents périphériques d'un réseau.

I.6 Conclusion

La sécurité du réseau est définie comme le processus de création d'une approche défensive stratégique qui sécurise les données et les ressources d'une entreprise sur son réseau. Dans ce premier chapitre, nous avons vu les différents aspects liés à la sécurité. Dans le chapitre suivant, nous abordons les deux domaines de l'intelligence artificielle : la vision par ordinateur et la reconnaissance facial.

CHAPITRE II : VISION PAR ORDINATEUR

CHAPITRE II : VISION PAR ORDINATEUR

II.1 Introduction

L’espace autour de nous a une structure tridimensionnelle 3D. Lorsqu'on demande à une personne d'écrire ce qu'elle voit, elle n'a aucun problème à nommer les objets qui l'entourent : téléphones, livres, etc. La rétine guide à la fois les images ou les informations encodées. Le processus visuel construit la perception.

La vision est une source d'informations très riche, et les machines ont le potentiel de se déplacer et d'explorer leur environnement afin qu'elles puissent reconnaître et s'adapter, et ce par le biais de plusieurs techniques et algorithmes.

Les visages sont l'un des moyens les plus simples pour différencier entre les individus. La reconnaissance faciale est un système d'identification personnel qui utilise les caractéristiques personnelles d'une personne pour les identifier. Le processus de reconnaissance du visage humain se compose de deux phases : la détection du visage puis la reconnaissance de visage en tant qu'individu. C’est l'un des biométries les plus étudiés et développé par les experts. L’identification d’une personne est une tâche facile pour les humains. Néanmoins ceci n’est pas pareil pour les machines, ce qui a engendré un grand nombre de travaux de recherche au cours des dernières années pour que la machine puisse reconnaitre et classifier les gens par des images numériques capturés, que nous allons présenter dans ce chapitre.

II.2 Vision par ordinateur

II.2.1 Définition

La vision par ordinateur est une technologie d’intelligence artificielle permettant aux ordinateurs d’imiter les êtres humains.

La vision permet aux machines de reconnaître des objets et des personnes. Cela permet un contrôle autonome avec un faible coût et une consommation d'énergie raisonnable. De plus, le traitement d'images et la vision par ordinateur utilisent les connaissances et les techniques de l'intelligence artificielle pour s'adapter à des environnements hostiles et

changeants, répondre aux exigences de prise de décision possible lors du déplacement de machines, de flexibilité et de sécurité.

II.2.2 Une théorie de la vision

La théorie scientifique passe par 3 étapes :

- ✓ Enoncer la théorie (le concept de base) ;
- ✓ Exprimer ces concepts de base sous forme mathématiques ;
- ✓ Réaliser un ensemble expérimental qui permet de vérifier la théorie.

II.2.3 Fondements de l'image numérique

Le terme image numérique, dans son sens le plus général, désigne toute image obtenue, traitée et sauvegardée sous une forme codée qui peut être représentée par des nombres (valeurs numériques). [4]

Un signal bidimensionnel 2D borné et discret : c'est une image obtenue, créée, traitée et conservée sous forme de nombres à codage numérique (binaire) et géométrique appelés pixels, l'unité principale de mesure de la définition de l'image. Chaque pixel est localisé par deux coordonnées X et Y dans le système de coordonnées de l'image. La numérisation d'une image passe du physique au numérique [5].

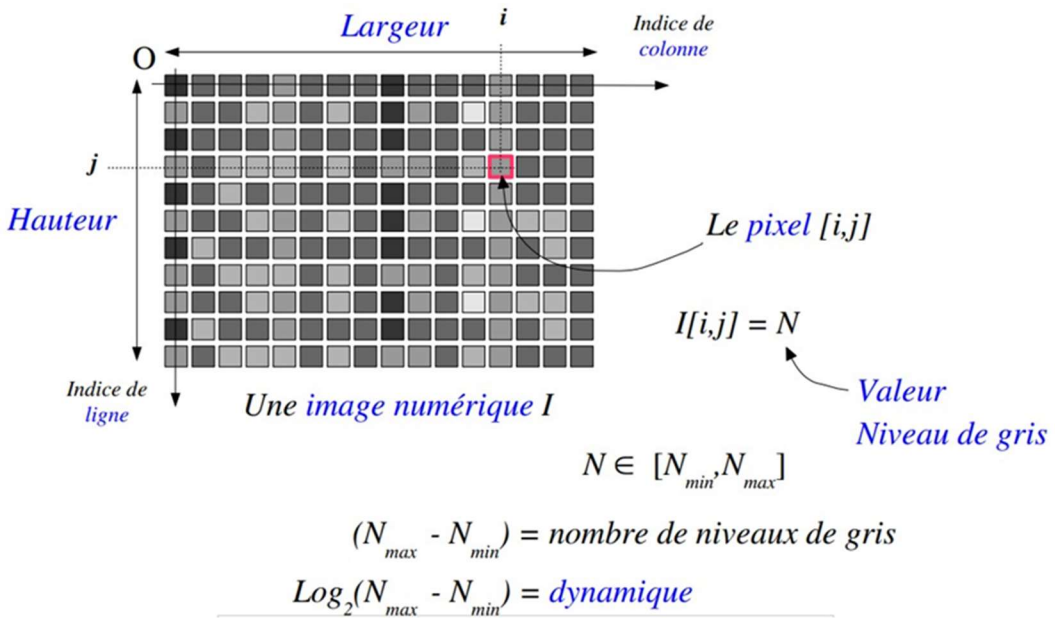

Figure II.2-1 : Pixels et niveaux de gris

II.2.3.1 Types d'images numériques

II.2.3.1.1 Image matricielle (bitmap)

Une image raster (mode pointillé, ou "bitmap" en anglais) est une image couramment utilisée pour le rendu de photographies numériques. Ils sont basés sur une grille de plusieurs pixels qui forment une image avec une définition très précise.

Lorsqu'on les agrandi trop, on perd de la qualité «pixellisation».

- **La définition :** définie en pixels sa hauteur H multipliée par sa largeur L exprimées en pixels.
- **La taille :** c'est la place nécessaire au stockage de l'image avec la règle suivante : Taille=Nbr d'octets par pixel X définition (en octets).
- **Le codage :** il s'agit du nombre de couleurs que peut prendre chaque pixel (en bits).

II.2.3.1.2 Image vectorielle

Son principe est de représenter des données d'image par des formes géométriques décrites d'un point de vue mathématique.

Son avantage c'est qu'elle peut être facilement redimensionnée. Son codage dépend directement du logiciel de sa création. [6]

II.2.3.2 Traitement d'image numérique

C'est une procédure qui permet l'extraction de l'information utile de l'image, l'amélioration, redimensionnement et éventuelle transformation (luminosité, netteté, etc.).

II.2.3.3 Niveaux de traitement d'image

Le traitement d'images peut être classé en trois niveaux différents comme montré dans la figure ci-dessous :

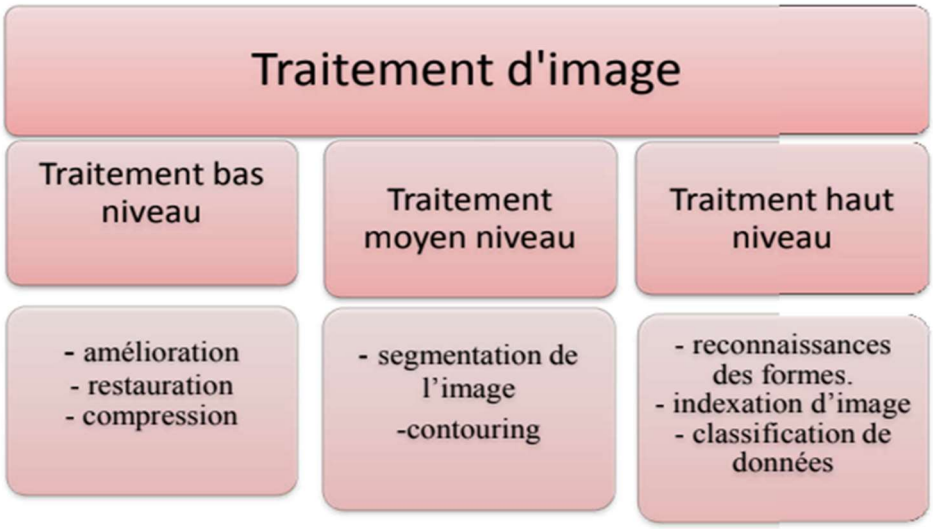

Figure II.2-2 : Niveaux traitement d'image

II.2.3.3.1 Traitement bas-niveau

Ce processus nous permet d'extraire des informations utiles (le contraste, la couleur, la direction et l'orientation de l'image). Il opère sur des données numérique, il modifie l'image sans trop la dégrader, comme il donne aussi une description structurelle de l'image non corrélé avec le contexte de la scène réelle.

Ces traitements comprennent :

- **Amélioration** : modification de l'image pour une meilleure visualisation.

- Restauration : correction des défauts causé par une source de dégradation.

- **Compression :** réduit le volume de l'image.[7]

Figure II.2-3:Traitement bas niveau compression.

II.2.3.3.2 Traitement moyen niveau (segmentation)

La segmentation d'images consiste à regrouper les pixels de ces images qui partagent les mêmes propriétés pour former des régions connexes. Il existe de nombreuses méthodes de segmentation qui se répartissent en quatre classes principales :

- ✓ Segmentation fondée sur les régions.
- ✓ Segmentation fondée sur les contours.
- ✓ Segmentation fondée sur la classification ou le seuillage des pixels selon leurs intensités.
- ✓ Segmentation fondée sur la coopération.

Entre les trois premières segmentations. Ils existent deux approches principales :

• **Approche (contour)** : les régions sont délimitées par les contours des objets qu'elles représentent (séparation).

• **Approches (région)** : les régions sont déterminées en fonction de leurs propriétés intrinsèques (agrégation de pixels avec le critère d'homogénéité)

Figure II.2-4: Traitement moyen niveau-contour.

II.2.3.3.3 Traitement haut niveau

Il s'agit d'une technique qui concerne des propriétés émergentes telles que la classification et la reconnaissance. Son rôle est la reconnaissance sémantique et l'interprétation des scènes par des techniques d'intelligence artificielle qui modélise le plus souvent les connaissances humaines à travers des modèles appropriés. [6]

• **Reconnaissances des formes :** est une identification du contenu de l'image, corrélation bidimensionnelle et normalisée.

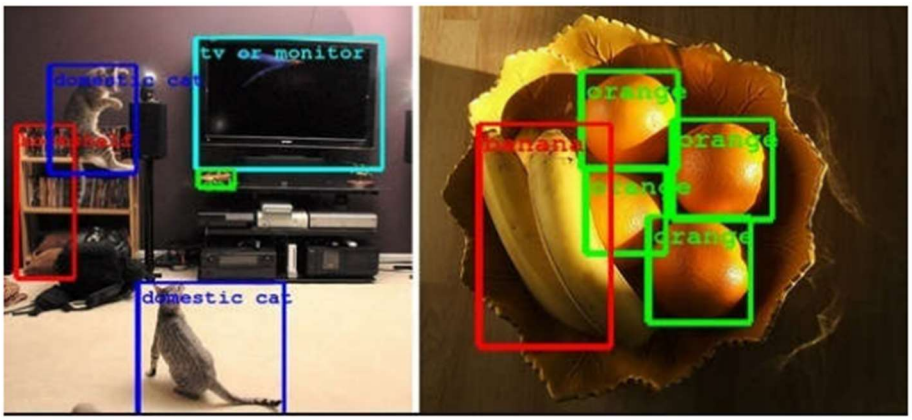

Figure II.2-5: Reconnaissance des formes.

• **Indexation d'image et recherche** : recherche dans l'image des caractérisations du contenu de l'image.

• **Classification de données :** cela comprend l'extraction de caractéristiques (mesures effectuées sur des objets) et les outils de classification (définition de critère de mesures).

II.2.3.4 Applications de traitement d'images

On retrouve le traitement d'images dans plusieurs domaines, à savoir : Sécurité et surveillance notamment détection d'intrusion et reconnaissance faciale, Vision industrielle (contrôle de marchandise), Imagerie médicale (détection des tumeurs cancérogène), et encore l'instrumentation en développement de capteurs et en aviation (système embarqué).

II.3 Reconnaissance faciale

II.3.1 Historique

La reconnaissance faciale automatique est un nouveau concept, le premier système de la reconnaissance faciale semi-automatique a été développé dans les années 1960. L'administrateur doit localiser les yeux, la bouche, le nez et les oreilles sur l'image et puis saisir les distances et les rapports calculés par rapport à un point de référence commun et les comparés aux données de référence. Ce problème a été abordé pour la première fois par Francis Galton en 1988. Il a proposé de collecter les traits du visage sous forme de courbes et de trouver leurs critères et de classer les nouveaux profils en fonction de leur dérivation à partir de ces critères. Depuis, une multitude d'approches et de multiples méthodes ont été

développées pour résoudre ce problème. En 1991, Turk et Pentland ont découvert que la technique Eigenfaces (ACP) peut être utilisée pour détecter des visages dans des images à l'aide de résidus, cette technologie a été testée lors de la finale du championnat de football américain ''SUPER BOWL'' de janvier 2001 en capturant des images de surveillance et en les comparant à une base de données numérique.[8]

II.3.2 Définition et problématique de reconnaissance faciale

La reconnaissance faciale est une catégorie de la sécurité biométrique et un moyen d'identifier ou de confirmer l'identité d'un individu à travers son visage. Les systèmes de reconnaissance faciale peuvent être utilisés pour identifier des personnes sur des photos, des vidéos numériques au sein de nombreuses applications notamment la surveillance. [9]

Étant donnée une ou plusieurs images d'un visage, la tâche consiste à trouver ou à vérifier l'identité d'une personne en comparant son visage à un ensemble d'images stockées dans une base de données. La reconnaissance faciale relève deux défis :

- ✓ Détection de visage en temps réel.
- ✓ Détection de visage dans les images. [10]

Un visage peut être défini comme : « Une structure tridimensionnelle avec une configuration 'externe'. Les contours du visage sont modelés par des crêtes osseuses et soulignés par des cheveux. Il est gravé d'une composition "interne" formée par un ensemble de traits. .Il y a aussi certaines caractéristiques comme les cheveux, les lunettes, les textures, la couleur de la peau..." [11]

La détection de visage dans les images est un processus essentiel et important avant la phase de reconnaissance. En effet, le processus de reconnaissance faciale ne peut être entièrement automatisé que s'il est précédé d'une étape de détection efficace. Les systèmes de reconnaissance faciale peuvent être divisés en deux grandes catégories selon la source de capture d'image : reconnaissance du visage dans une séquence vidéo ou bien à partir d'images fixes.

D'autres formes de logiciels biométriques incluent la reconnaissance de la voix, des empreintes digitales, de la rétine ou de l'iris. Cette technologie est principalement utilisée à

des fins de sécurité et d'application de la loi, bien que d'autres domaines s'y intéressent de plus en plus.

II.3.3 Étape de la reconnaissance Faciale

De nombreuses personnes connaissent la reconnaissance faciale en utilisant FaceID pour déverrouiller leur smartphone. Cette dernière ne repose pas sur une énorme base de données de photos pour identifier une personne : elle ne fait qu'identifier et reconnaître une personne comme propriétaire unique de l'appareil, pour empêcher autrui d'accéder à l'appareil.

Au-delà du déverrouillage des téléphones, la reconnaissance faciale fonctionne en comparant les visages des personnes passant devant des caméras spéciales aux photos des personnes surveillées. Ces listes de personnes surveillées peuvent contenir des photos de n'importe qui, même ceux qui n'ont jamais rien fait de mal. Les photos peuvent provenir de n'importe où, même des comptes de réseaux sociaux. Les systèmes de reconnaissance faciale varient, mais ils fonctionnent généralement comme ceci :

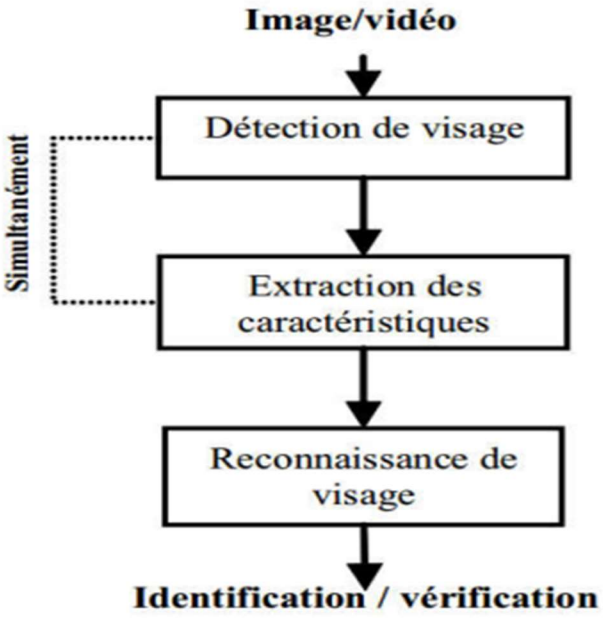

Figure II.3-1: Les étapes de reconnaissance Faciale.

II.3.3.1 Étape 1 : Détection du visage

La caméra détecte l'image d'un visage. L'image peut montrer la personne de face ou de profil. L'efficacité des systèmes biométriques basés sur l'authentification de visage dépend fortement de la méthode utilisée pour localiser le visage dans l'image ; tels que la bouche, le nez, les yeux, etc. Cependant, les solutions doivent répondre à la variabilité des conditions d'acquisition dans la vie quotidienne, surtout: La pose ; Avec ou sans éléments structuraux (barbes) ; L'éclairage; Les occultations (visages peuvent masquer d'autres visages), les propriétés de l'appareil photographique…etc.

II.3.3.2 Étape 2 : Extraction des caractéristiques du visage

L'extraction des caractéristiques telles que les yeux, le nez, la bouche, etc. est une étape de prétraitement nécessaire pour identifier les spécificités de visage. On peut distinguer deux pratiques différentes : La première est basée sur l'extraction de toute la région du visage, elle est souvent mise en œuvre avec une approche globale de la reconnaissance de visage. Le deuxième exercice extrait des points spécifiques de différentes zones caractéristiques du visage (l'analyse géométrique du visage) Les facteurs clés incluent la distance entre les yeux, la profondeur des orbites, la distance entre le front et le menton, la forme des pommettes, ainsi que le contour des lèvres, des oreilles et du menton. Elle est utilisée avec les méthodes de reconnaissance locale et également utilisée pour estimer la pose de visage. La plupart des technologies de reconnaissance faciale utilisent la 2D plutôt que la 3D, car il est plus pratique de faire la comparaison entre des images en 2D à des photos ou aux images d'une base de données.

II.3.3.3 Étape 3 : conversion d'image en données

Le processus de capture du visage transforme des informations analogues (un visage) en un ensemble d'informations numériques (les données) en fonction des caractéristiques faciales d'une personne. En fait, l'analyse du visage est transformée en formule mathématique. Le code numérique est appelé une empreinte faciale. Tout le monde a une empreinte faciale unique, tout comme votre empreinte digitale est unique.[12]

II.3.3.4 Étape 4 : La reconnaissance du visage

Le module de reconnaissance utilise les propriétés de visage extraites en tant que telles pour créer une signature numérique qu'il stocke dans une base de données. Chaque visage de la base de données est associé à une signature unique qui caractérise cette personne. [13]

L'empreinte de visage est ensuite comparée à une base de données avec d'autres visages connus. Si cette empreinte faciale correspond à une image de la base de données utilisée par la reconnaissance faciale, une correspondance est effectuée.

La reconnaissance faciale est considérée comme la mesure biométrique et le moyen le plus naturel. Ce qui est logique au niveau intuitif, car nous nous reconnaissons nous-mêmes et les autres par le visage plutôt que par les empreintes digitales ou l'iris.[12]

II.3.4 Les difficultés de la reconnaissance faciale

• Le changement d'éclairage rend la tâche de reconnaissance faciale beaucoup plus difficile.

• La variation de pose est considérée comme un problème majeur pour les systèmes de reconnaissance faciale. Surtout pour les rotations supérieures à 30°, la normalisation géométrique n'est plus possible.

•Les déformations du visage causées par les expressions faciales sont principalement localisées sur la partie inférieure du visage. Cela conduit inévitablement à des taux de reconnaissance plus faibles.

• Les jumeaux identiques : il est peu probable que la vérification automatique puisse détecter les différences très subtiles qui existent entre les jumeaux.

II.3.5 L'utilisation de la reconnaissance faciale

Cette technologie sert à bien des choses. Cela inclut :

- **Le déverrouillage des téléphones**

De nombreux smartphone utilisent la reconnaissance faciale pour déverrouiller les appareils. Cette technologie offre une manière efficace de protéger les informations privées et de garantir que les données sensibles restent inaccessibles en cas de vol de téléphone. Les chances de déverrouiller le téléphone avec un visage aléatoire sont d'environ une sur un million.

- **Application de la loi**

La reconnaissance faciale est couramment utilisée par les autorités. La police collecte des photos des personnes arrêtées et les compare aux bases de données locales, étatiques et

fédérales. Une fois prise, la photo du suspect est ajoutée à une base de données et scannée dès que la police effectue une autre enquête criminelle.

- **Contrôles dans les aéroports et aux frontières**

La reconnaissance faciale devient la plus courante dans les aéroports du monde entier, Car elle réduit les files d'attente et améliore la sécurité de l'aéroport.

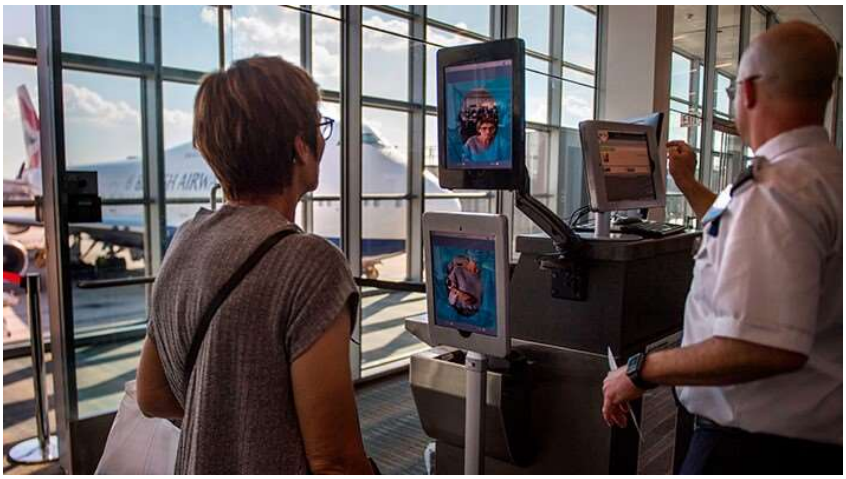

Figure II.3-2: Des contrôles à reconnaissance faciale aux aéroports.

- **Retrouver des personnes disparues**

La reconnaissance faciale peut être utilisée pour retrouver les personnes disparues et les victimes de la traite d'êtres humains. Supposant que les personnes disparues soient ajoutées à une base de données. La reconnaissance faciale peut être utilisée pour alerter les autorités dès que ces personnes sont repérées.

- **Santé**

Les hôpitaux utilisent la reconnaissance faciale pour améliorer la qualité des soins. L'utilisation de la reconnaissance faciale pour accéder aux dossiers des patients et simplifier leur enregistrement.

- **Surveiller l'addiction aux jeux d'argent**

La reconnaissance faciale aide les sociétés de jeux d'argent à mieux protéger leurs clients. Parmi les sociétés technologiques qui fournissent des services de reconnaissance faciale.

II.3.6 Les avantages de la reconnaissance faciale

En dehors du déverrouillage de votre smartphone, la reconnaissance faciale présente d'autres avantages :

- Une meilleure sécurité :

  Au niveau gouvernemental : la reconnaissance faciale permet d'identifier les criminels.

  Au niveau personnel : la reconnaissance faciale peut servir d'outil de sécurité pour verrouiller les appareils personnels et pour les caméras de surveillance des particuliers.
- Un traitement plus rapide :

  Le processus de reconnaissance du visage ne dure qu'une seconde, ce qui représente un avantage pour les entreprises qui utilisent la reconnaissance faciale.
- Intégration avec d'autres technologies :

  La plupart des solutions de reconnaissance faciale sont compatibles avec les logiciels de sécurité populaires. En fait, elles sont très faciles à intégrer. Cela permet de réduire les dépenses liées à leur intégration.
- Rationaliser la fouille des suspects.

II.3.7 Les inconvénients de la reconnaissance faciale

La reconnaissance faciale repose sur le Machine Learning, une technologie qui nécessite d'énormes ensembles de données pour apprendre à donner des résultats précis. Les petites et les moyennes entreprises n'ont souvent pas les ressources suffisantes pour stocker les données nécessaires. Ce qui précède, représente l'inconvénient de stockage massive des données, par ailleurs on peut citer les inconvénients suivants :

- La surveillance

Certains s'inquiètent de l'utilisation de la reconnaissance faciale et des caméras de surveillance, de l'intelligence artificielle et d'analyse des données qui créent une surveillance de masse potentielle et restreignent les libertés individuelles.

Alors que la reconnaissance faciale permet aux gouvernements de retrouver des criminels, elle peut aussi leur permettre de suivre des personnes ordinaires et innocentes à tout moment.

- La marge d'erreur

Les données de la reconnaissance faciale ne sont pas parfaites, ce qui pourrait mener à l'implication de personnes dans des crimes qu'elles n'ont pas commis. Par exemple, un simple changement de l'angle d'une caméra, comme une nouvelle coiffure, peut donner des erreurs.

- Violation de la vie privée

Le problème de l'éthique et de la vie privée se pose toujours. On sait que certains gouvernements possèdent des photos de plusieurs citoyens sans leur consentement.

II.4 Conclusion

Dans ce chapitre nous avons présenté l'image comme un objet à deux dimensions, projeter sur un espace à trois dimensions, de même, la reconnaissance faciale est une technologie qui a atteint une certaine maturité au point d'en faire un outil non seulement de la vie quotidienne mais également une solution parmi d'autres pour améliorer la sécurité. L'enjeu est celui des normes techniques qui garantissent un niveau de qualité et d'exploitation large. Il porte surtout sur l'évolution de la norme juridique. Elle doit garantir le juste équilibre entre liberté et sécurité dans un cadre transfrontalier. La détection et la reconnaissance d'individus demeurent des problèmes complexes, malgré les recherches actives actuelles. Il y a de nombreuses conditions réelles, difficiles à modéliser et prévoir, parmi lesquelles le mouvement du visage, qui limitent les meilleurs systèmes actuels.

CHAPITRE III : DEEP LEARNING

CHAPITRE III : DEEP LEARNING

III.1 Introduction

Le Machine learning (apprentissage automatique) et le Deep learning (apprentissage profond) sont les deux concepts les plus importants qui rendent l'intelligence artificielle possible. On confond bien souvent ces deux termes, alors qu'ils désignent deux méthodes bien distinctes employées dans des champs d'application différents.

Le Machine learning et le Deep learning font partie de l'intelligence artificielle. Ces approches ont toutes deux pour résultat de fournir aux ordinateurs la compétence de prendre des décisions intelligentes. Cependant, le Deep learning est une sous-catégorie du ML, car il s'appuie sur un apprentissage sans surveillance[14].

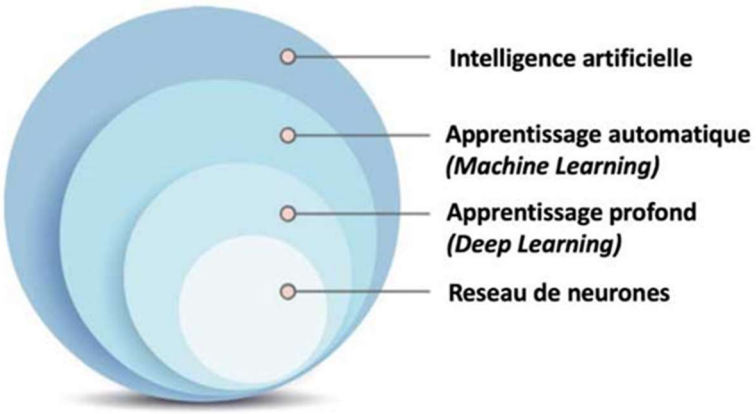

Figure III.1-1 : ML et Deep Learning dans l'IA.

Nous présentons donc dans ce chapitre les 3 concepts clés de l'intelligence artificielle : Machine Learning, Deep Learning et les réseaux de neurones artificiels. D'abord, nous rappellerons les définition de ces 3 concepts, avant de décrire l'architectures neuronales la plus utilisée CNN. Nous poursuivrons en exposant les différents algorithmes de détection.

III.2 Les concepts clés

III.2.1 Qu'est-ce que l'intelligence artificielle ?

La notion voit le jour dans les années 1950 grâce au mathématicien Alan Turing. Dans son livre Computing Machinery and Intelligence, ce dernier soulève la question d'apporter aux machines une forme d'intelligence. Il décrit alors un test aujourd'hui connu sous le nom «

Test de Turing » dans lequel un sujet interagit à l'aveugle avec un être humain, puis avec une machine programmée pour formuler des réponses sensées. Si le sujet n'est pas capable de faire la différence, alors la machine a réussi le test et, selon l'auteur, peut véritablement être considérée comme « intelligente ». [15]

L'intelligence artificielle est un processus qui consiste à développer des systèmes informatiques capables d'effectuer des taches qui nécessitent normalement l'intelligence humaine.

L'IA fonctionne en combinant de grandes quantités de données avec un traitement itératif rapide et des algorithmes intelligents, permettant aux logiciels d'apprendre automatiquement les modèles ou les caractéristiques des données. L' IA est un vaste domaine de recherche qui comprend de nombreuses théories, méthodes et techniques.

III.2.2 Machine Learning

L'apprentissage automatique (Machine Learning) est un domaine scientifique, plus précisément une sous-discipline de l'intelligence artificielle. Elle englobe de nombreuses méthodes permettant de créer automatiquement des « patterns » et d'effectuer des prédictions à partir de données. Ces données peuvent être des images, des mots, des nombres, des statistiques...

Les programmes informatiques traditionnels effectuent des tâches en suivant des instructions précises, donc toujours de la même manière. En revanche, les systèmes d'apprentissage automatique ne suivent pas les instructions, mais apprennent de manière autonome à faire des prédictions et à améliorer leurs performances dans l'exécution d'une tâche au fil du temps. Plus on le "nourrit" de données, plus il devient précis.

Le Machine Learning comporte généralement deux phases : Une phase d'apprentissage consiste à estimer un modèle lors la conception du système. L'estimation du modèle comprend la résolution de taches pratiques telle que la traduction du discours, l'estimation de la densité de probabilité, la reconnaissance de la présence d'un chat ou un chien dans une photo. La seconde phase est dite de vérification, elle correspond à la mise en production : le modèle est déterminé, de nouvelles données peuvent alors être soumises pour obtenir des résultats correspondant à la tache souhaitée.

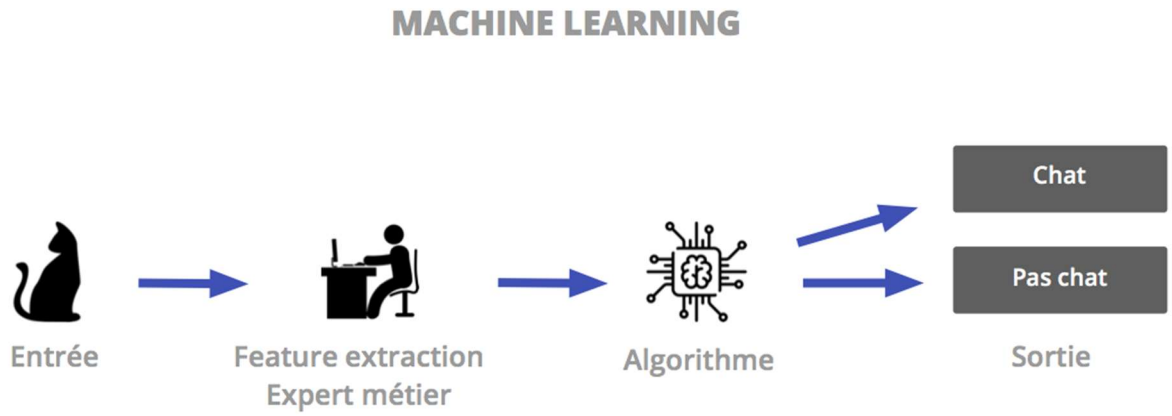

Figure III.2-1 : Machine Learning.

III.2.3 Réseaux de neurones artificiels

III.2.3.1 L'origine des neurones artificiels

Le concept des réseaux de neurones artificiels fut inventé en 1943 par deux chercheurs de l'Université de Chicago : le neurophysicien Warren McCullough, et le mathématicien Walter Pitts. Dans un article publié dans le journal Brain Theory, les deux chercheurs présentent leur théorie selon laquelle l'activation de neurones est l'unité de base de l'activité cérébrale.

En 1957, le Perceptron fut inventé. Il s'agit du plus ancien algorithme de Machine Learning, conçu pour effectuer des tâches de reconnaissance de patterns complexes. C'est cet algorithme qui permettra plus tard aux machines d'apprendre à reconnaître des objets sur des images.

Malheureusement, à l'époque, les réseaux de neurones étaient limités par les ressources techniques. Par exemple, les ordinateurs n'étaient pas assez puissants pour traiter les données nécessaires au fonctionnement des réseaux de neurones. C'est la raison pour laquelle la recherche dans le domaine des réseaux de neurones est restée en sommeil durant de longues années.

Il aura fallu attendre le début des années 2010, avec l'essor du Big Data et du traitement massivement parallèle, pour que les Data Scientists disposent des données et de la puissance de calcul nécessaires pour exécuter des réseaux de neurones complexes. En 2012, lors d'une compétition organisée par ImageNet, un réseau de neurone est parvenu pour la première fois à surpasser un humain dans la reconnaissance d'image.

C’est la raison pour laquelle cette technologie est de nouveau au cœur des préoccupations des scientifiques. A présent, les réseaux de neurones artificiels ne cessent de s’améliorer et d’évoluer de jour en jour.[16]

III.2.3.2 C’est quoi un réseau de neurones artificiel

III.2.3.2.1 Les réseaux de neurones biologiques

Le cerveau humain, est le meilleur modèle de la machine, polyvalente incroyablement rapide et surtout douée d’une incomparable capacité d’auto organisation. Son comportement est beaucoup plus mystérieux que le comportement de ses cellules de base. [17]

Un neurone biologique est caractérisé pas :

• Le corps cellulaire : Où se déroule toutes les activités vitales de la cellule.

• Des synapses : C’est une jonction entre les terminaisons axonales et les autres cellules.

• Des dendrites : les récepteurs du potentiel d'action.

• Les axones : un émetteur, conduit le potentiel d'action.

III.2.3.2.2 Les réseaux de neurones formels

Un réseau de neurones artificiels est un modèle de calcul informatique dont la conception est très inspirée du fonctionnement des neurones biologiques. Il s’agit d’une variété d’apprentissage en profond, qui font elles-mêmes partie de la sous-catégorie de l'intelligence artificielle d'apprentissage automatique.

Leur architecture était très innovante dans les années 1950, lorsqu'ils sont nés avec une structure très similaire au cerveau humain : de nombreux neurones sont connectés par des synapses, à travers lesquelles le calcul se propage en parallèle dans le cortex cérébral. Cela permet d'acquérir de nouvelles capacités : trouver des solutions à des problèmes complexes en temps réel, auto-apprentissage, résistance aux échecs et aux erreurs, etc.

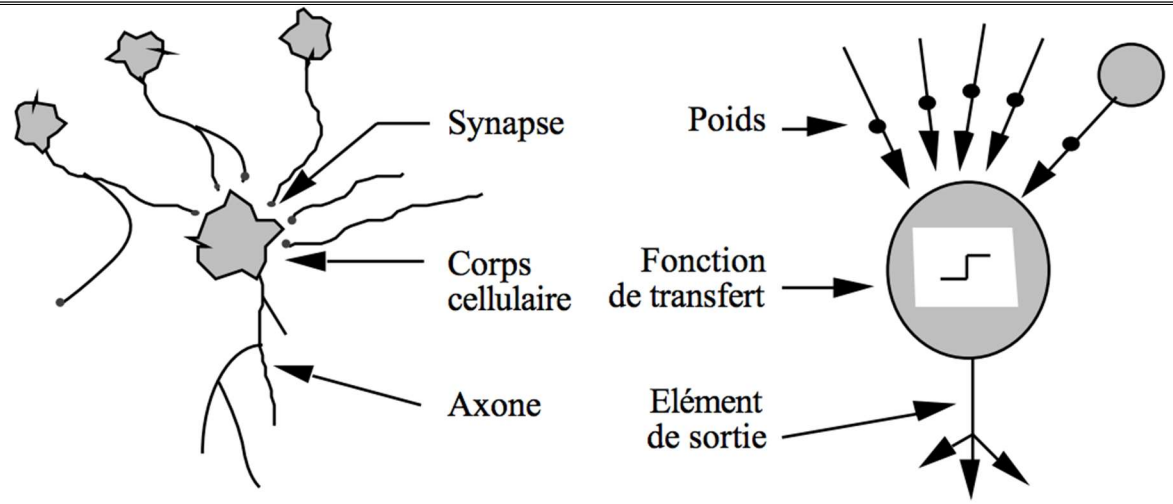

Figure III.2-2: Neurone artificiel et neurone biologique.

III.2.3.3 Fonctionnement des réseaux de neurones

Un réseau de neurones artificiels peut être décrit comme un système composé d'au moins deux couches, chacune contenant plusieurs neurones ou nœuds. D'une couche à l'autre, les nœuds sont reliés entre eux. Chaque nœud est associé à des données d'entrée, un poids, un seuil et des données de sortie (valeur de sortie est le résultat de la multiplication de la valeur d'entrée du neurone par son poids). Si les données de sortie d'un nœud dépassent un seuil spécifié, le neurone est activé et envoie ses données aux neurones de la couche suivante. Et un ainsi de suite.

L'apprentissage d'un réseau de neurones artificiel consiste à ajuster itérativement les poids associés à chacun de ses nœuds, en partant de petites valeurs. Dans l'objectif de minimiser les écarts par rapport aux résultats attendus. Pour notre cas, nous nous appuierons sur une base d'apprentissage constituée de couples associant des données d'entrée à des objectifs à atteindre. Pour chaque donnée d'entrée soumise, le réseau doit estimer les poids du réseau. La fonction d'erreur est utilisée pour calculer la différence entre la valeur prédite atteinte et la valeur cible afin d'ajuster les poids au fil du temps. Le processus est répété pour chaque paire de bases d'apprentissage.

III.2.3.4 Réseaux de neurones à multicouche

Un réseau de neurone multicouche est un type de réseau neuronal artificiel organisé en plusieurs couches au sein desquelles une information circule de la couche d'entrée vers la couche de sortie uniquement ; il s'agit donc d'un réseau à propagation directe (feedforward).

Chaque couche est constituée d'un nombre variable de neurones, les neurones de la dernière couche (dite « de sortie ») étant les sorties du système global.[18]

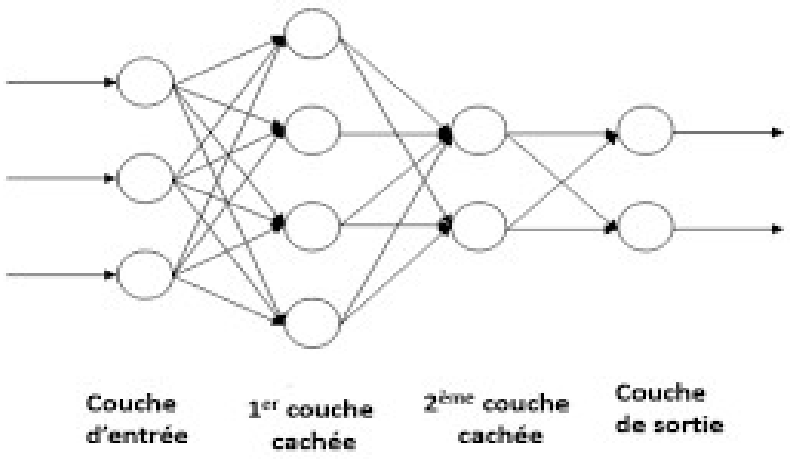

Figure III.2-3 : Un réseau de neurone avec deux couches cachées

III.2.3.4.1 Un réseau de neurones à propagation avant

Le réseau de neurones à propagation avant a été le premier et le plus simple type de réseau neuronal artificiel conçu. Typiquement, il ne comportait qu'une seule couche cachée et s'appelait Perceptron. Dans ce réseau, l'information ne se déplace que dans une seule direction, vers l'avant, à partir des nœuds d'entrée, en passant par les couches cachées (le cas échéant) et vers les nœuds de sortie. Il n'y a pas de cycles ou de boucles dans le réseau.[18]

III.2.3.4.2 Un réseau de neurone à rétro-propagation

La rétropropagation du gradient de l'erreur (ou backpropagation) est un algorithme d'optimisation permettant d'ajuster les paramètres d'un réseau de neurones multicouches pour mettre en correspondance des entrées et des sorties référencées dans une base d'apprentissage. Pour pouvoir entraîner ces systèmes, il faut savoir comment ajuster les paramètres de chaque couche de neurones. La rétropropagation permet de calculer le gradient de l'erreur pour chaque neurone, de la dernière couche vers la première. Le calcul de ce gradient se fait par la méthode de rétropropagation, pratiquée depuis le milieu des années 80. Cela permet de corriger les erreurs selon l'importance des éléments qui ont justement participé à la réalisation de ces erreurs. Ainsi, les poids synaptiques qui contribuent à engendrer une erreur importante se verront modifiés de manière plus significative que les poids qui ont engendré une erreur

marginale. Moyennant quelques précautions lors de l'apprentissage, les procédures d'optimisation finissent par aboutir à une configuration stable, généralement un extremum local, au sein du réseau de neurones. [19]

III.3 Deep Learning

Le Deep Learning est l'une des technologies principales du Machine Learning. Cette méthode d'apprentissage dit profonde repose plus précisément sur le concept de réseaux de neurones artificiels. Le système fonctionne à partir de plusieurs couches de réseaux neuronaux, qui combinent différents algorithmes inspirés du cerveau humain. Par conséquent, le système est capable de gérer des données non structurées. Cette approche est particulièrement utile pour les tâches complexes, lorsque tous les aspects de l'objet à traiter ne peuvent pas être catégorisés en amont. Le système d'apprentissage profond lui-même peut identifier des caractéristiques discriminantes. Dans chaque couche, il recherche de nouveaux critères spécifiques à l'objet comme une base pour décider du classement à conserver pour l'objet en fin de processus.

Dans l'apprentissage profond, le système reconnaît lui-même les caractéristiques discriminantes des données, sans classification préalable. Le système ne nécessite pas d'entrainement par les développeurs. Il évalue par lui-même s'il doit réviser la classification ou créer de nouvelles catégories en fonction des nouvelles données.

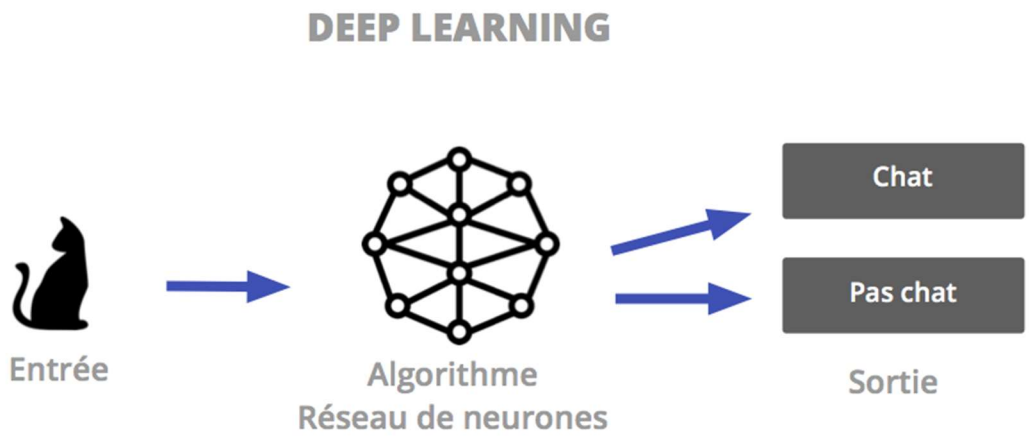

Figure III.3-1: Deep Learning.

III.3.1 Pourquoi le choix de deep learning

- **De meilleurs résultats qu'avec d'autres méthodes d'apprentissage machine**

Le plus grand point fort du Deep learning reste la qualité des résultats obtenus. Dans des secteurs tels que le traitement d'images ou la reconnaissance d'images, cette forme d'intelligence artificielle détrône toutes les autres.[20]

- **Une exécution efficace des tâches de routine, sans écarts de qualité**

Parce que basé sur un apprentissage routinier, ne montrant jamais aucun signe de fatigue et avec une qualité constante, celle-ci est beaucoup plus efficace et rapide que n'importe quelle autre méthode.

Puisque le système se forme de façon autonome (après une phase d'instruction initiale), il permet d'économiser beaucoup de temps et d'argent tout en garantissant un développement de ses fonctionnalités.[20]

- **Le traitement des données non structurées**

De plus, et contrairement à d'autres moteurs d'intelligence artificielle, l'apprentissage profond est capable d'analyser des données stockées sous un format non structuré (documents, photos, mails, etc.).[20]

III.3.2 Les algorithmes de Deep Learning

Il existe plusieurs algorithmes pour un réseau de neurone profond.

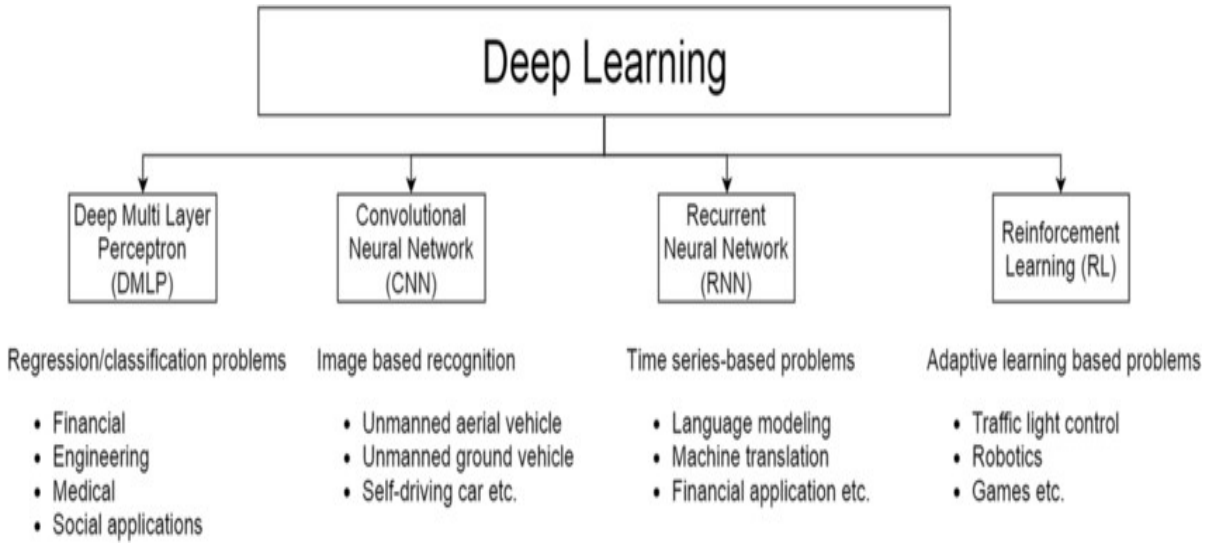

Figure III.3-2 : Différents modèles du deep learning

III.4 Réseaux neurones convolutionnels

Les réseaux de neurones convolutifs (CNN) désignent une sous-catégorie de réseaux de neurones dont l'architecture de connectivité s'inspire du cortex visuel des mammifères. Ils sont actuellement les modèles de classification d'images les plus efficaces. Ils comportent deux parties différentes. La première partie est appelée la partie convolutive du modèle, elle est basée sur le principe mathématique de convolution. Et la deuxième partie est appelée la partie classification du modèle, qui correspond au modèle MLP (Multi layer Perceptron).

En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a deux dimensions (largeur et hauteur) pour une image monochrome (en niveau du gris) ou trois dimensions pour une image en couleur (une troisième dimension qui correspond à la composante couleur : vert, bleu et rouge).

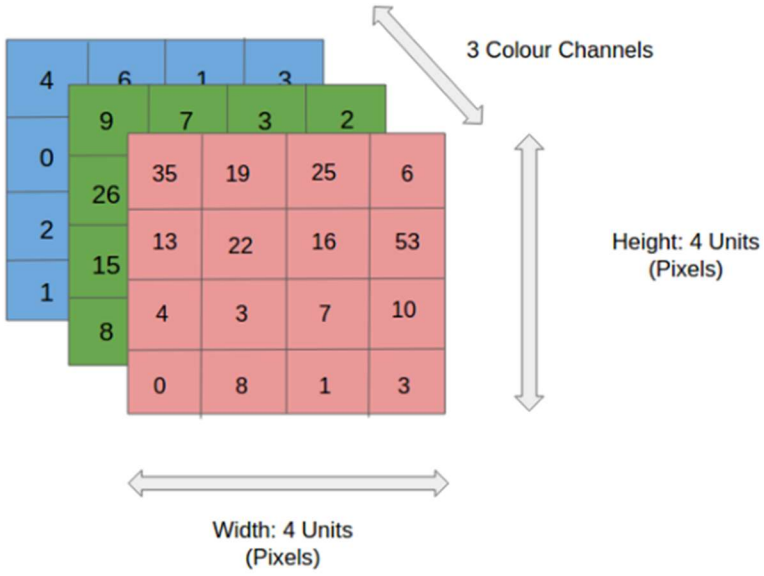

Figure III.4-1: image RVB 4*4*3.

La première partie du CNN fonctionne comme un extracteur de caractéristiques des images. L'image est passée à travers un ensemble de filtres ou de noyaux de convolution, créant une nouvelle image appelée carte de convolution. Et enfin, les cartes obtenues sont concaténées en vecteur de caractéristiques, appelé code CNN.

Dans la deuxième partie de classification, le code CNN obtenue de la première partie est fourni en entrée de cette seconde partie, composée de couches entièrement connectées appelées perceptron multicouche (MLP pour Multi Layers Perceptron). Son rôle est de combiner les caractéristiques du code CNN afin de classifier l'image.

A la sortie, on obtient des nombres généralement normalisés entre 0 et 1 pour représenter une distribution de probabilité à propos de la classe.

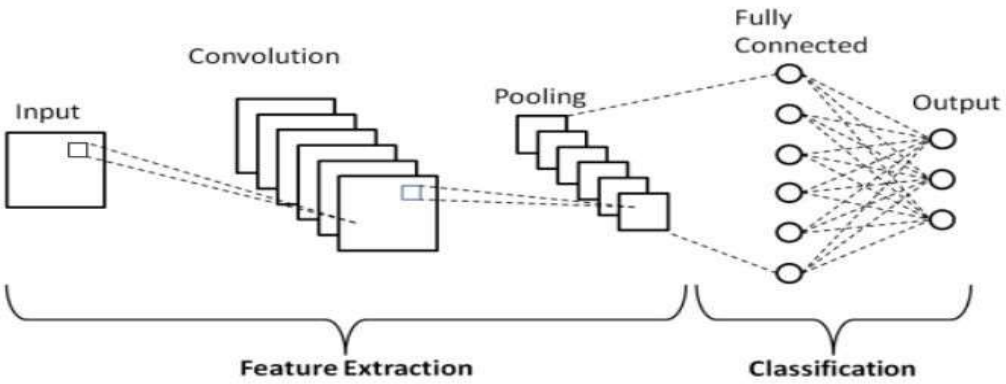

Figure III.4-2: Les 2 étapes de CNN.

III.4.1 Architecture d'un réseau de neurone convolutionnel

L'architecture CNN comprend plusieurs blocs de construction, tels que les couches de convolution, les couches de Pooling et les couches entièrement connectées (Fully Connected). Une architecture typique consiste en des répétitions d'un empilement de plusieurs couches de convolution et d'une couche de Pooling, suivies d'une ou plusieurs couches entièrement connectées (FC). L'étape où les données d'entrée sont transformées en sortie à travers ces couches est appelée propagation en avant.

III.4.1.1 Couche de convolution

Une couche de convolution est un composant fondamental de l'architecture CNN qui effectue l'extraction de caractéristiques, qui consiste généralement en une combinaison d'opérations linéaires et non linéaires, c'est-à-dire une opération de convolution et une fonction d'activation.

- **Terminologie**

Dans la terminologie du réseau convolutif, le premier argument de la convolution est souvent appelé l'entrée (input) et le second argument comme noyau (kernel). La sortie (output) est parfois appelée feature map.

- **Convolution**

La convolution est un type spécialisé d'opération linéaire utilisée pour l'extraction de caractéristiques, où un petit tableau de nombres, appelé noyau (filtre), est appliqué sur l'entrée, qui est un tableau de nombres, appelé tenseur. Un produit élément par élément entre chaque élément du noyau et le tenseur d'entrée est calculé à chaque emplacement du tenseur et additionné pour obtenir la valeur de sortie dans la position correspondante du tenseur de sortie, appelée carte de caractéristiques. Cette procédure est répétée en appliquant plusieurs noyaux pour former un nombre arbitraire de cartes de caractéristiques, qui représentent différentes caractéristiques des tenseurs d'entrée ; différents noyaux peuvent donc être considérés comme des extracteurs de caractéristiques différents. Deux hyperparamètres clés qui définissent l'opération de convolution sont la taille et le nombre de noyaux. Le premier est généralement 3 × 3, mais parfois 5 × 5 ou 7 × 7. Le second est arbitraire et détermine la profondeur des cartes d'entités en sortie.

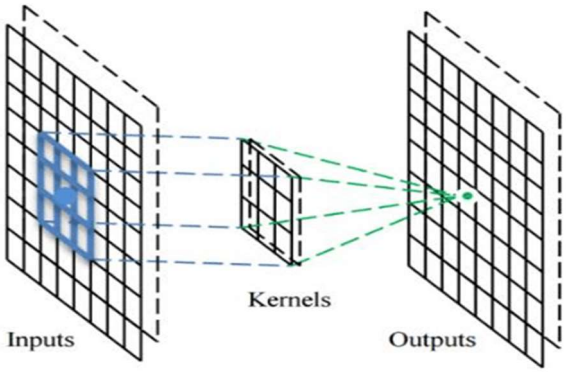

Figure III.4-3: Filtre de convolution.

Plusieurs noyaux fonctionnent comme différents extracteurs de caractéristiques, tels qu'un détecteur de bord horizontal (en haut), un détecteur de bord vertical (au milieu) et un détecteur de contour (en bas). Notez que l'image de gauche est une entrée, celles du milieu sont des noyaux et celles de droite sont des cartes d'entités de sortie.

La distance entre deux positions successives du noyau est appelée foulée, qui définit également l'opération de convolution. Le choix commun d'une foulée est 1; cependant, une foulée supérieure à 1 est parfois utilisée afin de réaliser un sous-échantillonnage des cartes d'entités. Une technique alternative pour effectuer un sous-échantillonnage est une opération de Pooling.

III.4.1.2 Fonction d'activation

La fonction d'activation est une fonction mathématique appliquée à un signal en sortie d'un neurone artificiel. Le terme de "fonction d'activation" vient de l'équivalent biologique "potentiel d'activation", seuil de stimulation qui, une fois atteint entraîne une réponse du neurone. La fonction d'activation est souvent une fonction non linéaire. Un exemple de fonction d'activation est la fonction de Heaviside, qui renvoie tout le temps 1 si le signal en entrée est positif, ou 0 s'il est négatif. [21]

Les sorties d'une opération linéaire telle que la convolution sont ensuite asses à travers une fonction d'activation non linéaire. Bien que les fonctions non linéaires douces, telles que la fonction sigmoïde ou tangente hyperbolique (tanh), aient été utilisées auparavant parce qu'elles sont des représentations mathématiques du comportement d'un neurone biologique, la fonction d'activation non linéaire la plus couramment utilisée actuellement est rectified linear unit (ReLU), qui calcule simplement la fonction : f(x) = max(0, x)

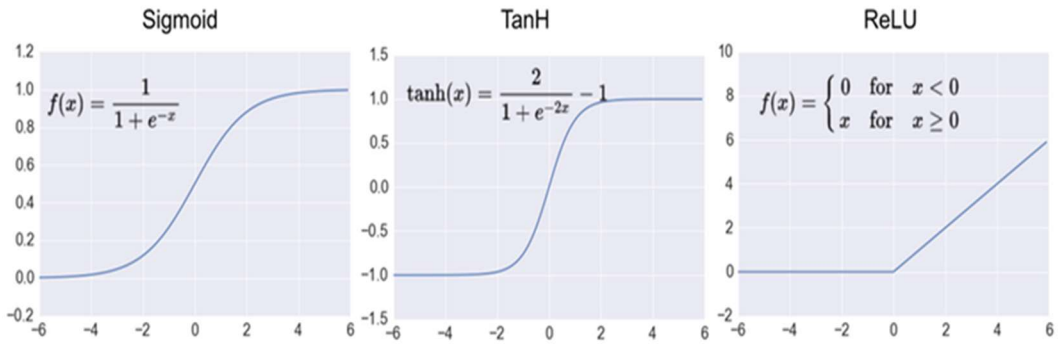

Figure III.4-4: Fonctions d'activation couramment utilisées

III.4.1.3 Couche de Pooling

L'architecture d'un réseau convolutif se compose de trois types de couches différentes. Une première pour générer un ensemble d'activations linéaires ensuite, une deuxième fonction d'activation non linéaire telle que Rectified Linear Unit (ReLu), et enfin on se projette dans une couche pooling.

Une couche de pooling fournit une opération de sous-échantillonnage typique qui réduit la dimensionnalité dans le plan des cartes de caractéristiques afin d'introduire une invariance de translation aux petits décalages et distorsions, et de diminuer le nombre de paramètres apprenables ultérieurs. Il est à noter qu'il n'y a aucun paramètre apprenable dans aucune des couches de regroupement, alors que la taille du filtre, la foulée et le rembourrage sont des

hyperparamètres dans les opérations de regroupement, similaires aux opérations de convolution.

- **Max Pooling**

Le max pooling est la forme la plus populaire de la couche pooling, qui extrait les correctifs des cartes de caractéristiques d'entrée, génère la valeur maximale dans chaque correctif et supprime toutes les autres valeurs. Un max pooling avec un filtre de taille 2 × 2 avec une foulée de 2 est couramment utilisé en pratique. Cela sous-échantillonne la dimension dans le plan des cartes d'entités par un facteur de 2. Contrairement à la hauteur et à la largeur, la dimension de profondeur des cartes d'entités reste inchangée.

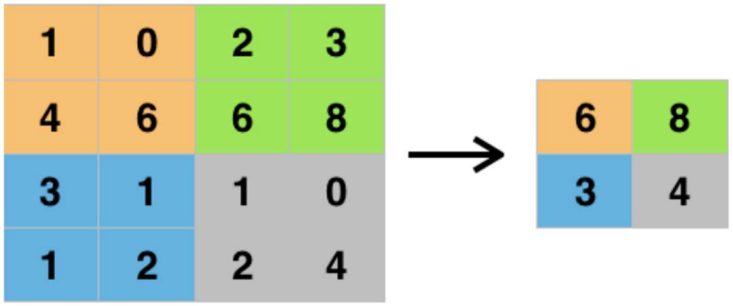

Figure III.4-5: Max Pooling avec un filtre 2x2 et un pas de 2.

III.4.1.4 Couche Entièrement Connectées (FC)

Les cartes de caractéristiques de sortie de la convolution finale ou de la couche de pooling sont généralement aplaties, c'est-à-dire transformées en un tableau unidimensionnel (1D) (vecteur), et connectées à une ou plusieurs couches Fully Connected (FC), dans lequel chaque entrée est connectée à chaque sortie par un poids apprenable. Une fois que les caractéristiques extraites par les couches de convolution et sous-échantillonnées par les couches de pooling sont créées, elles sont mappées par un sous-ensemble de couches entièrement connectées aux sorties finales du réseau, telles que les probabilités pour chaque classe dans les tâches de classification. La couche finale entièrement connectée a généralement le même nombre de nœuds de sortie que le nombre de classes. Chaque couche entièrement connectée est suivie d'une fonction non linéaire, telle que ReLU.

III.4.1.5 Fonction d'activation de la dernière couche

La fonction d'activation appliquée à la dernière couche entièrement connectée est généralement différente des autres. Une fonction d'activation appropriée doit être sélectionnée en fonction de chaque tâche. Une fonction d'activation appliquée à la tâche de classification

multiclasse est une fonction softmax qui normalise les valeurs réelles de sortie de la dernière couche FC aux probabilités de classe cible, où chaque valeur est comprise entre 0 et 1 et toutes les valeurs totalisent 1.

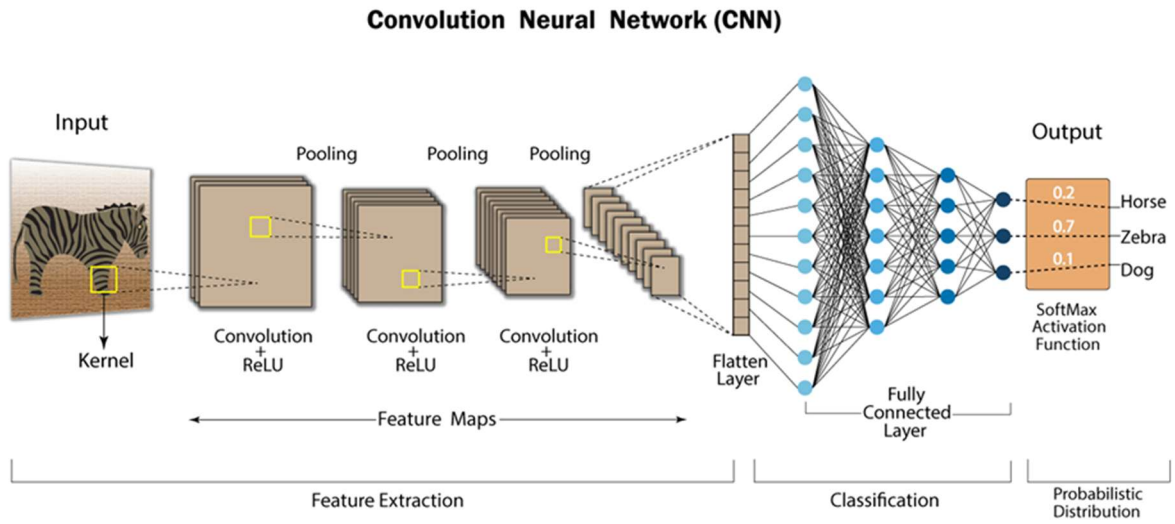

Figure III.4-6: architecture d'un réseau de neurone convolutif.

III.4.2 Les avantages de CNN

Le CNN offre de nombreux avantages :

- Le réseau fonctionne de manière robuste et est insensible à la distorsion ou à d'autres changements optiques.
- Il peut traiter des images enregistrées dans différentes conditions d'éclairage et dans différentes perspectives. Les caractéristiques typiques d'une image sont ainsi facilement identifiées.
- Il nécessite beaucoup moins d'espace de stockage que les réseaux de neurones entièrement maillés. Le CNN est divisé en plusieurs couches locales partiellement maillées. Les couches de convolution réduisent considérablement les besoins de stockage.
- Le temps de formation d'un CNN est également considérablement réduit. Grâce à l'utilisation de processeurs graphiques modernes, les CNN peuvent être formés de manière très efficace.[22]

III.5 Réseaux siamois

À l'ère moderne du Deep Learning, les réseaux de neurones sont presque bons dans toutes les tâches, mais ces réseaux de neurones s'appuient sur plus de données pour bien fonctionner. Mais, pour certains problèmes comme la reconnaissance faciale et la vérification de signature, nous ne pouvons pas toujours compter sur l'obtention de plus de données, pour résoudre ce genre de tâches, nous avons un nouveau type d'architecture de réseau neuronal appelé Siamese Networks. [23]

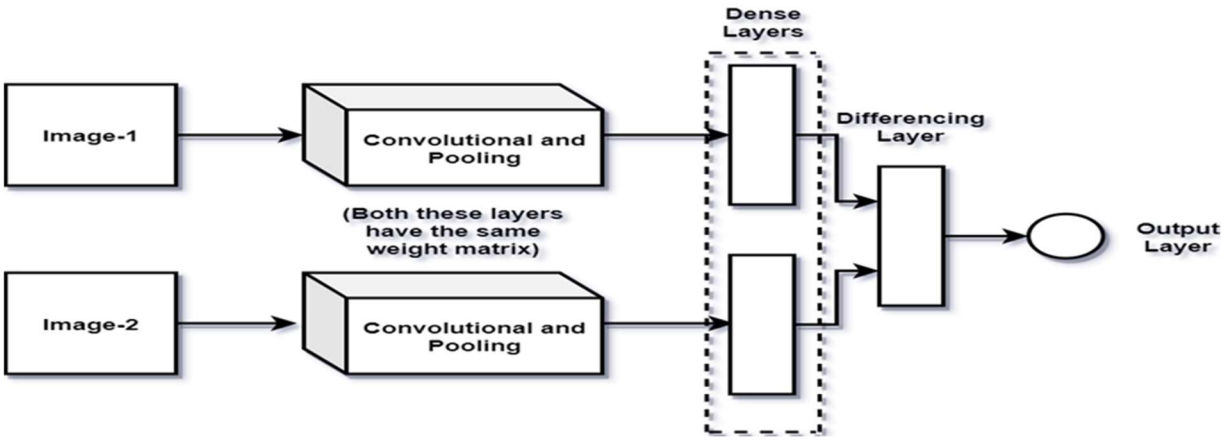

Figure III.5-1: réseaux siamois.

III.5.1 Que sont les réseaux siamois ?

Un réseau de neurones siamois est une classe d'architectures de réseaux de neurones qui « contiennent deux ou plus identique sous-réseaux ». ' identique' ici signifie qu'ils ont la même configuration avec les mêmes paramètres et poids. La mise à jour des paramètres est reflétée sur les deux sous-réseaux. Il est utilisé pour trouver la similitude des entrées en comparant ses vecteurs de caractéristiques, de sorte que ces réseaux sont utilisés dans de nombreuses applications. [23]

Habituellement, les réseaux siamois effectuent une classification binaire à la sortie, classant si les entrées sont de la même classe ou non. Ainsi, différentes fonctions de perte peuvent être utilisées pendant l'entraînement. L'une des fonctions de perte les plus populaires est la perte d'entropie croisée binaire. Cette perte peut être calculée comme

$$L = -y \log p + (1 - y) \log(1-p)$$

où L est la fonction de perte, «y» l'étiquette de classe (0 ou 1) et p est la prédiction. Afin d'entraîner le réseau à faire la distinction entre des objets similaires et différents, nous

pouvons lui donner un exemple positif et un exemple négatif à la fois et additionner les pertes :

$$L = L + + L -$$

Une autre possibilité consiste à utiliser la perte de triplet :

$$L = \max (d(a,p) - d(a,n) + m,0)$$

Ici, d est une fonction de distance (par exemple la perte L2), « a» est un échantillon de l'ensemble de données, p est un échantillon positif aléatoire et n est un échantillon négatif. m est une marge arbitraire et est utilisée pour accentuer la séparation entre les scores positifs et négatifs.

III.5.2 Les avantages du réseau siamois

- Le réseau siamois est un modèle de classification unique et peut effectuer des prédictions avec un seul exemple d'entraînement.
- Plus robuste au déséquilibre de classe car il nécessite très peu d'informations. Il peut être utilisé sur un ensemble de données où très peu d'exemples existent pour certaines classes.
- La fonction d'apprentissage ponctuel du réseau siamois ne repose pas sur des connaissances spécifiques à un domaine, mais exploite des techniques d'apprentissage en profondeur.

III.6 Algorithmes de détection

III.6.1 R-CNN, Fast R-CNN

- **R-CNN**: Le détecteur R-CNN génère d'abord des régions en utilisant un algorithme tel que ‘Edge Boxes’ , Ces régions sont rognées hors l'image et redimensionnées. Ensuite, le CNN classe les régions recadrées et redimensionnées. Enfin, les boîtes de délimitation de la proposition de région sont affinées par une machine à vecteurs de support (SVM) qui est formée à l'aide des fonctionnalités CNN.

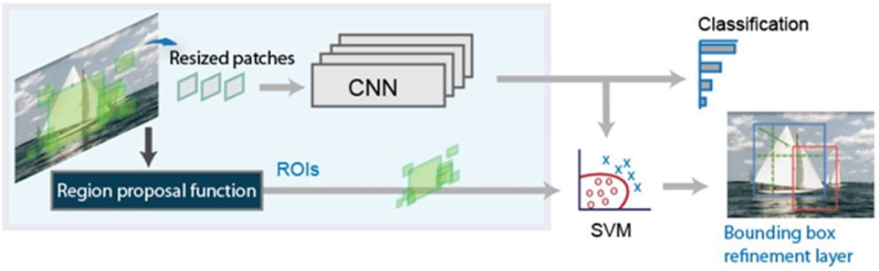

Figure III.6-1: Illustration du R-CNN.

- **Fast R-CNN** : Comme dans le détecteur R-CNN, le détecteur Fast R-CNN utilise également un algorithme comme le Edge Boxes pour générer des propositions de régions. Contrairement au détecteur R-CNN, qui recadre et redimensionne les propositions de région, le détecteur Fast R-CNN traite l'image entière. Alors qu'un détecteur R-CNN doit classer chaque région, Fast R-CNN regroupe les caractéristiques CNN correspondant à chaque proposition de région. Fast R-CNN est plus efficace que R-CNN, car dans le détecteur Fast R-CNN, les calculs des régions qui se chevauchent sont pas répétés.

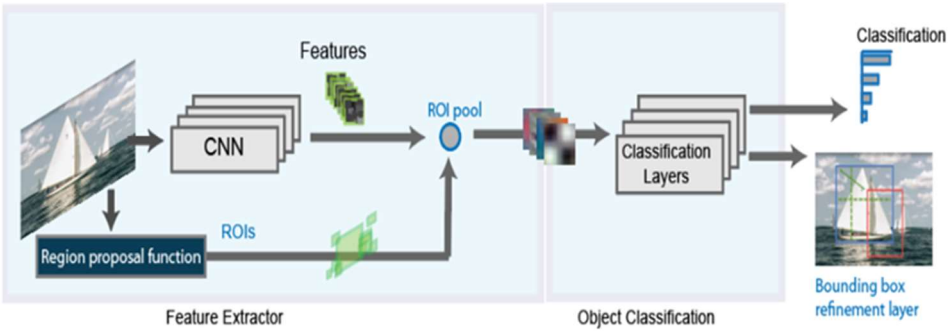

Figure III.6-2 : Illustration du Fast R-CNN.

III.6.2 SSD

SSD exécute un réseau convolutionnel sur l'image d'entrée une seule fois et calcule une carte des caractéristiques. Pour mieux comprendre le SSD, commençons par expliquer d'où vient le nom de cette architecture:

Single Shot : cela signifie que les tâches de localisation et de classification des objets sont effectuées en un seul passage avant du réseau.

MultiBox : c'est le nom d'une technique de régression par boîte englobante développée par Szegedy.

Detector : Le réseau est un détecteur d'objets qui procède aussi a leur classements.

III.6.3 YOLO

Yolo, qui veut dire “You Only Look Once”, en français : « vous ne regardez qu’une seule fois », c’est un réseau de neurones spécialisé dans la détection et l’analyse d’objets dans l’image. Sa grande force est la rapidité : il peut travailler en temps réel (à 45 im / sec). Yolo est plus rapide que des R-CNN, car il découpe l’image en petits blocs et génère des tenseurs pour chaque bloques. Pour le Yolo l’entrainement est essentiel. Les modèles doivent recevoir un jeu de données d’entrainement très proches des futures images à analyser. [24]

III.6.3.1 Comment fonctionne le Yolo ?

Notre système divise l'image d'entrée en une grille S × S. Si le centre d'un objet tombe dans une cellule de la grille, cette cellule de la grille est responsable de la détection de cet objet.[25]

Figure III.6-3 : Image YOLO (divisée en grille S*S)

Chaque cellule de la grille prédit B boîtes englobantes et les scores de confiance pour ces boîtes. Ces scores de confiance reflètent la confiance du modèle dans le fait que la boîte contient un objet et également la précision avec laquelle il pense que la boîte prédit. Formellement, nous définissons la confiance comme Pr(Object) ∗ IOU . Si aucun objet n'existe dans cette cellule, les scores de confiance doivent être nuls. Sinon, nous voulons que

le score de confiance soit égal à l'intersection sur l'union (IOU) entre la boîte prédite et la vérité terrain.[25]

Chaque boîte englobante se compose de 5 prédictions : x, y, w, h et confiance. Les coordonnées (x, y) représentent le centre de la boîte par rapport aux limites de la cellule de la grille. La largeur et la hauteur sont prédites par rapport à l'ensemble de l'image.

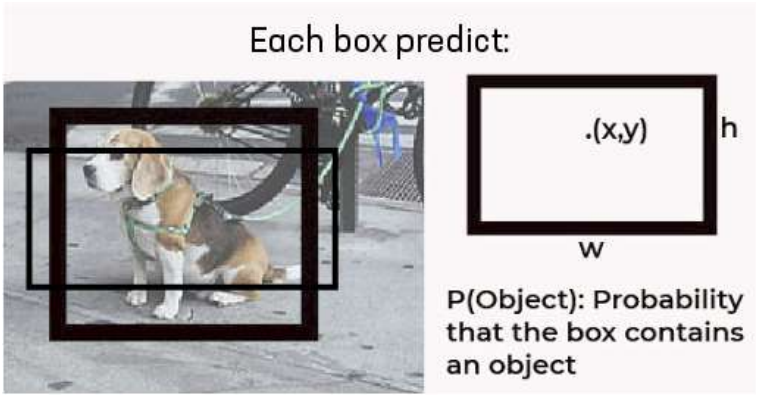

Figure III.6-4 : Boîte-boîte de délimitation de grille unique YOLO

Cela se traduit par une combinaison de boîtes englobantes de chaque grille comme celle-ci.

Figure III.6-5 : Boîte englobante YOLO Combinaison

Enfin, la prédiction de confiance représente l'IOU entre la boîte prédite et toute boîte de vérité terrain. Chaque cellule de la grille prédit également les probabilités de classe conditionnelle C, Pr(Classi |Object).

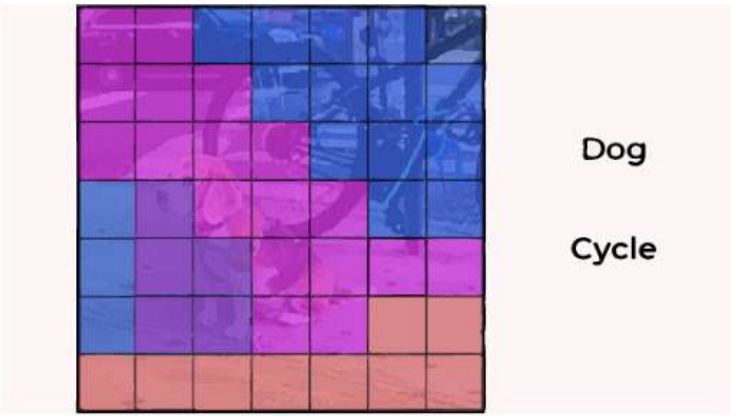

Figure III.6-6 : Carte de probabilité conditionnelle YOLO

Ces probabilités sont conditionnées sur la cellule de grille contenant un objet. Nous ne prédisons qu'un seul ensemble de probabilités de classe par cellule de grille, quel que soit le nombre de cases B.[25]

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

$$Pr(Classi|Object)*Pr(Object)*IOU = Pr(Classi)*IOU$$

, ce qui nous donne des scores de confiance spécifiques à chaque classe pour chaque case. Ces scores encodent à la fois la probabilité que cette classe apparaisse dans la boîte et la façon dont la boîte prédite correspond à l'objet.[25]

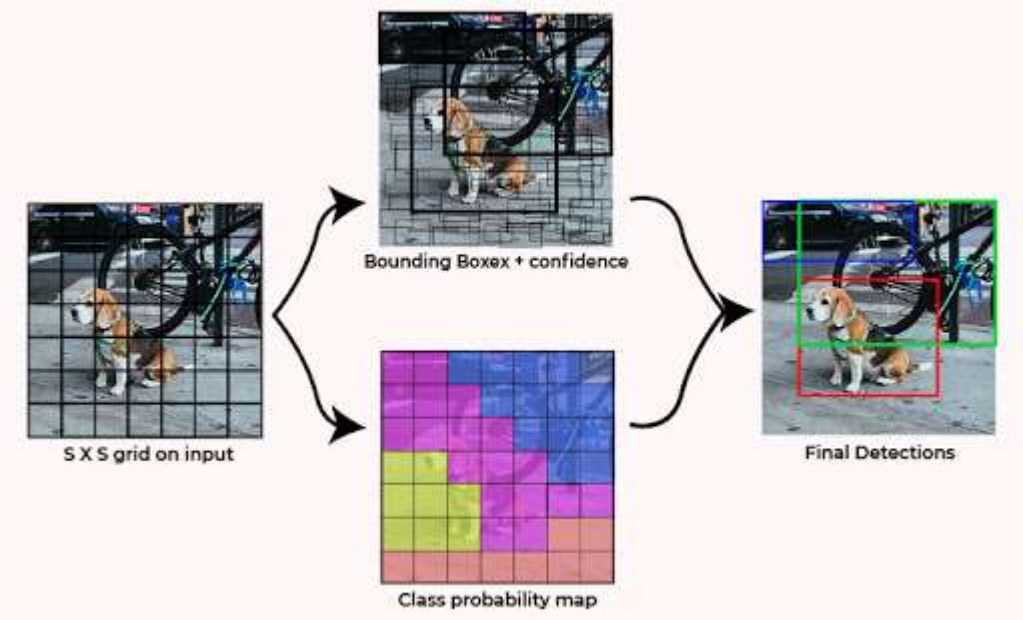

Figure III.6-7 : Résultat du test YOLO

III.6.3.2 Pourquoi choisir le Yolo ?

YOLO s'entraîne sur des images complètes et optimise directement les performances de détection. Ce modèle unifié présente plusieurs avantages par rapport aux méthodes traditionnelles de détection d'objets. Premièrement, YOLO est extrêmement rapide. Puisque nous considérons la détection comme un problème de régression, nous n'avons pas besoin d'un pipeline complexe. Nous exécutons simplement notre réseau de neurones sur une nouvelle image au moment du test pour prédire les détections. Notre réseau de base fonctionne à 45 images par seconde sans traitement par lots sur un GPU Titan X et une version rapide fonctionne à plus de 150 ips. Cela signifie que nous pouvons traiter la vidéo en streaming en temps réel avec moins de 25 millisecondes de latence.

Deuxièmement, YOLO raisonne globalement sur l'image lorsqu'il fait des prédictions. Contrairement aux techniques basées sur la fenêtre glissante et la proposition de région, YOLO voit l'intégralité de l'image pendant la formation et le temps de test, de sorte qu'il encode implicitement des informations contextuelles sur les classes ainsi que leur apparence. Fast R-CNN, une méthode de détection de pointe, confond les correctifs d'arrière-plan d'une image avec des objets car il ne peut pas voir le contexte plus large. YOLO fait moins de la moitié du nombre d'erreurs d'arrière-plan par rapport à Fast R-CNN.

Troisièmement, YOLO apprend des représentations généralisables d'objets. Lorsqu'il est formé sur des images naturelles et testé sur des illustrations, YOLO surpasse de loin les meilleures méthodes de détection telles que DPM et R-CNN. Étant donné que YOLO est hautement généralisable, il est moins susceptible de tomber en panne lorsqu'il est appliqué à de nouveaux domaines ou à des entrées inattendues.

III.6.3.3 YOLO 5

YOLO (You Only Look Once) Version 5 est une famille de modèles de détection d'objets publiée en juin 2020. Il s'agit d'une architecture en une seule étape qui va directement des pixels de l'image aux coordonnées de la boîte englobante et aux probabilités de classe. YOLO Version 5 utilise la technique Cross Stage Partial Networks (CSPNet) dans son épine dorsale pour extraire des fonctionnalités riches et informatives de l'image d'entrée, implémente le cou PA-NET pour l'agrégation de fonctionnalités et utilise la fonction SiLU pour ses activations. Sa formation s'appuie sur plusieurs nouvelles techniques d'augmentation de données telles que l'augmentation et la découpe de mosaïque (également utilisées dans la version 4 de YOLO), aidant le modèle à reconnaître de petits objets. Le modèle YOLO Version 5 S est environ 90 % plus petit que YOLOv4-custom (avec architecture Darknet), ce qui signifie qu'il peut être déployé beaucoup plus facilement sur des appareils embarqués.

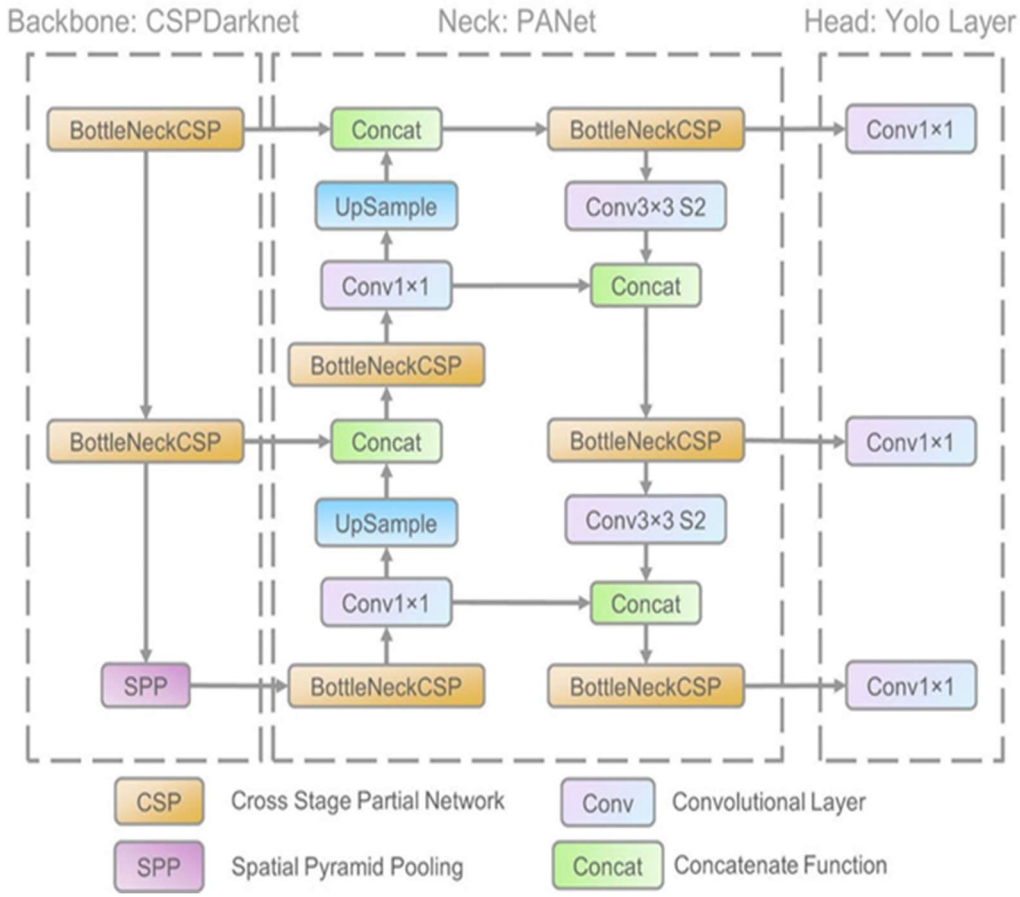

Figure III.6-8: Architecture globale de la version 5 du modèle YOLO.

III.7Conclusion

Dans ce chapitre on a fait le tour sur l'apprentissage profond d'une manière générale et on a abordé les définitions fondamentales relatives aux réseaux de neurones. On s'est basé en particulier sur les réseaux de neurones convolutifs CNN. Enfin on a parlé de quelques algorithmes de détection caractérisés par leur latence minimale.

CHAPITRE IV : ANALYSE DES BESOINS ET CONCEPTION DU SYSTEME

CHAPITRE IV : ANALYSE DES BESOINS ET CONCEPTION DU SYSTEME

IV.1 Introduction

Ce chapitre définit les fonctionnalités clés de notre système en utilisant les différentes versions proposées par les diagrammes d'UML qui seront par la suite utilisés dans la phase d'implémentation :

- ✓ Diagrammes de cas d'utilisation : représentation des fonctions du système du point de vue de l'utilisateur.
- ✓ Diagrammes de séquence : un type de diagramme comportemental en langage de modélisation unifié (UML) qui représente les transitions entre divers objets.

D'abord, nous présentons l'architecture globale de notre système.

IV.2 L'architecture générale du système

Nous proposons notre Conception de l'architecture générale du système, mettant en évidence le détail de fonctionnement tel qu'il est illustré par la figure ci-dessous.

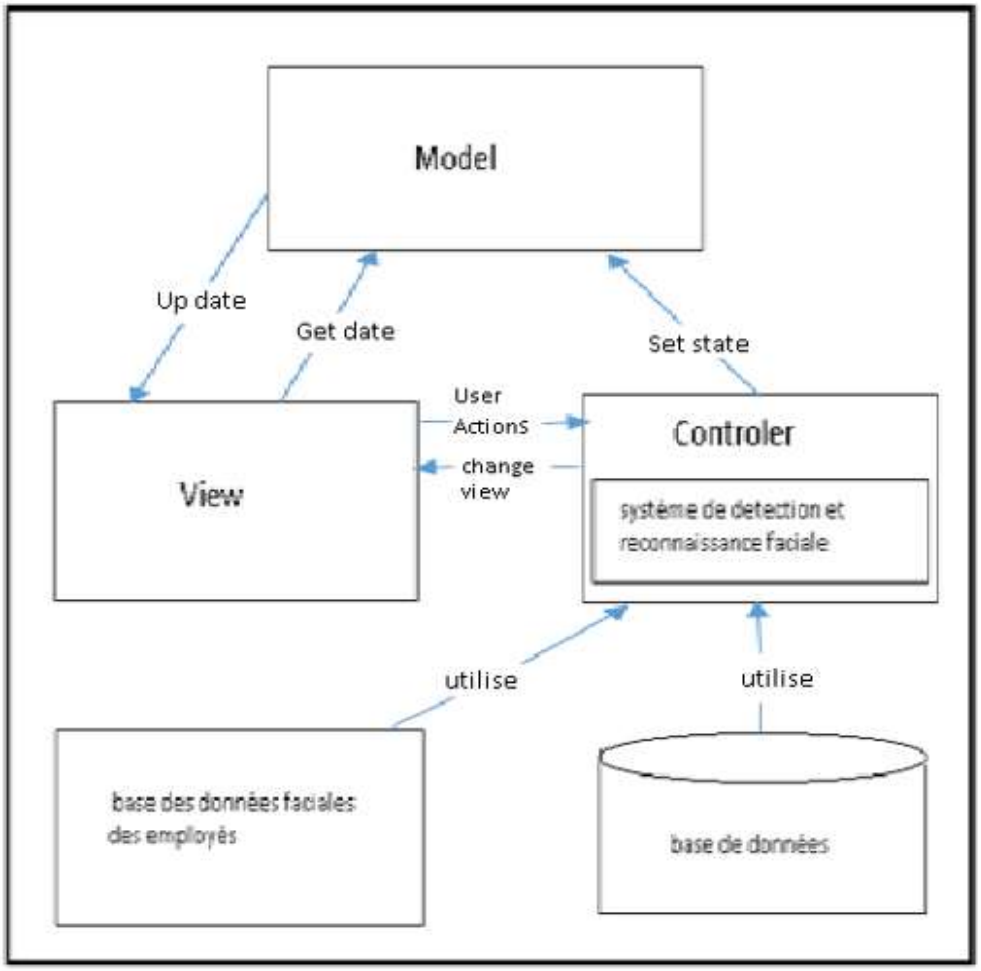

Figure IV.2-1: Architecture générale du système.

Le modèle MVC (Model View Controler) réspecter par django (framework)

IV.3 Diagrammes de cas d'utilisation

A ce stade du processus, nous présentons les différents cas d'utilisation qui seront réalisés, ainsi qu'une description de chaque cas.

Les cas d'utilisation constituent un moyen de recueil et de description des besoins des acteurs du système. Ainsi, ils permettent de décrire l'interaction entre ces acteurs (les utilisateurs du système) et le système. L'interaction est décrite selon le point de vue de l'utilisateur.

Dans ce qui suit, nous décrirons les différents cas d'utilisation et les principaux acteurs :

- **Administrateur :** il s'occupe principalement de la partie technique de notre application, la gestion des comptes des employés (les opérations de l'ajout et la suppression des comptes des employés).
- **Employé :** son accès est conditionné par l'introduction du son mot de passe adéquat de plus, il sera authentifié par le biais de sa caméra tout ça pour confirmer son compte.

Identifications des cas d'utilisation :

Le cas d'utilisation détermine la méthode d'utilisation du système et nous permet de décrire ses exigences fonctionnelles. Chaque cas d'utilisation contient un ou plusieurs scénarios qui définissent la façon dont le système interagit avec les utilisateurs (l'administrateur et les employés) pour atteindre un objectif ou une fonction spécifique d'un travail. Après avoir identifié les acteurs, nous déterminerons pour chaque représentant leurs propres cas d'utilisation.

L'employé : il a le rôle de :

- Accéder à son compte privé en toute sécurité.
- Modifier son profil : Changer sa photo.

L'administrateur : il a le rôle de :

- Gérer les comptes des employés (la suppression)
- Ajouter des nouveaux comptes pour les nouveaux employés de l'entreprise.
- S'assurer qu'une photo du nouvel employé est prise lors de la création de son compte.

IV.3.1 Diagramme des cas d'utilisation global

Ce diagramme de cas d'utilisation globale aide à expliciter comment notre système est utilisé. Selon notre diagramme, il existe deux acteurs principaux (employé, administrateur) qui agissent directement sur le système et ils ont besoin d'utiliser ce dernier. Dans notre diagramme, il y a plusieurs cas d'utilisation chacun représente un ensemble d'actions qui sont réalisées par le système et qui produisent un résultat observable les acteurs désignés.

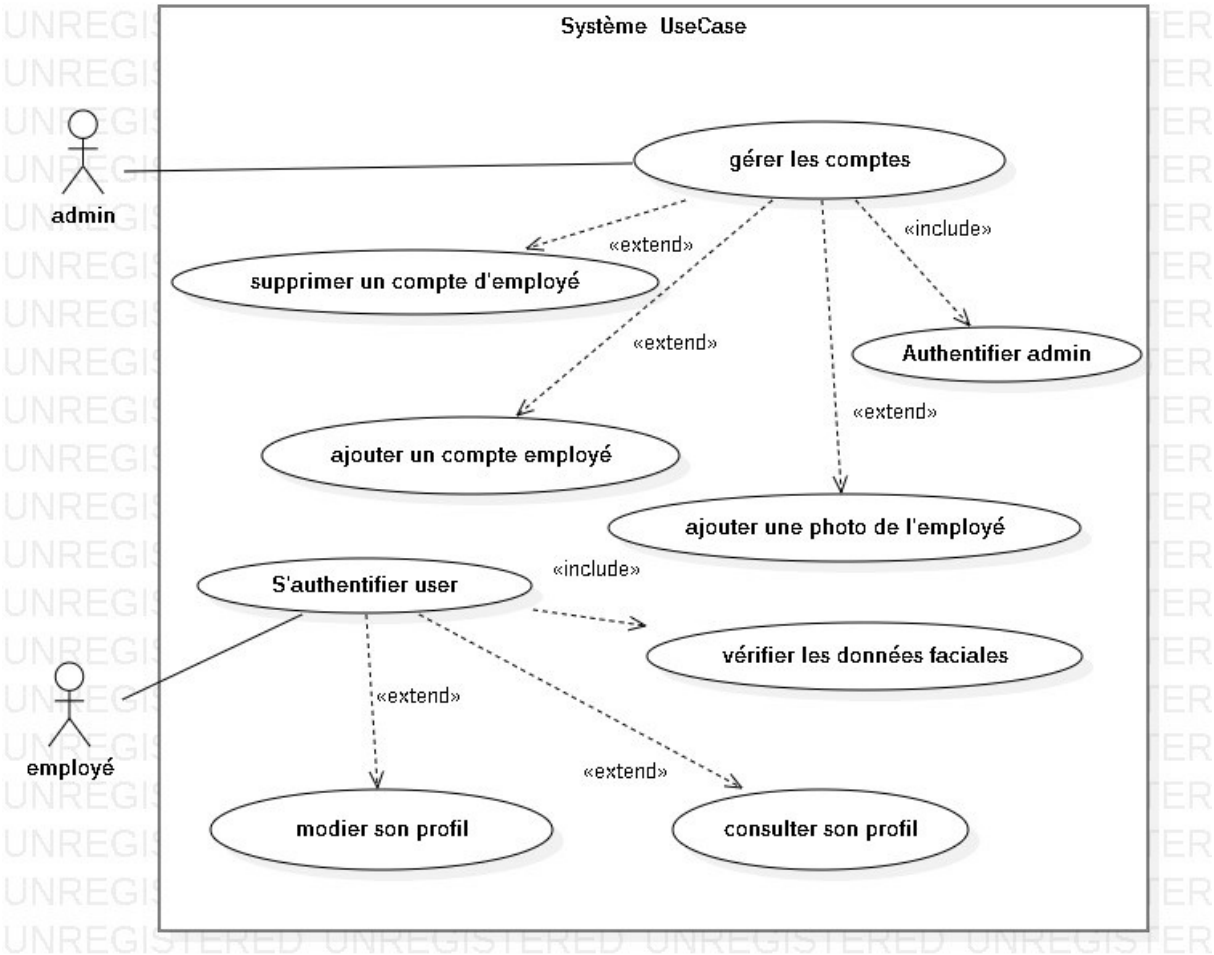

Figure IV.3-1: Diagramme de cas d'utilisation globale du 'système'

Nous identifions les cas d'utilisation qui donnent une vue d'ensemble des fonctionnalités que doivent être implémenté, puis nous allons détailler chaque cas d'utilisation qui fera l'objet d'une définition préalable. Nous décrirons l'intention des acteurs lorsqu'ils utilisent le système et la séquence des actions principales qu'il est susceptible d'effectuer. Ces définitions servent à clarifier les idées et ne visent pas à définir un processus complet et irréversible.

IV.3.2 Diagramme de cas d'utilisation de l'administrateur

La figure suivante représente le diagramme de cas d'utilisation de l'administrateur (admin).

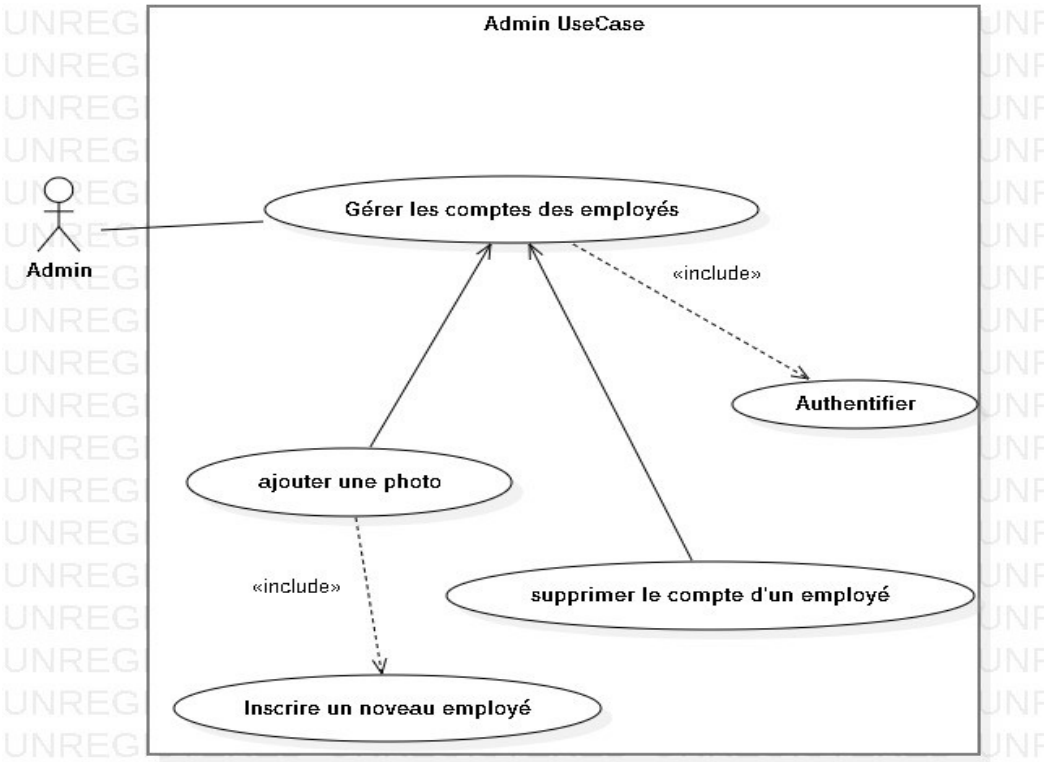

Figure IV.3-2: Diagramme de cas d'utilisation 'Admin'.

Un administrateur peut effectuer les taches présentées dans le tableau ci-dessous

Cas d'utilisation	Messages émis/reçus
créer un nouveau compte employé	**Emet :** demande d'inscription. **Reçoit :** validation d'inscription si : le nom de l'employé n'existe pas au niveau de la base de donnée et son mot de passe respecte les conditions notés sur le formulaire d'inscription ; Sinon Reçoit inscription non valide.
Ajouter une photo de l'employé (après la création de son compte)	**Emet :** demande d'ajouter photo. **Reçoit :** La caméra s'ouvre pour prendre des photos du nouvel employé en temps réel, afin de compléter le processus d'ouverture de son compte.
Supprimer un compte employé	**Emet :** demande de suppression. **Reçoit :** compte supprimé.

Tableau IV.3-1 : Privilèges de l'administrateur

IV.3.2.1 Créer un nouveau compte employé

- **Sommaire d'identification :** le sommaire résume les propriétés du cas d'utilisation 'Créer un nouveau compte employer'.
    - **But :** Ouvrir un nouveau compte sécurisé et personnel pour le nouveau employé de l'entreprise afin que ce dernier puisse entrer au système.
    - **Acteur :** administrateur (admin).
    - **Description des enchainements :** cette description a pour but de déterminer les conditions préalable au déclenchement du cas d'utilisation ; 'Créer un nouveau compte employer' doit être spécifié.
    - **Prés-condition :** L'administrateur doit s'authentifier.
    - **Scénario nominal :** Ce cas d'utilisation commence lorsque l'administrateur demande au système de créer un nouveau compte.

L'administrateur remplit le formulaire par le nom d'utilisateur (le nom et prénom de l'employé/pseudo) puis l'employé remplit le champ de mot de passe (Tout en respectant les conditions mentionnées) et valide l'inscription.

Pour finir l'inscription, l'administrateur ajoute une photo de l'employé en temps réel.

- **Post-condition :** L'utilisateur accède à son compte personnel.

IV.3.2.2 Ajouter une photo de l'employé

- **Sommaire d'identification :** présenter le but du cas d'utilisation « ajouter une photo employé » et les acteurs principaux.
    - **But :** prendre une photo de l'employé.
    - **Acteur :** Administrateur.
- **Description des enchainements :** l'objectif c'est de présenter les prés-conditions du cas d'utilisation « ajouter une photo employé» et le scénario nominal et les conditions.

- **Prés-condition :** - L'administrateur doit s'authentifier

-Il doit y avoir un compte valide pour le nouvel employé, et l'employé doit être présent pour être photographié (en temps réel).

- **Scénario nominal :** Ce cas d'utilisation commence lorsque l'administrateur et l'employé (remplit le champ mot de passe) terminent la création du compte.

- **Post-condition :** -La caméra fonctionne.

  -Prendre quelques photos.

Le cas d'utilisation « Ajouter une photo d'employé » : nous ne pouvons pas y arriver sans assurer que le cas d'utilisation « Créer un nouveau compte employer » est exécuté. En d'autres termes, le compte d'employé doit d'abord être disponible (crée avec succès).

IV.3.2.3 Supprimer un compte employé

- **Sommaire d'identification :** il permet de présenter le but du cas d'utilisation « Supprimer un compte employé » et les acteurs principaux.
  - **But :** Supprimer un compte d'un employé (qui ne travaille plus pour l'entreprise par exemple un retraité) pour qu'il ne puisse pas accéder au système.
  - **Acteur :** Administrateur.
- **Description des enchainements :** l'objectif est de présenter les prés-conditions du cas d'utilisation « Supprimer un compte employé » et le scénario nominal et les post-conditions.
  - **Prés-condition :** L'administrateur doit s'authentifier.
  - **Scénario nominal :** Ce cas d'utilisation commence lorsque l'administrateur accède au système. L'administrateur consulte la liste des comptes des employés de la société puis il sélectionne le compte qu'il souhaite. Enfin, il confirme la suppression.
- **Post-condition :** Le compte de l'employé est supprimé.

IV.3.2.4 Accès au système

Les employés et l'administrateur de l'entreprise doivent s'authentifier avant d'accéder au système.

La figure suivante représente le diagramme de cas d'accès au système par un employé.

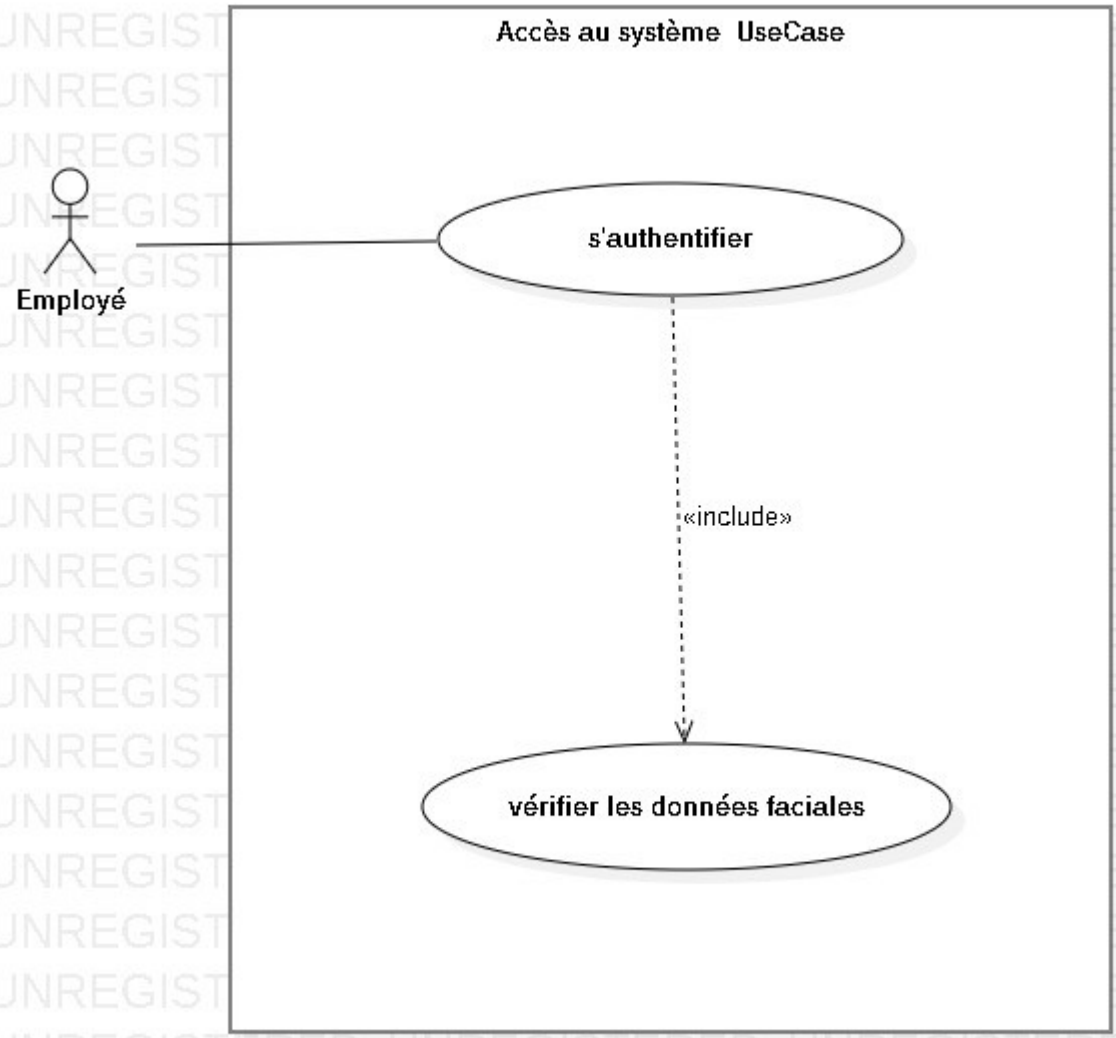

Figure IV.3-3: Diagramme de cas d'utilisation « accès au système ».

- **Sommaire d'identification :** l'objectif est de présenter le but du cas d'utilisation « Accès au système » ainsi que les acteurs.
  - **But :** Accéder à l'espace personnel.
  - **Acteur :** Employé.
- **Description des enchainements :** le but est de présenter des prés-conditions au déclenchement du cas d'utilisation « Accès au système », un scénario nominal.
  - **Prés-condition :** L'employé doit être inscrit (le compte existe).

- **Scénario nominal :** Ce cas d'utilisation commence lorsque l'employé accède au système. L'employé doit saisir ses informations (Nom d'utilisateur et mot de passe).
- **Condition (reconnaissance faciale) :** la caméra s'actionne pour détecter le visage de l'employé pour le comparer avec sa photo au niveau de la base de données.

S'il y a une correspondance entre les deux visages (de la même personne) le système accède au compte sinon il l'envoie à la page d'accueil. Ce processus vise à protéger le compte d'employé.

[Exception1 : champ obligatoire]: le système déclenche une exception sur un champ obligatoire en envoyant un message d'erreur à l'employé pour l'informer que des champs n'ont été pas remplis.

- **Post-condition :** Accéder à l'espace personnel de l'employé.
  - ✓ Le système ajoute l'employé à la liste des employés connectés.

IV.4 Diagrammes de séquence

Dans cette partie nous allons présenter les diagrammes de séquence qui détaillent les cas d'utilisation mentionnés ci-dessus.

IV.4.1 Diagramme de séquence « authentification admin »

Ce diagramme de séquence représente le scénario d'authentification de l'administrateur de système qui se résume dans les étapes suivantes :

1. L'administrateur demande la page d'authentification.
2. Le serveur application doit afficher login.
3. L'administrateur saisit son identifiant et mot de passe.
4. Le système vérifie identification et mot de passe saisi par l'administrateur.
5. Le serveur application doit vérifier l'existence des données de l'utilisateur.
6. Si les données sont erronées, le système réaffiche la fenêtre d'accueil.
7. Si les données sont correctes, le système donne la main à l'utilisateur pour accéder à son espace personnel.

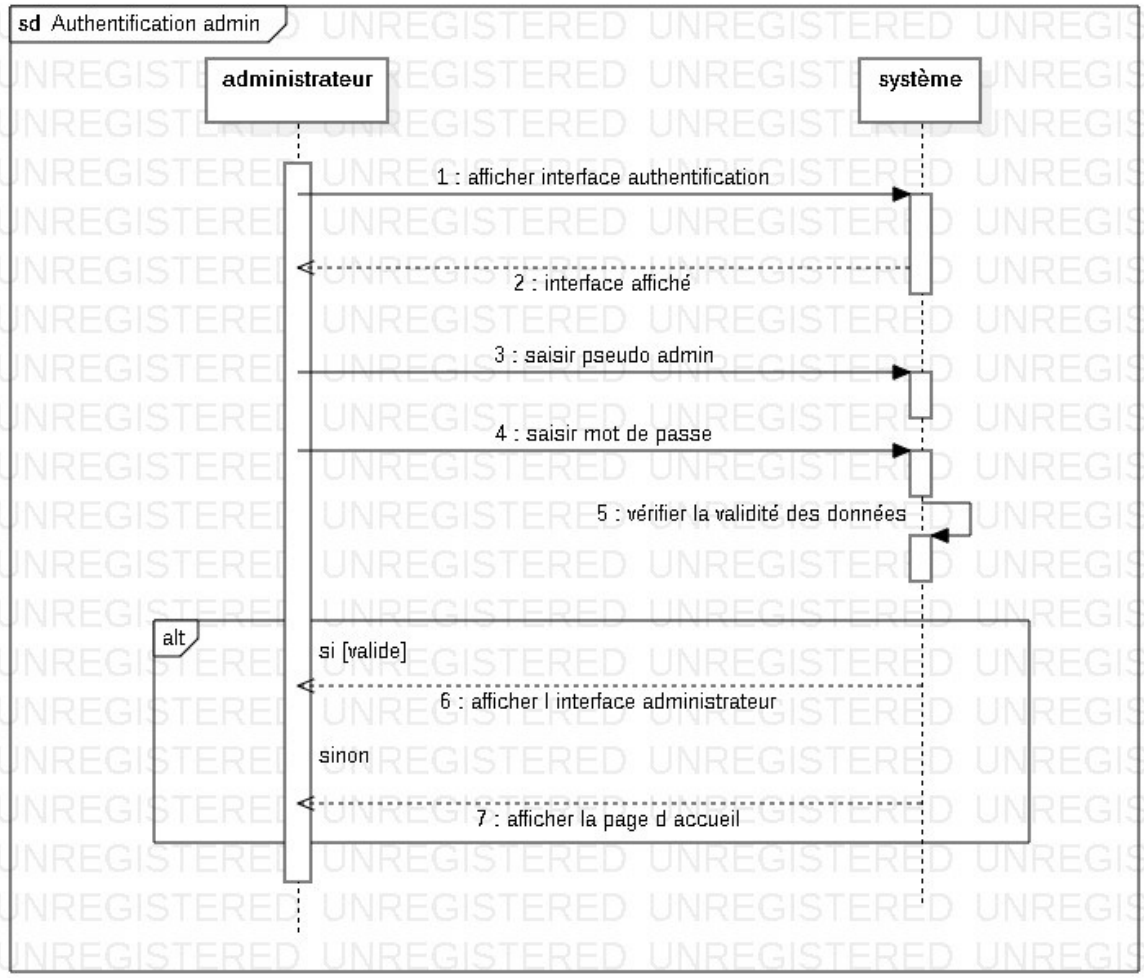

Figure IV.4-1: Diagramme de séquence « authentification admin ».

IV.4.2 Diagramme de séquence « ajouter compte employé »

Ce diagramme de séquence représente le scénario d'ajout d'un nouveau compte pour le nouvel employé par l'administrateur qui est déjà authentifié, et se résume dans les étapes suivantes :

1. L'administrateur demande le formulaire d'ajout au serveur application.
2. Le serveur application affiche le formulaire d'ajout.
3. L'administrateur remplie le formulaire.
4. Le système vérifie l'existence de ce compte.
5. Si le compte existe déjà, le système envoie un message à l'administrateur pour l'informé que le compte existe déjà « pas d'enregistrement ».
6. Sinon, le système enregistre les données.

7. Le système affiche la page insertion photo.
8. L'administrateur capture la photo de l'employé (le propriétaire de ce compte).
9. Le système doit insérer la photo de l'employé renommé avec son identifiant.
10. Le système affiche l'espace de l'administrateur.

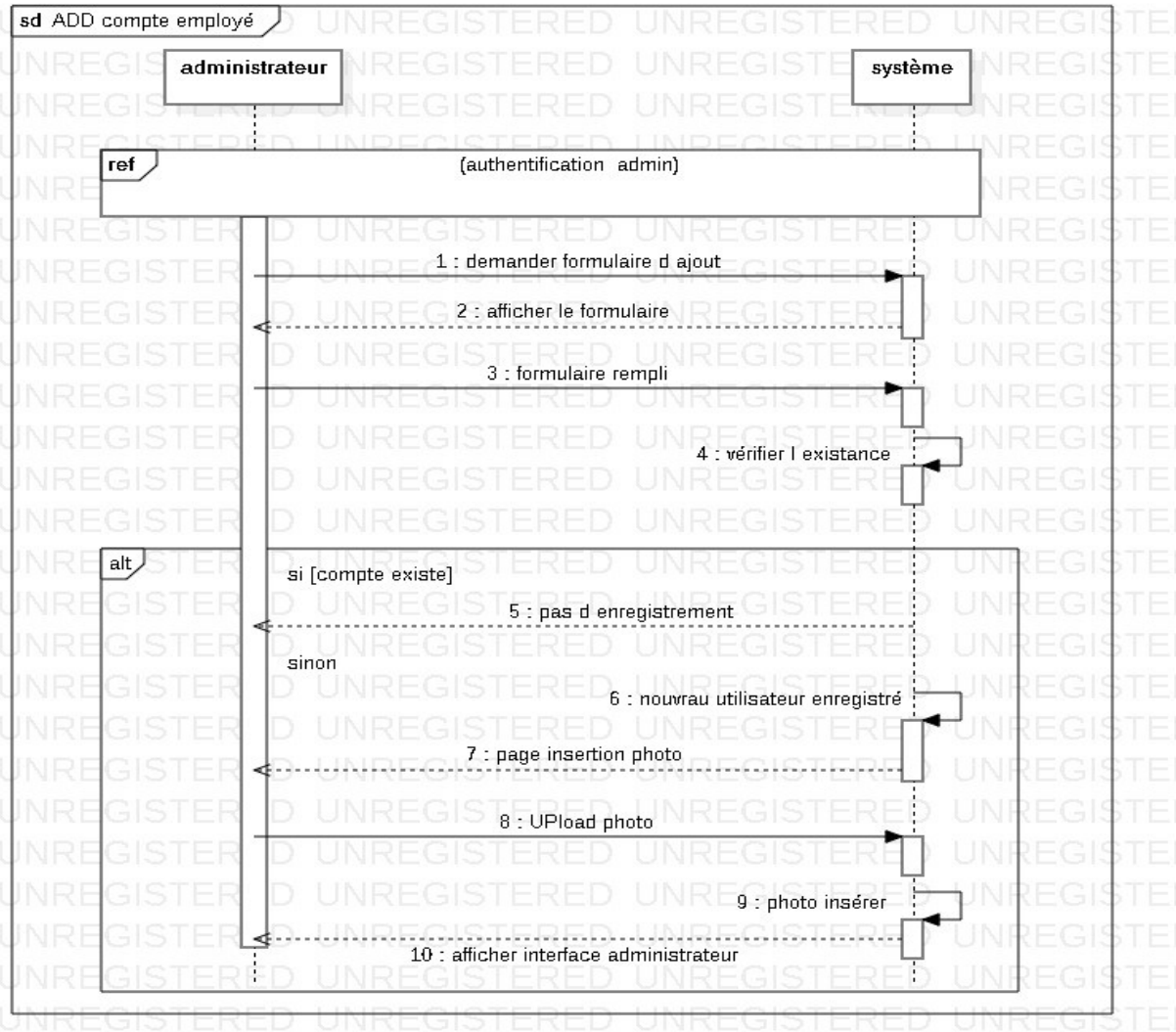

Figure IV.4-2: Diagramme de séquence « Ajouter compte employer ».

IV.4.3 Diagramme de séquence « supprimer compte employé »

Ce diagramme de séquence représente le scénario de suppression d'un compte par l'administrateur qui doit être authentifié, et se résume dans les étapes suivantes :

1. L'administrateur demande le compte de l'employé.
2. Le serveur fait une recherche pour vérifier l'existence de compte.
3. Si le compte existe, l'administrateur valide la suppression le compte sera supprimé.
4. Le serveur affiche l'espace de l'administrateur.

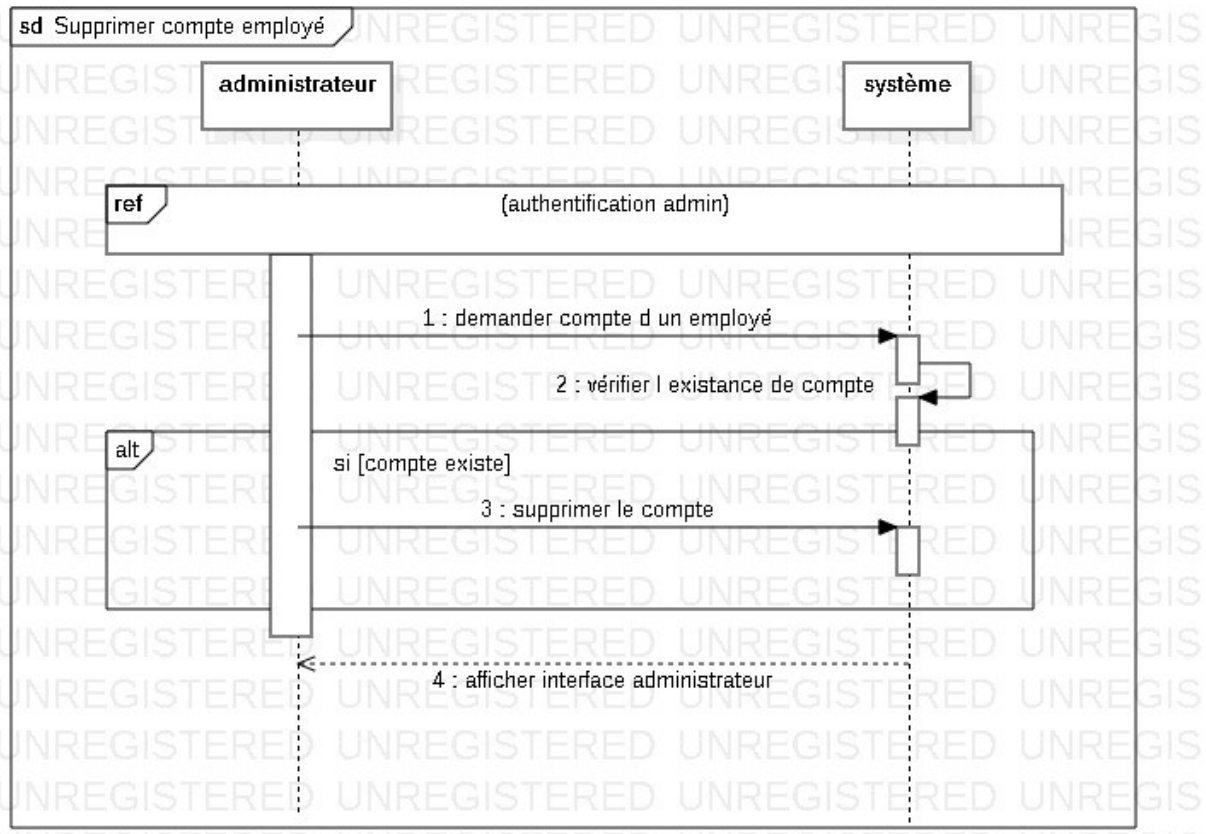

Figure IV.4-3: Diagramme de séquence « Supprimer compte employé ».

IV.4.4 Diagramme de séquence « authentifier l'employé »

Ce diagramme de séquence représente le scénario d'authentification des employés qui se résume dans les étapes suivantes :

1. L'employé demande la page login (envois l'URL au système).
2. Le système vérifie l'URL envoyé par l'utilisateur.
3. Le système répond par l'affichage de la page d'authentification.
4. L'employé saisit son identifiant et mot de passe.
5. Le serveur contacte la de base de données pour vérifier l'existence des données saisies de l'utilisateur et renvoie les résultats au système.

6. Si les données sont correctes, la caméra s'allume automatiquement pour détecter son visage.
7. Le serveur doit vérifier les données faciales.
8. Si les données faciales sont valides, Le système donne la main à l'employé pour accéder à son espace personnel
9. Si les données faciales sont erronées, le système réaffiche la fenêtre de l'accueil.
10. Si ce n'est le propriétaire de ce compte, le système réaffiche la fenêtre de l'accueil.

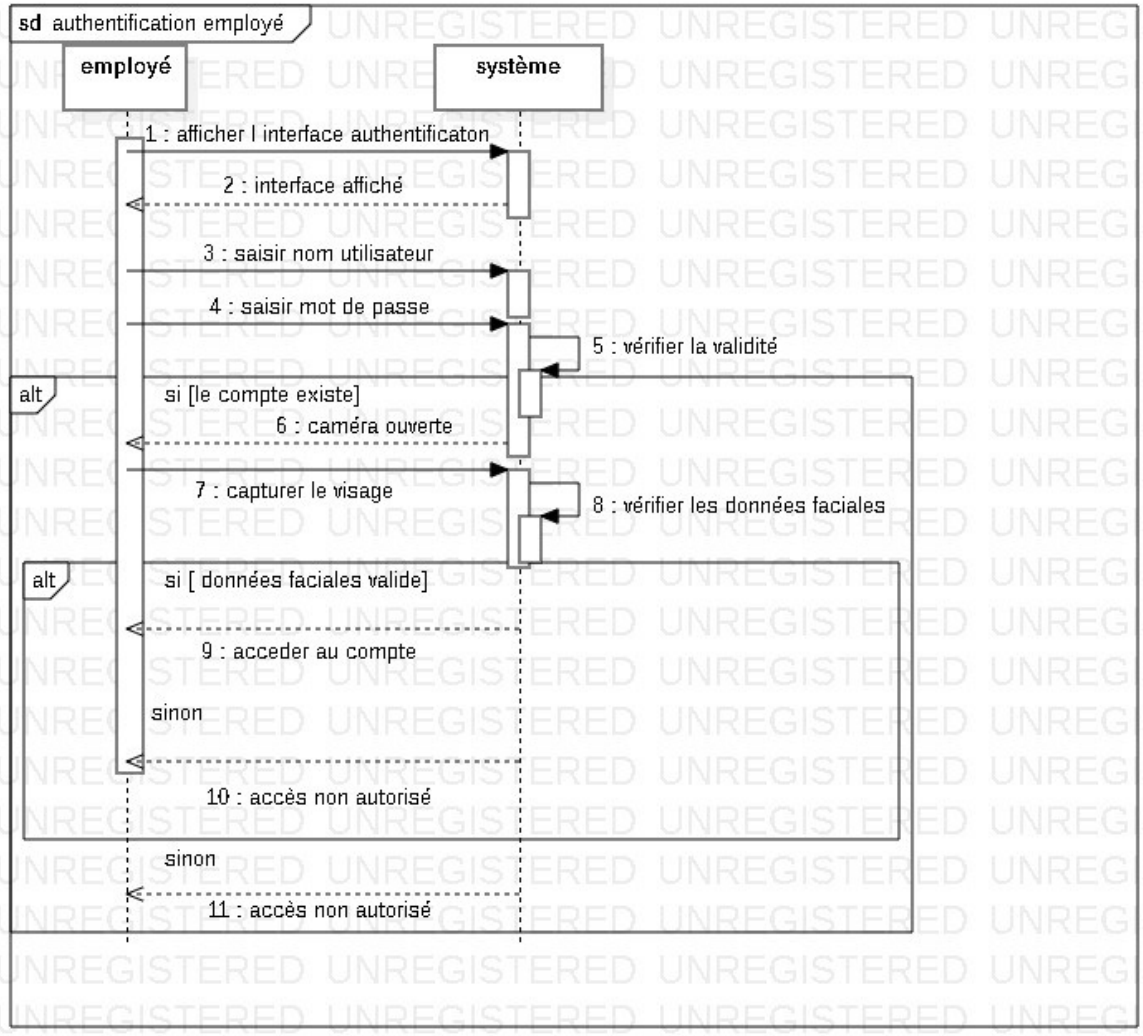

Figure IV.4-4 : Diagramme de séquence 's'authentifier'.

IV.5 Conclusion

Dans ce chapitre, nous avons présenté la conception de notre système de la reconnaissance faciale.

Nous avons présenté les éléments essentiels de l'architecture de notre système et décrit leurs fonctionnements en utilisant les diagrammes de UML. Nous avons aussi identifié les utilisateurs de notre application (l'administrateur et les employés), nous avons établi les diagrammes des cas d'utilisation ainsi que les diagrammes de séquence qui décrivent le comportement du système proposé.

Dans le chapitre suivant nous présenterons la partie réalisation que nous avons faite pour le bon fonctionnement de notre système de la reconnaissance faciale.

CHAPITRE V :
IMPLÉMENTATION ET RÉSULTATS

CHAPITRE V : IMPLÉMENTATION ET RÉSULTATS

V.1 Introduction

Après avoir achevé l'étape de la conception de notre application. Dans ce chapitre nous allons présenter les outils utilisés dans le développement, les fonctionnalités incluses dans le prétraitement des données et la modélisation de nos modèles ainsi que les étapes de l'implémentation de ces derniers.

Nous allons aussi présenter l'application finale et son interface afin d'introduire la méthode d'utilisation du logiciel. Pour finir nous allons voir et discuter les résultats obtenus et mesurer leurs fiabilités grâce aux tests réalisés.

L'objectif de ce travail consiste en l'implémentation d'algorithmes CNN pour la détection des visages sur des photos des employés de Sonatrach prises par la caméra d'ordinateur.

On a développé une application de sécurité afin d'éviter les vulnérabilités face à des menaces telles que les accès et les modifications non autorisés.

Le fonctionnement du système développé comporte trois étapes :

1. Création d'une base de données.

2. Implémentation du modèle de détection YOLO.

3. Reconnaître les données en temps réel.

Avant de commencer, il est important de citer le matériel utilisé et toutes les bibliothèques nécessaires.

V.2 Matériels et outils

Pour développer notre système nous avons utilisé le matériel et les logiciels suivants

V.2.1 Ressources matérielles

Nous avons travaillé sur un ordinateur personnel. Les informations suivantes représentent la configuration du matériel utilisé.

- **Ordinateur personnel**

Processeur : Intel® Core(TM) i5-3437U CPU @

1.90Hz 2.40 GHz

Mémoire RAM installé: 4.00 Go

Type du système : Système d'exploitation 64 bits

processeurx64

Édition Windows 10 Professionnel

Version 21H1

Build du système d'exploitation 19043.1889

V.2.2 Logiciels et librairies Utilisés dans l'implémentation

La mise en œuvre des différentes étapes discutées se faites avec plusieurs outils et logiciels de développement dans cette section on va parler sur tous les langages utilisés ainsi les logiciels pour implémenter et réaliser notre projet, l'utilisation des technologies reconnues et des versions stables sont nécessaires.

V.2.2.1 Python

Python est un langage de programmation de haut niveau interprété (il n'y a pas d'étape de compilation) et orienté objet avec une sémantique dynamique. Il est très sollicité par une large communauté de développeurs et de programmeurs. Python est un langage simple, facile à apprendre et permet une bonne réduction du cout de la maintenance des codes. Les bibliothèques (packages) python encouragent la modularité et la réutilisabilité des codes.

Python et ses bibliothèques sont disponibles (en source ou en binaires) sans charges pour la majorité des plateformes et peuvent être redistribués gratuitement.

V.2.2.2 Django (Framework)

Django (High-level Python web Framework) est un Framework Web Python de haut niveau gratuit et open source qui encourage un développement rapide et une conception propre et pragmatique. Il rendre le développement d'applications web simple et basé sur la réutilisation de code ; élaborer par des développeurs expérimentés, il prend en charge une grande partie des taches du travail de développement Web de sorte que vous pouvez vous concentrer sur l'écriture de votre application. La plupart des gens le connaissent sous le nom de framework MVC, Par contre Django utilise une approche légèrement différente et le framework réel utilisé est Model View Template.

Le fonctionnement de framework Django « MVC » est par suit :

- Un utilisateur demande https://localhost:8000 atterrit à notre application et se retrouve dans urls.py fichier où l'itinéraire « Home » est recherché.
- À views.py nous sommes sur le point de rechercher une vue qui pourrait traiter ces demandes.
- Si une vue est trouvée, elle tente d'interroger la base de données pour des données particulières pour ce modèle (models.py) et la renvoie à la vue.
- Une fois les données renvoyées par models.py, la vue affiche un Facing.html modèle et le renvoie au navigateur de l'utilisateur.

V.2.2.3 Dlib

Dlib est une bibliothèque moderne C++ open source implémentant une variété d'algorithmes d'apprentissage automatique, notamment la classification, la régression, le clustering, la transformation des données et la prédiction structurée. Dlib inclut une couverture étendue des tests unitaires et des exemples utilisant la bibliothèque. Chaque classe et fonction de la bibliothèque est documentée.

Dlib dispose également de fonctionnalités utilitaires, y compris :

- Filetage
- Réseautage
- Algorithmes numériques,
- Traitement d'images
- Algorithmes de compression et d'intégrité des données.

V.2.2.4 OpenCV

OpenCV (Open Source Computer Vision Library) vision par ordinateur open-source est une bibliothèque de fonctions de programmation destinées à la vision par ordinateur en temps réel. Il cible généralement le traitement d'images, la reconnaissance de visages, la capture vidéo, la recherche et la divulgation d'objets. Développé à l'origine par Intel, il a ensuite été soutenu par Willow Garage puis Itseez (qui a ensuite été acquis par Intel). La bibliothèque est multiplateforme et gratuite pour une utilisation sous la licence open-source Apache 2.

OpenCV est créé pour mettre en œuvre diverses opérations, notamment la reconnaissance faciale et la détection de visages, l'analyse des tâches humaines dans des vidéos, l'identification d'objets, l'enregistrement de mouvements de caméra, le suivi d'objets en mouvement et la combinaison d'images pour créer une image haute résolution pour une scène précise.

V.2.2.5 SQLiteStudio

SQLiteStudio est un gestionnaire de la base de données SQLite fiable et complet qui nous fournit un moyen très simple de gérer le contenu de chaque base de données, y compris les tables, les vues et les déclencheurs.

SQLiteStudio est pratique pour les utilisateurs qui ont besoin de modifier et de gérer les bases de données SQLite en insérant des nouveaux index, vues, tables et déclencheurs.

V.2.2.6 SQLite

SQLite est une bibliothèque écrite en « langage C » qui propose un moteur de base de données relationnelle accessible par le langage SQL.

SQLite implémente en grande partie le standard SQL-92 et des propriétés ACID. Contrairement aux serveurs de bases de données traditionnels, comme par exemple MySQL, sa particularité est de ne pas reproduire un schéma habituel client-serveur mais d'être directement intégrée aux programmes. L'intégralité de la base de données (les index, les tables, les déclarations et les données) est stockée dans un fichier indépendant de la plateforme.

V.2.2.7 CMake

Le nom « CMake » est l'abréviation de « cross platform make » est un système de construction logicielle multiplateformes ; Il permet de vérifier les prérequis nécessaires à la construction, de déterminer les dépendances entre les différents composants d'un projet, afin de planifier une construction ordonnée et adaptée à la plateforme. La construction du projet est ensuite déléguée à un logiciel spécialisé dans l'ordonnancement de taches et spécifique à la plateforme, Microsoft Visual Studio, Make, ou Ninja.

CMake reprend le concept de configuration initié par autoconf, mais s'en distingue par son caractère multiplateforme, le rendant particulièrement adapté à la construction des logiciels destinés à fonctionner sous Windows et Linux.

Exemples de quelques logiciels utilisant CMake :

- MySQL (sous Windows seulement)
- Netflix
- OGRE (à partir de la version 1.7)
- OpenCV
- Polycode
- ReactOS
- Scribus
- VTK
- wmfs
- Wormux

V.2.2.8 HTML

Le langage de balisage hypertexte, ou HTML, est le langage de balisage standard pour les documents conçus pour être affichés dans un navigateur Web. Il peut être assisté par des technologies telles que les feuilles de style en cascade (CSS) et des langages de script tels que JavaScript.

Les navigateurs Web reçoivent des documents HTML d'un serveur Web ou d'un stockage local et restituent les documents dans des pages Web multimédias. HTML décrit la structure d'une page Web de manière sémantique et inclut à l'origine des indices pour l'apparence du document.

V.2.2.9 CSS

Les feuilles de style en cascade (CSS) sont un langage de feuille de style utilisé pour décrire la présentation d'un document écrit dans un langage de balisage tel que HTML. CSS est une technologie de base du World Wide Web, aux côtés de HTML et JavaScript.

CSS est conçu pour permettre la séparation de la présentation et du contenu, y compris la mise en page, les couleurs et les polices. Cette séparation peut améliorer l'accessibilité du contenu, offrir plus de flexibilité et de contrôle dans la spécification des caractéristiques de présentation, permettre à plusieurs pages Web de partager le formatage en spécifiant le CSS pertinent dans un fichier .CSS séparé, ce qui réduit la complexité et la répétition du contenu structurel et permet le fichier .CSS à mettre en cache pour améliorer la vitesse de chargement des pages entre les pages qui partagent le fichier et sa mise en forme.

V.2.2.10 JavaScript

JavaScript, souvent abrégé en JS, est un langage de programmation conforme à la spécification ECMAScript. JavaScript est de haut niveau, souvent compilé juste à temps et multi-paradigme. Il a une syntaxe entre accolades, un typage dynamique, une orientation objet basée sur des prototypes et des fonctions de première classe.

Avec HTML et CSS, JavaScript est l'une des technologies de base du World Wide Web. Plus de 97 % des sites Web l'utilisent côté client pour le comportement des pages Web, incorporant souvent des bibliothèques tierces.

La plupart des navigateurs Web disposent d'un moteur JavaScript dédié pour exécuter le code sur l'appareil de l'utilisateur.

En tant que langage multi-paradigmes, JavaScript prend en charge les styles de programmation événementiels, fonctionnels et impératifs. Il dispose d'interfaces de programmation d'applications (API) pour travailler avec du texte, des dates, des expressions régulières, des structures de données standard et le modèle objet de document (DOM).

V.2.2.11 Bootstrap

Bootstrap est un Framework CSS gratuit et open source destiné au développement Web frontal réactif et mobile. Il contient des modèles de conception CSS et (éventuellement)

JavaScript pour la typographie, les formulaires, les boutons, la navigation et d'autres composants d'interface.

V.2.2.12 UML

Le langage de modélisation unifié (UML) est un langage de modélisation de développement à usage général dans le domaine du génie logiciel qui vise à fournir un moyen standard de visualiser la conception d'un système.

La création d'UML a été motivée à l'origine par le désir de standardiser les systèmes de notation disparates et les approches de conception de logiciels. Il a été développé chez Rational Software en 1994-1995, avec un développement ultérieur mené par eux jusqu'en 1996.

V.2.2.13 Microsoft visuel studio

Microsoft Visual Studio est un environnement de développement intégré (IDE) de Microsoft. Il est utilisé pour développer des programmes informatiques, ainsi que des sites Web, des applications Web, des services Web et des applications mobiles. Visual Studio utilise des plates-formes de développement de logiciels Microsoft telles que Windows API, Windows Forms, Windows Presentation Foundation, Windows Store et Microsoft Silverlight. Il peut produire à la fois du code natif et du code managé. Visual Studio inclut un éditeur de code prenant en charge IntelliSense (le composant de complétion de code) ainsi que la ré-factorisation de code. Le débogueur intégré fonctionne à la fois comme débogueur au niveau de la source et comme débogueur au niveau de la machine. D'autres outils intégrés incluent un profileur de code, un concepteur pour la création d'applications GUI, un concepteur Web, un concepteur de classe et un concepteur de schéma de base de données. Il accepte les plug-ins qui étendent les fonctionnalités à presque tous les niveaux, y compris l'ajout de la prise en charge des systèmes de contrôle de source (comme Subversion et Git) et l'ajout de nouveaux ensembles d'outils tels que des éditeurs et des concepteurs visuels pour des langages spécifiques à un domaine ou des ensembles d'outils pour d'autres aspects du développement logiciel. Cycle de vie (comme le client Azure DevOps : Team Explorer).

V.2.2.14 Pillow

La bibliothèque pillow pour les traitements d'image en langage python, cette bibliothèque permet l'ouverture, la manipulation et la sauvegarde des images et permet de

faire de nombreuse autre fonctionnalités, la version actuelle identifie et permit de lire un grand nombre de format d'images comme par exemple : jpg, png, tiff, gif …

Les étapes de l'utilisation :

- la création de l'objet image
- la création de la photo image
- la création d'un label sur lequel on affiche l'image

V.2.2.15 PyTorch

PyTorch est une bibliothèque d' IA, elle est développée par Meta (ex Facebook), écrite en langage Python pour se lancer dans le deep learning (l' apprentissage profond) et le développement

de réseaux de neurones artificiels. À partir de plusieurs variables, elle peut servir à réaliser des calculs de gradients ou à utiliser des tableaux multidimensionnels obtenus grâce à des tenseurs.

PyTorch est disponible depuis 2016 en open source sous licence BSD Modifiée ;En 2018 , La librairie est fusionnée par Meta avec Caffe2, son infrastructure de deep learning taillée pour les déploiements et capable de prendre en charge des algorithmes d'apprentissage ingérant jusqu'à des dizaines de milliards de paramètres.

V.2.2.16 TensorFlow

TensorFlow est un outil open source d'apprentissage automatique développé par Google. Il est fondé sur l'infrastructure DistBelief, et est doté d'une interface pour Python, Julia et R.

TensorFlow fournit des API stables en langage Python et en C. Ces API sans rétro-compatibilité garantie en C++, Java, JavaScript, Go et Swift. Des packages faits par des tiers sont disponibles sur plusieurs de langages de programmation comme par exemple Julia, R et Haskell.

V.2.2.17 Implémentation du YOLOv5

YOLOv5 est une famille d'architectures et de modèles de détection d'objets préformés sur l'ensemble de données COCO, et représente la recherche open source Ultra lytics sur les futures méthodes d'IA de vision, intégrant les leçons apprises et les meilleures pratiques évoluées au cours de milliers d'heures de recherche et développement.

Après implémentation de l'architecture YOLO version 5 sur notre plateforme et après entrainement sur cette base de données on a choisi quelques images de cette dernière hors celles utilisées pour l'entrainement.

V.3 Implémentation du système

Dans cette section nous allons présenter l'interface graphique du système réalisé.

V.3.1 Page d'accueil

Commençons par la première page essentielle : la page d'accueil

.

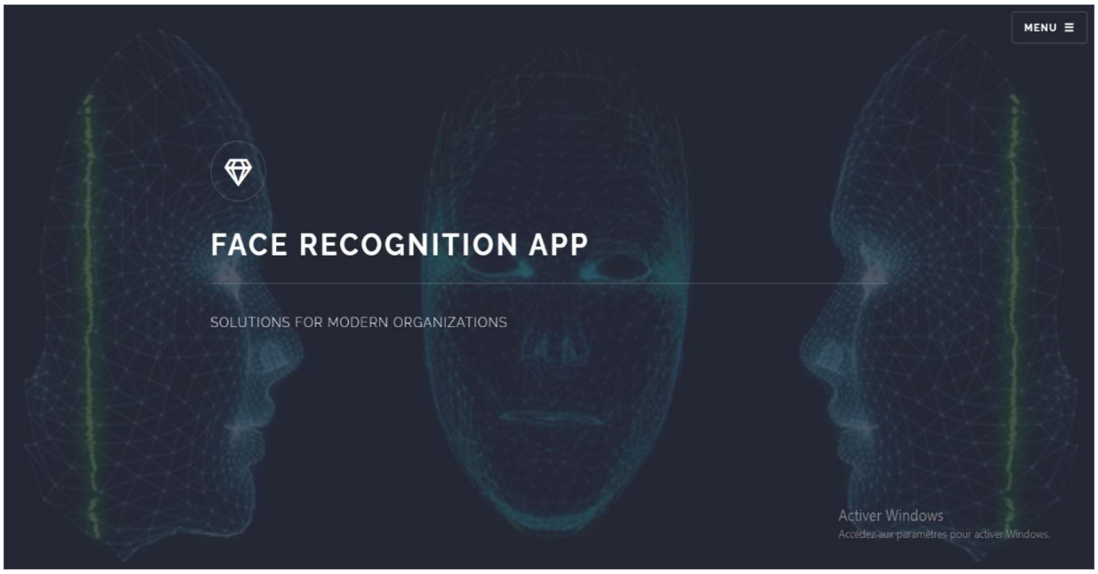

Figure V.3-1: Capture de la page d'accueil.

Cette interface contient le nom de notre application et un bouton essentiel « MENU», et ce dernier comprend les options suivantes :

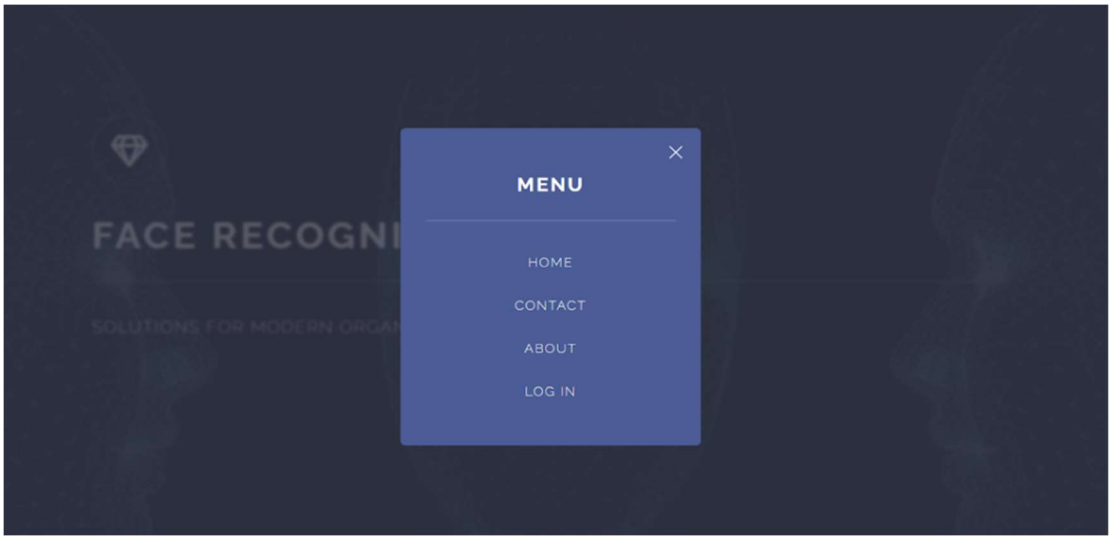

Figure V.3-2: Capture des options MENU.

V.3.2 Authentification

C'est la page d'authentification de tous les utilisateurs de l'application (administrateur et les employés). Il est montré dans la figure ci-dessous la capture de l'authentification de l'administrateur son pseudo est admin.

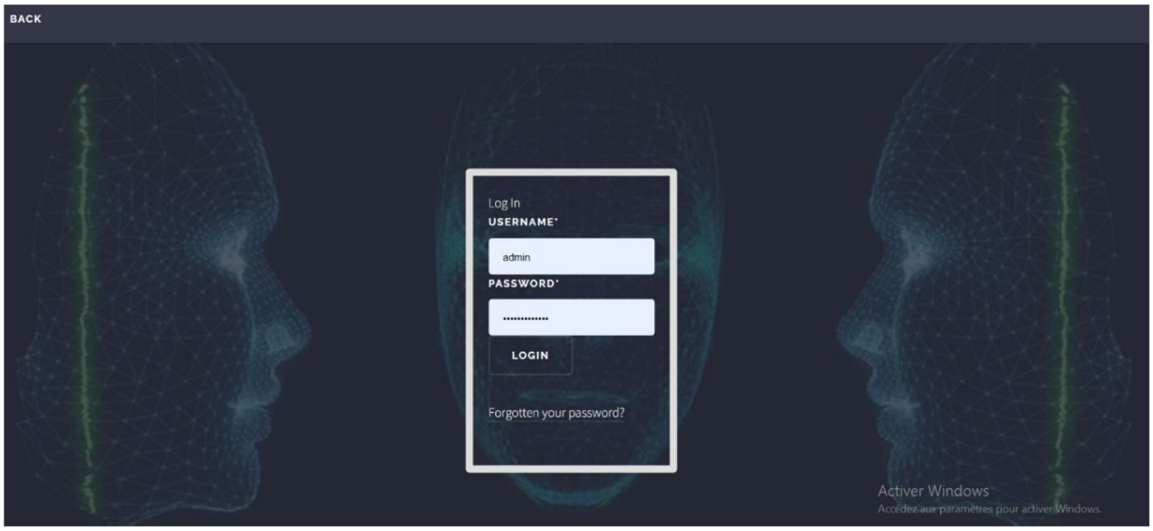

Figure V.3-3: capture de la page login de l'administrateur.

Lorsque l'administrateur est authentifié avec succès, il se dirige vers sa propre page (il sera sur la page WELCOME ADMIN) où il peut gérer les comptes des employés.

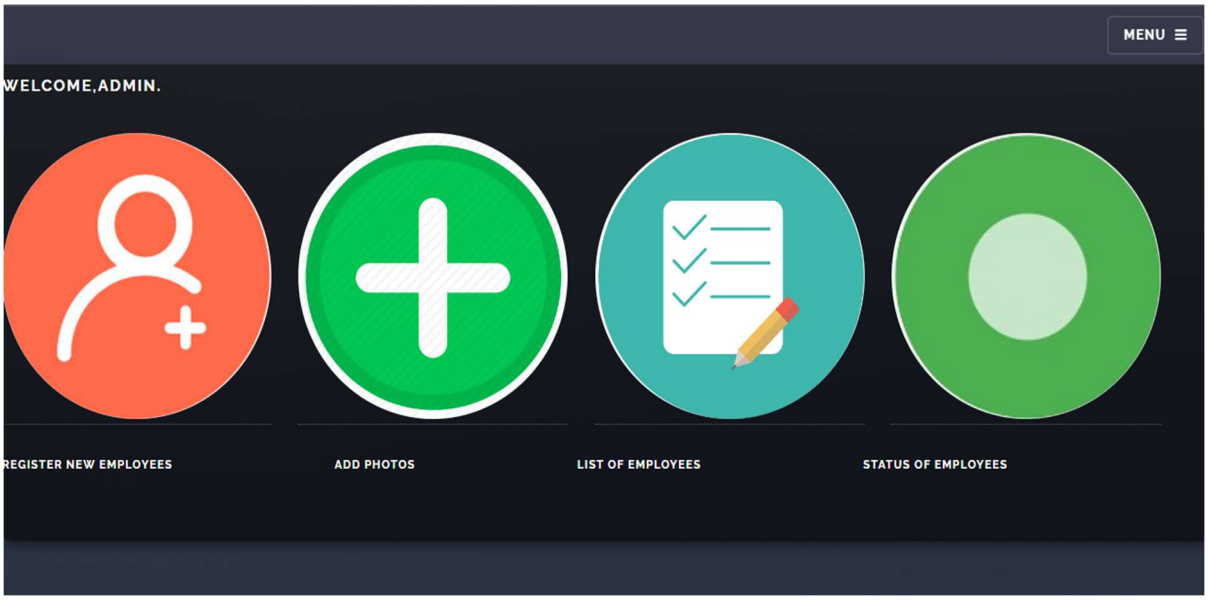

Figure V.3-4: Capture compte admin.

La figure ci-dessous montre que l'administrateur à cliquer sur le bouton REGISTER NEW EMPLOYEES (sur la figure compte admin)

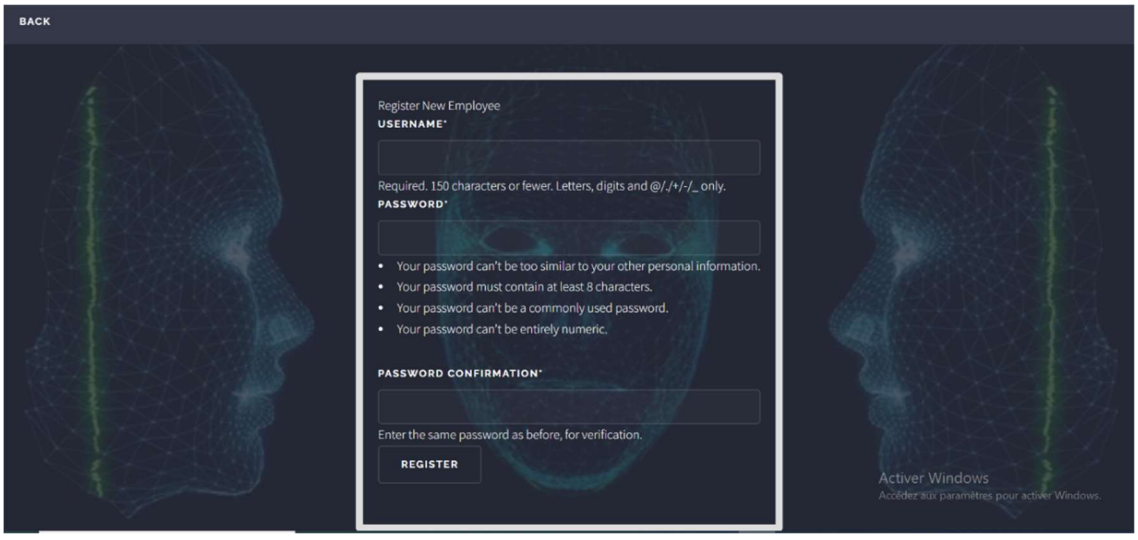

Figure V.3-5: Capture page d'inscription d'un nouvel employé.

Une fois que le processus de saisie le nom d'utilisateur et le mot de passe a été terminé avec succès, L'appareil photo s'ouvrira automatiquement pour prendre la première photo de l'employé et l'enregistrer dans un fichier Media/persons. Pour utiliser cette photo chaque fois qu'il veut se connecter à son propre compte via son ordinateur dans l'entreprise.

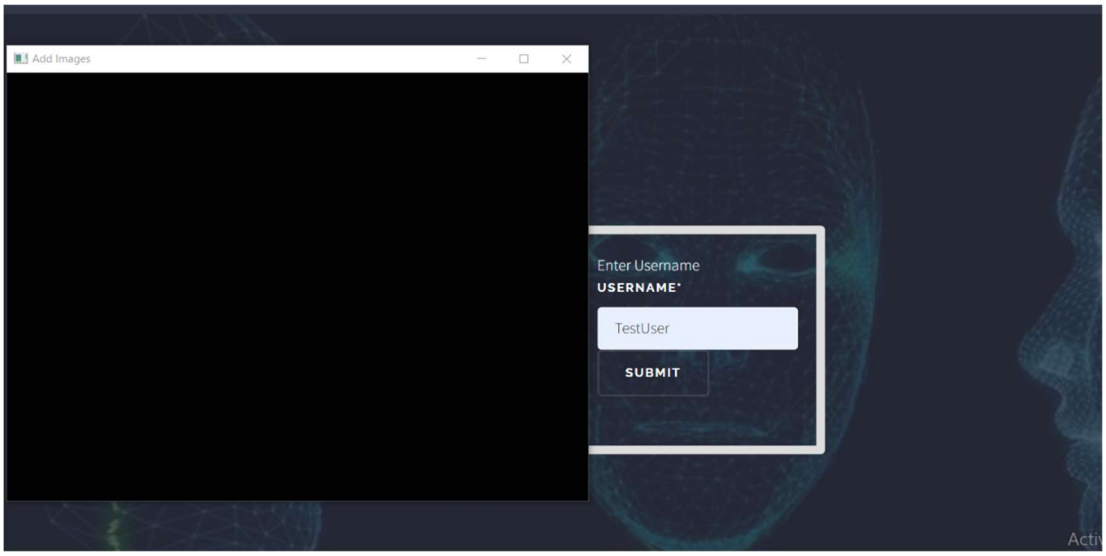

Figure V.3-6: Capture de fenêtre de caméra ouverte de l'employé TestUser.

L'administrateur peut consulter la liste de tous les comptes des employées. Une liste lui apparaît avec le nom des employées et leurs courriels ainsi qu'un bouton afin qu'il puisse effacer le compte d'un travailleur dont le rapprochement avec l'entreprise a expiré.

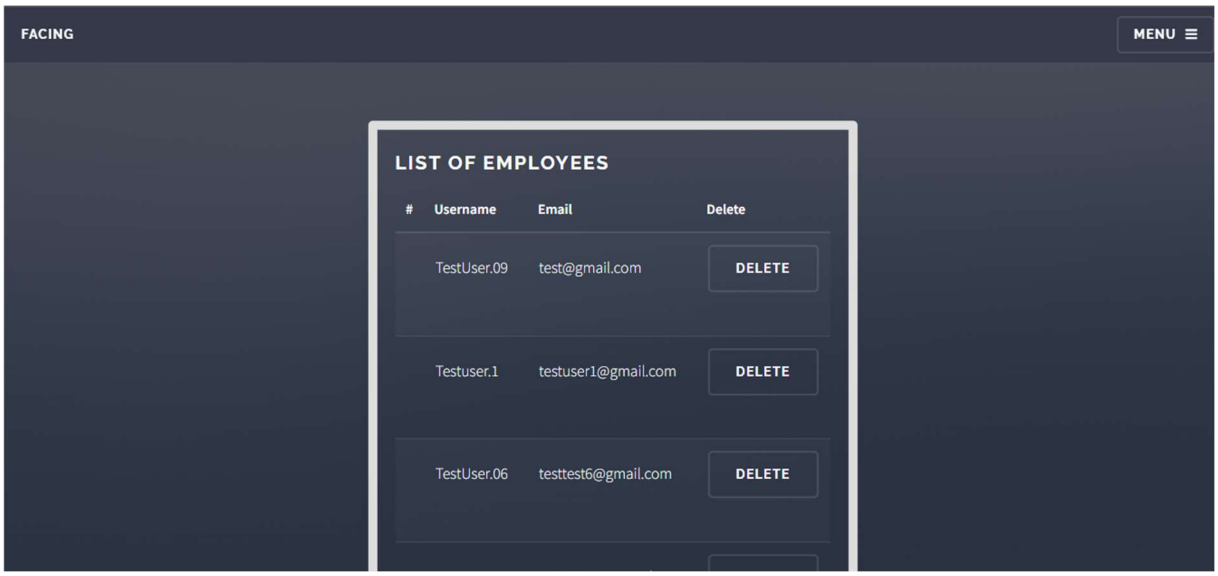

Figure V.3-7 : Capture de la liste des employées

L'administrateur peut aussi consulter la liste de tous les comptes des employées connecté au niveau de l'entreprise (les employées en ligne). La figure ci-dessus montre que l'employé TestUser.09 est en ligne.

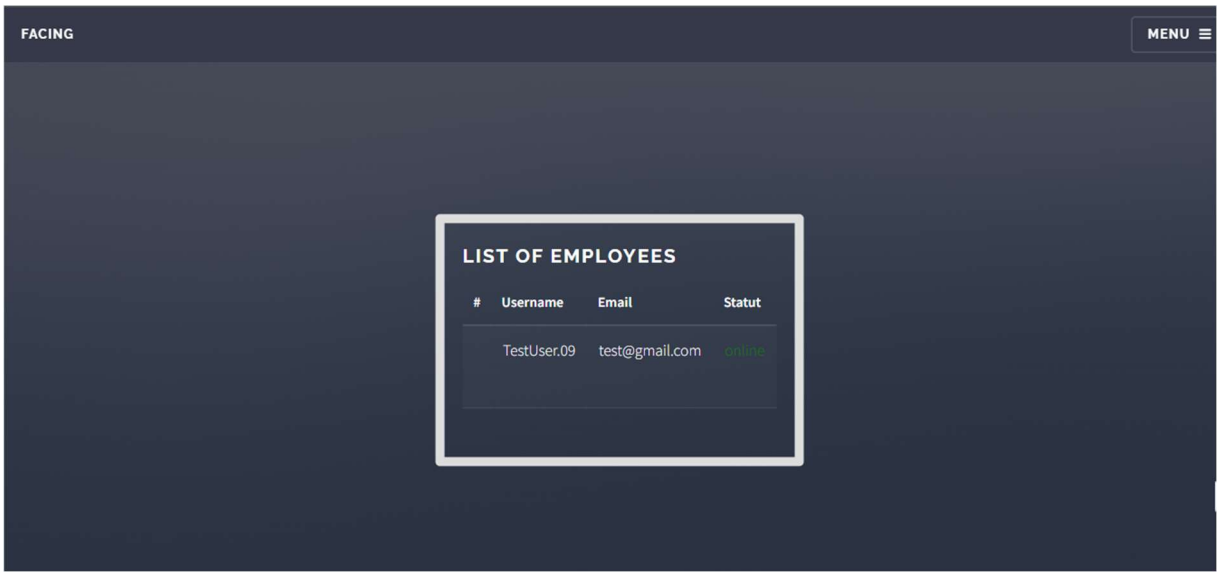

Figure V.3-8: Capture la liste des employées en ligne.

L'administrateur reçoit des notifications : Tentatives répétées d'entrer le compte de l'un des employés soit par erreur en tapant le mot de passe, soit même après l'absence de la reconnaissance de son visage.

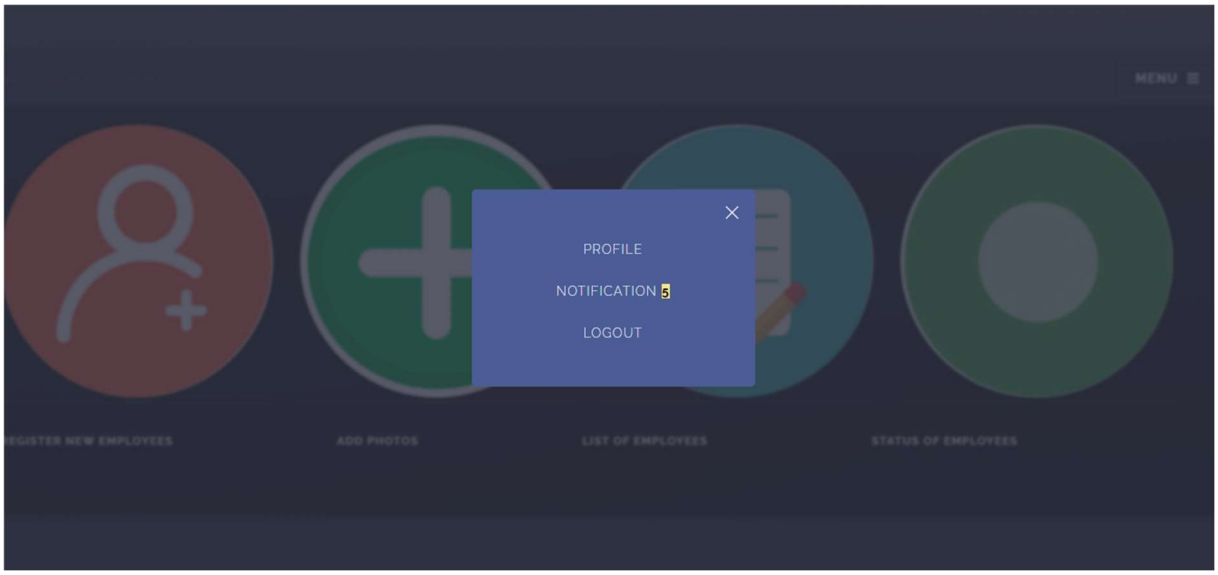

Figure V.3-9 : Capture de menu administrateur.

Figure V.3-10 : Capture de la liste des notifications reçus.

L'administrateur peut supprimer les notifications reçus bien sûr après avoir alerté l'employé de la tentative de quelqu'un d'autre d'accéder à son compte.

La figure ci-dessous montre la connexion d'une autre personne que le propriétaire du compte TestUser.09. Donc le résultat est « bad authentification ».

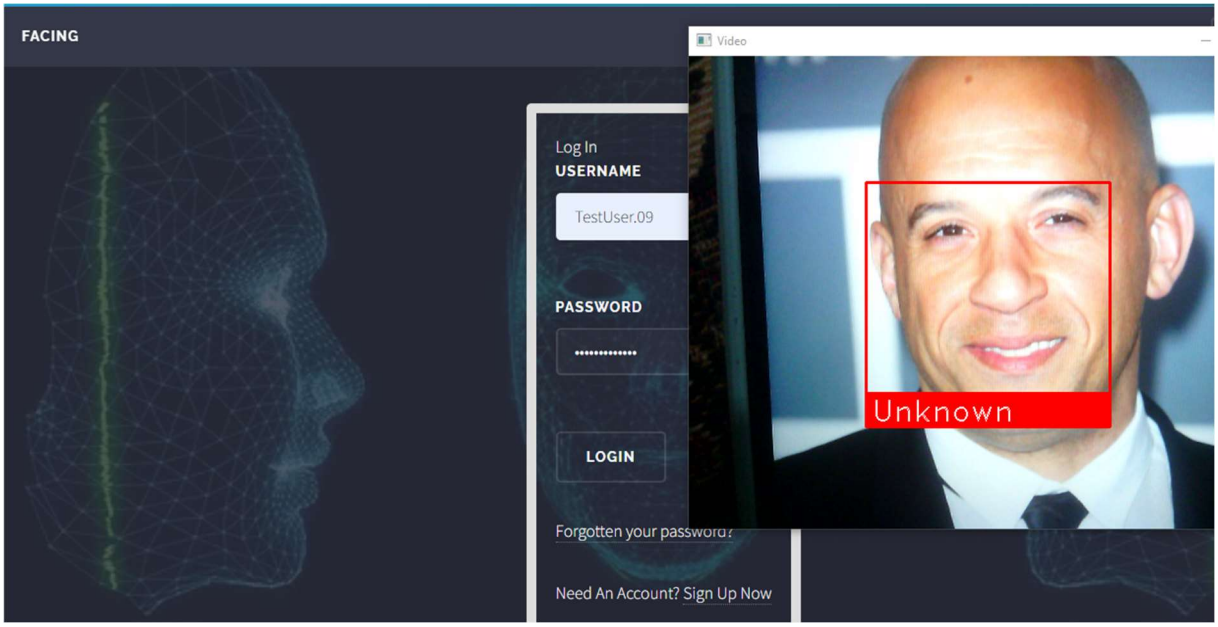

Figure V.3-11: Capture bad authentification.

La figure ci-dessous représente que l'authentification a été effectuée avec succès par le propriétaire TestUser.09.

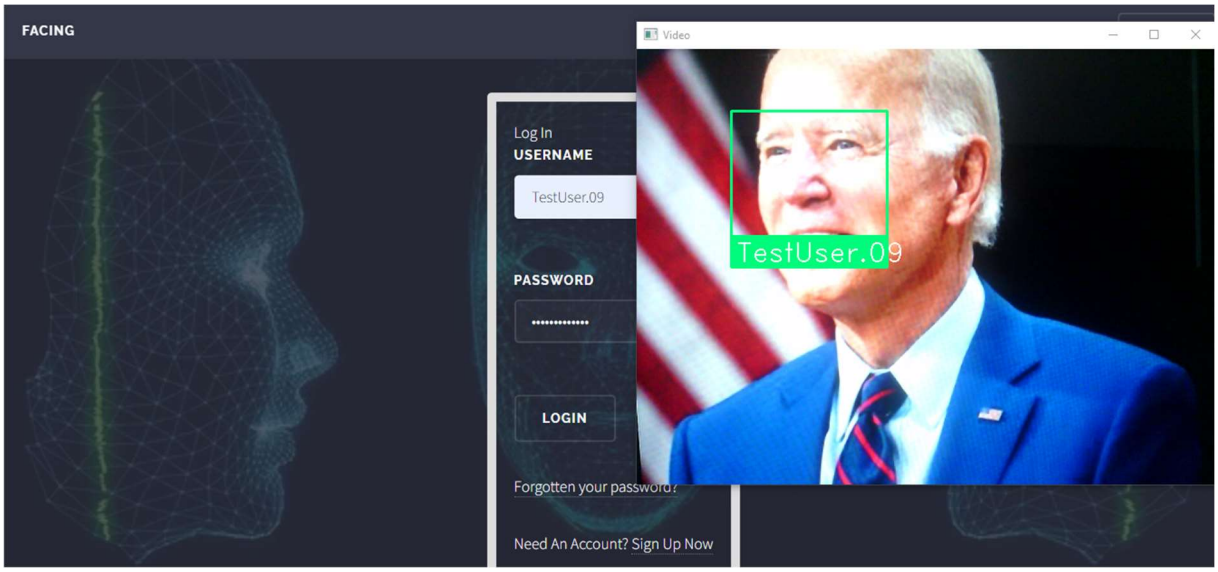

Figure V.3-12: Capture d'authentification avec succès.

V.4 Création de la base de données

On a utilisé Django comme Framework, l'architecture de Django basée sur le MVC.

Notre root contient manage.py pour exécuter la base de données par le cmd (l'invite des commandes).

V.4.1 Modèle

Sur le dossier modele.py au niveau de notre projet on a défini les classes ; ces derniers nous les considérons comme des tables de la base de données.

- **CMD :**

Au début, Nous utilisons l'invite des commandes pour accéder au dossier par la commande : cd chemin complet de dossier.

Après, la commande de la création de la base de données :

python manage.py makemigration nommigration

Ensuite la commande :

python manage.py migration.

V.4.2 View

Les vues sur notre projet sont également les Template (au niveau de dossier user).

V.4.3 Controler

C'est le traitement au niveau de « views.py ».

La figure ci-dessous montre une capture de notre base de données sur SQLiteStudio

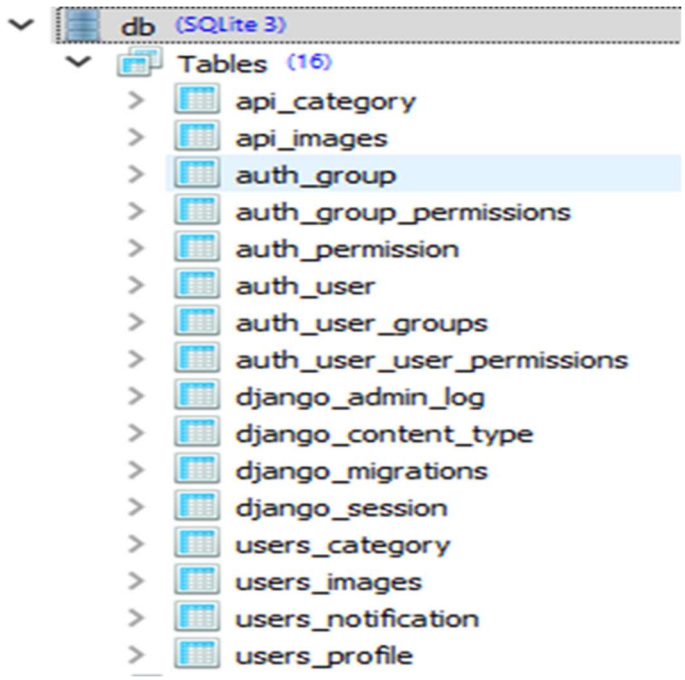

Figure V.4-1 : Capture les tables de la base de données

V.5 Test et résultats

V.5.1 Création d'un réseau de neurone siamois

La première chose qu'on a fait, on a créé notre propre réseau de neurone siamois pour la reconnaissance d'image à prise unique. Mais nous n'avons pas travaillé avec pour les raisons suivantes :

1. Une grande quantité de données d'entraînement en raison d'un si grand nombre de paires de classes.

2. Très consommateur en ressources de calcul

3. Non rapide.

4. Sensible à certaines variations de l'entrée.

5. et surtout le manque de matériel

V.5.1.1 L'implémentation

- **Pile technologique :**

Langage utilisé : Python

Packages utilisés : TensorFlow, Keras, Numpy, etc.

Pour créer notre propre SNN (Simease Neural Network), on a passé par trois étapes principales :

- La constitution d'une base de données d'images qui servira à l'entraînement et à l'évaluation du réseau,
- Le choix d'une architecture de réseau de neurones,
- L'entraînement du réseau à partir de la base de données.

➢ **Étape 01 :** Importation des packages

Tout d'abord, nous avons importé les packages requis. Pour une liste complète des versions de package utilisées sur la machine virtuelle pour exécuter ce code

```
#Import tensorflow dependencies -Functionnal API
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Layer, Conv2D, Dense, MaxPooling2D, Input, Flatten
import tensorflow as tf
```

Figure V.5-1 : code python « importer les packages »

- **Étape 02 :** Importation des données

Ensuite, nous avons importé un ensemble de données pour que notre SNN puisse travailler avec.

Les données sont organisées en triplets d'images : Positif, Négatif et Ancre(anchor).

Le dossier « anchor » contient 3000 captures de visage d'une seule personne

Le dossier « positive » contient 3000 captures de visage de la même personne que le dossier anchor

Le dossier « négative » contient 13000 photos de différents visages de personnes dans le monde

Nous avons généré des paires (positives et anchor), et des paires (négatives, anchor)

Entrée : image anchor, image positive (le même visage)

Sortie : 1

Entrée : image anchor, image négatif (des visages différents).

Sortie : 0

```
positives = tf.data.Dataset.zip((anchor, positive, tf.data.Dataset.from_tensor_slices(tf.ones(len(anchor)))))
negatives = tf.data.Dataset.zip((anchor, negative, tf.data.Dataset.from_tensor_slices(tf.zeros(len(anchor)))))
data = positives.concatenate(negatives)
```

Figure V.5-2 : code python « génération des paires »

Notre modèle standard est un réseau de neurones convolutionnels siamois avec des L couches chacune avec des N_l unités où $h_{1,L}$ représente le vecteur caché dans la couche l pour le premier jumeau, et $h_{2,l}$ dénote la même chose pour le deuxième jumeau. Nous utilisons exclusivement des unités linéaires rectifiées (ReLU) dans les premières L - 2 couches et des unités sigmoïdales dans les couches restantes.

- **Étape 03 :** Créer le modèle

Le modèle se compose d'une séquence de couches convolutionnelles, chacune utilisant un canal unique avec des filtres de taille variable et une foulée fixe de 1. Le nombre de filtres

convolutionnels est spécifié comme un multiple de 16 pour optimiser les performances. Le réseau applique une fonction d'activation ReLU aux cartes de fonctionnalités de sortie, éventuellement suivie d'un max pooling avec une taille de filtre et une enjambée de 2.

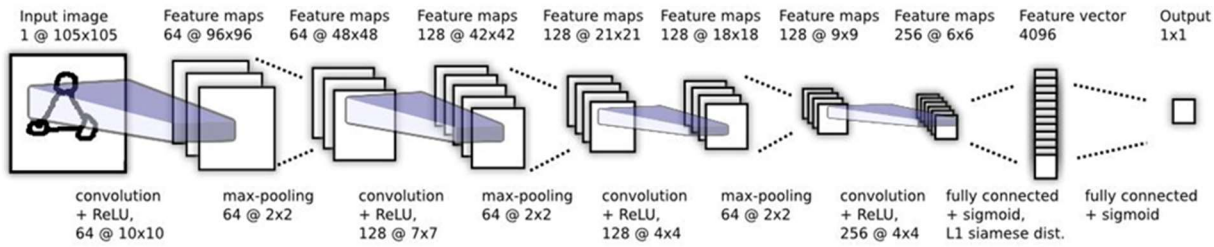

Figure V.5-3 : L'architecture convolutionnelle pour la tâche de vérification.

Le jumeau siamois n'est pas représenté dans l'image ci-dessus, mais se joint immédiatement après l'unité 4096 entièrement connectée couche où la distance L1 dans le sens des composants entre les vecteurs est calculé.

Les unités de la dernière couche de convolution sont aplaties en un seul vecteur. Cette couche convolutionnels est suivie d'une couche entièrement connectée, puis d'une couche supplémentaire calculant la métrique de distance induite entre chaque jumeau siamois, qui est donnée à une seule unité de sortie sigmoïdale

```
Entrée [103]: def make_embedding():
                  inp = Input(shape=(100,100,3), name='input_image')

                  # First block
                  c1 = Conv2D(64, (10,10), activation='relu')(inp)
                  m1 = MaxPooling2D(64, (2,2), padding='same')(c1)

                  # Second block
                  c2 = Conv2D(128, (7,7), activation='relu')(m1)
                  m2 = MaxPooling2D(64, (2,2), padding='same')(c2)

                  # Third block
                  c3 = Conv2D(128, (4,4), activation='relu')(m2)
                  m3 = MaxPooling2D(64, (2,2), padding='same')(c3)

                  # Final embedding block
                  c4 = Conv2D(256, (4,4), activation='relu')(m3)
                  f1 = Flatten()(c4)
                  d1 = Dense(4096, activation='sigmoid')(f1)


                  return Model(inputs=[inp], outputs=[d1], name='embedding')
```

Figure V.5-4 : Capture du code python utilisé pour créer les couches d'intégration

Ensuite, nous avons créé la couche L1 Dist qui calcule la distance de similarité entre 2 images.

```python
# Siamese L1 Distance class
class L1Dist(Layer):

    # Init method - inheritance
    def __init__(self, **kwargs):
        super().__init__()

    # Magic happens here - similarity calculation
    def call(self, input_embedding, validation_embedding):
        return tf.math.abs(input_embedding - validation_embedding)
```

Figure V.5-5 : code python "fonction distance"

Ensuite, nous avons créé un modèle siamois qui reçoit une image, le transmet séquentiellement au modèle d'intégration pour l'intégration, puis transmet les incorporations résultantes à la fonction de perte.

```python
def make_siamese_model():

    # Anchor image input in the network
    input_image = Input(name='input_img', shape=(100,100,3))

    # Validation image in the network
    validation_image = Input(name='validation_img', shape=(100,100,3))

    # Combine siamese distance components
    siamese_layer = L1Dist()
    siamese_layer._name = 'distance'
    distances = siamese_layer(embedding(input_image), embedding(validation_image))

    # Classification layer
    classifier = Dense(1, activation='sigmoid')(distances)

    return Model(inputs=[input_image, validation_image], outputs=classifier, name='SiameseNetwork')
```

Figure V.5-6 : code python « modèle siamois »

- **Étape 04 :** Fonctions de perte utilisées dans le réseau siamois

Le réseau siamois utilise le score de similarité pour prédire si les deux entrées sont similaires ou différentes en utilisant une approche d'apprentissage des métriques, qui trouve la distance relative entre ses entrées.

On a calculé le score de similarité à l'aide de l'entropie croisée binaire, « Binary Cross entropy »

```
binary_cross_loss = tf.losses.BinaryCrossentropy()
```

Figure V.5-7 : code python "fonction de perte"

➢ **Étape 05 :** Entrainement de modèle

On est maintenant prêts à former notre modèle.

```
@tf.function
def train_step(batch):

    # Record all of our operations
    with tf.GradientTape() as tape:
        # Get anchor and positive/negative image
        X = batch[:2]
        # Get label
        y = batch[2]

        # Forward pass
        yhat = siamese_model(X, training=True)
        # Calculate loss
        loss = binary_cross_loss(y, yhat)
    print(loss)

    # Calculate gradients
    grad = tape.gradient(loss, siamese_model.trainable_variables)

    # Calculate updated weights and apply to siamese model
    opt.apply_gradients(zip(grad, siamese_model.trainable_variables))

    # Return loss
    return loss
```

Figure V.5-8 : code python "Fonction d'entrainement"

V.5.1.2 Résultat

Pour le réseau de neurone siamois qu'on a créé, la reconnaissance faciale est plutôt satisfaisante, mais nous avons rencontré quelques problèmes dans la phase d'implémentation, l'utilisation d'un CPU a fait que le temps d'exécution était trop couteux. à cause de ressource matériel . Donc On n'a pas travaillé avec le réseau qu'on a créé.

On a trouvé une autre méthode permettant d'accélérer le calcul et d'étendre son emploi en automatisant l'augmentation des données afin de gagner en temps et en énergie.

V.5.2 Discussion de résultats

1. La reconnaissance faciale avec OpenCV :

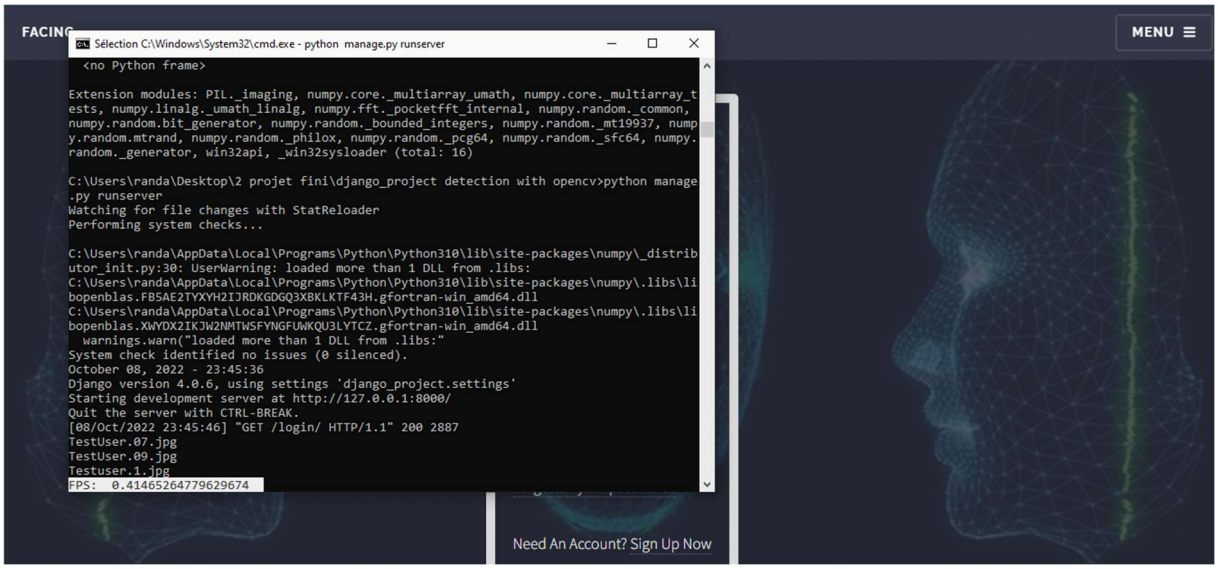

Figure V.5-9 : résultat de calcule FPS pour OpenCV

Comme montré sur la figure précédente on constate que la reconnaissance faciale avec OpenCV est plutôt satisfaisante, mais le temps de calcul pour classifier une image est long.

Il est préférable de trouver d'autres méthodes permettant d'accélérer le calcul et d'étendre son emploi, et de trouver des méthodes automatisant l'augmentation des données afin de gagner en temps et en énergie.

2. La reconnaissance faciale avec détection YOLO et identification avec l'algorithme siamois :

La figure ci-dessus montre le temps de détection des visages (FPS) par la caméra de chaque authentification de l'employée TestUser.09

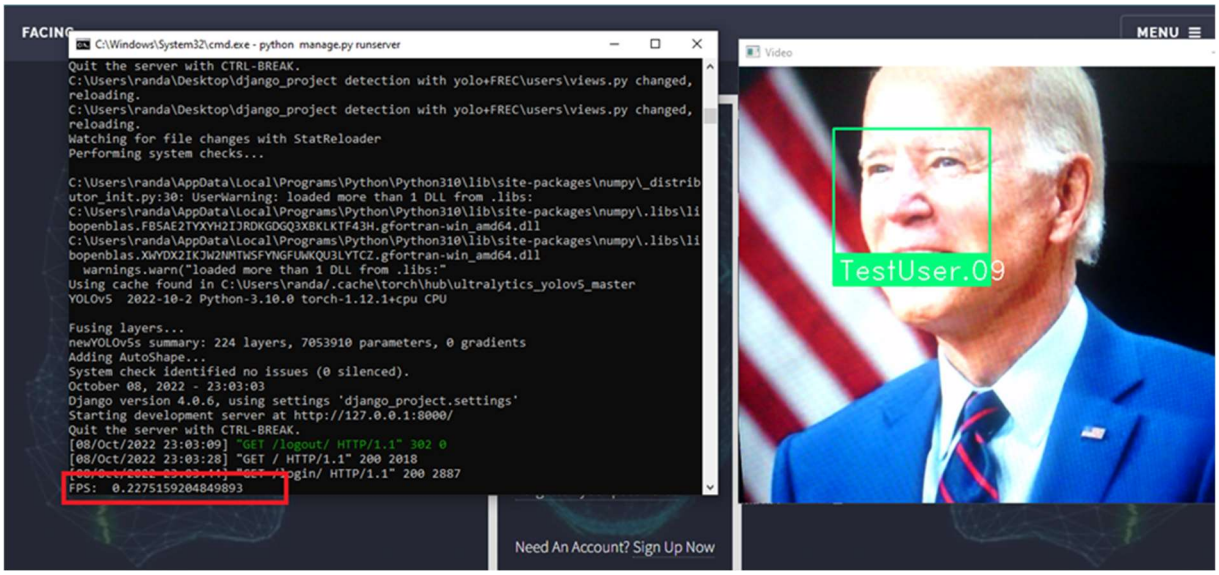

Figure V.5-10 : Résultat de calcul FPS pour l'algorithme siamois

Nous concluons donc que le FPS de la reconnaissance faciale avec l'algorithme siamois est plus petit par rapport à OpenCV, donc notre solution n'est pas couteuse en termes de temps.

V.6 Conclusion

La détection des visages est considérée comme l'un des problèmes les plus difficiles dans le domaine de la reconnaissance faciale, car ça implique la combinaison de la classification de visage de l'employé en temps réel (login) et son visage qui existe sur la base de données (un employé après l'inscription),dans ce chapitre on s'est focalisé sur l'architecture YOLO en version 5 et on l'a implémenté sur machine, et après avoir enrichi notre base de données et entrainé notre réseau de neurones qui consiste en une version de YOLO en version 5 modifiée et en utilisant cette base de données équipé d'une caméra au niveau de la machine (ordinateur…) de chaque employé.

On a effectué des tests qui ont donné des résultats concluants lesquels étaient affichés sur une interface graphique qu'on a conçue pour ces fins.

CONCLUSION GÉNÉRALE

CONCLUSION GÉNÉRALE

Ce travail illustre le fonctionnement d'un système de reconnaissance et d'authentification faciale. Il permet de reconnaître les intrus et d'authentifier le personnel d'une entreprise. L'application en question doit répondre à des exigences de rapidité et de robustesse des résultats.

L'application implémenté basée sur les réseaux de neurones convolutifs. Elle est capable de détecter des objets ainsi que de reconnaître des visages. Le modèle YOLO (You Only Look Once) a été utilisé pour détecter les objets dans l'image. Le réseau siamois identifie la personne en se référant à la base de données des personnes connues.

Dans ce mémoire, nous avons discuté des notions fondamentales de réseau et les différents mécanismes pour le sécurisé. On a abordé quelques notions de base en matière d'imagerie numérique et de la vision par ordinateur. On a parlé des réseaux de neurones en générale et des réseaux de neurones convolutionnels en particulier.

Ce mémoire de Master 2 nous a permet de :

- Effectuer des recherches sur les différents types de réseaux de neurones existants et collecter une base de données qu'on a enrichie et augmenté afin de l'utiliser dans l'entrainement de notre réseau de neurone convolutionnel.
- Implémenter l'architecture YOLO version 5 qui est un algorithme de détection d'objets caractérisé par sa rapidité ainsi que son utilisation en temps réel.
- Concevoir une interface graphique afin de pouvoir visualiser les performances de notre réseau de neurones artificiels convolutionnels aisément et ça nous a permis de maitriser un outil puissant en programmation orienté objet qui est le python.

RÉFÉRENCES BIBLIOGRAPHIQUES

[1] « Sécurité du réseau », *Wikipédia*. 30 janvier 2020. Consulté le: 9 mai 2022. [En ligne]. Disponible sur: https://fr.wikipedia.org/w/index.php?title=S%C3%A9curit%C3%A9_du_r%C3%A9seau&oldid=166883664

[2] « PFE-Master_Zineb_BENDELLA.pdf ». Consulté le: 11 mai 2022. [En ligne]. Disponible sur: http://dspace.univ-tlemcen.dz/bitstream/112/4798/1/PFE-Master_Zineb_BENDELLA.pdf

[3] S. Elgharbi, S. Samir, Y. Yasmine, et B. Ghenima, « Mise en place d'un IDS pour sécuriser un réseau en utilisant Snort », p. 81.

[4] « L'image numérique (archi.fr) & Computer vision ou vision par ordinateur : tout savoir sur cette technologie d'IA (lebigdata.fr) ».

[5] « Peyré, Gabriel. Le traitement numérique des images, Images des Mathématiques. Paris,France : CNRS, 2011 ».

[6] « C. Chan, J.K.A. Chu, J. K. Aggarwal. Transactions on Pattern Analysis and Machine Intelligence. s.l. : IEEE, 1993. ».

[7] « Lee Daechul, Park Moon Ho. Jacket Matrix Based Recursive Fourier Analysis and Its Applications. »

[8] « Bouchra, Khefif. Mise au point d'une application de reconnaissance faciale. 28 novembre 2013. »

[9] « Hanane, OUAMANE. Identification de reconnaissance faciale avec des expressions. Mémoire de Fin d'Etudes de diplôme Master. Biskra : s.n., 07 Juin 2012. »

[10] « M.Vineetha Sai, G.Varalakshmi, G.Bala Kumar, J.Prasad. FACE RECOGNITION SYSTEM WITH FACE DETECTION. 2011. Jawaharlal Nehru Technological University Kakinada. ».

[11] « Y.Ma, X.Ding. Face Detection based on hierarchical support vector machines. s.l. : IEEE, 2002. »

[12] « https://www.kaspersky.fr/resource-center/definitions/what-is-facial-recognition ».

[13] « Ababsa, Souhila Guerfi. Authentification d'individus par reconnaissance de caractéristiques biométriques liées aux visages 2D/3D. THÈSE pour obtenir le titre de Docteur de l'Université Evry Val d'Essonne. France : s.n., 2008. »

[14] « Quelles sont les différences entre le Deep learning et le Machine learning ? », *IONOS Digital Guide*. https://www.ionos.fr/digitalguide/web-marketing/search-engine-marketing/deep-learning-vs-machine-learning/ (consulté le 14 septembre 2022).

[15] la rédaction de Futura, « Définition | Intelligence artificielle - IA | Futura Tech », *Futura*. https://www.futura-sciences.com/tech/definitions/informatique-intelligence-artificielle-555/ (consulté le 8 septembre 2022).

[16] B. L, « Réseau de neurones artificiels : qu'est-ce que c'est et à quoi ça sert ? », *LeBigData.fr*, 5 avril 2019. https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition (consulté le 10 septembre 2022).

[17] « Réseaux de Neurone Artificiel », *pdfcoffee.com*. https://pdfcoffee.com/reseaux-de-neurone-artificiel-pdf-free.html (consulté le 11 septembre 2022).

[18] « Perceptron multicouche », *Wikipédia*. 13 avril 2022. Consulté le: 3 octobre 2022. [En ligne]. Disponible sur: https://fr.wikipedia.org/w/index.php?title=Perceptron_multicouche&oldid=192824773

[19] « Rétropropagation », *Data Analytics Post*. https://dataanalyticspost.com/Lexique/retropropagation/ (consulté le 4 octobre 2022).
[20] « Deep Learning : définition, applications, avantages et inconvénients », *Retengr*, 22 janvier 2021. https://www.retengr.com/2021/01/22/deep-learning-definitions-applications-avantages-inconvenients/ (consulté le 5 octobre 2022).
[21] « Fonction d'activation », *Wikipédia*. 30 décembre 2020. Consulté le: 13 septembre 2022. [En ligne]. Disponible sur: https://fr.wikipedia.org/w/index.php?title=Fonction_d%27activation&oldid=178207792
[22] « Qu'est-ce qu'un convolutional neural network ? », *Devenir Data Scientist*, 11 juin 2021. https://www.jeveuxetredatascientist.fr/convolutional-neural-network/ (consulté le 6 octobre 2022).
[23] « UNE INTRODUCTION CONVIVIALE AUX RÉSEAUX SIAMOIS - BLOG ». https://fr.quish.tv/friendly-introduction-siamese-networks (consulté le 13 septembre 2022).
[24] J. Redmon, S. Divvala, R. Girshick, et A. Farhadi, « You Only Look Once: Unified, Real-Time Object Detection », juin 2016, p. 779-788. doi: 10.1109/CVPR.2016.91.
[25] « YOLO : Vous ne regardez qu'une seule fois – Détection d'objets en temps réel – Acervo Lima ». https://fr.acervolima.com/yolo-vous-ne-regardez-quune-seule-fois-detection-dobjets-en-temps-reel/ (consulté le 7 octobre 2022).