République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université De Blida 1 – Saad Dahleb

Faculté des Sciences

Département Informatique

Mémoire de Master

Filière : Informatique

Spécialité Systèmes d'informatiques et réseaux

Thème

Développement d'un système de reconnaissance faciale.

Organisme d'accueil : Centre de développement des technologies avancées (CDTA).

Sujet Proposé par :

Encadreuse : Mme.AIT SADI Karima.

Promoteur : Mr.BENYAHIA Mohamed.

Présenté par :

AKCHA Ikram.

AMMARI Amira.

Soutenu le : 23/09/2020

Devant le jury composé de:

Mr. OULD KHAOUA Mohamed

Mme. FARAH Messaouda

2019/2020

Remerciements

« Avant toutes choses, nous remercions **DIEU** Grand et Puissant de nous avoir donné la volonté et le courage d'accomplir notre cursus universitaire et de mener ce travail jusqu'à la fin et cela en dépit des difficultés.

Nous tenons à remercier notre directrice de mémoire **Mme AIT SADI Karima** pour son aide ainsi que pour ses conseils précieux tout au long de cette expérience professionnelle et d'avoir relu et corrigé notre mémoire.Ses conseils de rédaction ont été très précieux. Veuillez accepter nos profonds respects.

Nous remercions également notre promoteur Mr BENYAHIA Mohamed pour son suivi tout au long de ce travail, pour ses encouragements et sa bonté.

Nos remerciements vont également aux membres du jury, vous nous faites honneur en jugeant notre humble travail. Veuillez accepter notre profonde reconnaissance.

Enfin, nous tenons à exprimer nos vifs remerciements à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail. »

Dédicaces

Je dédie ce travail, A mes très chers parents **Fatiha** et **Mohamed** qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance. J'espère qu'ils trouveront dans ce travail toute ma reconnaissance et tout mon amour, que Dieu vous accorde santé, longue vie et bonheur.

Je dédie ce travail aussi à mon très chers frère **Ouassim** pour son encouragement permanent, et son soutien moral.

Je dédie ce travail, à mes grands parents, à mes tantes et oncles, à ma cousine **Rania**, ainsi qu'à toute ma famille.

Je remercie Dieu de m'avoir entouré de personnes aussi pures, sages et sincères.

Et pour finir. Je dédie ce travail, à mon binôme et mon amie **Amira,** je la remercie pour son soutien, sa patience et son amitié.

AKCHA Ikram

Dédicaces

Je dédie ce humble travail, A mes très chers parents **Salima** et **Sohbi** et **Thouria** et **Djamel** pour tous leur sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tous au long de mes études.

Je dédie ce travail aussi à ma très chère sœur **Hanane** pour son appui et son encouragement et à mes très chers frères **Aymen** et **Mohamed** pour leur croyance en moi.

Je dédie ce travail, à ma très chère cousine **Awatif** qui a toujours été là pour moi et pour son soutien morale pendant tout au long de mon parcours universitaire, et à mes tantes et oncles ainsi qu'à toute ma famille.

Je remercie Dieu de m'avoir entouré de personnes aussi pures, sages et sincères et que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible.

Et pour finir. Je dédie ce travail, à mon binôme et mon amie **Ikram,** je la remercie pour son soutien, sa patience et son amitié.

Merci d'être toujours là pour moi.

AMMARI Amira

Résumé

Ce sujet de master s'inscrit dans la problématique de la sécurité des individus dans l'environnement extérieur ainsi que la sécurité des entreprises et la société qui consiste à contrôler l'accès à ces dernières. Afin de résoudre ce problème les techniques de l'apprentissage automatique, plus particulièrement l'apprentissage profond sont exploité ce qui surenchère des meilleurs résultats par rapport au techniques existantes dans l'état de l'art.

C'est dans ce contexte que s'insère ce projet de master 2, il s'agit de développer un système de reconnaissance faciale automatique capable de faire une détection et authentification en temps réel. En générale, le principe des approches d'authentification des visages présentées dans la littérature c'est d'extraire les caractéristiques du visage d'un individu et les compare à celle extraites et stockées au préalable dans la base de données. Le but est de trouver la meilleure ressemblance (matching).

Mot clés : Reconnaissance, apprentissage automatiques, apprentissage profond, détection

Abstract

This master's subject is related to the issue of the security of individuals in the external environment as well as the security of companies which consists in controlling access to them. In order to solve this problem the techniques of machine learning, more particularly deep learning are exploited which outbids better results compared to the techniques existing in the state of the art.

It is in this context that this master's 2 project fits, it is about developing an automatic facial recognition system capable of real-time detection and authentication. In general, the principle of the face authentication approaches presented in the literature is to extract the characteristics of an individual's face and compare them to those extracted and previously stored in the database. The goal is to find the best match.

Key words: Recognition, deep learning, machine learning, detection

ملخص

يرتبط موضوع الماستر هذا بمسألة أمن الأفراد في البيئة الخارجية بالإضافة إلى أمن الشركات التي يتمثل في التحكم في الدخول إليها. من أجل حل هذه المشكلة، يتم استغلال تقنيات التعلم الآلي، وبشكل خاص، يتم استغلال التعلم العميق الذي يعطي نتائج أفضل مقارنة بالتقنيات الموجودة من قبل. في هذا السياق، يناسب مشروع الماستر هذا، فهو يتعلق بتطوير نظام التعرف التلقائي على الوجه القادر على تحديد الوجه والتعرف عليه. بشكل عام، مبدأ أساليب التحقق من الوجه المقدمة في الأدبيات هو استخراج خصائص وجه الفرد ومقارنتها بتلك المستخرجة والمخزنة مسبقًا في قاعدة البيانات الهدف هو العثور على أفضل تطابق

كلمات مفتاحية: تحديد الوجه، التعرف التلقائي على الوجه، التعلم العميق، تقنيات التعلم الآلي

Sommaire

Table des Figures

Liste des tableaux

Introduction générale

Le monde dans lequel nous vivons aujourd'hui s'est marqué par son insécurité, et la recrudescence des attentats. La sécurité est devenus donc une préoccupation des instituions tant en étatique que en privé. En effet, Pour des raisons de sécurité publique et afin de lutter contre le risque de criminalité et de terroriste, le recours au système de la reconnaissance faciale est de plus en plus incontournable. Ce système de sécurité reste alors un privilège, notamment dans les lieux publics. La reconnaissance du visage est basée sur des données et des principes de l'intelligence artificielle. De plus en plus utilisée en raison de ses applications pratiques et efficaces dans la vie quotidienne, la reconnaissance faciale connaît aujourd'hui un grand succès en matière de surveillance et de sécurité. Au point vu recherche, cette application continue d'attirer des chercheurs du domaine de la vision artificielle et reste l'une des techniques les plus étudiées au cours des dernières décennies.

La reconnaissance faciale s'est discrètement installée dans nombreux lieux publics. C'est une technologie biométrique qui consiste à identifier un individu en fonction des caractéristiques de son visage. L'identification concerne notamment l'écartement entre les différents éléments qui forment la face du visage: les yeux, les oreilles, les lèvres, etc.).

En tant que mesure sécuritaire dans les établissements qui reçoivent du monde (les stades, salles de concert, aéroports, etc.), la reconnaissance faciale analyse plus rapidement la foule. Le principe consiste à examiner les caractéristiques du visage qui ne changent pas de manière significative lorsque l'individu vieillit, ou suite à une opération de chirurgie esthétique. Cette technologie représente un réel potentiel d'application dans divers domaines et devrait constituer un facteur de développement. Née dans les années 1970, la technologie de reconnaissance faciale est de nos jours en pleine évolution, avec des applications aussi bien pratiques que sécuritaires. En général, le système reconnaît une personne de façon automatique grâce à ses traits. Concrètement, cette technique permet d'identifier une personne donnée, de l'authentifier ou de vérifier son identité.

C'est dans ce contexte que s'insère notre mémoire de Master 2, il s'agit de développer un système de reconnaissance faciale selfie qui répond à des besoins de sécurité et de contrôle d'accès des employés d'une société quelconque. Dans notre cas le Centre de développement

des technologies avancées (CDTA), Centre de recherche où nous avons effectué notre stage de fin d'études.

Ce document est composé de quatre chapitres. Le premier chapitre est composé de deux section, la première exposera les notions basiques des systèmes biométriques en général et la deuxième portera uniquement sur la détection et la reconnaissance de visages. Dans le second chapitre, nous aborderons l'apprentissage automatique et les réseaux de neurones ainsi que les réseaux de neurones profonds et les différents modèles existants. Dans le chapitre suivant à savoir le chapitre trois, nous détaillerons les différentes modèles utilisées dans notre projet à savoir: Multi-Task Cascaded Convolutional Neural Network(MTCNN), You Only Live Once version 3 (YOLOv3) et le modèle Ulta-light. Nous présenterons ensuite dans le quatrième chapitre les tests et les résultats obtenus. Enfin, nous terminerons par une conclusion contenant des perspectives de l'application développée en termes d'amélioration des taux de réussite.

Chapitre 1 : La biométrie et le système de reconnaissance de visage.

1.1 Introduction

Traditionnellement, l'utilisation des caractéristiques personnelles comme les marques traditionnelles (permis de conduire, carte d'identité ou des connaissances (mot de passe) utilisés pour valider l'identité d'une personne n'est pas une solution fiable d'authentification. Les limites de ces approches traditionnelles sont qu'elles ne sont pas sûres et ne conviennent pas à l'authentification des personnes dans le monde moderne. Il est donc recommandé de développer des méthodes d'authentification personnelle plus cohérentes et plus réalistes pour contrôler la criminalité et la fraude quotidiennes dans diverses activités sociales et commerciales.

Ce chapitre comprend deux sections, la première est consacrée à la généralité sur les systèmes biométriques et la seconde portera uniquement sur la détection et la reconnaissance des visages. Nous commençons par la présentation de quelques généralités sur la biométrie telles que : sa définition, son fonctionnement, les différents types de la biométrie, les domaines d'applications, et enfin la mesure de performances des systèmes biométriques. Ensuite nous passerons à la deuxième partie de ce chapitre qui est consacrée à la reconnaissance des visages.

1.2 Section 1 : La Biométrie

1.2.1 Définition

Le terme «biométrie» est dérivé des mots grecs «bio» (vie) et «métriques» (mesure). Il se réfère à des métriques liées aux caractéristiques humaines, et les systèmes d'authentification biométrique, tels que les systèmes de reconnaissance faciale, les utilisent pour vérifier automatiquement les caractéristiques des personnes au moyen de l'informatique. Les identifiants biométriques [1] sont différents traits biologiques uniques à un seul individu qui nous permettent d'effectuer une identification.

Ces identifiants peuvent être différenciés entre des caractéristiques physiologiques, telles que l'empreinte digitale, les veines de la paume, le visage, l'ADN, l'empreinte de la paume, la géométrie de la main, l'iris ou la rétine, et les caractéristiques comportementales comme la signature, la voix ou la démarche.

1.2.2 Fonctionnement de la biométrie

1.2.2.1 Application d'Identification

L'identification d'une personne à plusieurs (1:N): La biométrie peut être utilisée pour déterminé l'identité d'une personne, même sans son consentement. Par exemple, la numérisation d'une foule avec une caméra et l'utilisation de la technologie de reconnaissance du visage peuvent contribuer à la détermination du sujet traité, en comparaison avec des profils stockés dans une ou plusieurs bases de données de référence [2].

1.2.2.2 Application d'authentification / vérification

L'authentification d'une à une personne (1:1): dans cette configuration, la biométrie est utilisée pour vérifier l'identité d'une personne. Par exemple, on peut assurer un accès physique à un espace sécurisé dans un bâtiment, par empreinte digitale ou bien on peut garantir l'accès à un compte bancaire ou à un guichet automatique par reconnaissance de l'iris.

L'authentification biométrique nécessite de comparer un échantillon biométrique préalablement enregistré (Modèle biométrique) à un autre échantillon biométrique nouvellement capturé (par exemple, celui capturé lors d'une connexion) [2].

1.2.3 Types de la biométrie

Les techniques biométriques peuvent être classées en trois catégories (Figure 1-1). La première repose sur les techniques d'analyse des comportements (Biométrie comportementale). Cette dernière concerne l'étude des actions répétitives et usuelles des personnes. Par exemples, l'analyse de la dynamique de la signature (vitesse de déplacement du stylo, accélérations,

pression exercée, inclinaison) ou la façon d'utiliser un clavier d'ordinateur (la pression exercée, la vitesse de frappe, les suites de caractères répétitives). La figure 1-2 illustre quelques exemples de cette catégorie de biométrie.

La seconde catégorie est basée sur les techniques d'analyse de la morphologie humaine (Figure 1-3) (Biométrie physique). Ces dernières utilisent comme moyens d'analyse les empreintes digitales, la forme géométrique de la main, les traits du visage, le dessin du réseau veineux, etc…. Ces éléments ont l'avantage d'être stables dans la vie d'un individu [3]. La troisième catégorie est basée sur tout ce qui est relié à la biologie humaine et qui est stable et permanent pendant toute la durée de vie (Figure 1-4).

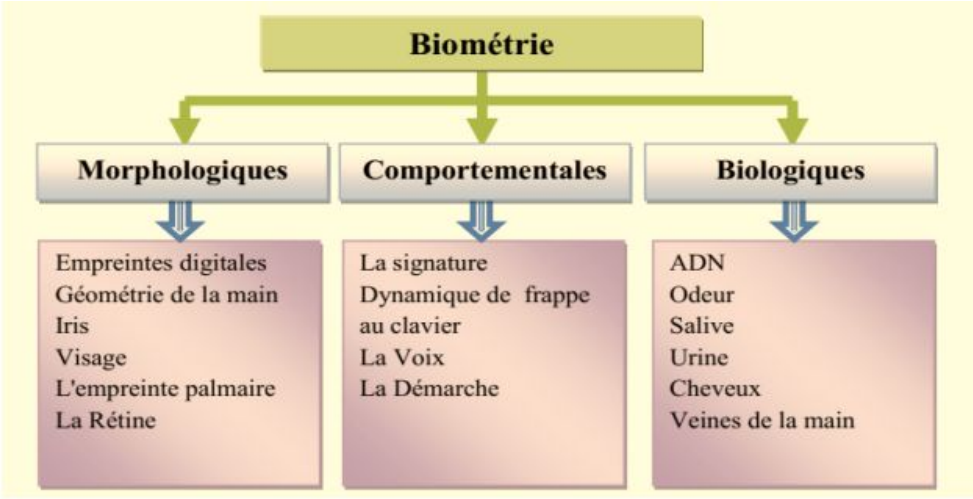
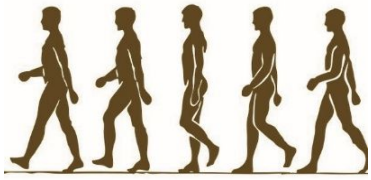


Figure 1-1 : Schéma globale de la biométrie.

Signature manuscrite.

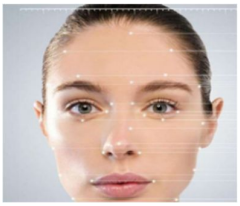


Système biométrique basé sur la démarche.

La Reconnaissance vocale.

Figure 1-2 : La biométrie comportementale[4].

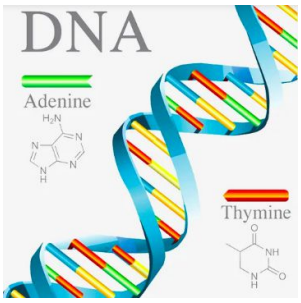
Les empreintes digitales.



Visage

Iris

Figure 1-3 : La biométrie morphologique [4].



Représentation d'une chaine d'ADN.

Figure 1-4 : La biométrie biologique [4].

Les principales contraintes liées à la biométrie sont dues à l'ergonomie et à l'acceptabilité de certaines modalités. Si la reconnaissance basée sur la biométrie comportementale ou la biométrie biologique sont généralement mal perçues par le public, la reconnaissance basée sur la morphologie particulièrement sur les modalités empreintes digitales et le visage est moins intrusive. Ces modalités présentent l'avantage d'être naturelles aux êtres humains, tout en apportant un niveau de sécurité suffisant pour un grand nombre d'applications. De plus, le matériel nécessaire caméra est actuellement intégré à la plupart des systèmes embarqués.

1.2.4 Les applications de la biométrie

Les techniques biométriques sont appliquées dans plusieurs domaines et leur champ d'application couvre potentiellement tous les domaines de la sécurité où il est nécessaire de

connaître l'identité des personnes. Les applications peuvent être divisées en trois groupes principaux [5] à savoir:

1. **Application commerciales:** Elle regroupe les applications telles que l'accès au réseau informatique, la sécurité de données électroniques, le commerce électronique, l'accès d'internet, l'ATM (automated teller machine), la carte de crédit, le contrôle d'accès physique, le téléphone portable, le PDA (personal digital assistant), la gestion des registres médicales, l'étude de distances, etc.…
2. **Applications gouvernementales:** Cette catégorie regroupe les applications telles que la carte nationale d'identifications, le permis de conduite, la sécurité sociale, le contrôle de passeport, etc.
3. **Applications juridiques**: Elle couvre les applications telles que l'identification de cadavre, la recherche criminelle, l'identification de terroriste, les enfants disparus, etc.

Le tableau 1-1 ci-dessous représente une comparaison de performance des différentes modalités biométriques mentionnées auparavant selon les propriétés suivantes : Universalité (Univ), Unicité (Unic), Permanence (Pm), Collectabilité (C), Acceptabilité (A) et Performance (Pf) qui notée par des étoiles pour expliquer le niveau de performances.

Tableau 1-1 : Tableau de comparaison des différentes modalités.

Modalité	Univ	Unic	Pm	C	A	Pf
ADN	Oui	Oui	Oui	Faible	Faible	*****
Démarche	Oui	Non	Oui	Faible	Non	*
Voix	Oui	Oui	Faible	Oui	Oui	***
Iris	Oui	Oui	Oui	Oui	Faible	*****
Visage	Oui	Non	Faible	Oui	Oui	****
Empreinte digitale	Oui	Oui	Oui	Oui	Moyenne	****
Signature	Faible	Non	Non	Oui	Oui	**

Dans cette section, nous avons défini la biométrie et présenté les différentes modalités utilisées dans les systèmes biométriques, nous avons aussi donné un aperçu des différents domaines

d'application des systèmes biométriques pour l'application de reconnaissance et l'authentification. Cette étude nous a permis de constater que la reconnaissance faciale suscite de plus en plus l'intérêt de la communauté scientifique, vu qu'elle présente plusieurs défis.

En effet, Le seul dispositif biométrique qui a émergé, est la reconnaissance facial grâce à son potentiel d'être utilisé dans de large zone comme des aéroports, des stades ou des centres commerciaux et d'autre part c'est un dispositif plus rapide et moins invasive (peu ou pas d'interaction avec le l'individu. Dans la section qui suit, nous nous aborderons uniquement le système biométrique pour l'application de reconnaissance de visage.

1.3 Section 2 : Système de reconnaissance de visage

1.3.1 Définition

La reconnaissance de visages consiste à associer un visage à une identité. Cette étape est effectuée chez les êtres humains d'une façon naturelle et évolutive. Mais, dans un système artificiel de reconnaissance faciale, cette application nécessite la construction d'une base de données des visages des individus.

Le système de reconnaissance de visage peut fonctionner sur deux modes: le mode temps réel (en-ligne) ou le mode a posteriori (hors-ligne). Au cours du mode hors-ligne, le système de reconnaissance récolte les informations de chaque visage qu'il détecte. Ces données sont ensuite notées dans une base de données qui est facilement accessible. En cas de besoin (enligne), un utilisateur est en mesure d'accéder à celle-ci et de choisir un visage en particulier pour une authentification ou une identification.

Dans chacun des deux modes, le système exécute des opérations essentielles, à savoir [6]: l'acquisition de l'image, le prétraitement, détection, extraction de caractéristiques (features), classification et la décision (Authentification /reconnaissance) (Figure 1-5).

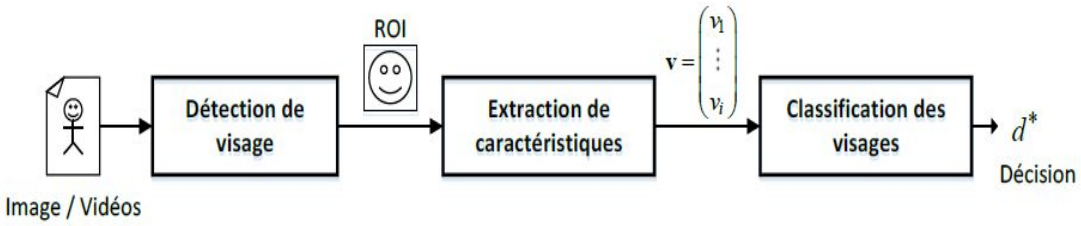


Figure 1-5 : Schéma standard d'un système de reconnaissance.

1.3.1.1 L'acquisition de l'image

L'acquisition de l'image c'est la première étape dans un système de reconnaissance, elle est faite en capturant l'image de l'individu du monde extérieur grâce aux appareils comme une caméra ou un appareil photo.

1.3.1.2 Le prétraitement

La phase de prétraitement vient après la phase de l'acquisition. Elle permet de préparer l'image du visage de telle sorte qu'elle soit exploitable. Son objectif est d'éliminer les bruits dans l'image d'entrée, causés par la qualité des dispositifs utilisés lors de son acquisition, pour ne conserver que les informations utiles et ainsi préparer l'image à l'étape suivante [7].

1.3.1.3 Détection de visage

Cette étape consiste généralement à déterminer la présence éventuelle d'un visage dans l'image et de le localiser de telles sortes à obtenir une région d'intérêt (ROI) visage sur laquelle l'extraction des vecteurs de caractéristiques pourra être accomplie [7].

1.3.1.4 L'extraction des caractéristiques

Elle consiste à extraire les caractéristiques intrinsèques du visage appelées « Landmarks » en anglais. Ces informations sont propres à chaque individu et peuvent donc le représenter d'une manière précise. Ces caractéristiques doivent être pertinentes, et aussi unique pour chaque individu, elles constituent le noyau du système de reconnaissance faciale [6].

1.3.1.5 La classification

Elle consiste à concevoir un des modèles à partir des caractéristiques d'un visage ou d'un ensemble de visages extraites dans la phase précédente en se basant sur les caractéristiques qu'ils ont en commun. Un modèle est un ensemble d'information unique caractérisant un ou

plusieurs individus possédant des caractères communs, ces individus sont groupés dans une même classe**.**

1.3.1.6 La décision

Dans cette étape, le système examine l'identité affirmée par un utilisateur ou détermine l'identité d'une personne basée sur le degré de similarité entre les caractéristiques extraites et celles des modèles stockés.

1.3.2 Les principales difficultés de la reconnaissance faciale

Pour le cerveau humain, le processus de la reconnaissance de visages est une tâche visuelle de haut niveau. Bien que les êtres humains puissent détecter et identifier des visages dans une scène sans beaucoup de peine, construire un système automatique qui accomplit de telles tâches représente un sérieux défi. Ce défi est d'autant plus grand lorsque les conditions d'acquisition des images sont très variables. Il existe deux types de variations associées aux images de visages: inter et intra sujet. La variation inter-sujet est limitée à cause de la ressemblance physique entre les individus. Par contre la variation intra-sujet est plus vaste. Elle peut être attribuée à plusieurs facteurs que nous analysons ci-dessous.

1.3.2.1 Changement d'illumination

La variation de l'éclairage lors de la capture des visages rend la tâche de la reconnaissance plus ardue, un visage de la même personne avec deux différents niveau d'éclairage peut être reconnue comme deux différentes personnes [8], la figure 1-6 montre un exemple.

Figure 1-6 : Variation de luminosité.

1.3.2.2 Variation de pose

La variation de pose est considérée comme un problème majeur pour les systèmes de reconnaissance faciale. Quand le visage est de profil dans le plan image (orientation < 30°), il peut être normalisé en détectant au moins deux traits faciaux (passant par les yeux). Cependant, lorsque la rotation est supérieure à 30°, la normalisation géométrique n'est plus possible [8], la figure 1-7 montre un exemple:

Figure 1-7 : Variation de posture.

1.3.2.3 Expressions Faciales

Un autre facteur qui affecte l'apparence du visage est l'expression faciale (voir Figure 1-8). La déformation du visage qui est due aux expressions faciales est localisée principalement sur la partie inférieure du visage. L'information faciale se situant dans la partie supérieure du visage reste quasi invariable. Elle est généralement suffisante pour effectuer une identification. Toutefois, étant donné que l'expression faciale modifie l'aspect du visage, elle entraîne forcément une diminution du taux de reconnaissance. L'identification de visage avec expression faciale est un problème difficile qui est toujours d'actualité et qui reste non résolu. L'information temporelle fournit une connaissance additionnelle significative qui peut être utilisée pour résoudre ce problème [8].

Figure 1-8 : Variation d'expression.

1.3.2.4 Présence ou absence des composants structurels

Des aspects particuliers tels que la barbe, la moustache, les lunettes, le style et la couleur des cheveux(Figure 1-9) provoquent des changements importants dans les composants structuraux du visage, notamment la forme, la couleur, la taille, etc.... [9].

Figure 1-9 : Une personne avec plusieurs composants structurels.

1.3.2.5 Occultations partielles

Le visage peut être partiellement masqué par des objets dans la scène, ou par le port d'accessoire tels que lunettes, écharpe, etc.… Dans le contexte de la biométrie, les systèmes proposés doivent être non intrusifs c'est-à-dire qu'on ne doit pas compter sur une coopération active du sujet.

1.3.3 États de l'art des techniques de reconnaissance faciale

Comme nous l'avons évoqué précédemment, un système automatique de reconnaissance de visages se décompose en trois sous-systèmes: détection de visage, extraction des caractéristiques et reconnaissance de visages. La mise en œuvre d'un système automatique et fiable de reconnaissance faciale est un verrou technologique qui n'est toujours pas résolu.

Plusieurs méthodes de reconnaissance de visages ont été proposées durant les vingt dernières années. La reconnaissance de visage est un axe de recherche ouvert attirant des chercheurs venants de disciplines différentes: psychologie, reconnaissance de formes, réseaux

neuraux, vision artificielle et infographie. Dans ce paragraphe, nous présenterons les approches de la reconnaissance de visage les plus connues. Ces dernières peuvent être subdivisées en trois catégories: les approches holistiques ou globales, les approches locales et les approches hybrides.

1.3.3.1 Approches globales

Ces approches sont également appelées méthodes basées sur l'apparence. Ces méthodes identifient un visage en utilisant l'image entière de ce dernier comme entrée du système de reconnaissance. Le principe est comme suit : Chaque image de visage de dimension (n,m) est représentée par un vecteur simple de dimension n m, en concaténant les valeurs du niveau de gris de tous les pixels de l'image du visage. L'espace I contenant tous les vecteurs images de visages est appelé espace images. L'avantage de cette représentation est qu'elle préserve implicitement les informations de texture et de forme nécessaire pour la reconnaissance de visages. De plus, elle permet une meilleure capture de l'aspect global du visage que les représentations locales [10]. Cette catégorie de techniques peut être divisée en deux classes: linéaire et non linéaire.

- **Techniques linéaires**

Ces techniques consistent à réaliser une projection linéaire de l'image d'une grande dimension dans un espace de petite dimension. Cependant, une telle projection ne peut pas conserver les variations de visage non convexes, qui permettent de distinguer différents individus [11] et la distance euclidienne ne peut pas être utilisée pour comparer les pixels car la classification visage, non visage ne sera pas effective et conduit à une détection et reconnaissance insatisfaisantes. Les techniques les plus utilisées sont:

- Eigenface [12]: Son principe est le suivant: étant donné un ensemble d'images de visages exemples, il s'agit tout d'abord de trouver les composantes principales de ces visages. Ceci revient à déterminer les vecteurs propres de la matrice de covariance formée par l'ensemble des images exemples. Chaque visage exemple peut alors être décrit par une combinaison linéaire de ces vecteurs propres. Pour construire la matrice de covariance, chaque image de visage est transformée en vecteur. Chaque élément du vecteur correspond à l'intensité lumineuse d'un pixel (Figure 1.10).

Figure 1-10 : Représentation de 12 valeurs de eigenface[13].

- **Techniques non-linéaires**

Des techniques globales non linéaires ont été développées, souvent à partir de techniques linéaires en utilisant la fonction « kernel » pour étendre les techniques linéaires. On mentionne quelques techniques non linéaires à savoir:

- Analyse en Composantes Principales à Noyaux (KACP) [13].
- Support vector machine (SVM) [14].
- Kernel independent component analysis (KICA) [15].
- Locality preserving projection (LPP) [16].

1.3.3.2 Approches locales

Les approches locales de la reconnaissance de visage sont basées sur des modèles et reposent sur un traitement séparé appliqué sur les différentes régions de l'image contenant un visage. Ce processus conduit à un vecteur caractéristique pour chaque région du visage [10]. Ces approches peuvent être classifiées en deux catégories : Méthodes basées sur l'apparence locale e méthodes basées sur les points d'intérêts.

- **Méthodes basées sur l'apparence locale**

Les approches locales de la reconnaissance de visages sont basées sur des modèles et utilisent des connaissances que nous possédons, à priori, de la morphologie des visages. Elles consistent

à détecter les points caractéristiques du visage, tels que les yeux, la bouche, le nez et les oreilles ; et ensuite de mesurer chaque position de ces points dans l'espace du visage ; puis à les comparer avec les paramètres extraits d'autres visages.

Le gros avantage des méthodes locales de reconnaissance de visages est qu'elles sont robustes aux problèmes posés par les variations de pose, d'illumination ou encore d'expressions que peuvent subir un visage, car elles peuvent modéliser facilement ces variations. Cependant, ces méthodes sont plus difficiles à mettre en place car elles nécessitent souvent le placement manuel de nombreux points d'intérêts pour une bonne précision, et sont donc lourdes à mettre en œuvre. Dans cette catégorie, nous trouvons plusieurs méthodes comme les plus répandues sont Local Binary Pattern [17], Histogramme Orienté Gradient (HOG) [18, 19].

- **Local Binary Patterns (LBP)**

Descripteur LBP [17] est conceptualisé dans l'objectif de caractériser des modèles (patterns) de textures très spécifiques, à partir des niveaux de gris dans l'image. La technique se base sur un seuillage des pixels autour d'une valeur centrale dans une fenêtre localisée, afin d'encoder des données binaires si les pixels voisins sont supérieurs ou inférieurs à celle-ci. Ce procédé d'encodage est représenté dans la Figure 1-11.

Les occurrences de combinaisons binaires observées sur l'image sont ensuite accumulées dans un histogramme, tel que montré à la Figure 1-12, après filtrage à un maximum de 59 caractéristiques pour supprimer les patrons irréalistes selon les observations accomplies dans le mode réel.

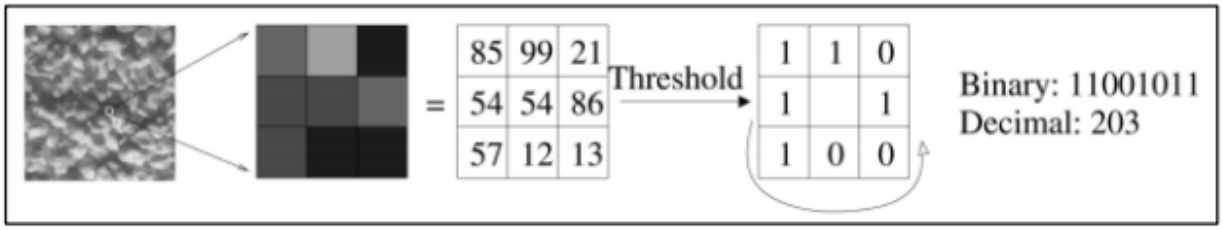


Figure 1-11 : Encodage des pixels par extraction des caractéristiques.

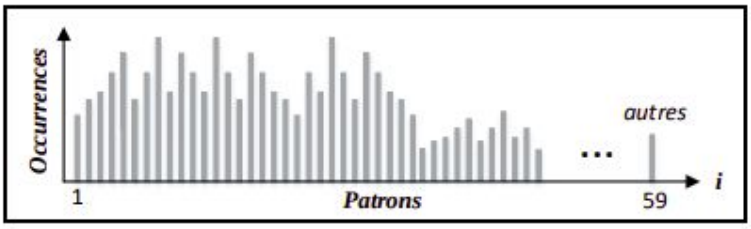


Figure 1-12 : Histogramme de caractéristique LBP.

- HOG (Histogramme Orienté Gradient)

HOG fait partie des méthodes conçues pour l'extraction de caractéristiques [20]. L'extraction de caractéristiques par HOG est aussi grandement employée dans le contexte de la reconnaissance de visages [18, 19]. Le principe de son fonctionnement consiste à parcourir l'image pixel par pixel tout en comparant le degré d'obscurité de chaque pixel avec les pixels qui l'entoure, puis dessiner une flèche représentant le gradient et indiquant dans quelle direction l'image devient plus sombre. Les gradients sont sélectionnés et regroupés par orientations communes successivement selon des sous-divisions de l'image en fenêtres, blocs, et cellules, avant d'être concaténés sous forme de vecteur caractéristiques (Figure 1-13).

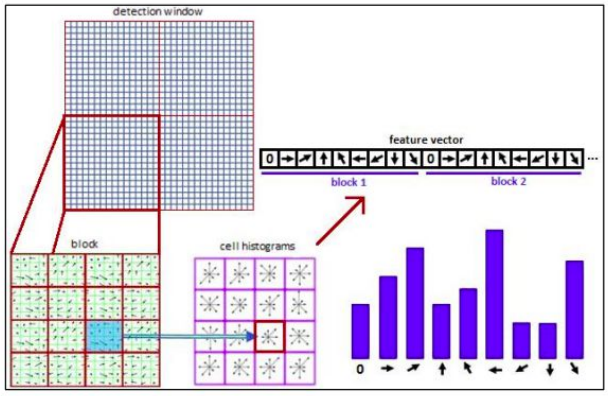


Figure 1-13 : Procédé du descripteur HOG.

1.3.3.3 Approche hybride

Les approches hybrides sont des approches qui combinent les caractéristiques globales et locales afin d'améliorer les performances de la reconnaissance de visages. En effet, les caractéristiques locales et les caractéristiques globales ont des propriétés tout à fait différentes.

Les méthodes hybrides permettent d'augmenter la stabilité de la performance de reconnaissance lors de changements de pose, d'éclairage et d'expressions faciales.

- Analyse en Composantes Locales (LCA) [21]**:** cette technique réalise plusieurs analyses en composants principales pour extraire les caractéristiques qui sont ensuite combinées avec une autre méthode pour minimiser l'erreur de reconstruction [10].

- HMM-LBP [22] **:** cette technique permet de faire une classification des images 2D en utilisant la technique locale LBP pour l'extraction des caractéristiques et les HMM pour la classification [11].

Nous avons présenté les principales approches utilisées dans la littérature pour la reconnaissance faciale automatique. Les méthodes peuvent principalement se classer en trois catégories : les méthodes globales, les méthodes locales et les méthodes hybrides, mais les méthodes hybrides essayent simplement de faire un lien entre les deux types d'approches précédents. Les méthodes globales présentent un certain nombre d'avantages : le problème de la reconnaissance faciale automatique est transformé en un problème d'analyse de sous-espaces de visages, pour lequel de nombreuses méthodes statistiques existent ; les méthodes globales sont souvent applicables à des images basses résolutions ou de mauvaises qualités. Certains inconvénients se posent cependant avec les méthodes globales : il est nécessaire de disposer suffisamment de données représentatives des visages ; ces méthodes ne sont robustes qu'à des variations limitées (pose, illumination, expression). De la même manière, les méthodes locales présentent certains avantages : le modèle créé possède des relations intrinsèques bien définies avec les visages réels ; les modèles créés peuvent prendre en compte explicitement les variations telles que la pose, l'illumination ou les expressions. La reconnaissance est ainsi plus efficace dans le cas de fortes variations et la reconnaissance a priori sur les visages peut être intégrée aux modèles afin d'améliorer leur efficacité. Les méthodes locales présentent cependant quelques inconvénients : la construction du modèle, reposant souvent sur la détection de points caractéristiques faciaux, peut être laborieuse ; l'extraction des points caractéristiques peut être difficile dans le cas de variations de pose, d'illumination, d'occlusion . . . ; les images doivent être relativement de bonne qualité, et/ou être de résolution suffisante afin de pouvoir extraire les points caractéristiques.

D'autre part, avec les méthodes d'apprentissage automatique traditionnelles, le système de reconnaissance fonctionne bien avec des données numériques, mais pour des données plus complexes, notamment ceux d'origine analogique comme les photos, les méthodes classiques exigent au préalable les éléments de base. Par exemple, pour la reconnaissance faciale, les chercheurs utilisent des algorithmes spécifiques pour analyser des éléments du visage comme les yeux ou la bouche. C'est là qu'intervient l'apprentissage profond ou deep learning en anglais. C'est une classe de méthodes dont les principes sont connus depuis la fin des années 1980, mais dont l'utilisation ne s'est généralisée que depuis 2012, environ.

Avec le « deep learning », les chercheurs ne fournissent aucune indication. Le réseau neuronal artificiel reçoit par exemple des photos avec pour seule information «voiture» ou « pas de voiture ». Il devra ensuite analyser les images pour déterminer seuls quels éléments constituent une voiture, et comparer aux images sans voitures pour vérifier lui-même ses résultats. Plus le réseau reçoit d'images d'entraînement, plus il affine ses algorithmes. Cela lui permet d'atteindre des performances d'une précision sans égal par rapport aux autres méthodes d'analyse de données complexes.

1.4 Mesure de performance d'un système de reconnaissance faciale

La performance d'un système de reconnaissance est mesurée en basant sur trois 3 critères principaux à savoir le TAR, le FAR et le FRR.

- **TAR (True Acceptance Rate) :** Il représente le taux de réussite. Dans le cas d'un système de reconnaissance faciale le TAR représente le taux de reconnaissance en pourcentage (%). il peut être calculé selon l'équation suivante où T représente le nombre total de tentatives:

$$TAR = \frac{Nombre\ \ de\ personnes\ correctement\ reconnues}{T}$$

- **FRR (False Reject Rate) :** Il représente le taux de faux rejet noté par le pourcentage de personnes censées être reconnues mais qui sont rejetées par le système. Il est calculé selon l'équation suivante :

$$FRR = \frac{Nombre\ de\ personnes\ faussement\ rejetées}{T}$$

- **FAR (False Acceptance Rate) :** Il représente le taux de fausse acceptation. Ce taux représente le pourcentage de personnes censées ne pas être reconnues mais qui sont tout de même acceptées par le système, Il est calculé selon l'équation suivante :

$$FAR = \frac{Nombre\ de\ personnes\ faussement\ acceptées}{T}$$

1.5 Conclusion

Ce chapitre est composé de deux sections. Dans la première, nous avons abordé les concepts de base de la biométrie en évoquant les différentes modalités ainsi que les applications des systèmes biométriques. Dans la deuxième section, nous avons introduit le fonctionnement d'un système de reconnaissance faciale, les différentes techniques répandues dans la l'littérature pour la reconnaissance de visage. Une synthèse des avantages et inconvénients des différentes catégories de techniques de reconnaissance de visage a été effectués. Suite à cette synthèse, nous nous sommes arrivées à la conclusion que la reconnaissance de visage ainsi la vision par ordinateur suscite de plus en plus l'intérêt de la communauté scientifique, vu qu'elle présente plusieurs défis. Et l'utilisation des principes d'intelligence artificielle et les algorithmes basés sur le Deep Learning, particulièrement les CNN est la meilleure solution pour concevoir et réaliser un système de reconnaissance faciale. Dans le chapitre suivant, nous aborderons les la technologie de l'apprentissage automatique et particulièrement les réseaux de neurones profonds.

Chapitre 2 : Les réseaux de neurones.

2.1 Introduction

L'intelligence artificielle (IA) est une technique informatique [23] qui permet à une machine quelconque d'imiter les tâches que seul l'être humaine peut effectuer à cause de leur complexité. Pour effectuer ces tâches, la machine a besoin d'être entraînée pour qu'elle être capable de reproduire sa propre représentation du monde et c'est ce qu'on appelle « l'apprentissage automatique » (en anglais: machine learning).

L'apprentissage automatique est un ensemble de méthodes qui permettre à une machine d'apprendre à partir des échantillons où L'apprentissage profond est le sous ensemble de l'apprentissage automatique qui est basé sur les réseaux de neurones. Comme son nom l'indique ce sont des réseaux de neurones artificiels imitant le système nerveux de l'être humain.

Dans ce chapitre, nous commençons par l'apprentissage automatique et son principe, passant aux réseaux de neurones ensuite l'apprentissage profond et les différentes architectures les plus utilisées.

2.2 L'apprentissage automatique (Machine Learning)

L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques. Son objectif est d'extraire et d'exploiter automatiquement l'information présenté dans un jeu de données. La figure 2-1 montre la relation entre Intelligence artificiel (IA), apprentissage automatique (ML) et apprentissage profond (DL).

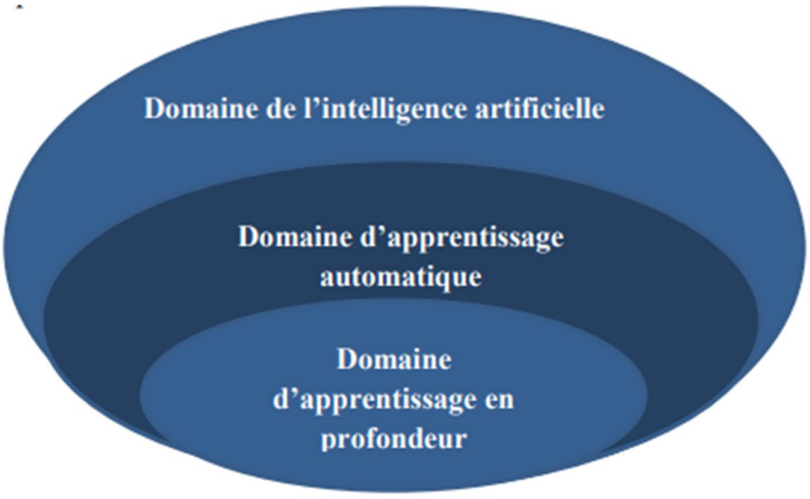


Figure 2-1 : La relation entre IA, ML, DL.

Le principe de l'apprentissage automatique peut être définit de la manière suivante : un programme peut apprendre d'une expérience E à effectuer une tâche T, si sa mesure de performance P, mesuré par la tâche T, s'améliore avec l'expérience E. en d'autre terme le programme peut imiter et généraliser un comportement pour des nouvelles situations similaire à l'expérience E [24].

La tâche T représente un processus qui regroupe une ou plusieurs fonctions. A savoir :

- **Régression :** La régression peut être définie comme une méthode ou un algorithme dans l'apprentissage automatique qui modélise une valeur cible basée sur des prédicteurs indépendants. Il s'agit essentiellement d'un outil statistique utilisé pour découvrir la relation entre une variable dépendante et une variable indépendante [25].
- **Classification :** La classification est un processus de catégorisation d'un ensemble de données en classes. Elle peut être effectuée sur des données structurées ou non structurées. Le processus commence par la prévision de la classe des points de données. Les classes sont souvent appelées cible, étiquette ou catégories (en anglais : Label).
- **Segmentation :** La segmentation est le processus de séparation des données en groupes distincts. L'un des plus grands défis de la création de segments est de déterminer combien de segments existent réellement dans les données [26].

L'expérience E représente l'algorithme de l'apprentissage. Les algorithmes d'apprentissage peuvent être classés en trois catégories principales :

- **L'apprentissage supervisé :** un apprentissage est dit supervisé si les classes sont prédéterminées et les exemples connus dont le système apprend à classer selon un modèle de classification ou de classement. Le processus se passe en deux phases, la première phase est hors ligne c.-à-d. les données sont statiques pendant l'apprentissage. Il s'agit de déterminer un modèle à partir des données étiquetées. La deuxième phase qui est en ligne c.-à-d. les données sont présentés les uns après les autres au fur et à mesure de leur disponibilité, cette phase consiste à prédire une étiquette d'une nouvelle donnée, cette phase est généralement connue sous le nom 'test'.
- **L'apprentissage non supervisé :** un apprentissage est dit non supervisé si le système ne dispose d'aucun étiquetage préalable des données, et que le nombre de classes et leur nature n'ont pas été prédéterminés. Aucun expert n'est requis. L'algorithme doit découvrir par lui-même la structure plus ou moins cachée des données. Le partitionnement des données (en anglais : data clustering) est un exemple d'algorithme de l'apprentissage non supervisé [23].
- **L'apprentissage par renforcement :** c'est une méthode d'apprentissage ou la machine se comporte comme un agent qui apprend de son environnement d'une manière interactive jusqu'à ce qu'il découvert les comportements qui produisent des récompenses. La figure 2-2 montre la différence entre l'apprentissage supervisé et non supervisé.

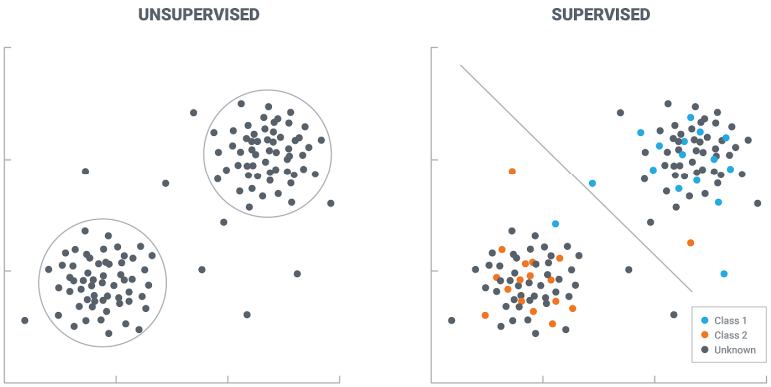
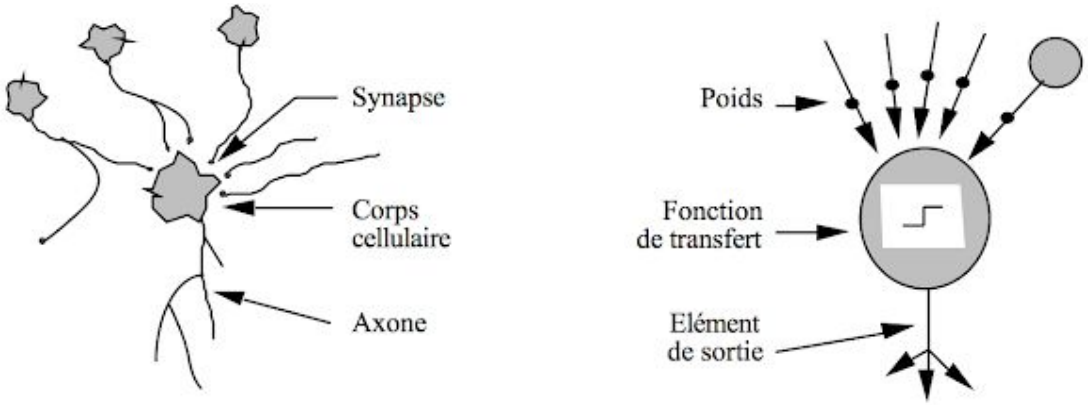


Figure 2-2 : l'apprentissage supervisé et non supervisé.

La performance P désigne des mesures quantitatives qui servent à évaluer l'algorithme de l'apprentissage effectué par la tâche.

2.3 Les réseaux de neurones

Les réseaux neuronaux, aussi appelés RNA, sont des modèles de traitement de l'information qui simulent le fonctionnement d'un système nerveux biologique (Figure 2-3). C'est similaire à la façon dont le cerveau manipule l'information au niveau du fonctionnement. Tous les réseaux neuronaux sont constitués de neurones inter connectés qui sont organisés en couches [27].



Neurone biologique. **Neurone artificiel.**

Figure 2-3 : Similitude entre le neurone biologique et le neurone artificiel.

Le but, d'un point de vue global, est d'exécuter des calculs complexes et de trouver, par apprentissage, une relation non linéaire entre des données numériques et des paramètres [27].

L'un des neurones artificiels simplifié c'est le perceptron (figure 2-4) qui a été introduit par Frank Rosenblatt en 1957 [28]. Il contient plusieurs entrées ou chacune possède un poids. Les entrées sont toutes connectées à une seule sortie en passant par une fonction de transfert qui est appelée aussi fonction d'activation [29].

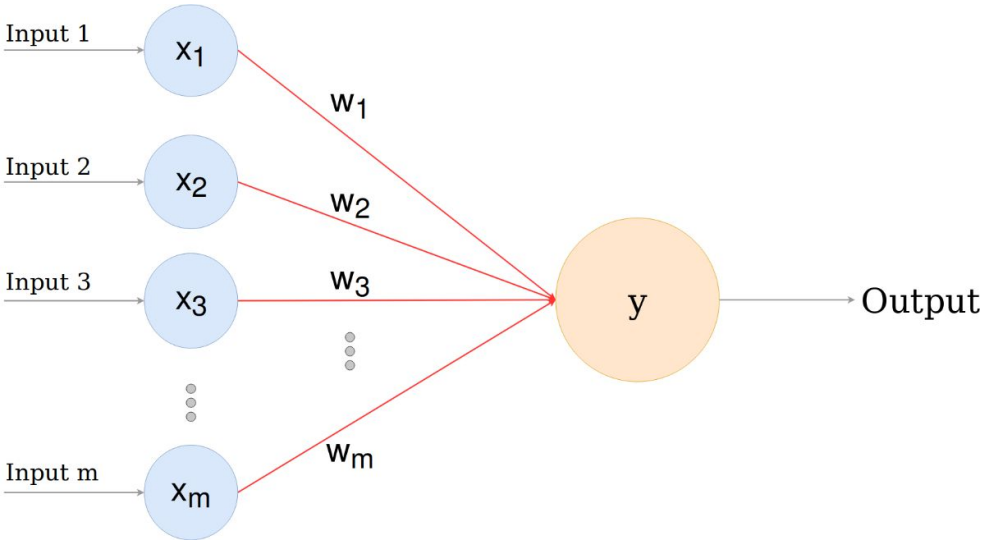


Figure 2-4: Représentation d'un perceptron.

Où :

Xi : représente les entrées de perceptron.

Wi : représente les poids associés à ces entrées.

Y : c'est la fonction de transfert ou la fonction d'activation.

Pour simplifier le principe du travail du perceptron, on donne l'exemple suivant :

On pose les entrés Xi, on lui associés les poids Wi respectivement, donc on calcule d'abord ce qu'on appelle le potentiel post-synaptique de la manière montré dans l'équation 2-1:

$$x = \sum_{i=1}^{n} Wi.Xi \quad \text{..... (2-1)}$$

Ensuite on applique la fonction d'activation comme montré dans l'équation 2-2 :

$$f(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{sinon} \end{cases} \quad \text{..... (2-2)}$$

2.3.1 Algorithme d'apprentissage par correction d'erreur

Le principe de cette algorithme consiste à trouver les poids d'un perceptron qui classifient correctement un ensemble d'entrés [30]. L'algorithme peut être décrit de la façon suivante où R représente le taux d'apprentissage et T c'est le seuil désiré.

Algorithme d'apprentissage avec correction d'erreur

Début

Initialiser tous les poids Wi à des valeurs aléatoires

TANTQUE pas tous les exemples sont prédit correctement **FAIRE**

POUR chaque exemple d'apprentissage X **FAIRE**

Calculer la sortie O(x) ;

POUR (i= 1 à n) **FAIRE**

Wi = Wi + r(t- o) Xi ;

FIN

2.3.2 Algorithme d'apprentissage par la descente du gradient

Cet algorithme consiste à trouver un vecteur pour minimiser l'erreur (E[x]). Quand la valeur de E[x] tant vers 0, c'est à dire que le perceptron a réussie à classifier l'échantillon correctement.

Une des méthodes qui permet de chercher le minimum d'une fonction c'est l'algorithme de la descente du gradient, on distingue deux types de cet algorithme :

La descente du gradient stochastique : Cet algorithme calcule la moyenne des gradients de la fonction d'erreur pour chaque paramètre du réseau à partir de tous les exemples

d'apprentissage puis minimise la fonction en chaque itération en mettant à jour les paramètres en fonction des moyennes calculées [31].

La descente du gradient ordinaire : L'algorithme prend en considération les exemples d'apprentissage un par un. Il calcule à chaque itération les valeurs des gradients des paramètres du réseau en fonction d'un seul exemple pour ensuite mettre à jours ces paramètres. Le processus s'arrête lorsqu'une valeur négligeable de l'erreur globale est atteinte [31].

2.3.3 Réseaux de neurones à multicouche

Un réseau de neurones à multicouche (Figure 2-5) est un type de réseaux de neurones artificiel organisé en plusieurs couches cachées 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) [23].

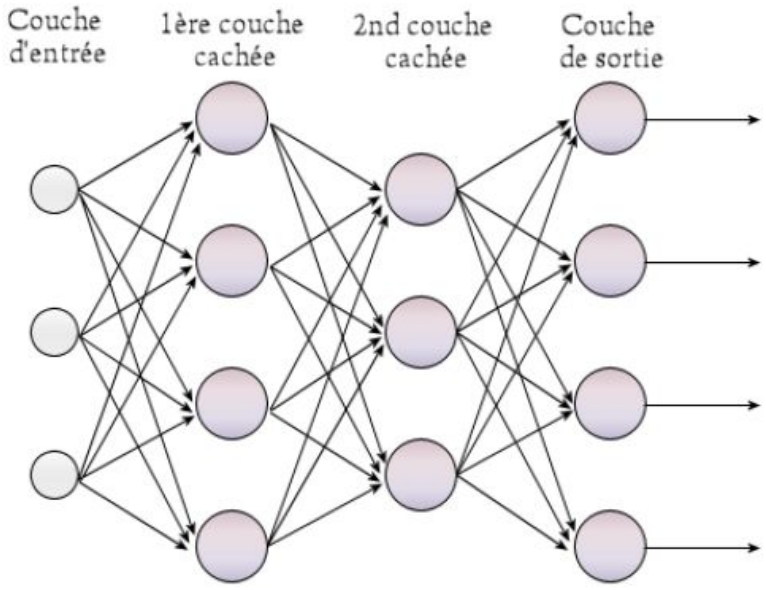


Figure 2-5: Un réseau de neurone avec deux couches cachées.

2.3.3.1 Propagation (forward- propagation)

Le réseau de neurones à propagation avant était le premier et le plus simple type de réseau neuronal artificiel conçu [32]. 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.

2.3.3.2 Rétro-propagation (backward propagation)

En statistiques, la rétro-propagation du gradient est une méthode pour calculer le gradient de l'erreur pour chaque neurone d'un réseau de neurones, de la dernière couche vers la première. En pratique, les coefficients du réseau sont modifiés de façon à corriger les erreurs de classification rencontrées, selon une méthode de descente de gradient. Ces gradients sont rétro propagés dans le réseau depuis la couche de sortie [23].

2.3.3.3 La fonction d'activation

Après que le neurone a effectué le produit entre ses entrées et ses poids, il applique également une non-linéarité sur ce résultat. Cette fonction non linéaire s'appelle la fonction d'activation. La fonction d'activation est une composante essentielle du réseau neuronal. Ce que cette fonction a décidé est si le neurone est activé ou non. Il calcule la somme pondérée des entrées et ajoute le biais. C'est une transformation non linéaire de la valeur d'entrée.

Après la transformation, cette sortie est envoyée à la couche suivante. La non-linéarité est si importante dans les réseaux de neurones. Sans la fonction d'activation, un réseau de neurones est devenu simplement un modèle linéaire. Il existe de nombreux types de ces fonctions, parmi lesquelles nous trouvons :

- **La fonction Sigmoïde :** Cette fonction (Figure 2-6) est l'une des plus couramment utilisées. Il est borné entre 0 et 1, et il peut être interprété stochastique-ment comme la probabilité que le neurone s'active, et il est généralement appelé la fonction logistique ou le sigmoïde logistique [23].

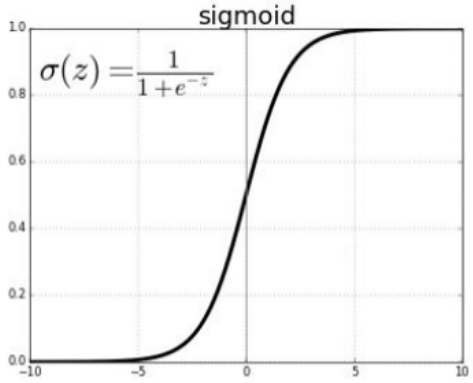


Figure 2-6: Représentation graphique de la fonction sigmoïde.

- **La fonction ReLu :** La fonction RELU (Figure 2-7) est probablement la plus proche de sa correspondante biologique [33]. Cette fonction est récemment devenue le choix de nombreuses tâches (notamment en computer vision) [34]. Comme dans la formule ci-dessus, cette fonction renvoie 0 si l'entrée z est inférieure à 0 et retourne z lui-même s'il est plus grand que 0.

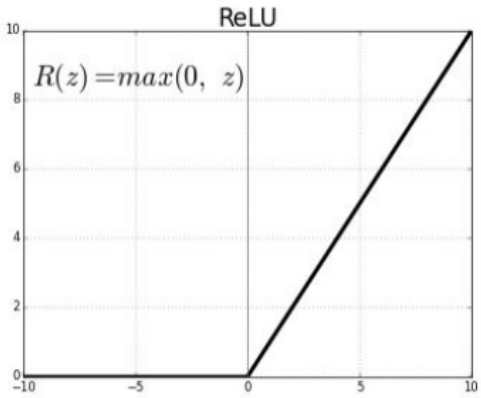


Figure 2-7: Représentation graphique de la fonction ReLu.

- **La fonction Softmax :** le rôle de cette fonction est de calculer la distribution des probabilités de l'événement sur n différents événements. En règle générale, cette fonction calcule les probabilités de chaque classe cible sur toutes les classes cibles possibles. Elle est utilisée généralement dans les couches de sorties [35].

2.4 L'apprentissage profond

L'apprentissage en profondeur est un ensemble d'algorithmes d'apprentissage automatique qui tentent d'apprendre à plusieurs niveaux, correspondant à différents niveaux d'abstraction. Il a la capacité d'extraire des caractéristiques à partir des données brutes grâce aux multiples couches de traitement composé de multiples transformations linéaires et non linéaires et apprendre sur ces caractéristiques petites à petit à travers chaque couche avec une intervention humaine minime [36]. La figure 2-8 montre un schéma d'un apprentissage profond avec plusieurs couches.

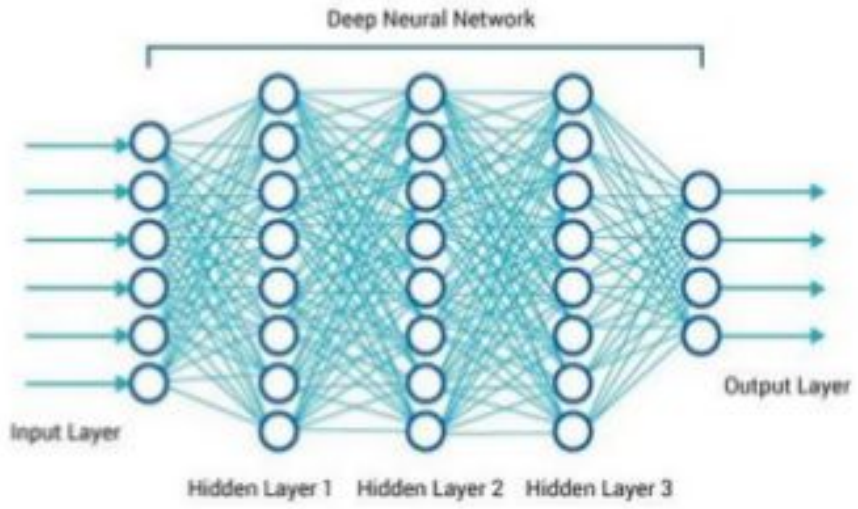


Figure 2-8: Schéma illustratif de DL avec plusieurs couches.

- **Pourquoi le choix deep learning**

Tout d'abord les différents algorithmes du deep learning ne sont apparus qu'à l'échec de l'apprentissage automatique tentant de résoudre une grande variété de problèmes de l'intelligence l'artificielle (l'IA) [37] :

- Afin d'améliorer le développement des algorithmes traditionnels dans telles tâches de l'IA.
- De développer une grande quantité de données telle que les big data.
- De s'adapter à n'importe quel type de problème.
- D'extraire les caractéristiques de façon automatique.

La figure 2-9 montre une comparaison entre l'apprentissage automatique et l'apprentissage profond :

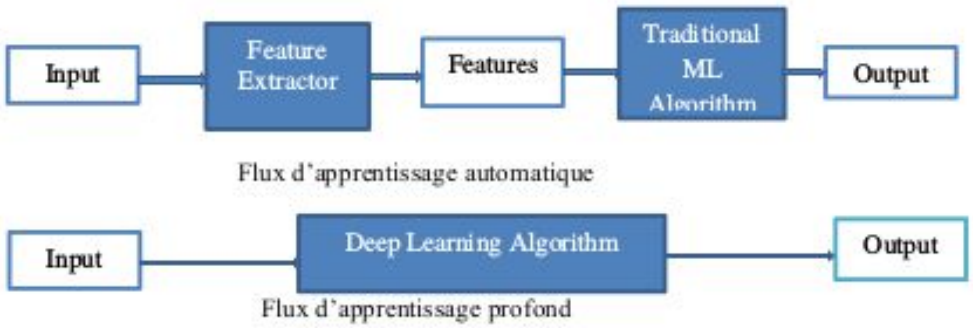


Figure 2-9: Comparaison entre la machine Learning et le deep learning.

.

2.4.1 Les différentes modèles du réseau profond

Il existe plusieurs modèles pour un réseau de neurones profond (figure 2-10).

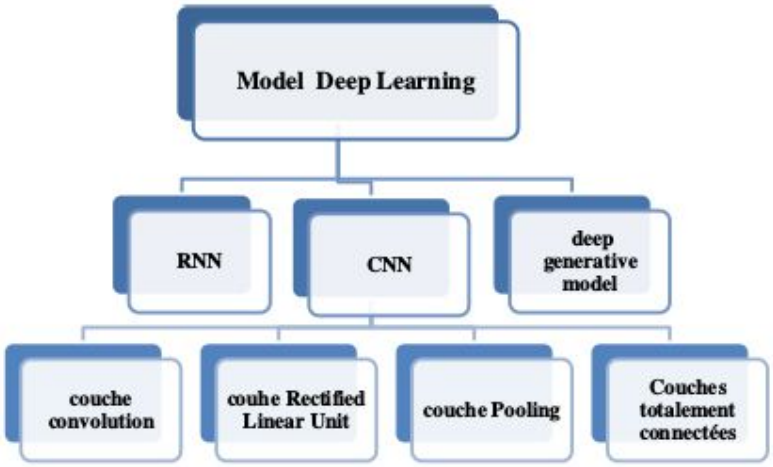


Figure 2-10: Différents modèles du Deep Learning.

2.4.1.1 Réseau de neurones récurrents (RNN)

Un réseau de neurones récurrent (Recurrent Neurel Network (RNN)) est un réseau de neurones dont le graphe de connexion contient au moins un cycle. De nombreux types de RNN

ont été développés au cours des 30 dernières années tels que les réseaux d'ELman, les réseaux de Jordan et les Echo StateNetworks [38].

Au cours de ces dernières années, un type de RNN qui est le réseau de neurones à base de cellules Long Short-Term Memory (LSTM) est devenu une norme dû aux performances élevées obtenus des applications aussi nombreuse que variées.

2.4.1.2 Modèle génératif

Si les modèles CNN et RNN sont utilisés pour prédire les données du label et de l'entrée, Le modèle génératif est un modèle probabiliste qui décrit comment générer les données, il apprend et fait des prédictions en utilisant la loi de Bayes [39]. En plus, les modèles génératifs sont capables de faire une simple classification comme par exemple générer de nouvelles observations.

Voici quelque exemple de modèle génératif :

- Boltzmann Machines [40 ,41].
- Restricted Boltzmann Machines [42,43].
- Deep Belief Networks [44,45].
- Deep Boltzmann Machines [46,47].
- Generative Adversarial Networks [48, 49,50].
- Generative Stochastic Networks [51].
- Adversarial autoencoders [52].

2.4.1.3 Réseau de neurones convolutionnels

Les réseaux de neurones convolutionnels sont à ce jour les modèles les plus performants pour classer des images. Désignés par l'acronyme CNN, de l'anglais Convolutional Neural

Network qui est un réseau constitué de plusieurs couches de convolutions dont chacune un traitement sur l'image est effectué [53]. En entrée, une image est fournie sous la forme d'une matrice de pixels. Elle a 2 dimensions pour une image en niveaux de gris. La couleur est représentée par une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales [Rouge, Vert, Bleu]. L'image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques. La figure 2-11 représente un réseau de neurone CNN.

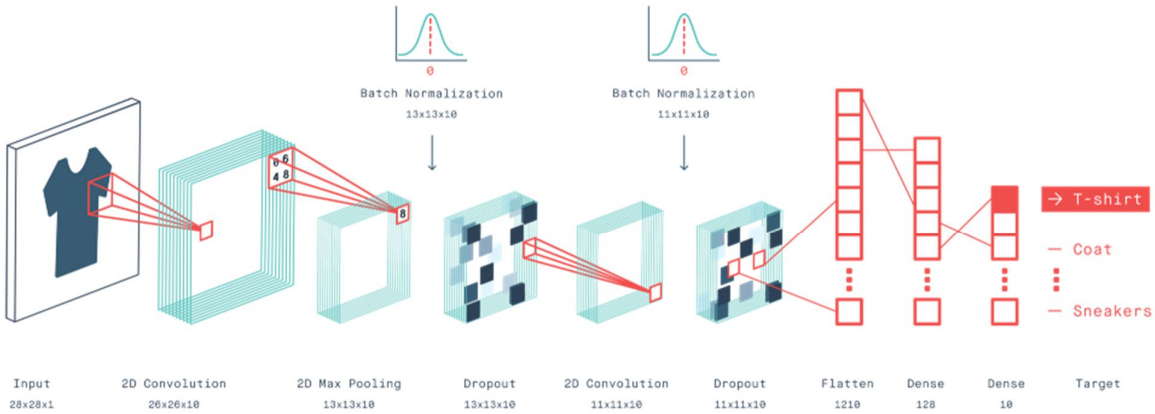


Figure 2-11 : Une représentation d'un réseau de neurone CNN.

2.4.2 Architecture de Réseau de neurones convolutionnels

Il existe plusieurs architectures des CNNs qui varient des plus basiques aux plus complexes. Une architecture spécifique est plus performante dans certains domaines que d'autres. Une architecture CNN est formée par un empilement de couches de traitement indépendantes :

- La couche de convolution (CONV), qui traite les données d'un champ récepteur.
- La couche de pooling (POOL), qui permet de compresser l'information en réduisant la taille de l'image intermédiaire (souvent par sous-échantillonnage).
- La couche de correction (ReLU) qui applique la fonction d'activation (Unité de rectification linéaire).

- La couche « entièrement connectée » (FC), qui est une couche de type perception.
- La couche de perte (LOSS) responsable pour la correction des erreurs.

- **Couche de convolution (CONV)**

La couche de convolution est la composante principale des réseaux de neurones convolutifs. Son but est de repérer la présence d'un ensemble des caractéristiques dans les images reçues en entrée [54].

Pour cela , on réalise un filtrage par convolution : le principe est de faire « glisser » une fenêtre représentant la caractéristique sur l'image , et de calculer le produit de convolution entre la caractéristique et chaque portion de l'image balayée et produire une carte de caractéristiques (features map) à sortie [54].

Avant l'étape de la convolution, il faut déterminer quelques paramètres, ces paramètres font partie du calcul de la taille des images à la sortie de chaque couche (la carte des caractéristiques), Ces paramètres sont :

- **Le pas (Stride) :** Stride est le nombre de pixels décalés sur la matrice d'entrée. Lorsque le pas est de 1, nous déplaçons les filtres à 1 pixel à la fois. Lorsque le pas est de 2, nous déplaçons les filtres à 2 pixels à la fois et ainsi de suite [55].
- **Le padding :** le padding est le rajout des pixels aux bords de l'image pour pouvoir appliquer la convolution sur les bordures de l'image (Figure 2-12). Le nombre de lignes et de colonnes rajoutés dépend de la taille du filtre de la convolution [23].

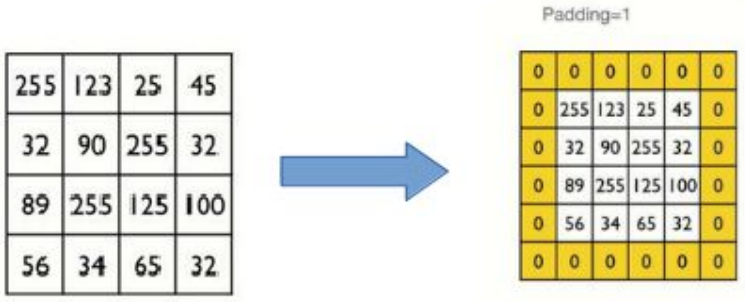


Figure 2-12 : Représentation d'un padding.

Le principe de la convolution est illustré sur la figure 2-13 :

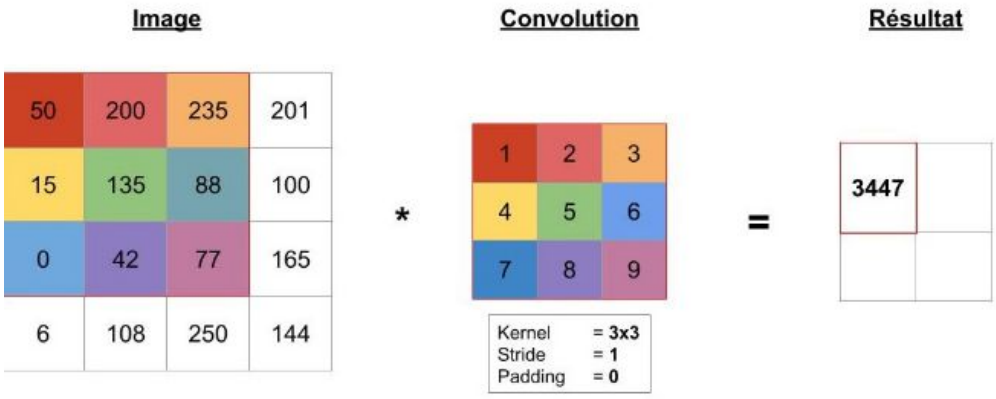


Figure 2-13 : Principe de la convolution.

Selon les valeurs présentées dans la figure 2.13, le calcul de la convolution est effectué comme suit :

50*1+200*2+235*3+15*4+135*5+88*6+0*7+42*8+77*9=3447.

A la sortie de chaque couche de convolution la taille de l'image est réduite avec la formule suivante (2-3), on peut savoir la taille de l'image en sortie d'une convolution à partie de la taille de l'image (W), la taille du filtre (F), le padding (P) et le stride (S) [56] :

$$Taille\ sortie\ covolution = 1 + \frac{W - F + 2P}{S} \quad \text{..... (2-3)}$$

- **Couche de pooling (POOL)**

C'est une méthode mathématique permettant de réduire la taille de l'image sans perdre les informations les plus importantes dans l'image. Son principe consiste à diviser l'image en petites matrices par exemple 2x2 et choisir la valeur maximale de chaque matrice, pour obtenir une image réduite [23].

La figure 2-14 suivante montre un exemple de l'application d'un sous échantillonnage 2x2 avec un pas égal à 2.

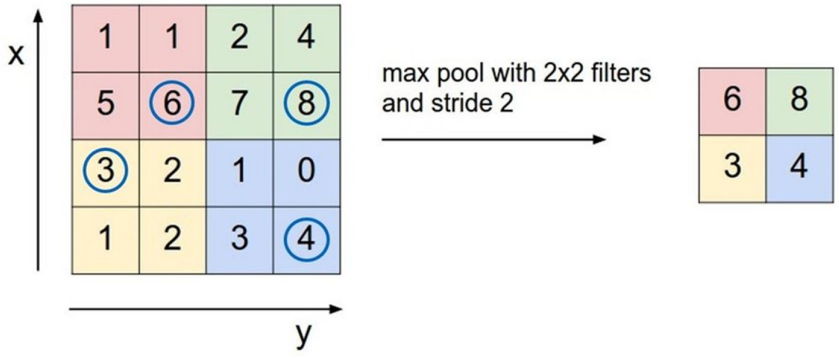


Figure 2-14 : Principe de pooling.

- **Couche de correction (ReLu)**

Souvent, il est possible d'améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie. La couche de correction ReLu remplace donc toutes les valeurs négatives reçues en entrées par des zéros [53].

- **Couche de « entièrement connectée » (FC)**

La couche entièrement connectée est un traditionnel perceptron multicouche (Multi Layer Perceptron). Le terme «entièrement connecté » implique que chaque neurone dans la couche précédente est connecté à chaque neurone sur la couche suivante. La figure 2-15 représente un exemple d'une couche entièrement connectée [57].

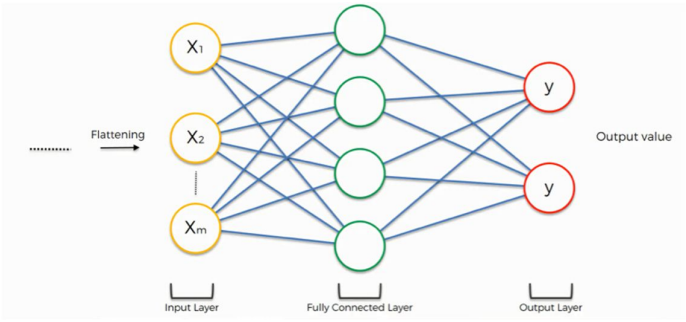


Figure 2-15 : Exemple des couches entièrement connectées.

La sortie des couches de convolution et de Pooling représente les fonctions de haut niveau de l'image d'entrée. Le but de la couche entièrement connectée est de pouvoir utiliser ces fonctions pour classer l'image d'entrée dans différentes classes en fonction de l'ensemble de données d'apprentissage [57].

- **Couche de perte (LOSS)**

La couche de perte spécifie comment l'entraînement du réseau pénalise l'écart entre le signal prévu et réel. Elle est normalement la dernière couche dans le réseau. Diverses fonctions de perte adaptées à différentes tâches peuvent y être utilisées. La fonction «Softmax» permet de calculer la distribution de probabilités sur les classes de sortie pour la correction des erreurs [53].

- **La Normalisation**

Il est connu depuis longtemps que l'apprentissage du réseau converge plus rapidement si ses entrées sont centrées réduites (moyenne = 0, variance = 1). Comme chaque couche observe des entrées produites par des couches qui la précèdent, il serait avantageux d'obtenir des entrées centrées et réduites pour chaque couche.

Batch normalisation est une composante qui se trouve entre les couches du réseau de neurones et qui prend continuellement la sortie d'une couche particulière et la normalise avant de l'envoyer à la couche suivante comme entrée [23].

- **Le Dropout**

Le Dropout est une technique où des neurones sélectionnés au hasard sont ignorés pendant l'apprentissage. Cela signifie que leur contribution à l'activation des neurones qui leur succède est temporairement supprimée lors de la phase de propagation et toutes les mises à jour de poids ne sont pas appliquées aux neurones lors de la phase de rétro-propagation [23].

2.4.3 Les architectures des CNN les plus connus

Plusieurs architectures dans le domaine des réseaux convolutifs existent, les plus utilisées sont :

2.4.3.1 LeNet-5

LeNet un réseau de convolution à 7 niveaux pionnier de LeCun et al en 1998, qui classifie les chiffres, a été appliqué par plusieurs banques pour reconnaître les numéros manuscrits sur les chèques numérisés en 32x32 pixels. La capacité à traiter des images à plus haute résolution nécessite des couches plus convolutives et plus grandes, cette technique est donc limitée par la disponibilité des ressources informatiques [58]. La figure 2-16 montre l'architecture de LeNet-5.

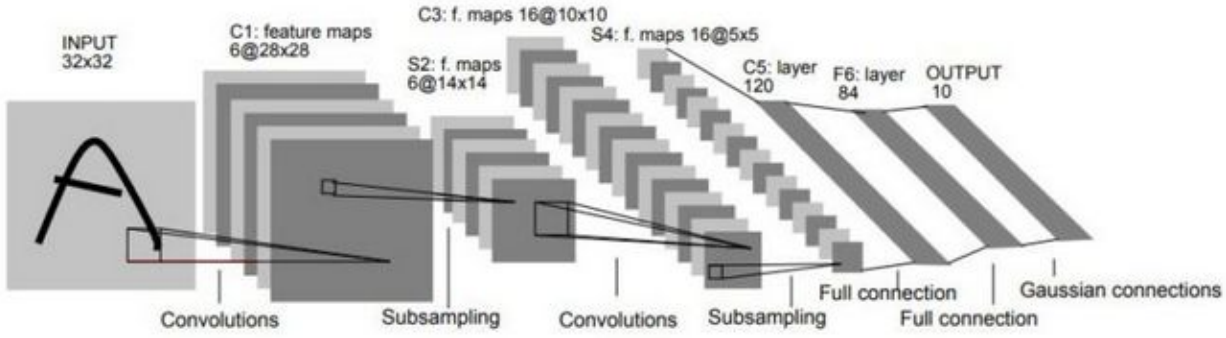


Figure 2-16 : Architecture LeNet.

Description des couches du réseau LeNet [59] :

- **Première couche :** Cette première couche est appliquée sur une image en niveau de gris de taille 32x32, avec six filtres de convolution de taille 5x5 et un pas (stride) égal à 1. La dimension de l'image passe de 32x32x1 à 28x28x6.
- **Deuxième couche:** Le LeNet-5 applique une couche de sous-échantillonnage avec une taille de filtre de 2x2 et un pas de deux. Les dimensions de l'image résultante seront réduites à 14x14x6.
- **Troisième couche :** C'est une deuxième couche de convolution avec 16 cartes de caractéristiques (filtres) de taille 5x5 et un pas de 1, les dimensions de l'image passent à 10x10x16.

- **Quatrième couche :** La quatrième couche est à nouveau une couche de sous échantillonnage avec une taille de filtre de 2 x 2 et un pas de 2. Cette couche est identique à la deuxième couche, la différence est qu'elle comporte 16 cartes de caractéristiques de sorte que la sortie soit réduite à 5x5x16.
- **Cinquième couche :** La cinquième couche (C5) est une couche convolutionnelle entièrement connectée avec 120 cartes de caractéristiques de taille 1x1. Chacune des 120 unités est connectée à tous les 400 nœuds (5x5x16) de la quatrième couche.
- **Sixième couches :** La sixième couche est une couche entièrement connectée avec 84unités
- **Couche de sortie :** Enfin, il existe une couche de sortie softmax entièrement connectée avec 10 valeurs possibles correspondant aux chiffres de 0 à 9.

2.4.3.2 AlexNet

Le premier travail qui a popularisé les réseaux convolutionnels dans la vision par ordinateur, développé par Alex Krizhevsky , Ilya Sutskever et Geoff Hinton . L'AlexNet a été soumis au **Challenge ImageNet ILSVRC** en 2012 et a nettement surperformé le deuxième finaliste (erreur de top 5 de 16 % par rapport à la deuxième place avec une erreur de 26%). Le réseau avait une architecture très similaire à celle de LeNet , mais il était plus profond , plus grand et comportait des couches convolutives empilées les unes sur les autres (auparavant , il était courant de n'avoir qu'une seule couche CONV immédiatement suivie d'une couche POOL) [60]. La figure 2.17 montre l'architecture AlexNet.

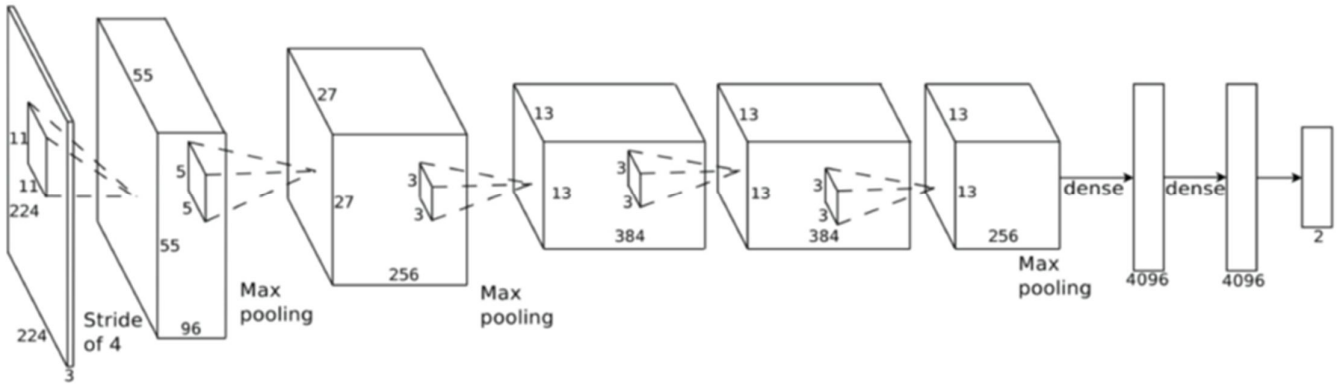


Figure 2-17 : Architecture AlexNet.

Résumé de l'architecture AlexNet est présenté dans le tableau 2-1 ci-dessous [61] :

Tableau 2-1 : Résumé de l'architecture AlexNet.

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	227x227x3	-	-	-
1	Convolution	96	55 x 55 x 96	11x11	4	relu
	Max Pooling	96	27 x 27 x 96	3x3	2	relu
2	Convolution	256	27 x 27 x 256	5x5	1	relu
	Max Pooling	256	13 x 13 x 256	3x3	2	relu
3	Convolution	384	13 x 13 x 384	3x3	1	relu
4	Convolution	384	13 x 13 x 384	3x3	1	relu
5	Convolution	256	13 x 13 x 256	3x3	1	relu
	Max Pooling	256	6 x 6 x 256	3x3	2	relu
6	FC	-	9216	-	-	relu
7	FC	-	4096	-	-	relu
8	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

2.4.3.3 VGGNet

Le finaliste au concours ILSVRC 2014 est surnommé VGGNet par la communauté et a été développé par Simonyan et Zisserman. VGGNet se compose de 16 couches convolutives et est très attrayant en raison de son architecture très uniforme. Similaire à AlexNet, seulement 3x3 convolutions, mais beaucoup de filtres. Formé sur 4 GPU pendant 2 à 3 semaines. C'est actuellement le choix préféré de la communauté pour l'extraction de fonctionnalités à partir d'images. La configuration de poids du VGGNet est accessible au public et a été utilisée dans de nombreuses autres applications et défis comme extracteur de fonctionnalités de base. Cependant, VGGNet se compose de 138 millions de paramètres, ce qui peut être un peu difficile à gérer [58]. La figure 2-18 montre l'architecture de VGGNet.

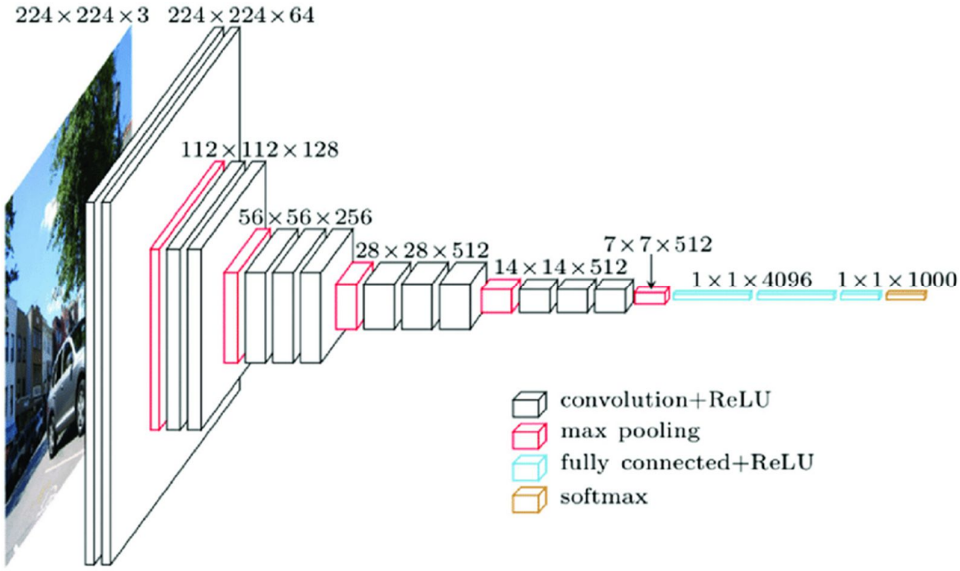


Figure 2-18 : Architecture du réseau VGGNet.

Résumé de l'architecture VGG16 est présenté dans le tableau 2-2 [62]:

Tableau 2-2 : Résumé de l'architecture VGG16.

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	224 x 224 x 3	-	-	-
1	2 X Convolution	64	224 x 224 x 64	3x3	1	relu
	Max Pooling	64	112 x 112 x 64	3x3	2	relu
3	2 X Convolution	128	112 x 112 x 128	3x3	1	relu
	Max Pooling	128	56 x 56 x 128	3x3	2	relu
5	2 X Convolution	256	56 x 56 x 256	3x3	1	relu
	Max Pooling	256	28 x 28 x 256	3x3	2	relu
7	3 X Convolution	512	28 x 28 x 512	3x3	1	relu
	Max Pooling	512	14 x 14 x 512	3x3	2	relu
10	3 X Convolution	512	14 x 14 x 512	3x3	1	relu
	Max Pooling	512	7 x 7 x 512	3x3	2	relu
13	FC	-	25088	-	-	relu
14	FC	-	4096	-	-	relu
15	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

2.4.3.4 GoogleNet

GoogleNet appelé aussi Inception Module, il contient 22 couches, ces couches ne sont pas toutes séquentielles comme dans les architectures précédentes. Il comporte des couches qui se traitent en parallèle. Son architecture est montrée sur la figure 2-19 [23].

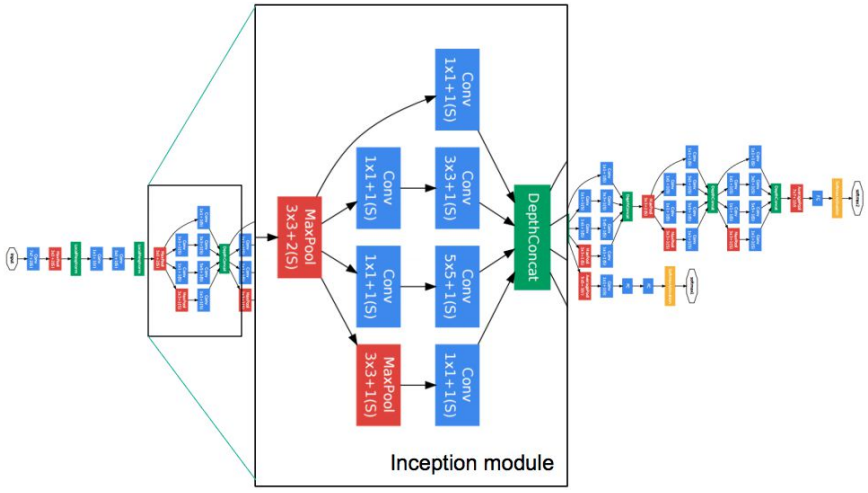


Figure 2-19 : Architecture du réseau GoogleNet.

La partie encadré s'appelle « **inception module** », la figure 2.20 l'énonce avec plus précision.

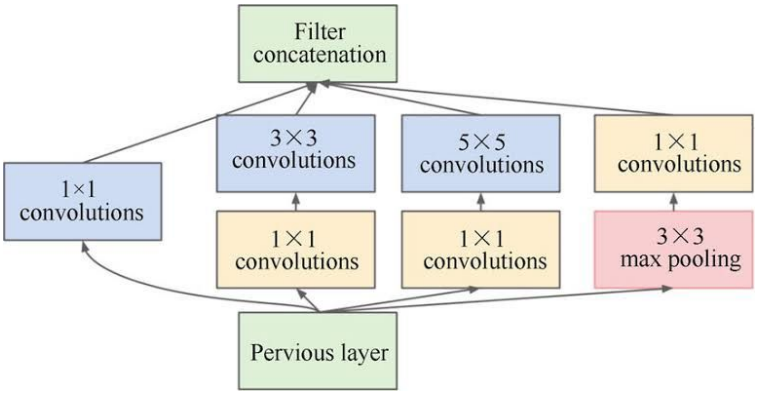


Figure 2-20 : Architecture d'un module « inception ».

2.5 Conclusion

Dans ce chapitre nous avons présenté en détails le principe de l'apprentissage automatique et les réseaux de neurones, ainsi que les réseaux de neurones profonds et les différents modèles existants.

Par la suite, nous avons introduit le principe et les architectures des réseaux révolutionnaires utilisés dans le domaine de la vision par ordinateur qui sont les CNN.

Dans ce projet on a travaillé avec trois différentes architectures CNN pour la détection qui sont : « Ultra-light, Yolov3 et Mtcnn », nous allons expliquer en détails ces méthodes dans le chapitre suivant.

Chapitre 3 : Plateforme de détection et de reconnaissance de visage selfie.

3.1 Introduction

La reconnaissance faciale est largement appliquée dans le domaine de la sécurité où la détection des visages est une première étape pour effectuer la reconnaissance faciale.

Ce chapitre vise à expliquer plus en détail les différentes architectures des réseaux de neurones à convolution utilisées dans la plateforme de détection et l'alignement à savoir les modèles Ultra-light, YOLO et MTCNN ainsi que l'architecture FaceNet dédiée pour la reconnaissance de visage selfie.

L'application de reconnaissance et d'authentification élaborée comprend quatre modules principaux :

- Création de la base de données.
- Phase de détection, alignement et d'extraction des caractéristiques. Pour cette étape de travail, nous avons testé les modèle basé sur les CNN qui sont MTCNN YOLO et Ultra-light.
- Phase d'apprentissage : Cette phase de travail n'a pas été effectuée pour les modèles MTCNN et YOLO, alors que pour le modèle Ultra-light une inférence du modèle est utilisée pour effectuer l'apprentissage sur les nouvelles données à l'aide de FaceNet (Annexe A).
- Phase de comparaison et de décision, elle constitue la phase de reconnaissance en temps réel. Cette étape consiste à créer une incrustation de visage. Une incrustation de visage est un vecteur qui représente les caractéristiques extraites du visage. Ceci peut ensuite être comparé aux vecteurs générés pour d'autres faces. Par exemple, un autre vecteur proche (d'une certaine mesure) peut être la même personne, tandis qu'un autre vecteur éloigné (d'une certaine mesure) peut être une personne différente. Le modèle de classificateur utilisé prend une incorporation de visage comme entrée et prédit l'identité du visage. Dans notre application, le modèle FaceNet est utilisé comme classifieur, il prétraite un visage comme entrée et crée une imbrication de visage qui est stockée et utilisée comme entrée à notre système de reconnaissance.

3.2 Description du système de reconnaissance réalisé

Le schéma bloc de l'application est illustré sur la figure 3-1 :

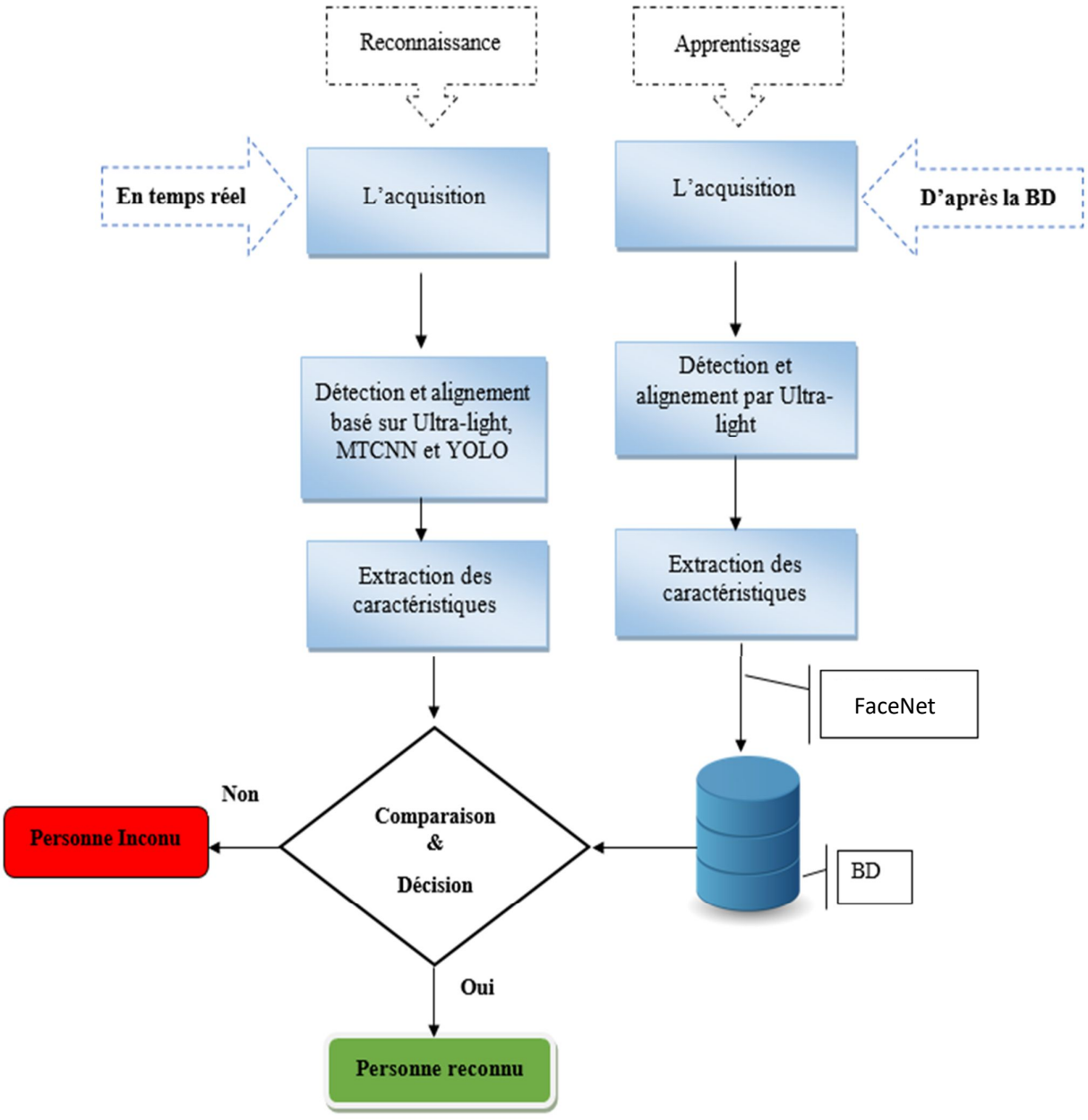


Figure 3-1: Diagramme blocs du système développé.

3.2.1 Création de la base de données

Cette étape est dédiée à stocker les informations lors de l'ajout d'un nouveau utilisateur dans le système.

Pour la détection par les architecture MTCNN et YOLO, l'étape d'apprentissage, n'a pas été effectuée, pour se faire, nous avons utilisé des modèles prédéfinis basés sur FaceNet et Darknet respectivement. En revanche, Pour Ultra-light, nous avons utilisé une inférence à travers la plateforme d'apprentissage profond « Open Neural Network Exchange (ONNX) pour effectuer l'entraînement sur une base de données que nous avons rassemblé et qui est constituée de 7 personnes dont chacun possède environ 35 images et 2 vidéos.

3.2.2 Phase d'apprentissage

Cette phase est effectuée seulement pour le modèle Ultra-light à travers une inférence en utilisant le modèle pré-entraîné d'Ultra-light chargé par la plateforme d'apprentissage profond ONNX.

ONNX est développé en 2016 à l'USA, c'est un format standard ouvert (Open source) pour représenter des modèles d'apprentissage automatique. C'est un modèle de deep learning et machine learning interopérable supporté par Caffe2, Chainer, MxNet, PyTorch, mais pas TensorFlow. La solution permet l'import et l'export de modèles de et vers ces différents outils [63] .Le modèle ONNX est accompagné de blocs-notes Jupyter pour la formation du modèle et l'exécution de l'inférence avec le modèle entraîné. Les blocs-notes sont écrits en Python et incluent des liens vers le jeu de données visage ainsi que des références au document d'origine qui décrit l'architecture du modèle.

- **Bloc inférence**

  Le traitement du bloc inférence est comme suit : Le modèle apprend les caractéristiques distinctives des visages et produit des images de visage en entrée. Pour chaque image de visage, le modèle produit un vecteur d'incorporation de longueur fixe, comme une caractéristique unique du visage. Les vecteurs générés à partir de différentes images de la même personne ont une plus grande similitude que ceux issus de différentes personnes. Ces plongements sont utilisés pour comprendre le degré de similitude entre deux visages à l'aide de métriques de distance euclidienne ou la similitude cosinus. La

figure 3-2 représente la relation entre la phase d'apprentissage et l'inférence.

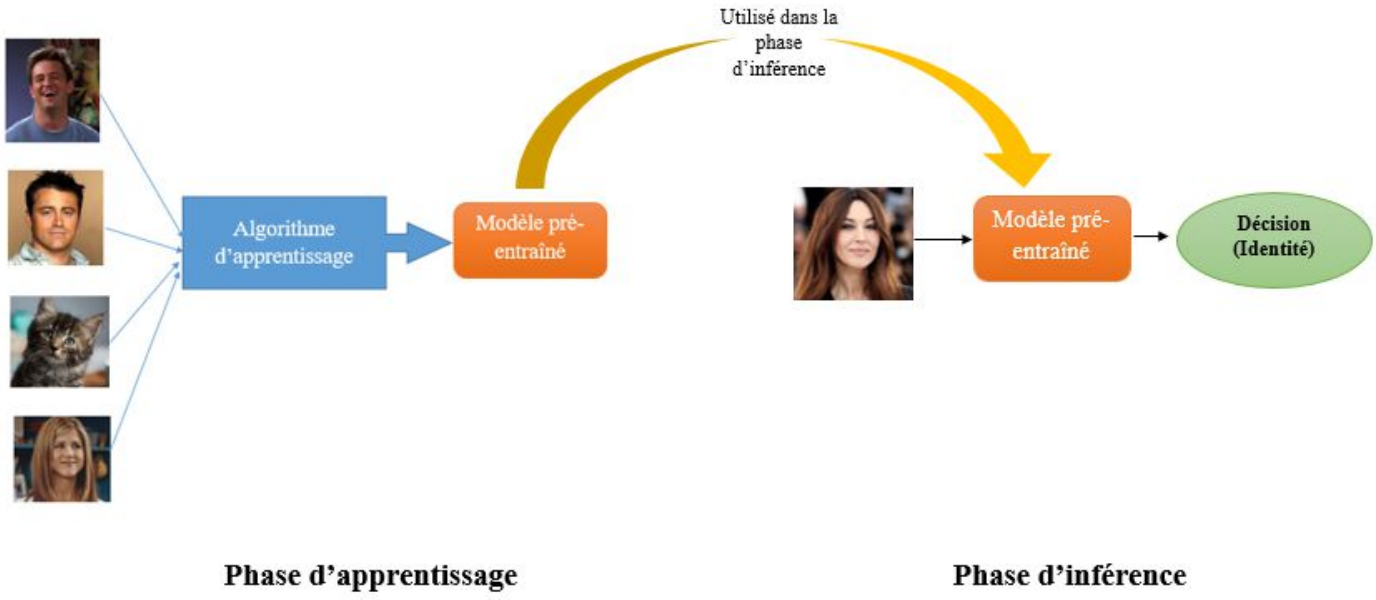


Figure 3-2: La relation entre la phase d'apprentissage et l'inférence.

Le processus d'apprentissage par inférence suit le même processus de la reconnaissance à savoir : l'acquisition de l'image, la détection et l'extraction des caractéristiques. Dans notre cas, l'acquisition des images ou trames pour les séquences vidéo est faite à partir des images et des vidéos capturées en temps différé. Ensuite, un prétraitement est effectué. Il consiste à redimensionner les images en entrées et à les normaliser en enlevant la valeur moyenne des pixels. Ainsi, toutes les régions d'intérêt (ROI) détectés sont redimensionnées pour être compatible à l'entrée du CNN. Enfin l'extraction des caractéristiques (vecteurs caractéristiques) est effectuée. Ces dernières sont stockées dans la base de données pour être par la suite utilisée dans la phase de comparaison des caractéristiques.

La plupart des approches de l'apprentissage profond nécessitent un GPU pour effectuer les traitements nécessaires en raison de la quantité de données. Le calcul mathématique de Deep Learning sur un CPU peut prendre des mois ! Mais ces calculs peuvent être assignés à des GPU pour un calcul plus rapide. Comme, les GPU sont chers, de même que les plates-formes cloud. Nous avons effectué l'apprentissage sur la plateforme Colab. Celle-ci offre un GPU Nvidia Tesla K80 gratuit.

3.2.3 Phase de détection

3.2.3.1 Phase de détection basée sur MTCNN

MTCNN est un modèle de détection développé en 2016 (en anglais : Multi-Task Cascaded Convolutional Neural Network) [64]. Son principe consiste à détecter des boites englobantes de visage ("faces bouding box") dans une image et cinq points de repère qu'on appelés "Landmarks". Il comprend trois étapes de traitements (Figure 3-3) pour effectuer simultanément la détection des visages et celle des repères faciaux. Dans le premier bloc de traitement, plusieurs fenêtres candidates via un réseau CNN peu profond est proposé, puis un deuxième réseau CNN plus complexe qui consiste à affiner les fenêtres pour rejeter un grand nombre de fenêtres ne contenant pas de visage. Dans le troisième bloc de traitement, un CNN plus puissant est utilisé pour affiner le résultat et afficher les positions des repères faciaux.

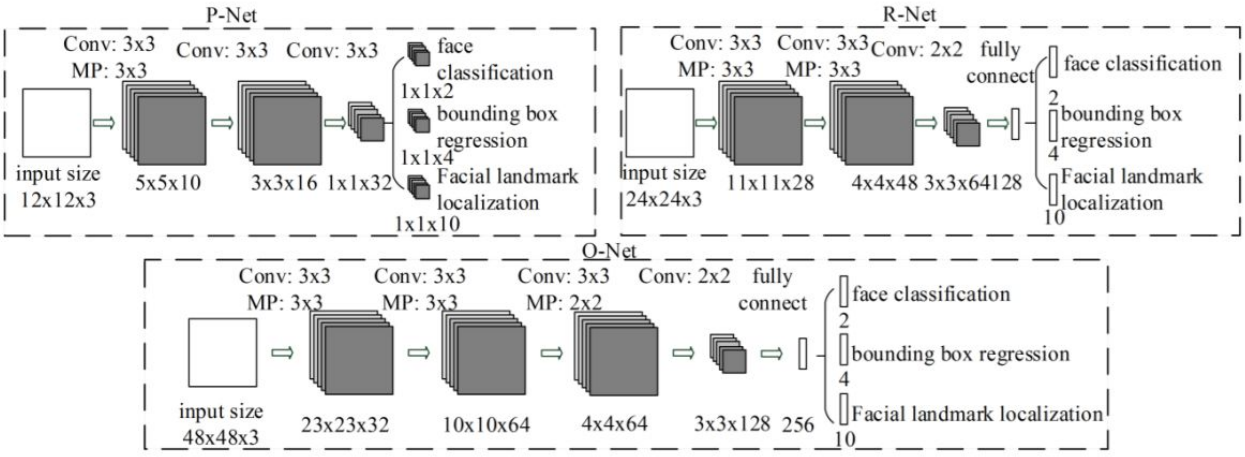


Figure 3-3: L'architecture du modèle de détection de visage MTCNN.

a. Le fonctionnement du bloc P-Net

Dans la première étape, un réseau entièrement convolutionnel appelé réseau de propositions (P-Net) est utilisé pour obtenir les régions proposées et leurs vecteurs de régression par boite de délimitation. Les vecteurs de régression obtenus sont utilisés pour calibrer les régions proposées, puis pour appliquer une suppression non maximale (NMS) afin de fusionner des régions fortement Chevauchées. Le principe de cette étape de traitement consiste à créer une pyramide d'images à partir d'une image en entrée afin de détecter les visages de différentes tailles (Figure 3-4). En d'autres termes, différentes copies de différentes tailles d'une même image sont créés. Ceci afin de chercher des visages de différentes tailles dans l'image. Pour

chaque copie du visage mise à l'échelle, un filtre (noyau) de Kernel de taille 12x12 parcourt toute l'image de manière incrémentale en recherchant les visages [65].

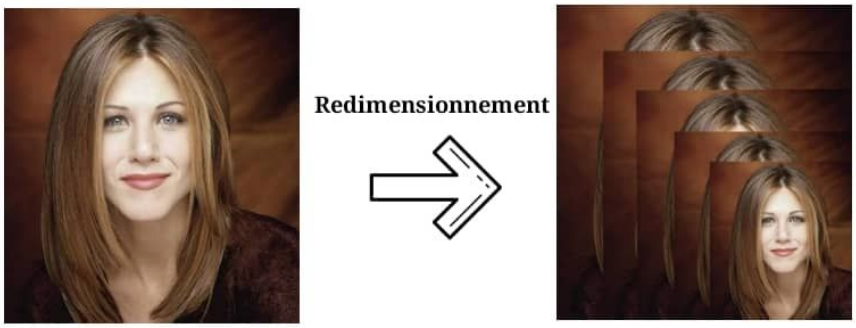


Figure 3-4: Image en entrée organisée en pyramide.

Le traitement commence par une région de l'image comprise entre (0,0) et (12,12) située dans le coin supérieur gauche (Figure 3-5). Celle-ci est transmise au bloc P-Net, qui renvoie les coordonnées d'un cadre de sélection s'il détecte un visage. Ensuite, il répète ce processus avec les régions (0 + 2a, 0 + 2b) à (12 + 2a, 12 + 2b), en décalant les pixels du noyau 12x12 avec un pas de 2 pixels vers la droite ou vers le bas. Le décalage de 2 pixels est appelé pas (stride ou pas), ou le nombre de pixels que le noyau déplace à chaque fois. Dans ce cas, le temps de traitement est réduit. Le seul inconvénient de ce processus est de recalculer tous les index liés au pas de déplacement. La méthode MTCNN permet de faire la détection et index liés au pas de déplacement. La méthode MTCNN permet de faire la détection et l'alignement des visages [65].

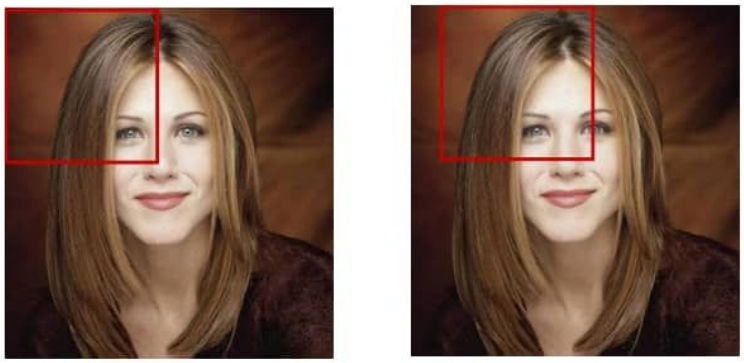


Figure 3-5: Le noyau et la fenêtre glissante.

Chaque noyau serait plus petit par rapport à une image de grande taille en entrée, de sorte qu'il serait capable de trouver des visages plus petits. De la même manière, le noyau serait plus gros par rapport à une image de taille plus petite, ce qui lui permettrait de trouver des visages plus grands dans l'image de plus petite taille. Plusieurs copies de l'image de différentes tailles sont créées et transmises au premier réseau de neurones, P-Net, et rassembler en sortie du bloc de traitement. Les poids et les biais de P-Net ont été formés de manière à produire un cadre de sélection relativement précis pour chaque noyau de 12 x 12 [64].

Cependant, le réseau est plus confiant sur certaines boites que sur d'autres. Par conséquent, il est nécessaire d'analyser la sortie P-Net pour obtenir une liste de niveaux de confiance pour chaque cadre de sélection et supprimer les cadres avec un niveau de confiance inférieur (c'est-à dire que les cases dont le réseau n'est pas tout à fait sûr de contenir un visage). Après avoir sélectionné les cases avec une confiance accrue, les coordonnées des cadres de sélection sont normalisées en les convertissant en ceux de l'image réelle non mise à l'échelle. Comme la plupart des noyaux sont dans une image réduite, leurs coordonnées seront basées sur la plus petite image. Cependant, il reste encore beaucoup de cadres de sélection, qui se chevauchent souvent. Le traitement basé sur la méthode suppression non maximale (NMS) [66] est utilisé pour réduire le nombre de cadres de sélection (Figure 3-6). Le processus de NMS est effectué en triant d'abord les boites englobantes (et leurs noyaux respectifs 12 x 12) en fonction de leur confiance ou de leur score. Dans d'autres modèles, le système NMS utilise la plus grande boite englobante au lieu de celle sur laquelle le réseau a le plus confiance [64].

Par la suite, la surface de chacun des noyaux, ainsi que la zone de chevauchement entre chaque noyau et le noyau ayant le score le plus élevée sont calculées. Les noyaux qui se chevauchent beaucoup avec le noyau très performant sont supprimés. Enfin, le traitement NMS renvoie une liste des boites englobantes survivantes. Le processus NMS est effectué une fois pour chaque image mise à l'échelle, puis une fois de plus avec tous les noyaux survivants de chaque échelle.

Cela supprime les cadres de sélection redondants, ce qui permet de limiter la recherche à une boite précise par visage [64].

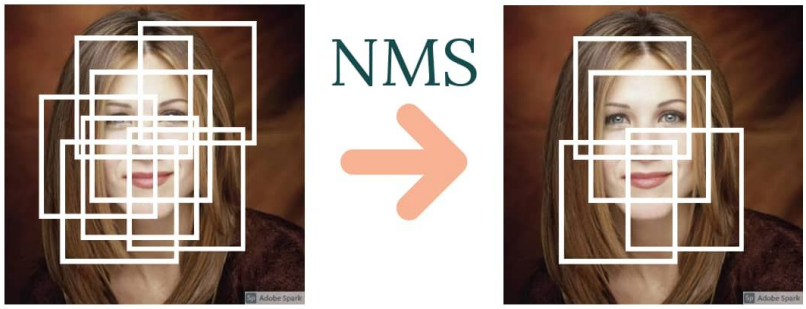


Figure 3-6: Le réseau analyse la sortie P-Net.

b. Fonctionnement du bloc R-Net

A la sortie du bloc P-Net, tous les candidats sont acheminés vers un autre CNN, appelé réseau raffiné (R-Net), qui rejette en outre un grand nombre de faux candidats et effectue un étalonnage avec une régression du cadre de sélection et une NMS [64].

Parfois, une image peut contenir uniquement une partie du visage qui s'observe du cotée du cadre. Dans ce cas, le réseau peut renvoyer une zone de sélection partielle en dehors du cadre. Pour chaque cadre de sélection, un tableau de la même taille est créé et les valeurs de pixel (l'image dans le cadre de sélection) sont copies dans le nouveau tableau. Si le cadre de sélection est en dehors des bords, uniquement la partie de l'image du cadre de sélection est copiée dans le nouveau tableau et le reste du tableau est complété par des zéros. Une fois que les tableaux des cadres de sélection sont remplis, ils sont redimensionnés à 24 x 24 pixels et normalisés à des valeurs comprises entre -1 et 1 [65].

Une fois que tous tableaux d'images sont de taille 24 x 24 (autant que le nombre de boites englobantes ayant survécu à l'étape 1, puisque chacune de ces boites englobantes a été redimensionnée et normalisée dans ces noyaux) ; ils sont rassemblés et transmis au réseau R-Net. La sortie de R-Net est similaire à celle de P-Net, elle inclut les coordonnées plus précises des nouveaux cadres de sélection, ainsi que le niveau de confiance de chacun de ces derniers. Une fois encore, les boites avec moins de confiance sont éliminées (Figure 3-7). Après

normalisation des coordonnées, les boites englobantes sont transformées en un carré à transmettre au réseau O-Net [65].

Figure 3-7: Le réseau rejette un grand nombre de faux candidats.

c. Fonctionnement du bloc de traitement O-Net

Cette étape est semblable à la deuxième étape, mais dans celle-ci, les régions de visage avec les cinq positions de points de repère faciaux sont identifiées (les deux yeux, le nez et les extrémités de la bouche) (Figure 3-8). Avant de traiter les boites englobantes de R-Net, les boites qui sont hors limites sont remplies de zéros et sont redimensionnées à la taille 48 x 48 pixels, puis transmis au réseau O-Net. Les résultats du bloc O-Net différent légèrement de ceux de P-Net et de R-Net. Le traitement effectué par le bloc O-Net fournit trois sorties : les coordonnées du cadre de sélection (out [0]), les coordonnées des cinq points de repère faciaux (out [1]) et le niveau de confiance de chaque cadre (out [2]). Une fois encore, les boites dont le niveau de confiance est faible sont éliminées et les coordonnées des boites englobantes et celles du repère facial sont standardisées. Enfin, le traitement NMS est appliqué. A ce stade, il ne doit exister qu'un seul cadre de sélection pour chaque visage de l'image. La toute dernière étape consiste à regrouper toutes les informations dans un dictionnaire comportant trois clés à savoir : (1) box, (2) confiance et (3) points-clés. `Box' contient les coordonnées du cadre de sélection, `confiance' contient le niveau de confiance du réseau pour chaque case, et `points-clés' inclut les coordonnées de chaque repère facial (yeux, nez et extrémités de la bouche) [64].

Figure 3-8: Le cadre de sélection, les 5 points de repère faciaux.

3.2.3.2 Détection basée sur le modèle YOLO

Le modèle YOLO (You Only Look Once) a été créé en 2015 par Joseph Redmon et Ali Farhadi et a subi de nombreuse évolutions afin d'arrivé à sa version 3 en 2018[67], nouvelles versions ont été développé en 2020 (version 4 et 5) et ils sont en cours de tests. La tâche de détection d'objet consiste à déterminer l'emplacement des objets sur l'image, ainsi qu'à les classifier.

- **Fonctionnement du modèle**

Le fonctionnement de YOLO peut être résumé de la manière suivante : L'image en entrée du modèle est divisée en grilles de cellules (SxS) dont chaque grille de cellule est responsable de la prédiction d'un seul objet présent dans l'image ainsi que les boites englobantes (boundry boxes) et les probabilités de classes. Le résumé du fonctionnement est présenté dans la figure 3-9.

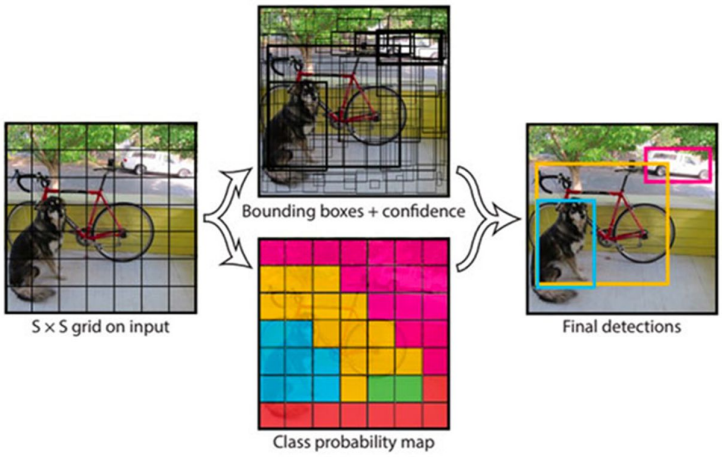


Figure 3-9: La distribution des grilles de cellules dans l'image pour déterminer les objets présents.

Pour La classe de détection visage, On a le vecteur suivant [68] :

Pc	Bx	By	Bh	Bw	C

Dont les éléments sont :

Pc : probabilité qu'il existe un objet dans la cellule.

Bx, By : les coordonnées normalisées entre 0 et 1 représentant le centre de la cellule.

Bw, Bh : la largeur et la hauteur normalisées de la cellule sélectionnée.

C: représente la classe visage.

La figure 3-10 illustre la détection avec YOLO.

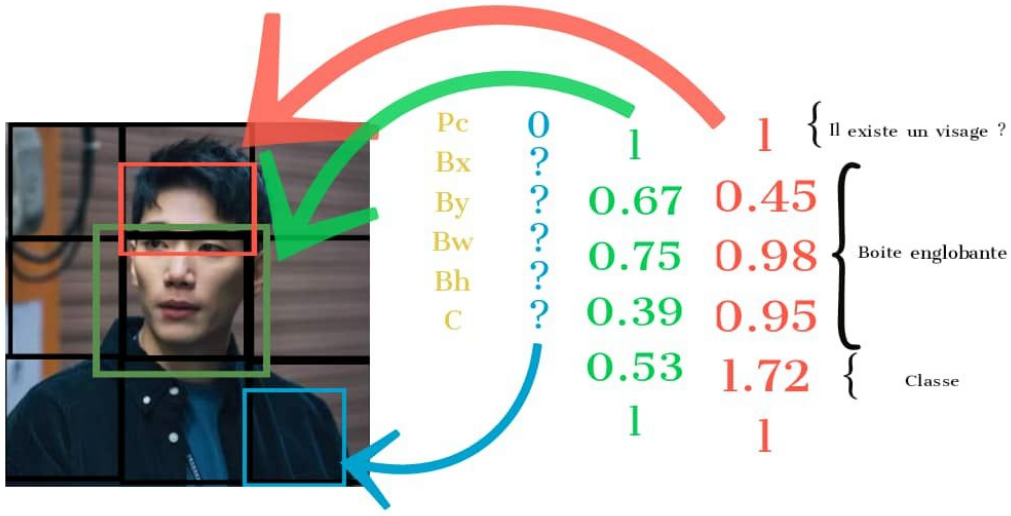


Figure 3-10: Détection par YOLO.

YOLO effectue de multiples détections pour cela il applique le processus NMS afin de les boites dupliquées ayant faible.

YOLO est un modèle conçu pour la détection des objets d'une manière générale et très connue pour sa haute précision. Pour bénéficier de ses avantages, on a pris un cas particuliers de ce modèle pour détecter seulement les visages dont on utilise une seule classe qui est la classe « Face » ou « visage ».

- **Les variations de YOLO**
  - **YOLO version 1**

YOLOv1 est constitué de 24 couches de convolution suivie par 2 couches entièrement connectées [69]. La figure 3-11 représente l'architecture de YOLOv1.

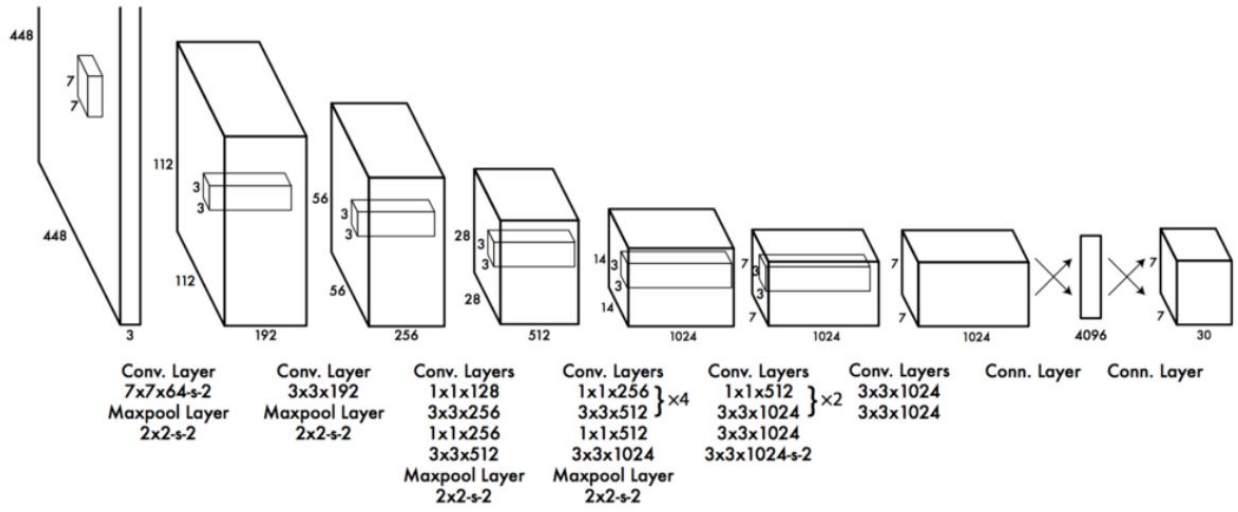


Figure 3-11: Architecture de YOLOv1.

L'entrée du modèle est une image de taille 448x448x3 et la sortie est un vecteur de taille 7x7x30. Il est constitué de trois types de couche : **convolution, max pooling et entièrement connecté.**

- La première couche est une couche de convolution de taille 7x7x64, elle applique un filtre de taille 7x7 64 fois avec un pas (stride) de 2 suivie d'une couche de max pooling de taille 2x2 avec un pas de 2.
- La 2ème couche de convolution est de taille 3x3x192 suivie par un max pooling de taille 2x2 avec un pas de 2.
- Les quatre couches de convolution suivantes sont de taille 1x1x128, 3x3x256, 1x1x256, 3x3x512 respectivement suivies par max pooling de taille 2x2 avec un pas de 2.
- Un bloc de couches de convolution qui se répète 4 fois et qui est suivie par un max pooling de taille 2x2 avec un pas de 2.
- Un 2ème bloc qui se répète 2 fois suivie par deux couches de convolutions puis deux autres couche de convolution de taille de 3x3x1024
- Enfin, deux couches entièrement connectées qui utilise une fonction d'activation linéaire ReLU [23].

- **YOLO version 2**

La 2ème version du modèle YOLO est développée en 2016 dans le but d'améliorer la précision et la vitesse du traitement, pour cela des modifications sont appliquées au YOLOv1. YOLOv2 a remplacé le modèle ImageNet utilisé dans YOLOv1 avec DarkNet-19 qui comporte 19 couches de convolution et cinq (05) couches de max pooling [70]. Le tableau 3-1 représente l'architecture de YOLOv2.

YOLOv2 a apporté d'autres modifications telles que «batch normalization » qui consistent à normaliser la sortie de chaque couche avant de l'envoyer aux couches suivantes. Le tableau 3-1 représente l'architecture de YOLOv2.

Tableau 3-1: L'architecture de YOLOv2.

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

- **YOLO version 3**

Le modèle YOLO version 3 a été développé en 2018 en effectuant plusieurs modifications par rapport aux versions antérieures. YOLOv3 emploie le framework de DarkNet-53 qui est

composé de 53 couches de convolution conduisant à une architecture de 106 couches. Tableau 3-2 représente l'architecture du modèle YOLOv3 [71].

Les modifications apportées au modèle YOLOv2:

- **Détection multi échelles :** YOLOv3 permet de détecter des objets à trois échelles différentes.
- **Meilleur détection pour les petits objets :** grâce à la détection en multi échelles, la détection de petits objets est plus facile. Les filtres de taille 13x13 permettent la détection des grands objets, les filtres de taille 52x52 détectent des petits objets, tandis que les filtres de 26x26 sont responsables de la détection des objets moyens taille [72].
- **Remplacer la fonction de perte :** Les versions antérieures de YOLO utilisaient la fonction softmax comme fonction de perte, alors que le modèle YOLOv3 remplace la fonction softmax par un classificateur logistique indépendant qui calcule la vraisemblance de l'objet en entrée pour qu'il appartienne à une classe spécifique.

Tableau 3-2: L'architecture de YOLOv3.

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

3.2.3.3 Détection basée sur le modèle Ultra-light

Le modèle Ultra-light est un nouveau modèle de détection développé et publié par Linzaer en Avril, 2019 [73].Il est conçu pour les applications de détection de visage à usage général dans les appareils informatiques à faible puissance et il est applicable aux téléphones Android et iOS ainsi qu'aux PC. Le modèle Ultra-light est sous deux versions à savoir :

- **Modèle version-slim :** Il accepte en entré des images de taille 320x240 pixels dont la taille en mémoire ne dépasse pas 900 Ko. Il est utilisé pour peu d'images (trame pour la vidéo) où figurent des visages de large dimension dont la distance entre le système d'acquisition (caméra, mobile....) et le visage des personnes a détectés est moyenne [74].
- **Modèle RFB (Receptive Field Block):** Il accepte en entré des images de taille 320x240 pixels et 640x480 pixels de haute résolution dont la taille en mémoire est 1.6 Mo. Cette version est utilisée pour les images où figurent des petits visages acquises à une certaine distance par rapport au système [74].

1. **Architecture du modèle Ultra-light**

Dans notre application, nous avons opté pour le modèle RFB pour détecter des visages dans des images de différentes tailles et de haute résolution. L'architecture de ce modèle est basée sur l'architecture de RFB-Net [76]. Cette architecture consiste à remplacer les couches de convolutions conv8_2 et conv9_2 existantes dans l'architecture SSD basée sur le VGG16 [77] par le module RFB à fin de construire une architecture légère. La figure 3-12 représente l'architecture de RFB-Net.

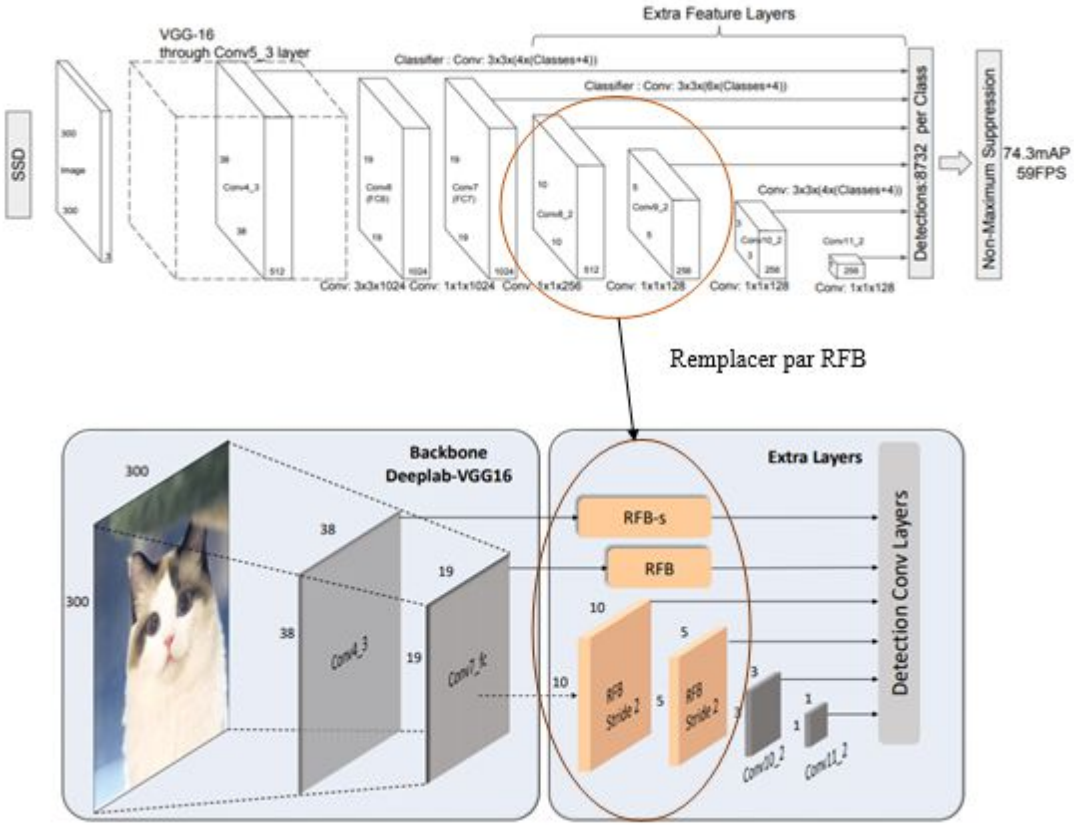


Figure 3-12: Construction de l'architecture de RFB-Net.

2. Fonctionnement

Le fonctionnement de ce modèle peut être résumé en 2 phases principales.

- **Prétraitement**

  Le prétraitement constitue une étape de paramétrage des images en entrée du réseau. Cette étape comprend deux traitements à savoir le redimensionnement des images en entrée et l'opération de normalisation. Dans notre cas, les images à l'entrée du modèle Ultra-light sont redimensionnées à la taille (640x480) pixels. Puis une normalisation est appliqué en soustrayant 127 de la valeur du pixel et divisant le résultat obtenu par 128 pour avoir toutes les valeurs comprises dans l'intervalle [0, 1].

- **Détection**

  Le processus de détection d'Ultra-light consiste à balayer l'image de gauche à droite en associant une probabilité pour chaque boite englobante. Puis les probabilités sont comparés à un seuil prédéfini dans notre cas il est égal à la valeur 0.7. Cette valeur nous permet de garder que l'ensemble des boites englobantes qui ont une haute probabilité de contenir un visage. Une fois les boites englobantes sont déterminées, le processus NMS est appliqué pour filtrer ces boites.

3.2.4 Extraction des caractéristiques

L'alignement est une technique qui applique une rotation normalisée sur le visage qui s'appuie sur les repères faciaux. Pour faire l'alignement, il faut d'abord trouver les repères faciaux et pour cela nous avons utilisé deux modèles de la bibliothèque dlib [78]:

- Le modèle de 5 repères faciaux.
- Le modèle de 68 repères faciaux.

Le modèle de 5 repères faciaux est 10 fois plus petit que le modèle de 68 repères en termes de taille et rapidité d'exécution. Alors que la précision de la structure faciale du modèle 68 repères est plus haute. Les figures 3-13 montrent la représentation des repères faciaux de 68 et 5 repères sur le visage.

68 repères faciaux.

5 repères faciaux.

Figure 3-13: Différence entre la représentation de 68 repères et 5 repères faciaux.

Ces modèles servent à traiter l'image afin d'extraire uniquement les caractéristiques (les yeux, le contour du visage, le nez, la bouche…) sous forme d'un vecteur, qui peuvent être par la suite utilisées dans la phase de comparaison des caractéristiques pour sortir avec une décision.

3.2.5 Phase de comparaison des caractéristiques et décision

Les caractéristiques extraites sont comparées à celles qui sont stockées dans la base de données visages en moyennant une métrique de similitude, dans notre cas, nous avons opté pour la distance euclidienne.

Après avoir exploré les détecteurs de visage MTCNN, YOLO et Ultra-light en incluant un pack de reconnaissance faciale FaceNet, nous avons constaté que la reconnaissance basée sur MTCNN est rapide mais moins précis, la reconnaissance basé sur la détection YOLO a la plus haute précision et le temps d'exécution est long, alors que le détecteur de visage Ultra-light est inégalé en termes de vitesse et produit une précision relativement bonne. Afin d'augmenter le taux de reconnaissance, nous avons opté pour fusionner MTCNN et Ultra-light.

3.2.6 La stratégie de fusion

La fusion des scores est une méthode qui consiste à fusionner les scores de plusieurs modèles à fin de gagner plus de précision [79]. Dans notre cas, les scores sont les vecteurs de distance (distance euclidienne) noté par la formule 3-1. Ils sont calculés en comparant les caractéristiques extraites du visage avec celles de la base de données. Dans ce travail, nous avons adopté pour un schéma de fusion au niveau de score utilisant une règle de somme pondérée.

$$distance\ euclidienne = \sqrt{\sum_{i=1}^{n} (xi - yi)^2} \qquad \text{.... (3-1)}$$

Avant la fusion des scores une normalisation Min-Max (MM) [80] est appliquée sur le vecteur des scores issus des algorithmes MTCNN et Ultra-light. La méthode (MM) projette les scores

dans l'intervalle [0,1], avec max(s) and min(s) sont respectivement le maximum et le minimum de l'ensemble S, le score normalisé n est donné par la formule 3-2 suivante :

$$n = \frac{S - min(S)}{max(S) - min(S)} \qquad \text{.... (3-2)}$$

a. Technique de fusion

Après la normalisation des vecteurs de distances, nous utilisons les méthodes de combinaison des deux vecteurs résultants (n_{ul} – n_{mt}) (« *n* » est assigné à la normalisation). Afin de trouver le vecteur fusionné Df, nous avons utilisé la somme simple et la Somme Pondérée.

- Dans le cas de la fusion par la somme simple, Df est égale à la moyenne des deux vecteurs de distance. Nous choisissons la classe correspondante à la valeur minimale de Df.

- Dans le le cas de la fusion par somme pondérée, nous attribuons un poids *w* à chaque méthode (D_{Ul} - Dmt) basé sur leur taux de reconnaissance. Ainsi, le vecteur Df est calculé comme suit :

$$Df = Dul * 0.750 + Dmt * 0.250 \qquad \text{.... (3-3)}$$

3.2.7 Processus de Reconnaissance basé sur FaceNet

FaceNet est un système de reconnaissance faciale écrit par Florian Schroff et al.à Google dans leur article de 2015 intitulé «FaceNet: A Unified Embedding for Face Recognition and Clustering» [81]. C'est une intégration unifiée pour la reconnaissance faciale et le clustering. C'est un système qui, étant donné une image d'un visage, extrait des caractéristiques de haute qualité du visage et prédit une représentation vectorielle de 128 éléments de ces caractéristiques, appelée incrustation de visage. FaceNet, apprend directement une scène (images ou vidéo) des images de visage à un espace euclidien compact où les distances correspondent directement à une mesure de similitude de visage.

FaceNet est une modèle de réseau neuronal convolutif profond formé via une fonction de perte de triplets qui encourage les vecteurs de la même identité à devenir plus similaires (distance plus petite), alors que les vecteurs d'identités différentes devraient devenir moins similaires (distance plus grande). Les détails de l'architecture du modèle FaceNet sont donnés en annexe.

3.3 Conclusion

Dans ce chapitre nous avons détaillé les différents modèles utilisées dans notre application en commençant par MTCNN passant à YOLOv3 et terminant par Ultra-light. Dans le chapitre suivant nous allons présenter notre application ainsi que les tests et les résultats expérimentaux des modèles en faisant une comparaison entre eux.

Chapitre 4 : Réalisation et résultats.

4.1 Introduction

Après avoir décrit dans le chapitre précédent les différentes méthodes proposées pour construire un système de détection et de reconnaissance de visage, nous allons dans ce chapitre présenter les différents résultats obtenus pour chaque étape du système. Nous commençons par la description du langage de programmation python et Tensorflow et les outils de traitement d'images utilisés. Nous détaillons par la suite, la plateforme réalisée et enfin nous terminons par l'analyse des résultats et quelques illustrations de reconnaissance issues du système de reconnaissance faciale développé.

Le fonctionnement du système développé comporte trois étapes :

1. Création d'une base de données.
2. Formation du modèle particulièrement pour la détection par Ultra-light.
3. Reconnaître les données en temps réel.

Avant de commencer, il est important de citer toutes les bibliothèques nécessaires. Les bibliothèques comprennent,

1. Opencv: pour traiter l'image / vidéo.

2. Numpy: pour travailler avec des pixels sous forme de tableaux.

3. Onnx: pour travailler avec les modèles onnx.

4. Dlib: pour les mappages faciaux (points de repère).

5. Os: pour changer / créer un répertoire et aussi pour ouvrir des fichiers dans le répertoire.

6. Imutiles: pour manipuler l'image (prétraitement).

7. Tensorflow: pour créer des couches et créer / charger des modèles.

8. Darknet.

9. Pickle.

4.2 Environnements de travail

Dans cette section, nous présentons l'environnement matériel utilisé pour réaliser l'application. Les caractéristiques sont les suivantes :

4.2.1 Environnement matériel

La configuration de la machine utilisée est comme suit :

- Processus : Intel Core i7 – 8550U CPU 1.80 GHz x 8
- Système d'exploitation : Ubuntu 16.04 64 bit.
- RAM : 8.00 GO
- Disque Dur : 1.00 TO
- Carte graphique accélératrice de traitements en temps réel (GPU) : AMD 4G.

4.2.2 Environnement logiciel

Les langages de programmation utilisés dans ce projet sont Python version 3.6.10 et OpenCv version 4.1.0 avec leurs différentes bibliothèques. La bibliothèque « QtDesigner » est utilisée pour la création de l'interface graphique. Cette bibliothèque est liée avec python à travers le module PyQt5 [82].

a. Le langage Python

Python est un langage de programmation créé en 1989 par Guido van Rossum au centre de recherche CWI au pays-bas. Il est connu par les caractéristiques suivantes [23] :

- Il est gratuit
- Il est multiplate-forme, il fonctionne sur plusieurs systèmes d'exploitation à savoir Windows, Mac OS X, Linux, Android, iOS,
- C'est un langage de haut niveau. Il demande relativement peu de connaissance sur le fonctionnement d'un ordinateur pour être utilisé.
- C'est un langage interprété. Un script Python n'a pas besoin d'être compilé pour être exécuté, contrairement à des langages comme le C ou le C++.

- Il est orienté objet. Il est possible de concevoir en Python des entités qui miment celles du monde réel (une cellule, une protéine, un atome, etc.) avec un certain nombre de règles de fonctionnement et d'interactions.

b. Logiciel de gestion de base de donnée (MySQL)

Le logiciel MySQL est un logiciel open source et un serveur de base de données relationnelle développé dans un souci de performance élevée en lecture. Il est multi-thread et multi-utilisateurs. Il fonctionne sur nombreux systèmes d'exploitation et il fait partie du quatuor LAMP (linux, apache,php,mysql) ainsi que ses variantes comme WAMP sous windows [83].

4.2.3 Plateforme d'apprentissage profond

Il existe plusieurs outils pour utiliser des modèles d'apprentissage profond, parmi ces outils on peut citer : Caffe, Matlab, Tensorflow, Darknet. Chacun d'eux possède ses avantages et inconvénients. Dans notre projet, nous avons utilisé ONNX version 1.6.0 pour charger l'inférence du modèle Ultra-light et Darknet pour le modèle YOLO et Tensorflow pour le modèle de reconnaissance FaceNet.

a. Tensorflow

Tensorflow est l'un des outils les plus utilisés en IA dans le domaine de l'apprentissage profond. Il est développé par Google et il est open source depuis le 9 novembre 2015. Son architecture flexible permet le développement sur plusieurs variétés de plateformes (CPU, GPU, TPU), allant du PC de bureaux à des clusters de serveurs et des mobiles aux dispositifs de bords [84].

b. ONNX

L'Open Neural Network Exchange (ONNX) est un format open source pour les modèles d'IA. Il prend en charge l'interopérabilité entre les framworks. Cela signifie que vous pouvez entraîner un modèle dans l'un des nombreux framworks d'apprentissage automatique populaires tels que PyTorch, le convertir au format ONNX [85].

c. Darknet

Darknet peut être construit avec plusieurs paramètres facultatifs qui sont désactivés par défaut tels que GPU (accélérateur matériel assurant les fonctions de calcul parallèle), CUDNN

(bibliothèque de primitives accélérées par GPU pour les architectures d'apprentissage profond), OPENCV (bibliothèque graphique libre, spécialisée dans le traitement d'images en temps réel), etc...[23].

d. Google Colaboratory (Google Colab)

Google Colaboratory est un service cloud gratuit de Google pour les développeurs d'IA permettant de développer des applications d'apprentissage profond sur un GPU tesla K80 d'environ 12 GO pendant 12 heures [23].

4.3 Etapes de programmation

- **Création d'une base de données.**

Tout d'abord, une base de données doit être créée, où les vidéos ou des photos des personnes sont stockées. Il est préférable que les séquences vidéo soient au format.mp4 ou format.avi, pour que le modèle extraie les fonctionnalités. Plus la vidéo est claire et variée, plus elle peut générer de fonctionnalités. Et chaque dossier est attribué à chaque personne pour avoir sa vidéo correspondante. Les modèles de détections utilisées sont pré-entrainés avec des bases de données différentes. Le modèle MTCNN est entrainé sur une base de données WIDER FACE contenant 32,203 visages publics et 393,703 visages étiquetés. Le modèle Ultra-light est entrainé sur la base de données VOC généré par la base de données WIDER FACE. Tandis que le modèle YOLO est entrainé sur la base de données Pascal VOC créée en 2012.

- **Apprentissage**

Pour la reconnaissance par Ultra-light, le modèle doit être entrainé. Ce dernier est entrainé avec les images extraites des vidéos et photos enregistrées. Avant son entrainement, les images sont prétraitées pour extraire des caractéristiques de la vidéo et pour générer des variations ou des alignements à partir des images source. Cela est effectué en utilisant la fonctionnalité face_mutils feauture de la bibliothèque des imutils le fichier shape_predictor_68_face_landmarks.dat et shape_predictor_5_face_landmarks.dat.

Ensuite, ces fonctionnalités extraites sont sauvegardées dans un fichier.pkl, avec les noms et les images correctement mappés à l'aide de pickle.

Pour utiliser le modèle ultra-light, les packages python suivants (python version 3.6) sont requis: onnx == 1.6.0, onnx-tf == 1.3.0, onnxruntime == 0.5.0, opencvpython == 4.1.1.26, Tensorflow == 1.13.1.

Comme les modèles pré-entraînés ultra_light_640.onnx, ultra_light_320.onnx et version-RFB-320.onnx sont utilisés, donc les images devront être redimensionnées à 640x480 pixels et 320x240 pixels respectivement. Après le prétraitement de l'image, le modèle ONNX est préparé pour créer une session inférence ONNX.

- **Reconnaissance**

La reconnaissance du visage est effectuée grâce FaceNet, en raison de sa vérification de visage en temps réel et de sa haute précision.

Pour reconnaître un visage, il suffit de charger l'ensemble de données dans le « fichier .pkl », puis d'utiliser la distance euclidienne et un seuil prédéfini pour déterminer la différence entre le visage cible et celui d'une personne dans la base de données. Le visage dont la différence est minimale est considéré comme la personne appartenant à la base de données visage. La différence minimale doit être supérieure à celui du seuil pour le considérer comme personne connue. Lorsque la différence est plus grande, la personne est considérée comme inconnue.

4.4 Analyse des résultats

Nous avons testé les trois modèles de détection dans différents scénarios avec des seuils de comparaison et des données différents, nous avons évalué par la suite les résultats de ces tests pour chaque modèle selon les mesures de performances mentionnées dans le chapitre 1. Les tests sont appliqués en utilisant le modèle de 68 repères faciaux. La raison pour laquelle nous avons choisi ce modèle, c'était après un test de détection en utilisant les deux modèles des repères faciaux mentionnés dans le chapitre 3, en affichant les repères faciaux sur le visage. Dans la figures 4-1 nous pouvons voir que les repères des yeux du modèle 5 repères se décalent lorsque le visage s'incline qui mène par la suite à un changement de l'information concernant le visage(différent vecteur caractéristiques), par contre dans la figure 4-2, nous pouvons voir que malgré l'inclinaison du visage les repères faciaux du modèle 68 repères restent presque inchangeable qui est favorable dans le cas d'un système de reconnaissance faciale car les

repères faciaux représentent l'information de base pour effectuer la phase de comparaison des caractéristiques nécessaire à la reconnaissance.

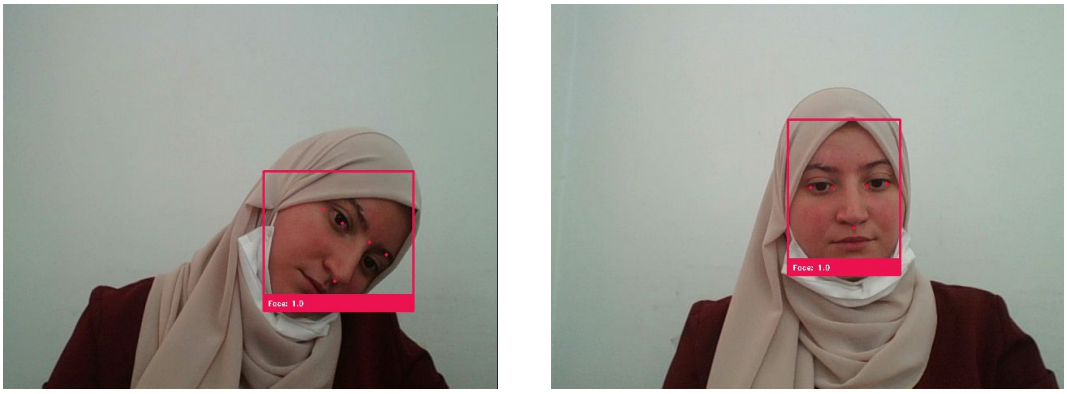


Figure 4-1: Représentation des 5 repères faciaux.

Figure 4-2: Représentation des 68 repères faciaux.

4.4.1 Tests et résultats d'Ultra-Light

Dans ce modèle nous avons commencé à travailler avec une base de données minimale où nous avons utilisé une seule vidéo pour chaque personne, par la suite nous avons ajouté plus de données en marquant simultanément les résultats des tests. Le tableau 4-1 représente les premiers tests du système en temps réel ainsi que les résultats et les données utilisées.

Tableau 4-1: Premiers tests et leurs résultats.

Test #	Données	Résultats
Test 1	Une seule vidéo de 12s – 37s.	Faible
Test 2	Une seule vidéo de 2 min et 15 images.	Acceptable 50 %
Test 3	Deux vidéos et 25 images.	Acceptable 70 %
Test 4	3 vidéos et 35 images.	Acceptable 79 %

Nous avons remarqué que l'entraînement du modèle avec plusieurs données mène à une meilleure reconnaissance. Pour ce modèle, nous avons testé trois versions, à savoir : RFB640, RFB320 et Slim320. Les résultats des versions RFB320 et Slim320 ont été approximables en terme de taux de réussite mais pour la version Slim320, nous observons une augmentation de 1.5% de la valeur du FAR .Tandis que pour la version RFB640, le taux de réussite été élevé par rapport aux autres versions. Nous avons rencontré des difficultés lors de la reconnaissance de plusieurs personnes à la fois où le système mélange les reconnaissances. Pour résoudre ce problème nous avons changé le seuil de comparaison des caractéristiques pour stabiliser le système. Nous avons testé trois seuils différents pour le modèle RFB640 dont leurs valeurs sont: 0.630, 0.512 et 0.320, deux seuils différents pour les modèles RFB320 et Slim320 dont leurs valeurs sont : 0.512, 0.450. Le choix de diminution du seuil de comparaison est fait après une observation de distances calculées entre les caractéristiques extraites et les caractéristiques stocké dans la base de données dont nous avons trouvé que la distance se varie dans un intervalle de [0.27, 0.51]. Le tableau 4-2 représente les résultats d'un test qui a duré 2 minutes pour la version RFB640 avec les différents seuil dont nous pouvons voir que le système a marqué un taux de reconnaissance élevé dans le cas où le seuil de comparaison est égal à 0.512 par rapport aux autres seuils. Le tableau 4-3 représente les résultats des versions RFB320 et Slim320 pour un test d'une durée de 2 minutes avec les différents seuils.

Tableau 4-2: Résultats finaux des tests pour la version RFB640.

Matching	Seuil de comparaison	Temps d'exécution	TAR	FRR	FAR
En temps réel (avec webcam)	0.630	120 s	79 %	21 %	0.06 %
	0.512	120 s	87.23 %	12.77 %	0.005 %
	0.320	120 s	70.52 %	29.48 %	0.4 %

Tableau 4-3: Résultats des tests des versions RFB320 et Slim320.

Matching	Version	Seuil	Temps d'exécution	TAR	FRR	FAR
En temps réel (avec webcam)	**RFB320**	0,512	120 s	67,35 %	32.65 %	0,005
		0,450	120 s	71,23%	28.77 %	
	Slim320	0.450	120 s	66,95 %	33.05 %	0.02
		0.512	120 s	72,28 %	27.72 %	

4.4.2 Tests et résultats de YOLO

Nous avons utilisé la même base de données avec une valeur du seuil qui est également changé pour ce modèle à 0.512. Le tableau 4-4 représente les résultats d'un test qui a duré 2 minutes dont nous pouvons voir que le taux de reconnaissance a marqué 89.69 % qui est très élevé par rapport aux résultats d'Ultra-light malgré que son FAR a marqué une valeur 0.05 %.

Tableau 4-4: Résultats de YOLO.

Matching	Données	Temps d'exécution	TAR	FRR	FAR
En temps réel (avec webcam)	Minimal	120 s	66.30 %	33.7 %	0.10 %
	Plus de données	120 s	89.69 %	10.31 %	0.05 %

4.4.3 Tests et résultats de MTCNN

Le tableau 4-5 représente les résultats d'un test qui a duré 2 minutes avec un seuil de comparaison qui est égale à 0.512 dont nous remarquons que les résultats de MTCNN sont proches des résultats de Ultra-light en terme du taux de reconnaissance qui a marqué un pourcentage de 85.26 %. Le FAR a marqué un pourcentage de 0.06 % qui est proche de la valeur de FAR du modèle YOLO.

Tableau 4-5: Les résultats de MTCNN.

Matching	Données	Temps d'exécution	TAR	FRR	FAR
En temps réel (avec webcam)	Minimal	120 s	80.30 %	19.7 %	0.09 %
	Plus de données	120 s	85.26 %	14.74 %	0.06 %

4.4.4 Tests et résultats de Fusion

Comme mentionné dans le chapitre précédent. Nous avons essayé de fusionner MTCNN et Ultra-light dans le but d'augmenter le taux de reconnaissance. Après l'implémentation, nous avons remarqué que la distance minimale de la comparaison des caractéristiques d'ultra-light est égale à 0.27 et la distance minimale de MTCNN est égale à 0.33 qui montre que le modèle Ultra-light est plus précis que MTCNN, c'est pour cela que nous avons opté pour la sommation pondéré mentionné auparavant.

Lors des tests nous avons remarqué que le taux de reconnaissance a augmenté, tandis que le taux de fausse acceptation a marqué un taux très élevé qui pose un problème, pour cela nous avons diminué le seuil de comparaison à 0.130 mais le problème reste persistant. En observant les valeurs de la distances calculé à partir de la différence entre les caractéristiques, nous avons trouvé que ses valeurs se varie dans un intervalle de [0, 0.033] qui est la raison du choix du

seuil de comparaison d'une valeur de 0.028. Le tableau 4-6 représente les résultats d'un test d'une durée de 2 minutes avec les différents seuils de comparaison dont nous avons remarqué une amélioration de 3 % de taux de reconnaissance mais le problème de fausse acceptation reste persistant.

Tableau 4-6: Résultats de la fusion de MTCNN et Ultra-light.

Matching	Seuil de comparaison	Temps d'exécution	TAR	FRR	FAR
En temps réel (avec webcam)	0.512	120 s	84.30 %	15.70 %	100 %
	0.130	120 s	87.45 %	12.55 %	99.82 %
	0.028	120 s	89.27 %	10.73 %	62.5%

4.4.5 Test Bavette (en plus)

Tenant compte la situation sanitaire actuelle et la propagation du covid-19, nous avons décidé d'ajouter un test pour reconnaître les personnes masqués (avec bavette). Nous avons essayé de faire le test dans deux cas, à savoir : sans apprentissage et avec apprentissage. Les résultats sont présentés dans le tableau 4-7 ci-dessous.

Tableau 4-7: Résultats de test bavette.

Test	Résultats
Sans apprentissage	Taux de réussite = 16.66 %
Avec apprentissage	Taux de réussite = 80.20 %

4.5 Comparaison

En comparant les résultats présentés auparavant pour chaque modèle avec une base de données minimale puis avec plus de données, nous pouvons voir que le taux de reconnaissance a connu une augmentation qui peut être également augmenté en entraînant le système sur une grande base de données. Parmi les modèles utilisés, YOLO a marqué un taux de reconnaissance très élevé et les résultats de la fusion de MTCNN avec Ultra-light a augmenté le taux de reconnaissance de 3 % qui est considérable et proche aux résultats de YOLO mais le taux de

fausse acceptation doit diminuer pour que nous pouvons prendre cette amélioration en considération. Le graphe présenté dans la figure 4-3 montre l'augmentation du taux de reconnaissance par rapport aux données d'apprentissage pour chacun des trois modèles.

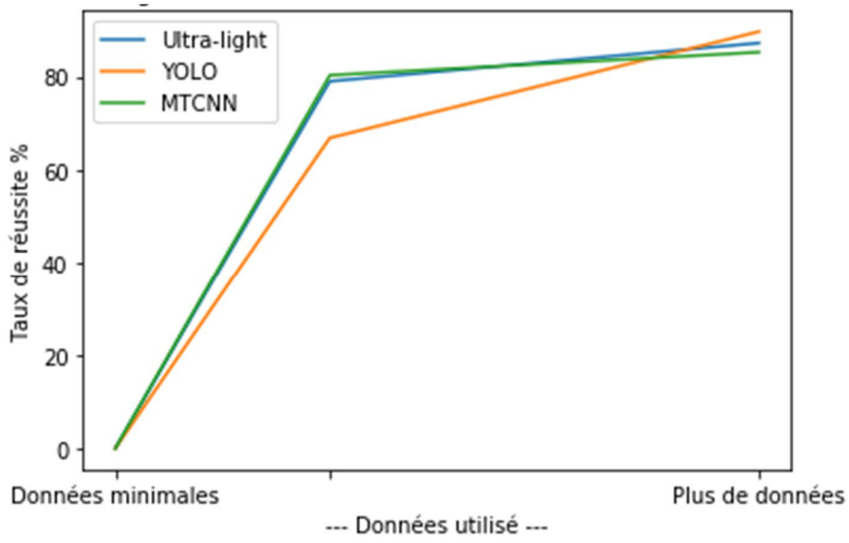


Figure 4-3: Augmentation du taux de reconnaissance par rapport aux données d'apprentissage.

4.6 L'interface de l'application

L'application est composée de 4 interfaces. La première interface est une page d'accueil (Figure 4-4) contenant des informations concernant le projet ainsi qu'un bouton pour accéder à la page suivante.

Figure 4-4: Page d'accueil de l'application.

L'interface suivante est considérée comme la page principale de l'application (Figure 4-5). Elle est composée de trois boutons (Apprentissage, Reconnaissance et Recherche) dont les boutons reconnaissance et recherche mène à d'autres interfaces tandis que le bouton apprentissage lance l'entraînement de nouvelles données.

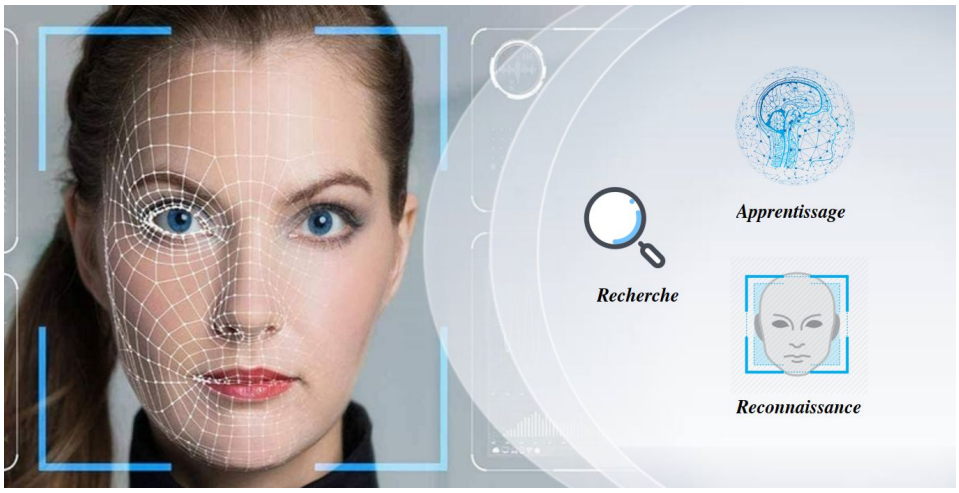


Figure 4-5: La page principale de l'application.

Pour l'entraînement on aura deux choix, soit on fait un entraînement de ≤ 5 personnes et ça sera effectué directement sur l'appareille (PC) sinon l'entraînement sera lancé sur la plateforme Colab car nos appareilles ne sont pas puissant pour effectuer un apprentissage qui demande beaucoup d'espace mémoire. La figure 4-6 illustre le choix d'entraînement et la figure 4-7 montre un message qui s'affiche quand l'apprentissage se termine.

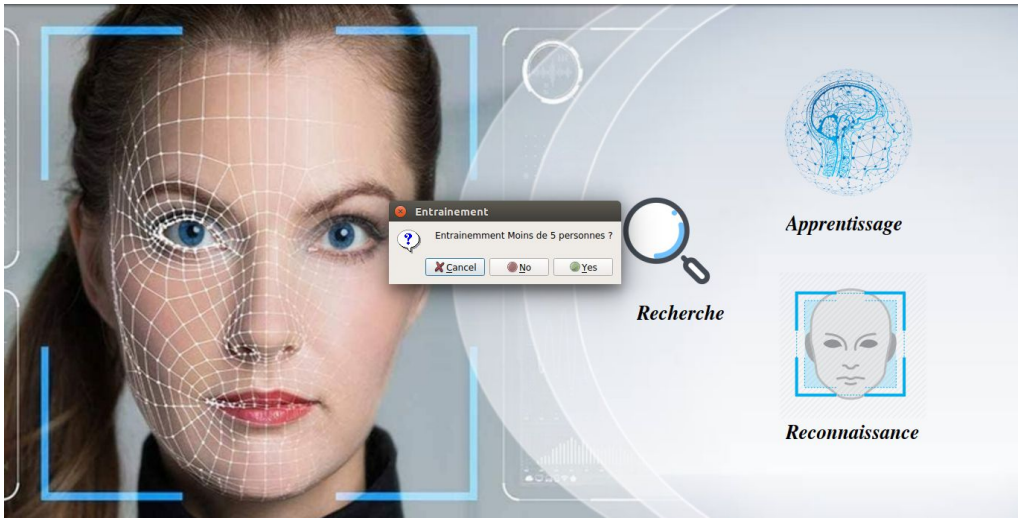


Figure 4-6: Le choix de l'apprentissage.

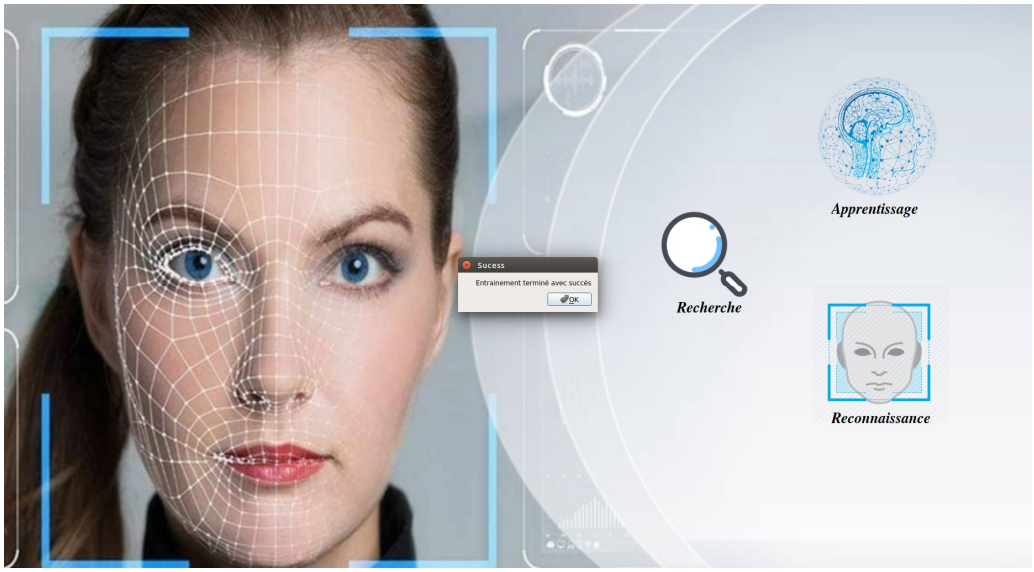
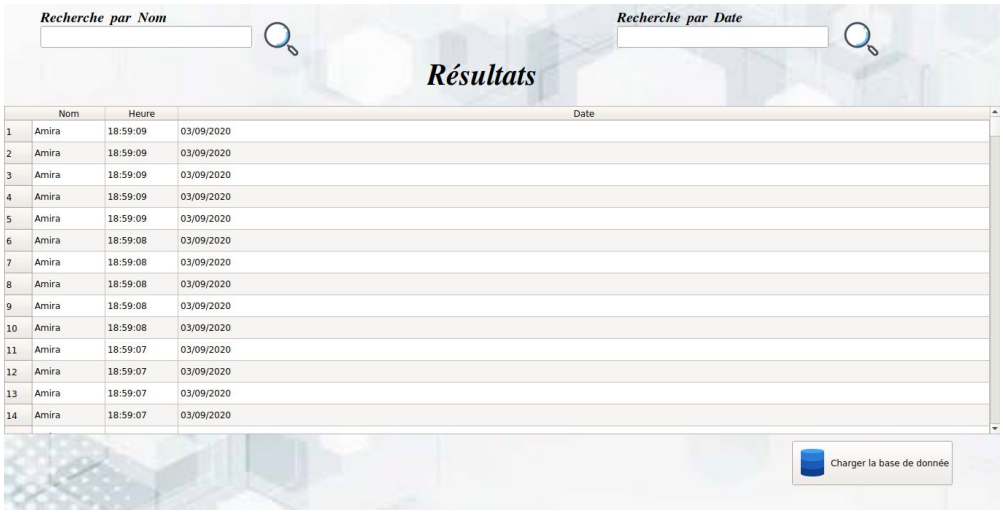


Figure 4-7: Entraînement terminé.

Le bouton « Reconnaissance » mène à l'interface responsable pour faire la reconnaissance. Cette reconnaissance est faite en temps réel. Elle est composée d'un espace réservé pour l'affichage de la caméra et quatre boutons pour les trois modèles utilisés ainsi que la fusion mentionné dans le chapitre 3. La figure 4-8 montre l'interface et un exemple d'une reconnaissance.

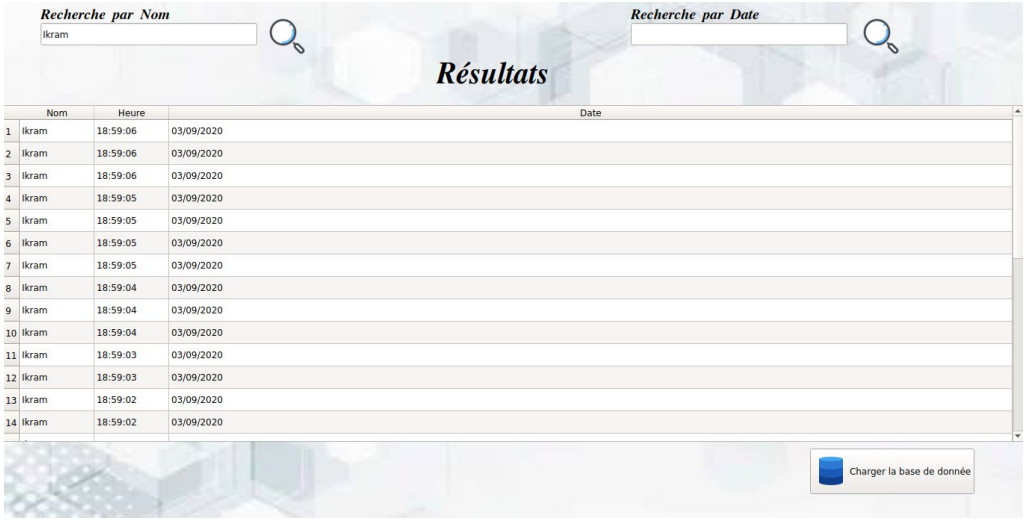
Figure 4-8: Interface de la reconnaissance.

Le bouton « Recherche » mène à la dernière interface qui est l'interface qui nous permet de garder trace de toutes les reconnaissances faite (Figure 4-9) et d'effectué une recherche par nom de la personne reconnue ou par la date (Figure 4-10 et Figure 4-11).



	Nom	Heure	Date
1	Amira	18:59:09	03/09/2020
2	Amira	18:59:09	03/09/2020
3	Amira	18:59:09	03/09/2020
4	Amira	18:59:09	03/09/2020
5	Amira	18:59:09	03/09/2020
6	Amira	18:59:08	03/09/2020
7	Amira	18:59:08	03/09/2020
8	Amira	18:59:08	03/09/2020
9	Amira	18:59:08	03/09/2020
10	Amira	18:59:08	03/09/2020
11	Amira	18:59:07	03/09/2020
12	Amira	18:59:07	03/09/2020
13	Amira	18:59:07	03/09/2020
14	Amira	18:59:07	03/09/2020

Figure 4-9: La quatrième interface avec la base de données chargée.



	Nom	Heure	Date
1	Ikram	18:59:06	03/09/2020
2	Ikram	18:59:06	03/09/2020
3	Ikram	18:59:06	03/09/2020
4	Ikram	18:59:05	03/09/2020
5	Ikram	18:59:05	03/09/2020
6	Ikram	18:59:05	03/09/2020
7	Ikram	18:59:05	03/09/2020
8	Ikram	18:59:04	03/09/2020
9	Ikram	18:59:04	03/09/2020
10	Ikram	18:59:04	03/09/2020
11	Ikram	18:59:03	03/09/2020
12	Ikram	18:59:03	03/09/2020
13	Ikram	18:59:02	03/09/2020
14	Ikram	18:59:02	03/09/2020

Figure 4-10: Recherche par nom de la personne.

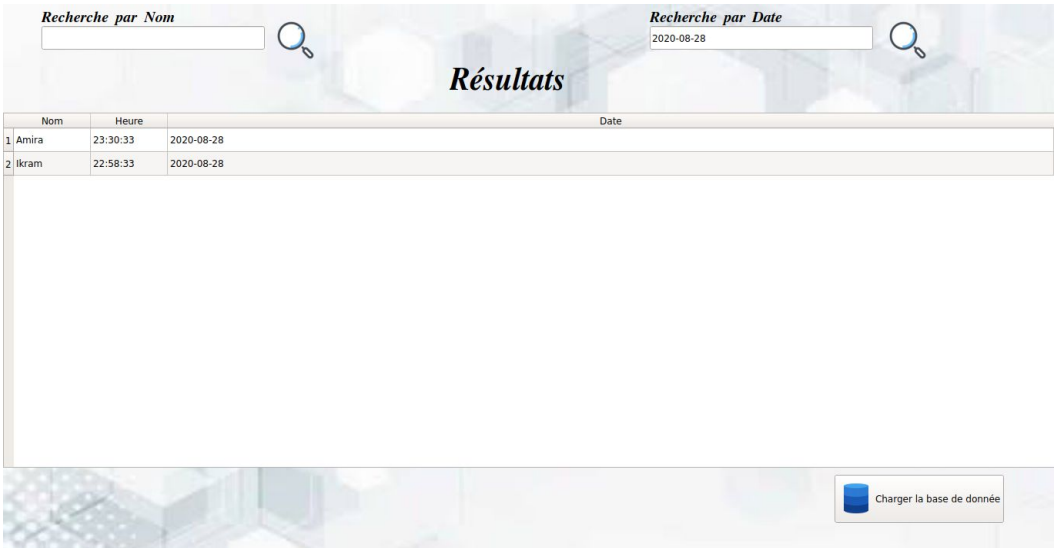


Figure 4-11: Recherche par date.

4.7 Conclusion

Dans ce chapitre nous avons commencé par introduire la configuration de nos appareilles ainsi que les différentes outils utilisés dans ce projet passant à la présentation des interfaces de l'application réalisé et ses différente fonctionnalités, en suite nous avons présenté les tests appliqué lors de la phase de l'implémentation ainsi que leur résultats en résumant le tout dans des tableaux et enfin nous avons terminé par faire une petite comparaison d'où nous avons conclus que les taux de reconnaissance ont augmentés dont YOLO a marqué un taux très élevé par rapport aux autres modèles.

Conclusion générale

L'objectif de ce mémoire est d'implémenter une application de détection, de reconnaissance et d'authentification faciale en temps réel qui répond à des besoins de sécurité et contrôle d'accès d'une société quelconque. L'application en question doit répondre à des exigences de rapidité et de robustesse des résultats.

L'application implémentée est basée sur trois modèles de détection MTCNN, YOLOv3, Ultra-light et une fusion. Les raisons de ce choix de modèles, d'une part, les modèles exploités dans notre application sont imposés par les encadreurs, et d'autre part, pour voir les différences ainsi que les avantages et les inconvénients de chacun d'eux

L'application de détection d'objets en temps réel développée répond aux objectifs du projet et ses performances en termes de taux de reconnaissance sont satisfaisantes mais reste à améliorer.

Ce mémoire de Master 2 nous a permet de:

- Approfondir nos connaissances théoriques et pratiques déjà acquises , maîtriser les nouvelles techniques et compléter notre initiale pour atteindre ainsi un niveau de perfection supérieur et de pouvoir apprendre d'autres nouveautés dans les différents domaines de la science en générale et de l'informatique en particulier.
- Construire des savoirs et des savoir-faire dans le domaine de traitement d'images et de vidéo particulièrement, la détection, la reconnaissance.
- Découvrir une nouvelle technologie de l'intelligence artificielle qui est l'apprentissage profond. Cette dernière utilise des outils statistiques pour découvrir des corrélations et établir des modèles dans des données.
- Comprendre les différentes étapes de l'apprentissage profond et afin de créer des algorithmes de deep-learning en Python.

Un prolongement de ce travail pourrait s'axer sur plusieurs points à savoir:

- Augmenter la précision en entraînant le modèle sur une grande base de données.
- Améliorer la reconnaissance de loin pour le modèle ultra-light pour bénéficier de sa légèreté.
- Utiliser un GPU de haute performance pour pouvoir entraîner une grande base de données et réduire la durée d'entraînement de réseaux de Deep Learning.
- Augmenter la précision de la fusion en essayant d'autres méthodes comme le Z-score et Tanh à fin de résolue le problème du FAR.
- Ajouter un aspect sécuritaire au système pour crypté les vecteurs caractéristiques à fin d'interdire l'accès à la base de données (ajout et/ou modification).

Bibliographie

[1] B.Ibtissam « Etude et mise au point d'un procédé biométrique multimodale pour la reconnaissance des individus » THÈSE En vue de l'obtention du Diplôme de Doctorat, Université des Sciences et de la Technologie d'Oran Mohamed Boudiaf) -2016.

[2] Y.Kabbara « Caractérisation des images à Rayon-X de la main par des modèles mathématiques : application à la biométrie »Thèse de doctorat université paris-est université libanaise -2016.

[3] F. Dib « Identification des personnes par le réseau veineux de la main », Université Ferhat Abbas –Setif1- 2013.

[4] D. Douaidi, S. Grini « Identification et Reconnaissance Biométrique par L'utilisation des Empreintes Palmaires », Université Akli Mohand Oulhadj de Bouira, 2017.

[5] A. Berredjem « La reconnaissance des individus par leur empreinte des articulations des doigts », Université 8Mai 1945 – Guelma, 2019.

[6] B.Abdessettar et S.Fathi « Extraction des caractéristiquespour l'analyse biométrique d'un visage » Mémoire de Master, Université Kasdi MerbahOuargla -2014.

[7] A.Amel « Reconnaissance Bimodale de Visages par Fusion de Caractéristiques Visuelles et de Profondeur » Thèse de doctorat, Université Lille A -2014.

[8] G.BENCHERKI, B.MOUSTAFA « Implémentation d'un système de reconnaissance de visages à base de PCA » Université Djilali Bounaama Khemis Miliana -2018.

[9] D.Boukhlouf « Résolution de problèmes par écosystèmes : Application au traitement d'images.» Université de Biskra-2005.

[10] P.Buyssens « Fusion de différents modes de capture pour la reconnaissance du visage appliquée aux transactions » Université de Caen-2011.

[11] M.Chihaoui, A.Elkefi, W.Bellil and C.Ben Amar « A Survey of 2D Face Recognition Techniques » University of Sfax, National School of Engineers (ENIS) -2016.

[12] A. Matthew, P.Alex « Face recognition using eigenfaces. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition », Maui, HI, USA, 3–6 June 1991.

[13] H.Hoffmann « Kernel PCA for novelty detection » Pattern Recognit-2007.

[14] N.Vladimir « The Nature of Statistical Learning Theory » New York, NY, USA-1995.

[15] F.Bach, M.Jordan « Kernel independent component analysis » Learn. Res-2002.

[16] Y.Hu « Learning a locality preserving subspace for visual recognition ». In Proceedings of the 9th IEEE International Conference on Computer Vision, Nice, France, 13–16 October-2003.

[17] D.Huang, C.Shan, M.Ardabilian Y.Wang, L.Chen « Local binary patterns and its application to facial image analysis » - 2011.

[18] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol, "Face recognition using HOG–EBGM," Pattern Recognition Letter - 2008.

[19] O. Déniz, G. Bueno, J. Salido, and F. De la Torre « Face recognition using Histograms of Oriented Gradients » Pattern Recognition Letters - 2011.

[20] N. Dalal and B. Triggs « Histograms of Oriented Gradients for Human Detection in Computer Vision and Pattern Recognition » - 2005.

[21] P. S Penev and J. J Atick « Local feature analysis : a general statistical theory for object representation. Network : Computation in Neural Systems » - August 1996.

[22] M.Chihaoui, W.Bellil, A.Elkefi, C.B.Amar « Face recognition using HMM-LBP. In Hybrid Intelligent Systems » Springer: Cham, Switzerland - 2015.

[23] I. Chaibeddra et S. Madene « Détection visuelle d'objets statiques et dynamiques dans un environnement de type route et classification en exploitant l'apprentissage profond » Université des Sciences et de la Technologie Houari Boumediene - 2019.

[24] L. Nehemy « Estimation de modèles autorégressifs vectoriels à noyaux à valeur opérateur», Docteur de l'Université d'Évry Val d'Essonne - 2015.

[25] A. Freedman. «Statistical models: Theory and practice» Cambridge University Press - 2012.

[26] J.Dean «Big Data, Data Mining, and Machine Learning: Value Creation for Business Leaders and Practitioners » - 2014: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118691786.ch6.

[27] C.Touzet. « Les réseaux de neurones artificiels, introduction au connexionnisme » Collection de l'EERIE – 1992.

[28] F. Rosenblatt « The perceptron: a probabilistic model for information storage and organization in the brain » -1988.

[29] G.Saporito « What is a perceptron? » - 2019 : https://towardsdatascience.com/what-is-a-perceptron-210a50190c3b.

[30] P. Habermehl et D. Kesner « Réseaux de neurones », cours programmation logique et IA-2016

[31] D. Moualek « Deep Learning pour la classification des Images », Master en Informatique, Université Abou Bakr Belkaid -2017.

[32] V. Iuhaniwal « Forward propagation in neural networks — Simplified math and code version » - 2019 : https://towardsdatascience.com/forward-propagation-in-neural-networks-simplified-math-and-code-version-bbcfef6f9250.

[33]V.Zocca, G.Spacagna, D.Slater, P.Roelants « Python Deep Learning » - 2017.

[34] N.Buduma « Fundamentals of Deep Learning Designing Next-Generation Machine Intelligence Algorithms » - 2017.

[35] S. Polamuri. « Difference Between Softmax Function and Sigmoid Function » -2017: https://dataaspirant.com/2017/03/07/difference-between-softmax-function-and-sigmoid-function/.

[36] L.Deng, D.Yu « Deep learning: methods and applications » - 2014.

[37] D.Moualek « Deep Learning pour la classification des images » Thèse de doctorat, Université Abou Bakr Belkaid– Tlemcen - 2017.

[38] M. Gregory Gelly. « Réseaux de neurones récurrents pour le traitement automatique de la parole », Thèse de doctorat de l'Université Paris-Saclay préparée à l'Université Paris-Sud - 2017.

[39] Y.Andrew, M.Jordan « On discriminative vs. generative

classifiers: A comparison of logistic regression and naive bayes. In Advances in neural information processing systems » - 2002.

[40] G.Hinton « Boltzmann Machines » University of Toronto, Toronto, ON, Canada - 2014.

[41] H. Ackley, E. Hinton, J. Sejnowski, « A learning algorithm for boltzmann machines » Cognitive science - 1985.

[42] G.Hinton « A Practical Guide to Training Restricted Boltzmann Machines » - 2010.

[43] R. Salakhutdinov, A. Mnih, G. Hinton « Restricted boltzmann machines for collaborative filte-ring » - 2007.

[44] R.Khandelwal. « Deep learning –deep belief network » -2018: https://medium.com/datadriveninvestor/deep-learning-deep-belief-network- dbn-ab715b5b8afc.

[45] G. Hinton, S. Osindero « A fast learning algorithm for deep belief nets » Neuralcomputation - 2006.

[46] G.Hinton, R.Salakhutdinov. « Deep Boltzmann Machines» in Artificial Intelligence and Statistics - 2009.

[47] N. Srivastava, R. Salakhutdinov « Multimodal learning with deep boltzmann machines » in Advances in neural information processing systems - 2012.

[48] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S.Ozair, A. Courville, Y. Bengio « Generative adversarial nets » in Advances in neural information processing systems - 2014.

[49] L. Denton, S. Chintala, R. Fergus, al « Deep generative image models using a laplacian pyramid of adversarial networks » in Advances in neural information processing systems - 2015.

[50] A. Radford, L. Metz, and S. Chintala, « Unsupervised representation learning with deep convolutional generative adversarial networks » - 2015.

[51] Y. Bengio, E. Laufer, G. Alain, and J. Yosinski « Deep generative

stochastic networks trainable by backprop » in International Conference on Machine Learning - 2014.

[52] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey « Adversarial autoencoders » -2015.

[53] S. Albawi, T. Mohammed and S. Al-Zawi « Understanding of a convolutional neural networ » - 2017.

[54] P.Monasse, K.Nadjahi « Découvrez les différentes couches d'un CNN» -2020 : https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5083336-decouvrez-les-differentes-couches-dun-cnn.

[55] Prabhu « Understanding of Convolutional Neural Network (CNN) Deep Learning » - 2018 : https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148.

[56] R.Lambert « Réseau de neurones convolutifs» -2019 : https://www.aspexit.com/reseau-de-neurones-on-va-eesayer-de-demystifier-un-peu-tout-ca-3/.

[57]D.Nene, A.Dian. « La reconnaissance des expressions faciales» Université 8 Mai 1945 – Guelma – 2019.

[58] S. Das « CNN Architectures : LeNet, AlexNet, VGG, GoogLeNet,ResNet and more... » - 2017 : https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5.

[59] Y.LeCun, L.Bottou, Y.Bengio, P.Haffner « LeNet-5 – A Classic CNNArchitecture » - 1990 : http://engmrk.com/lenet5-a-classic-cnn-architecture/

[60] N.Foued « Reconnaissance d'expression faciale à partir d'un visage réel » Université de 8 Mai 1945 – Guelma - 2019.

[61] A.Krizhevsky, G.Hinton, I.Sutskever «Implémentation AlexNet à l'aide de Keras » - 2012: https://engmrk.com/alexnet-implementation-using-keras/

[62] K.Simonyan, A.Zisserman « VGG16 – Implementation Using Keras » - 2018 : https://engmrk.com/vgg16-implementation-using-keras/

[63] O.Ezratty « Les usages de l'intelligence artificielle » novembre, 2018.

[64] V.Mühler « Realtime JavaScript Face Tracking and Face Recognition using face-api.js' MTCNN Face Detector » - 2018 : https://itnext.io/realtime-javascript-face-tracking-and-face-recognition-using-face-api-js-mtcnn-face-detector-d924dd8b5740.

[65] C.F.Wang « How Does A Face Detection Program Work? (Using Neural Networks) » - 2018 : https://towardsdatascience.com/how-does-a-face-detection-program-work-using-neural-networks-17896df8e6ff.

[66] S.Goswami « Reflections on Non Maximum Suppression(NMS) » - 2020 : https://medium.com/@whatdhack/reflections-on-non-maximum-suppression-nms-d2fce148ef0a.

[67] M.Menegaz « understanding yolo » - 2020: https://hackernoon.com/understanding-yolo-f5a74bbc7967.

[68] G. Nishad. « you only look once: implementing yolo in less than 30 lines of pyhthon code » -2019 : https://towardsdatascience.com/you-only-look-once-yolo-implementing-yolo-in-less-than-30-lines-of-python-code-97fb9835bfd2

[69] R.Joseph chet " yolo: real time object detection" – 2016 : https://pjreddie.com/darknet/yolo/.

[70] J.Hui « Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3 » - 2018.

[71] J.Redmon, A.Farhadi « YOLOv3: An Incremental Improvement » University of Washington - 2018.

[72] A.Kathuria « what's new in yolo v3 ? » - 2018 : https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b.

[73] Linzaer « Ultra-Light-Fast-Generic-Face-Detector-1MB » - 2019 : https://github.com/Linzaer/Ultra-Light-Fast-Generic-Face-Detector-1MB

[74] Y.Yuan et M.Sarazen « Smaller Is Better: Lightweight Face Detection For Smartphones » - 2019 : https://syncedreview.com/2019/11/01/smaller-is-better-lightweight-face-detection-for-smartphones/.

[75] « Smaller Is Better: Lightweight Face Detection For Smartphones », novembre, 2019: https://syncedreview.com/2019/11/01/smaller-is-better-lightweight-face-detection-for-smartphones/ -

[76] S.Liu, D.Huang, Y.Wang « Receptive Field Block Net for Accurate and Fast Object Detection » Beihang University, Beijing 100191, China - 2018.

[77] W.Liu, D.Anguelov, D.Erhan, C.Szegedy, S.Reed, C-Y.Fu, Alexander C. Berg

«SSD: Single Shot MultiBox Detector » University of Michigan -2016.

[78] Florent « Détecter les repères faciaux (facial landmarks) avec Dlib » - 2017 : https://pymotion.com/detecter-facial-landmarks/

[79] W.Ben Soltana, D.Huang, M.Ardabilian, Liming C.LIRIS « Laboratoire d'InfoRmatique en Image et Systèmes d'information » Ecole Centrale Lyon 36 Avenue Guy de Collongue, 69134, Ecully, France - 2005.

[80] A. Mian, M. Bennamoun,R. Owens « Face recognition using 2d and 3d multimodal local features » - 2006.

[81] F.Schroff, D. Kalenichenko, J.Philbin «FaceNet: A Unified Embedding for Face Recognition and Clustering» - 2015.

[82] https://fr.wikipedia.org/wiki/PyQt

[83] https://en.wikipedia.org/wiki/MySQL

[84] https://fr.wikipedia.org/wiki/TensorFlow

[85] Microsoft Ignite « Tutorial : Detect objects using ONNX in ML.NET » - 2020 : https://docs.microsoft.com/en-us/dotnet/machine-learning/tutorials/object-detection-onnx.

Annexe A

FaceNet

1. Architecture

L'architecture de FaceNet est composé de 22 couches en totale dont quelques couches de convolution avec un filtre de 1x1 sont ajoutées entre les couches de convoltions régulières de l'architecture. Le module inception (Figure 1) est adapté à l'architecture à fin de pouvoir l'utilisé dans des appareilles comme le mobile. Le tableau A-1 représente l'architecture du modèle.

Tableau A-1 : Architecture de FaceNet.

layer	size-in	size-out	kernel	param	FLPS
conv1	220×220×3	110×110×64	7×7×3, 2	9K	115M
pool1	110×110×64	55×55×64	3×3×64, 2	0	
rnorm1	55×55×64	55×55×64		0	
conv2a	55×55×64	55×55×64	1×1×64, 1	4K	13M
conv2	55×55×64	55×55×192	3×3×64, 1	111K	335M
rnorm2	55×55×192	55×55×192		0	
pool2	55×55×192	28×28×192	3×3×192, 2	0	
conv3a	28×28×192	28×28×192	1×1×192, 1	37K	29M
conv3	28×28×192	28×28×384	3×3×192, 1	664K	521M
pool3	28×28×384	14×14×384	3×3×384, 2	0	
conv4a	14×14×384	14×14×384	1×1×384, 1	148K	29M
conv4	14×14×384	14×14×256	3×3×384, 1	885K	173M
conv5a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv5	14×14×256	14×14×256	3×3×256, 1	590K	116M
conv6a	14×14×256	14×14×256	1×1×256, 1	66K	13M
conv6	14×14×256	14×14×256	3×3×256, 1	590K	116M
pool4	14×14×256	7×7×256	3×3×256, 2	0	
concat	7×7×256	7×7×256		0	
fc1	7×7×256	1×32×128	maxout p=2	103M	103M
fc2	1×32×128	1×32×128	maxout p=2	34M	34M
fc7128	1×32×128	1×1×128		524K	0.5M
L2	1×1×128	1×1×128		0	
total				140M	1.6B

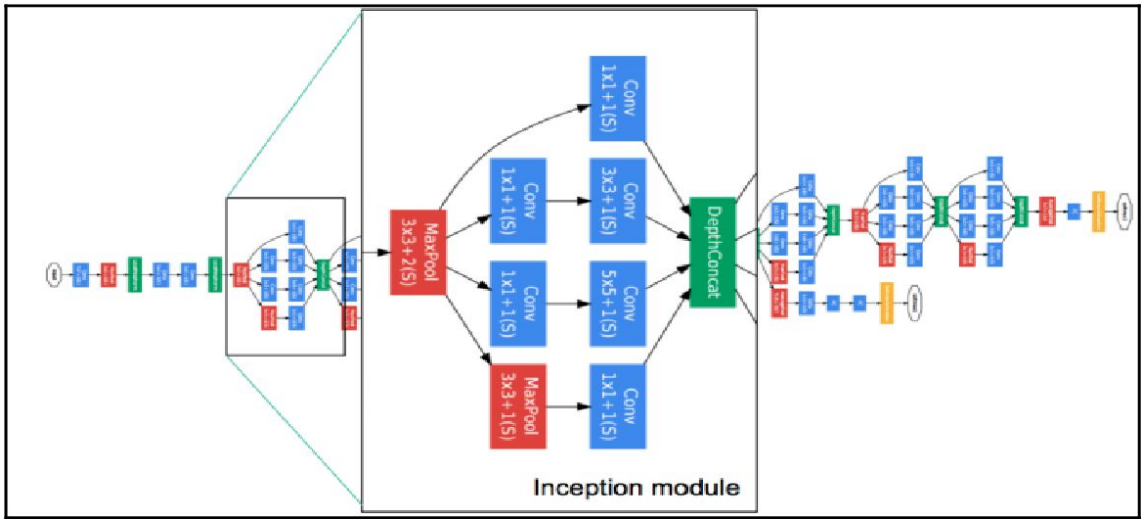


Figure A-1 : Module inception.