كلية العلوم و التقنيات فاس
ⵜⴰⵖⵚⵍⵓⵍⵜ ⵏ ⵜⵉⵡⵙⵙⵉⵏ ⴰ ⵜⵓⵙⵙⵉⵔⵜ
Faculté des Sciences et Techniques de Fès

جامعة سيدي محمد بن عبد الله
ⵜⴰⵡⴰⵍⵜ ⵓⵙⴰ ⵉⵙⵎⵓⵃⴰⵏ ⵓ ⵄⴰⴱⴷⵍⵍⴰ
Université Sidi Mohamed Ben Abdellah

DÉPARTEMENT D'INFORMATIQUE

PROJET DE FIN D'ÉTUDES

MASTER SCIENCES ET TECHNIQUES

SYSTÈMES INTELLIGENTS & RÉSEAUX

SYSTÈME DE RECONNAISSANCE FACIALE

التحول الذكي
Smart Transformation

LIEU DU STAGE : SMART TRANSFORMATION

Réalisé par :

- OUBAHA AMINA
- SABRI ZINEB

Encadré par :

- Pr. NAJAH SAID
- Pr. ZENKOUAR KHALID
- Mme. EL MANIOUI NOUHAILA

Soutenu le 14.07.2021 devant le jury composé de :

- Pr. Ab. Ben Abbou	Faculté des Sciences et Techniques de Fès	(Président)
- Pr. K. Abbad	Faculté des Sciences et Techniques de Fès	(Examinateur)
- Pr. S. Najah	Faculté des Sciences et Techniques de Fès	(Encadrant)
- Pr. K. Zenkouar	Faculté des Sciences et Techniques de Fès	(Encadrant)

Année Universitaire 2020 – 2021

REMERCIEMENT

Le travail présenté dans ce mémoire a été effectué dans le cadre de la préparation du diplôme du Master Sciences et Techniques intitulé Systèmes Intelligents et Réseaux à la Faculté des Sciences et Techniques de Fès.

Au terme de ce projet, nous tenons à remercier les personnes qui nous ont permis de mener à bien notre projet de fin d'étude, ceux dont l'intervention a aidé et a favorisé son aboutissement.

Une pensée particulière est adressée tout spécialement à nos parents, nos frères, sœurs et amis respectifs qui nous ont toujours encouragés et soutenus lors de nos choix.

A cette occasion, nous exprimons notre profonde gratitude et notre immense respect à Mr NAJAH Said et Mr ZENKOUAR Khalid nos encadrants à la faculté des sciences et techniques pour tous leurs conseils, leurs avis éclairés et leur précieux soutien.

Nous tenons également à remercier Mr. EL KANDRI Mohamed le Directeur Général de la société SMART Transformation pour l'opportunité de passer notre stage dans sa société, et Mme EL MANIOUI Nouhaila notre encadrante pour son suivi, sa sympathie, sa disponibilité tout au long de notre stage de fin d'études, et surtout sa patience.

Nous voudrions remercier tout le personnel de SMART Transformation pour leur soutien et leur bienveillance.

Nous voudrions également adresser toute notre gratitude à Mr ZAHI Azzedine chef du département pour le dynamisme de ce département d'études, à Madame BEGDOURI Ahlam responsable du master pour ses conseils, à tous les enseignants qui nous ont accompagnés durant ces années, pour leur gentillesse, leur efficacité, et pour la qualité de l'enseignement qui nous a été dispensée.

Un remerciement particulier aux membres du jury Pr Ab. Ben Abbou, Pr. K. Abbad pour avoir accepté de juger ce travail et de nous faire profiter de leurs remarques et conseils.

RESUME

Eviter la redondance d'images, éviter qu'une personne s'inscrive dans l'application avec l'image d'une autre; ce que l'on peut qualifier d'usurpation de l'identité et s'assurer qu'en cas de modification, la nouvelle photo entrée par un utilisateur lui appartient sont les tâches ou les conditions que nous avons dû respecter lors de notre période de stage, et que nous présentons dans ce rapport. Nous avons implémenté et entrainé différentes méthodes et architectures de la reconnaissance faciale pour pouvoir arriver à une meilleure précision compte tenu de la difficulté principale de cette tâche, qui est en même temps la richesse et la pauvreté au niveau des données. Une situation qui a nécessité l'utilisation de quelques techniques pour arriver à un meilleur compromis.

Mots clés : Reconnaissance faciale – détection de visage – apprentissage profond – réseau neuronal convolutif - apprentissage par transfert.

ABSTRACT

Avoiding image redundancy, avoiding that a person registers in the application with the image of another; what can be qualified as identity theft and ensuring that in case of modification, the new photo entered by a user belongs to him are the tasks or conditions that we had to respect during our internship period, and that we present in this report. We have implemented and trained different methods and architectures of facial recognition in order to achieve a better accuracy considering the main difficulty of this task, which is at the same time the richness and poverty of the data. A situation that required the use of some techniques to achieve a better compromise.

Keywords: Face recognition - face detection - deep learning - convolutional neural network - transfer learning.

TABLE DES MATIERES

REMERCIEMENT i
RESUME ii
ABSTRACT iii
TABLE DES MATIERES iv
LISTE DES FIGURES vi
LISTE DES TABLEAUX ix
LISTE DES FIGURES x
INTRODUCTION GENERALE 1
Chapitre 1 : CONTEXTE GENERAL DU PROJET 3
1. Introduction 3
2. Présentation de l'organisme d'accueil 3
2.1. Présentation de l'entreprise 3
2.2. Organigramme 4
3. Contexte général 4
4. Conclusion 5
Chapitre 2 : ETAT DE L'ART 6
1. Introduction 6
2. Définition de la reconnaissance facial 6
3. Historique 6
4. Etapes de La reconnaissance Faciale 8
4.1. Détection de visage 8
4.2. Extraction Des Caractéristiques 16
4.2.1. Définition 16
4.2.2. L'approche Peu Profonde 16
L'approche Profonde 19
5. Correspondance des visages 33
5.1. Distance euclidienne 33
5.2. Similitude en cosinus 33

6. Conclusion 33
Chapitre 3 : Le SYSTEME DE RECONNAISSANCE FACIALE PROPOSE 34
1. Introduction 34
2. Les outils d'environnement 34
2.1. Python 34
2.2. Numpy 34
2.3. Matplotlib 34
2.4. Opencv 34
2.5. TensorFlow 35
2.6. Keras 35
2.7. Dlib 35
2.8. Streamlit 36
2.9. Angular 36
3. Reconnaissance faciale 36
3.1. Notions utilisées 36
3.2. La Détection de visage 38
3.2.1. Comparaison entre les détecteurs de visage sur plusieurs niveaux 38
3.2.2. Discussion des résultats 44
3.3. L'Extraction Des Caractéristiques 45
3.3.1. Comparaison entre les modèles de reconnaissance faciale 45
3.3.2. Discussion des résultats 52
3.4. Approche Proposée 52
3.5. Application 53
4. Application « Doccerts » 55
4.1. Système de messagerie 56
4.2. Profil de l'utilisateur 57
5. Conclusion 62
CONCLUSION 63
REFERENCES 64

LISTE DES FIGURES

Figure 1: Logo d'entreprise 3

Figure 2: Organigramme de Smart Transformation Maroc....4

Figure 3: Evolution De La Reconnaissance Faciale [4] 7

Figure 4: Les 5 différents types de caractéristiques de type Haar extraites d'un patch d'image [19] 9

Figure 5: Introduites pour la première fois en 2001, les cascades de Haar sont une classe d'algorithmes de détection d'objets [20]. 10

Figure 6: examen d'un pixel et ceux qui l'entoure directement [52]. 11

Figure 7 : l'image s'assombrit vers le haut....11

Figure 8 : ensemble de flèches montrant le passage du clair au foncé de l'œil 11

Figure 9 : Image originale....12

Figure 10 : Transformée en une représentation HOG capturant les principales caractéristiques de l'image,....12

Figure 11 : Comparaison avec un modèle de visage généré de plusieurs images pour détecter le visage [52]. 13

Figure 12: Différentes étapes de MTCNN [24] 15

Figure 13 : Aplatir une image kxk en un seul vecteur caractéristique. [40] 16

Figure 14 : Matrice représentant l'ensemble de données du visage. [40] 17

Figure 15 : Matrice V contenant N lignes (nos vecteurs propres), chacune de dimension K^{2}. [40] 17

Figure 16 : À gauche : image moyenne de visage. À droite : les 16 meilleures représentations de visage. Les régions claires montrent une forte variation dans l'ensemble de données, tandis que les zones sombres suggèrent une faible variation. [40] 18

Figure 17 : Calcul de la distance euclidienne (d) entre les représentations de visages.[40].....19

Figure 18 : Deep Learning [25] 20

Figure 19: Deep Neural Network....21

Figure 20: Architecture d'une couche de CNN [27] 21

Figure 21 : Architecture d'un réseau de neurones convolutif basique. 22

Figure 22 : Allure de la fonction RELU [58]....24

Figure 23 : Exemple d'un volume d'entrée passant par une activation ReLU, max (0, x). Les activations sont effectuées sur place, il n'est donc pas nécessaire de créer un volume de sortie distinct, même s'il est facile de visualiser le flux du réseau de cette manière. [29].....25

Figure 24 : Schéma fonctionnel d'Alexnet. [50] 26

Figure 25: Architecture du réseau VGG 16 27

Figure 26: Module Inception [34] 28

Figure 27: GoogleNet avec toutes ses caractéristiques. [34] 29

Figure 28: Structure du modèle FaceNet. [32] 30

Figure 29: les 128 mesures générées par FaceNet pour un visage. [52] 30

Figure 30: Triplet loss. [52] 31

Figure 31 : La cellule ResNet. La connexion d'identité du côté droit permet à une version non modifiée de l'entrée de passer à travers la cellule. Cette modification permet l'entraînement efficace d'architectures convolutionnelles très profondes. [50] 32

Figure 32 : Apprentissage par Transfert.....36

Figure 33 : Exemple D'augmentation d'image.....37

Figure 34: Exemple d'alignement de visage 38

Figure 35 : Précision pour chaque détecteur de visage.....39

Figure 36 : Exemple de Détection de visage de petite taille (25*50).....39

Figure 37 : Exemple de Détection de visage de petite taille (55*60).....40

Figure 38 : Exemple de Détection de visage non frontal, regard vers la gauche.....41

Figure 39 : Exemple de Détection de visage non frontal, regard vers le bas 41

Figure 40 : Exemple de Détection de visage non frontal, regard vers la droite.....42

Figure 41: Détection de visage sous une occlusion.....43

Figure 42 : Détection de visage avec port du masque.....43

Figure 43 : Comparaison des différentes méthodes de détection en cas de détection de visage avec occlusion 43

Figure 44 : le temps de détection pour chaque détecteur de visage sur GPU 44

Figure 45 : le temps de détection pour chaque détecteur de visage sur CPU.....44

Figure 46 : Architecture CNN Utilisée 46

Figure 47 : Précision de CNN 47

Figure 48 : Architecture d'AlexNet 48
Figure 49 : Précision d'AlexNet 48
Figure 50: Architecture de VGG 16 49
Figure 51 49
Figure 52 : Représentation de Data Augmentation 50
Figure 53 : Générateur de données. 50
Figure 54 : Le modèle créé pour exécuter model.fil_generator 50
Figure 55 : Précision de VGG16 51
Figure 56 : images de notre base de données 51
Figure 57 : Schéma de l'approche proposée 52
Figure 58 : Les 68 points clés de visage 53
Figure 59 : structure de l'application 54
Figure 60: la détection de visage 54
Figure 61 : L'identification du visage existant 55
Figure 62 : L'identification du visage non existant 55
Figure 63 : Diagramme de cas d'utilisation du système de messagerie. 56
Figure 64 : Interface de système de messagerie. 57
Figure 65 : La boite d'envoi de message 57
Figure 66 : Diagramme de cas d'utilisation de gestion du profil 58
Figure 67 : Informations personnelles 59
Figure 68 : Paramètres du compte 60
Figure 69 : Changement de mot de passe. 60
Figure 70 : Interface de l'identité blockchain. 61
Figure 71 : Clé publique copiée. 61
Figure 72 : Profil du Verifier. 62

LISTE DES TABLEAUX

Tableau 1: Conclusion de notre étude comparative entre les détecteurs de visage.....45

Tableau 2 : Précision de FaceNet et VGGFACE 52

LISTE DES FIGURES

CNN	Convolutional Neural Network
VGG	Visual géométrie groupe
RESNET	Residual Neural Network
MTCNN	Multi-task Cascaded Neural Network
HOG	Histogram Of Gradient
DNN	Deep Neural Network
OpenCV	Open Source Computer Vision Library
GPU	Graphics Processing Unit
CPU	Central Processing Unit

INTRODUCTION GENERALE

La notion de l'extraordinaire se dissipe avec le pouvoir du possible, l'énorme capacité de la blockchaine et l'immense puissance de l'intelligence artificielle ne laissent plus de place à l'impossible, et fournissent de plus en plus d'émerveillement. Actuellement, nous vivons dans une ère de technologie par excellence, et avec l'épidémie par laquelle passe le monde entier et les restrictions sanitaires prises durant cette période à savoir les confinements, le télétravail et les études à distance, etc., le monde entier a été obligé de la suivre, ne laissant plus place à l'analphabétisme numérique et technologique qualifié d'illectronisme ou encore d'illettrisme numérique.

Depuis un moment la digitalisation a été adoptée dans les différents domaines de la vie que ce soit domestique, professionnelle ou estudiantine. Ces deux derniers aspects nous intéressent particulièrement, surtout au Maroc où certes nous nous détachons de la version papiers à grands pas, mais elle reste présente et par-dessus tout obligatoire en ce qui l'en est des diplômes et des certificats. Et là l'étudiant n'est pas le seul à être traumatisé par la quantité de la paperasse à gérer, mais aussi les établissements, les sociétés, etc. et c'est ici que « Doccerts », une application en cours de conception par la société Smart Transformation ,entre en jeu, avec la notion de la blockchain, qui en gros permet la sauvegarde éternelle du document en gardant trace de toute modification faite, mais surtout en garantissant une facilité d'accès pour l'étudiant en prime abord, un établissement susceptible à le recevoir et un recruteur potentiel. La blockchain est d'ailleurs parfois décrite comme la plus grande invention depuis internet ou comme une réelle révolution qui va changer notre mode de vie. Le célèbre journal anglais le Guardian n'a d'ailleurs pas hésité à publier un titre provocateur : « blockchain : the answer to life, the universe and everything ? »[1].

Les inventions révolutionnaires et les plus marquantes sont le fruit de l'intelligence humaine, en plus d'internet et de la blockchaine, l'intelligence artificielle se manifeste comme une humble reproduction de l'intelligence humaine, une reproduction présente depuis une décennie, certes qui n'est pas aussi intelligente que le cerveau humain, mais avec des capacités de simulation de cette dernière, la dépassant même entre autre en vitesse, en taux de données traitables et parfois en précisions des informations de sortie surtout quand le deep learning lui prête main-forte.

Notre travail dans cette application consiste en la création d'un système de reconnaissance faciale comme sorte de vérification de l'identité de la personne inscrite (étudiant,

administrateur, etc.). Essayant ainsi d'apporter des éléments de réponse à notre problématique qui consiste en l'usurpation de l'identité.

Comme beaucoup de technologies, la reconnaissance faciale est devenue disponible, applicable dans différents domaines, les plus sensibles et les beaucoup moins, en passant par la sécurité (investigations, recherche de fugitifs ou de personnes portées disparues, traque de terroristes, etc.), l'identification et le contrôle d'identité (dans les entreprises, ou pour déverrouiller un smartphone, etc.), la classification et le tague (dans les smartphones, les réseaux sociaux …).

Le but de l'intégration de cette technologie dans notre application n'est pas une question de sécurité nationale, c'est plutôt pour garantir une certaine crédibilité, et ceci non seulement dans le cadre national mais au niveau international.

Plusieurs applications et plateformes permettent l'accès gratuit ou payant à cette technologie au grand publique, à savoir Google Lens, Facebook, Amazone et autres. Quelques modèles et architectures sont également disponible pour les intéressés ; chercheurs, étudiants ou amateurs.

Ce présent rapport est scindé en quatre chapitres, le premier comportera une présentation de la société Smart Transformation, un aperçu sur la problématique et la solution proposée. Le deuxième se présentera sous forme d'un état de l'art de l'évolution de la reconnaissance faciale en se basant sur un corpus composé de travaux les plus imminent dans le domaine. Dans le troisième, nous traiterons différentes méthodes de la détection et de la reconnaissance faciale, en gardant la meilleure pour son implémentation dans l'application, nous projetterons la lumière également sur notre modeste contribution dans la partie front-end développée avec Angular. Le quatrième chapitre sera sous forme de présentation de l'environnement de travail, des données utilisées et des différentes méthodes et API qui nous ont facilités l'implémentation.

Chapitre 1 : CONTEXTE GENERAL DU PROJET

1. Introduction

Dans ce chapitre, nous allons donner une présentation générale de la structure d'accueil du stage, et nous allons parler du contexte général de notre projet.

2. Présentation de l'organisme d'accueil

2.1. Présentation de l'entreprise

Figure 1: Logo d'entreprise

Smart Transformation est une entreprise technologique de renommée, née de la passion des nouvelles technologies telles que Blockchain, AI et IoT.

Depuis la création de Smart Transformation l'objectif principale qui vise principalement à le réaliser est d'utiliser des technologies de pointe pour développer des solutions futuristes.

La mission de Smart Transformation concerne plusieurs axes tels que :

- Développement
- Applications décentralisées
- Développement de contrat intelligent
- Développement d'application web
- Déploiement d'application

Smart Transformation occupe une position mondiale en tant que partie intégrante de la Dhahran Techno Valley qui est un parc scientifique de premier plan regroupant des sociétés géantes telles qu'ARAMCO, SIPCHEM, Schlumberger et Honeywell, où elle collabore étroitement avec ses partenaires en Arabie saoudite pour favoriser la transformation numérique dans le secteur de L'industrie pétrolière et gazière.

Smart Transformation collaborent aussi étroitement avec l'Université Roi Fahd du pétrole et des minéraux - KFUPM (Qui occupe la Septième place mondiale en ce qui concerne le nombre de brevets délivrés par l'Office des brevets américain aux universités) afin de mettre

Au point un système de certification intelligent. Smart Transformation compte aussi plusieurs partenaires stratégiques au Canada.

Smart Transformation a également ouvert des bureaux au Maroc afin de renforcer les capacités locales, de participer à la transformation numérique en cours dans le pays et de faire le transfert de l'expertise internationale pour accélérer cette transformation.

2.2. Organigramme

L'organigramme illustré dans la figure 2 ci-dessous, représente la structure hiérarchique de Smart Transformation Maroc :

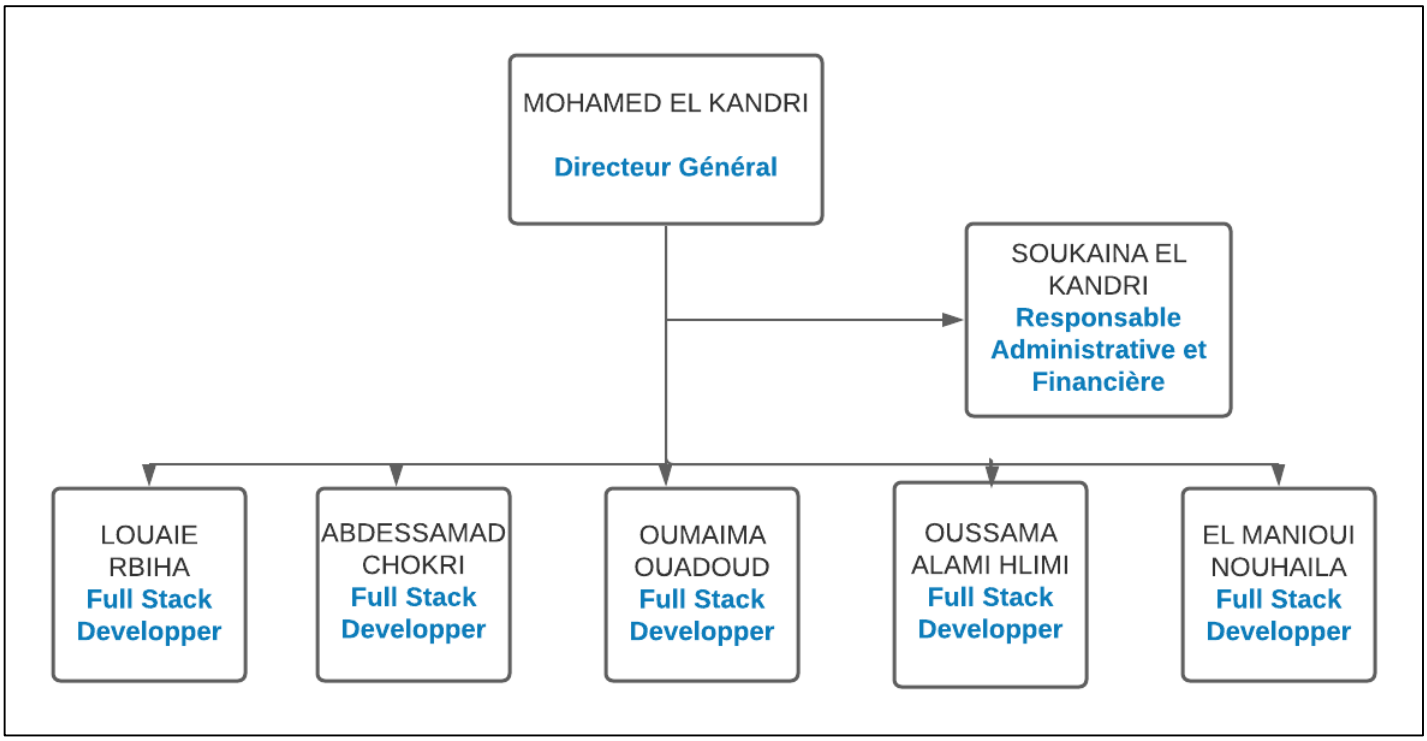

Figure 2: Organigramme de Smart Transformation Maroc

3. Contexte général

3.1. Problématique

Plusieurs personnes s'amusent à créer des comptes avec des images qui ne sont pas les leurs, que ce soit dès le départ ou après modifications de leurs photos de profils, que ce soit dans des environnements importants ou pas. Nous remarquons ce fléau dans les réseaux sociaux, il est tellement présent que ceci ne choque plus l'auditoire. Sauf que dans des environnements professionnels, ceci nuira à la personne et est susceptible de toucher à l'intégrité de la

plateforme en question, et de diminuer la crédibilité de la personne concerné et de la plateforme qui ne met pas en points des méthodes pour empêcher ce genre de nuisibilité. Il est important de préciser que les lois sanctionnent ce genre de comportement.

3.2. Solution

Pour essayer de remédier à ce type de problèmes, nous proposons l'accomplissement d'un système de reconnaissance et de vérification faciale qui permet d'abord de vérifier si l'image entrée par un utilisateur durant son inscription n'appartient pas à une personne déjà inscrite sur l'application, et puis en cas de modification de la photo, de s'assurer que la nouvelle image est en effet celle de la personne en question.

4. Conclusion

Ce chapitre a été consacré à d'abord la présentation de l'organisme de la société d'accueil, ensuite nous avons présenté une description de la problématique traité et la solution proposée.

Chapitre 2 : ETAT DE L'ART

1. Introduction

La recherche dans le domaine de la reconnaissance faciale ne date pas d'hier, depuis les années 60 la reconnaissance du visage a fait beaucoup parler d'elle et plusieurs chercheurs se sont mis à la tâche donnant naissance à plusieurs méthodes de plus en plus précises et performantes avec le temps et le développement technologique.

Dans ce chapitre, nous présentons un état de l'art de la reconnaissance faciale, nous nous intéressons à différentes méthodes d'abord de détections de visage puis de reconnaissance en passant par les différentes méthodes d'extractions des caractéristique.

2. Définition de la reconnaissance facial

La reconnaissance des visages est un processus qui consiste à analyser des visages humains dans des images et à identifier à qui ils appartiennent, ceci peut être fait à travers :

- Une vérification quand nous souhaitons savoir si un visage existe parmi ceux d'une base de données ;
- Une reconnaissance quand la base de données est sous formes d'images libellées et que nous voulons connaître le nom de la personne sur une image.

3. Historique

De nos jours la reconnaissance faciale peut sembler omniprésente, mais ce n'était pas le cas depuis toujours.

L'histoire de la reconnaissance des visages remonte aux années 1950 et 1960, mais on considère que les recherches sur la reconnaissance automatique des visages ont débuté dans les années 1970 quand Goldstein et al. Ont publié « Identification of human faces » étant une tentative grossière d'indentification de visages. Cette méthode proposait l'étude de 21 caractéristiques subjectives du visage, telles que la couleur des cheveux et l'épaisseur des lèvres, pour identifier un visage sur une photographie, sauf que ceci a été le principal inconvénient de cette méthode ; car en plus d'être subjectives, elles étaient calculées manuellement.

En 1987, Sirovich et Kirby ont publié leur ouvrage fondamental, « A Low-Dimensional Procedure for the Characterization of Human Faces »[2], suivi par Turk et Pentland en 1991 avec « Face Recognition Using Eigenfaces »[3], d'ailleurs les études de recherche sur la reconnaissance des visages se sont multipliées depuis le début des années 1990, suite aux développements du matériel et à l'importance croissante des applications liées à la sécurité.

Les progrès des techniques de reconnaissance des visages basées sur l'image grossièrement divisés en quatre phases principales de développement conceptuel par Wang et Deng [4], ce qui reflète le développement historique des principales méthodes :

i) Les approches holistiques ou basées sur l'apparence utilisent la région du visage dans son ensemble et utilisent des méthodes linéaires ou non linéaires pour représenter le visage dans un sous-espace de dimension inférieure [5]. L'une des premières méthodes réussies est Eigenfaces, développée par Turk et Pentland. Il existe d'autres approches qui utilisent des sous-espaces linéaires [6] [7] [8], l'apprentissage par les collecteurs [9] [10] [11] et les représentations éparses [12] [9] [13] [14].

ii) Les méthodes de reconnaissance des visages basées sur les caractéristiques locales sont devenues populaires après les années 2000 et utilisent des caractéristiques créées à la main pour décrire le visage, telles que les caractéristiques Ga-bor, les modèles binaires locaux (LBP) et leurs variantes [15] [16].

iii) Les méthodes qui utilisent des descripteurs locaux basés sur l'apprentissage sont apparues après les années 2010 et elles apprennent les filtres d'image discriminants en utilisant des techniques peu profondes.

iv) Les méthodes basées sur l'apprentissage profond ont gagné en popularité après le grand succès d'AlexNet dans la compétition ImageNet en 2012 [17], et ont apporté une nouvelle perspective au problème de la reconnaissance des visages. Une stabilité sans précédent a été atteinte pour la reconnaissance des visages, de sorte que leurs performances sont similaires à celles des humains sur des ensembles de données à grande échelle collectés dans des contextes non contraints [18].

L'apprentissage profond est désormais responsable d'une précision sans précédent dans la reconnaissance des visages. Des architectures spécialisées sont entraînées et ont atteint des précisions très proches voir meilleures que la performance humaine. Cette précision a atteint 97,35% puis a dépassé les 99,80% en seulement trois ans.

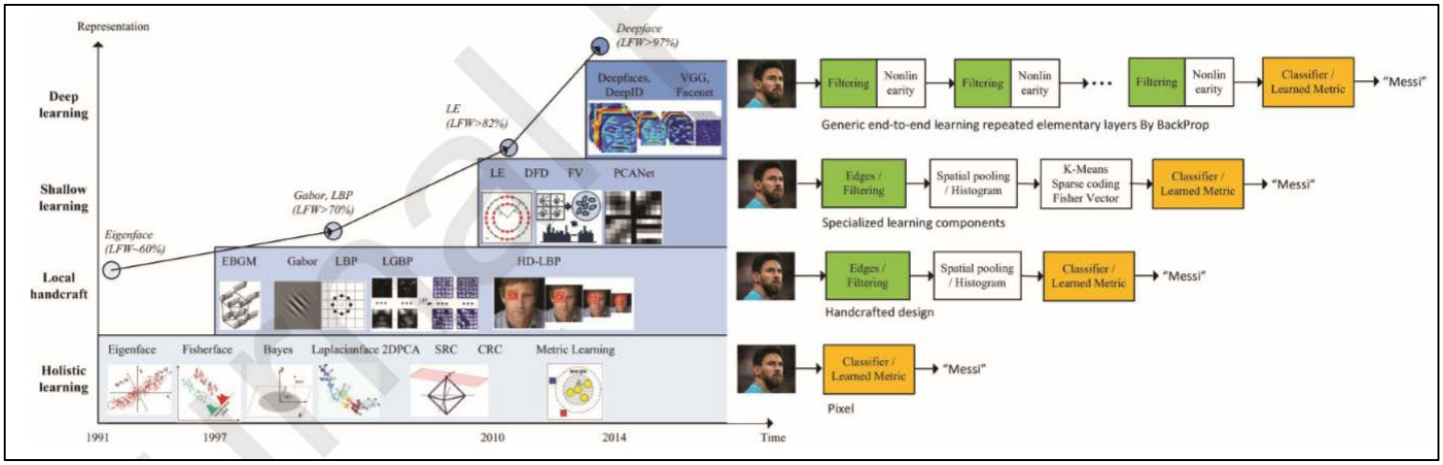

Figure 3: Evolution De La Reconnaissance Faciale [4]

4. Etapes de La reconnaissance Faciale

Un système de reconnaissance faciale se compose essentiellement des étapes suivantes :

- L'entrée de l'image
- La détection du visage
- L'extraction des caractéristiques
- La reconnaissance ou la vérification

4.1. Détection de visage

- Définition

La détection des visages consiste à estimer la boîte de délimitation du visage dans une image donnée. Le but est l'extraction du visage pour pouvoir mieux l'utiliser dans la reconnaissance. Afin de rendre le système de reconnaissance des visages plus robuste et plus facile à concevoir, quelques systèmes incluent un alignement des visages. La détection des visages doit faire face à plusieurs difficultés à savoir :

- La variation de la pose : La condition idéale pour la détection des visages que l'image soit frontale. Sauf que cela est très peu probable dans des conditions générales non contrôlées.
- L'occlusion d'éléments : La présence d'éléments comme des barbes, des lunettes ou des chapeaux introduit une forte variabilité. Les visages peuvent également être partiellement couverts par des objets ou d'autres visages.
- Les expressions faciales : Les traits du visage varient également beaucoup en raison des différents gestes du visage.
- Les conditions d'imagerie : Caméras différentes et conditions d'obscurité, puisque les conditions d'éclairage ainsi que le type de la caméra affectent la qualité d'une image, et affecte donc l'apparence d'un visage.

S'il y a plusieurs visages dans une image, ils doivent tous être détectés.

L'étape de la détection est très essentielle dans le procès de la reconnaissance faciale, et peut également avoir des applications très utiles. L'application la plus réussie de la détection des visages est probablement la prise de photos.

Plusieurs méthodes de détection ont pu être utilisés et tester, ci-dessous nous présentons quatre de ces méthodes :

- **Haar Cascade :** Il s'agit d'un algorithme de détection d'objets utilisé pour identifier des visages dans une image ou une vidéo en temps réel.

Publié pour la première fois par Paul Viola et Michael Jones dans leur article de 2001, « Rapid Object Detection using a Boosted Cascade of Simple Features », ce travail original est devenu l'un des articles les plus cités dans la littérature sur la vision par ordinateur.

Dans leur article, Viola et Jones proposent un algorithme capable de détecter des objets dans des images, quels que soient leur emplacement et leur échelle dans l'image. Cet algorithme peut également fonctionner en temps réel, ce qui permet de détecter des objets dans des flux vidéo. Viola et Jones se concentrent plus particulièrement sur la détection des visages dans les images. Néanmoins, le cadre peut être utilisé pour former des détecteurs d'objets arbitraires.

Le principe de la méthode est le glissement d'une fenêtre de taille fixe sur une image à plusieurs échelles, comme le montre la figure 2. À chacune de ces phases, la fenêtre s'arrête, calcule certaines caractéristiques, puis classe la région comme suit : Oui, cette région contient un visage, ou Non, cette région ne contient pas de visage.

Cela nécessite un peu d'apprentissage automatique. Nous avons besoin d'un classificateur entraîné à utiliser des échantillons positifs et négatifs d'un visage :

- Les points de données positifs sont des exemples de régions contenant un visage.
- Les points de données négatifs sont des exemples de régions qui ne contiennent pas de visage.

Avec ces points de données positifs et négatifs, nous pouvons "entraîner" un classificateur à reconnaître si une région donnée d'une image contient un visage.

Pour chacun des arrêts le long du chemin de la fenêtre glissante, cinq caractéristiques rectangulaires sont calculées :

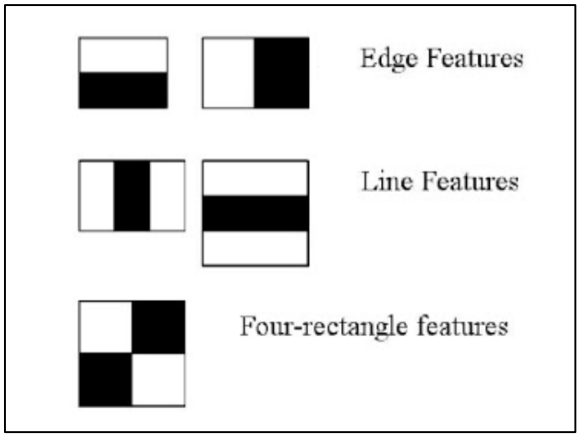

Figure 4: Les 5 différents types de caractéristiques de type Haar extraites d'un patch d'image [19]

L'obtention des caractéristiques se fait par la soustraction de la somme des pixels dans la région blanche et celle des pixels dans la région noire. Ces caractéristiques ont une réelle importance dans le contexte de la détection des visages puisqu'elles permettent de classer les parties d'un visage, à savoir :

- Les régions des yeux ont tendance à être plus sombres que celles des joues.
- La région du nez est plus claire que la région des yeux.

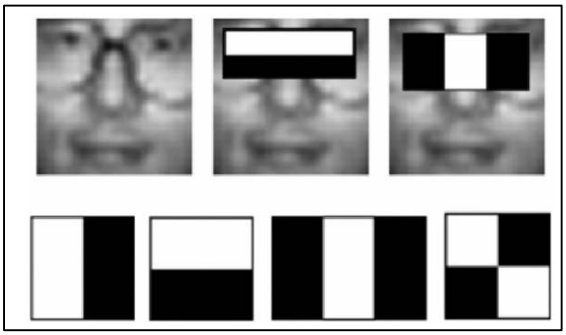

Figure 5: Introduites pour la première fois en 2001, les cascades de Haar sont une classe d'algorithmes de détection d'objets [20].

Ensuite, pour un ensemble complet de caractéristiques, nous utilisons l'algorithme AdaBoost pour sélectionner celles qui correspondent aux régions du visage d'une image. On obtient ainsi un classificateur composé de classificateurs faibles, puisqu'à eux seuls ils ne pourraient pas classer l'image, mais qui ensemble permettent une classification avec une grande précision. [20]

L'utilisation d'une fenêtre coulissante fixe et son glissement sur chaque coordonnée (x, y) d'une image, suivie du calcul de ces caractéristiques de type Haar, et enfin de la classification proprement dite, peut être coûteuse en termes de calcul, pour ce, Viola et Jones ont introduit le concept de cascades ou d'étapes. À chaque étape du parcours de la fenêtre glissante, la fenêtre doit passer une série de tests où chaque test suivant est plus coûteux en calcul que le précédent. Si l'un des tests échoue, la fenêtre est automatiquement rejetée et les caractéristiques y restantes ne sont pas considérées. Si elle réussit, elle passe à la deuxième étape de caractéristiques et continue le processus. La fenêtre qui passe toutes les étapes est une région de visage. [20]

Les avantages de la cascade de Haar sont qu'elle permet de calculer très rapidement des caractéristiques de type Haar grâce à l'utilisation d'images. Elles sont également très efficaces pour la sélection des caractéristiques. Grâce à l'utilisation de l'algorithme AdaBoost. Ainsi, ils peuvent détecter les visages dans les images, indépendamment de l'emplacement ou de l'échelle du visage. Pour ce qui est des limitations, le détecteur a tendance à être le plus efficace pour les images frontales du visage. Les cascades de Haar sont notoirement sujettes aux faux positifs, ce qui veut dire que l'algorithme de Viola-Jones peut facilement signaler un visage dans une image alors qu'il n'y en a pas. Comme il peut parfois entièrement manqués des visages. [19]

- **Histogramme de gradients orientés :** Un histogramme de gradient orienté (*HOG*) est une caractéristique utilisée en vision par ordinateur pour la détection d'objet, il a été présenté

pour la première fois par Dalal et Triggs dans leur ouvrage fondamental de 2005 « Histogrammes of Oriented Gradients for Human Detection »[21].

Similaire aux cascades de Haar, HOG + SVM linéaire s'appuie sur des fenêtres glissantes pour détecter des objets/faces dans une image.

La méthode consiste à diviser l'image en régions adjacentes de petite taille, appelées cellules, et calculer pour chaque cellule l'histogramme des orientations des contours pour les pixels à l'intérieur de cette cellule. La combinaison des histogrammes forme alors le descripteur HOG. [52]

Pour ce faire, il faut d'abord mettre l'image en blanc et noir. Après quoi il faut examiner chaque pixel de l'image, et pour chacun il faut examiner chaque pixel qui l'entourent directement.

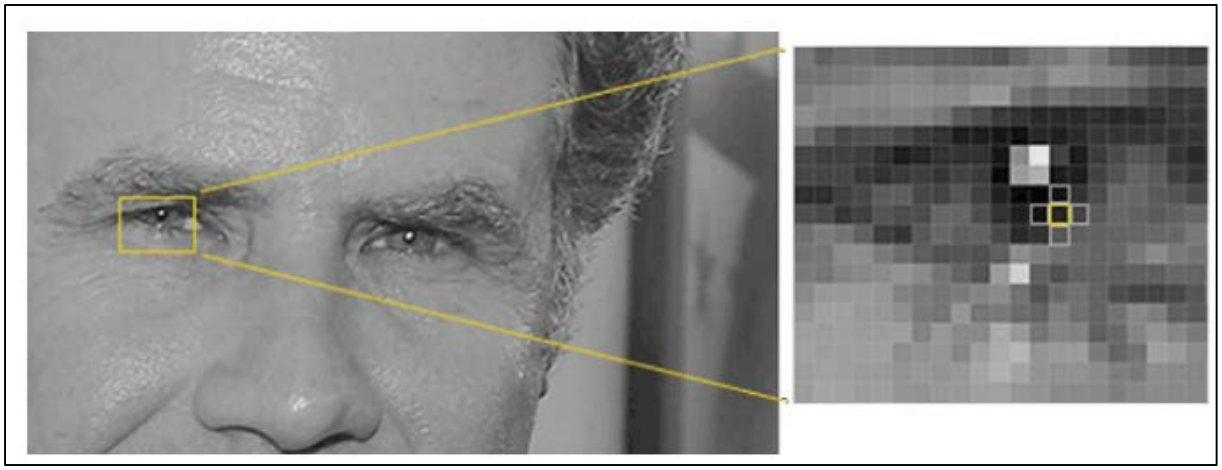

Figure 6: examen d'un pixel et ceux qui l'entoure directement [52].

Le but est d'abord déterminer à quel point un pixel est sombre par rapport à ceux qui l'entourent directement, puis dessiner une flèche dans le sens où l'image s'assombrit. En répétant ce processus, les pixels seront remplacés par des flèches appelées gradients, qui montrent le passage du clair au foncé sur l'ensemble de l'image.

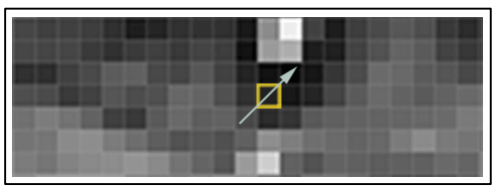

Figure 7 : l'image s'assombrit vers le haut

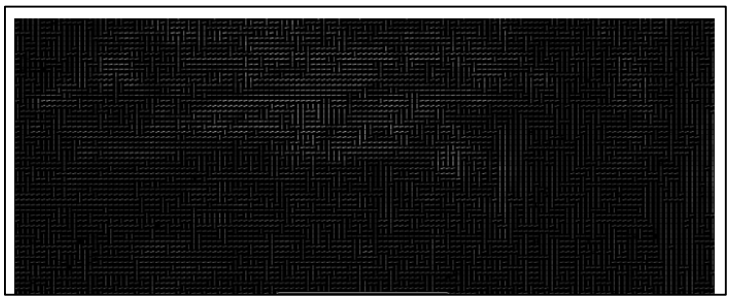

Figure 8 : ensemble de flèches montrant le passage du clair au foncé de l'œil

Sauf que ceci nous donne beaucoup trop de détails. Il serait préférable de voir le flux de base de la clarté et de l'obscurité à un niveau plus élevé, afin de voir le modèle de base de l'image. Pour cela, l'image est divisée en petits carrés de 16x16 pixels chacun. Dans chaque carré, il faut voir combien de dégradés pointent dans chaque direction principale (combien pointent vers le haut, vers le haut-droit, vers la droite, etc...). Ensuite, il faut remplacer ce carré dans l'image par les flèches de direction qui étaient les plus fortes. [52]

Le résultat final la transformation de l'image originale en une représentation très simple qui capture la structure de base d'un visage d'une manière simple :

Figure 9 : Image originale

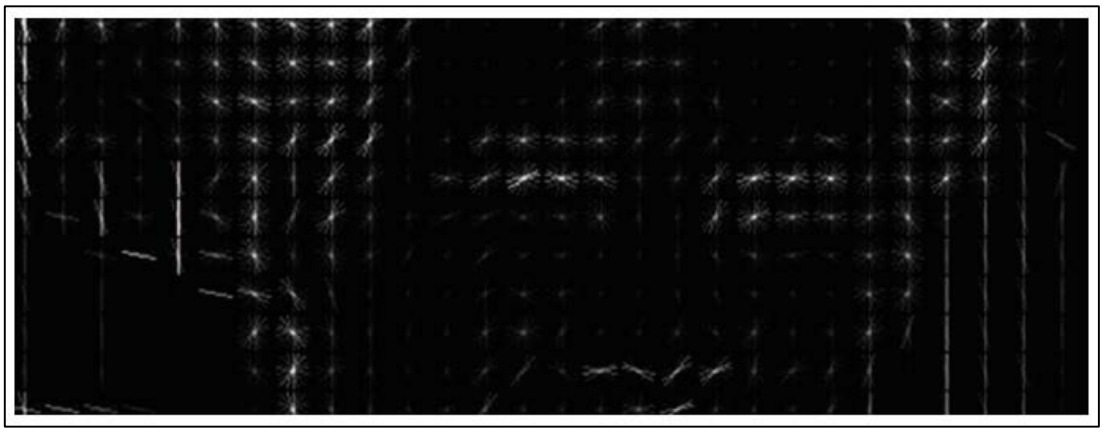

Figure 10 : Transformée en une représentation HOG capturant les principales caractéristiques de l'image,

Tout ce qui précède représente le travail fait pour entrainer le système, du coup pour trouver des visages dans cette forme HOG, tout ce que nous devons faire est de trouver la partie de notre image qui ressemble le plus à un modèle HOG connu qui a été extrait d'un tas d'autres visages d'entraînement:

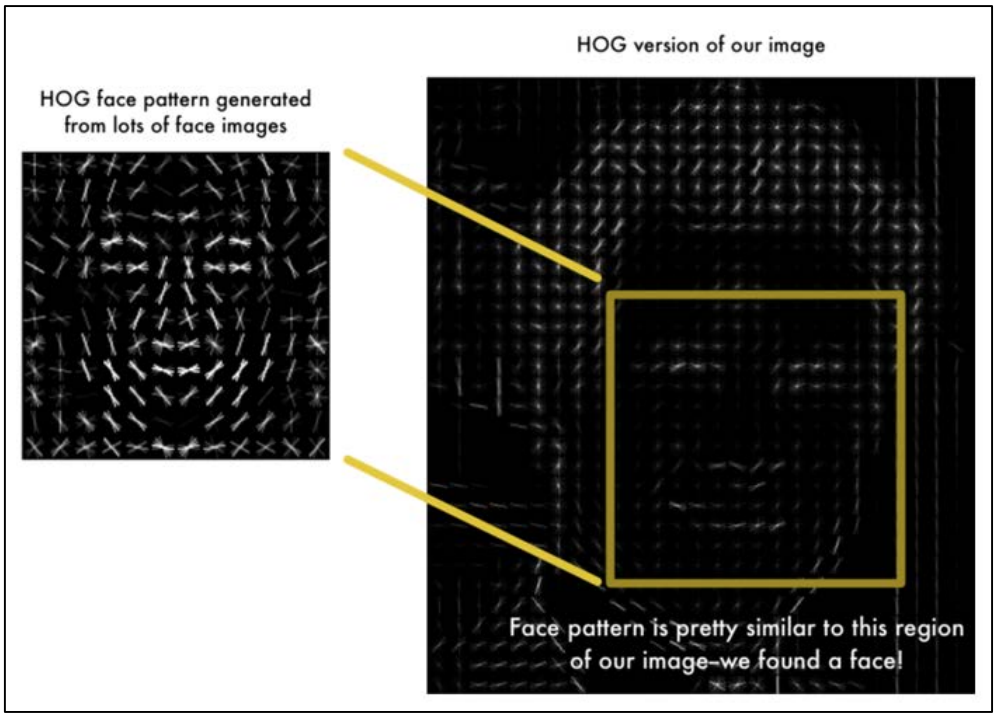

Figure 11 : Comparaison avec un modèle de visage généré de plusieurs images pour détecter le visage [52].

Les algorithmes de détection de visage qui utilisent le descripteur caractéristique histogramme des gradients orientés (HOG) et le classifier linéaire machine à vecteur de support (SVM) se sont avérés efficaces et généralistes, car ils se concentrent sur les contours du visage dans l'image. Le descripteur caractéristique HOG ne détecte pas directement les contours du visage, mais le vecteur normal de l'hyperplan obtenu dans la phase d'apprentissage du SVM accorde un poids important aux caractéristiques des contours du visage[21] [22].

Cet algorithme est un classique de la littérature sur la vision par ordinateur et est toujours utilisé aujourd'hui.

La méthode a des avantages à savoir qu'elle est plus précise que les cascades de Haar et a une détection plus stable que les cascades de Haar (c'est-à-dire moins de paramètres à régler). Elle est normalement utilisée pour la détection des images frontales mais marche également pour les images légèrement non frontales [52]. Par contre elle ne fonctionne que sur des vues frontales du visage - les visages de profil ne seront pas détectés car le descripteur HOG ne tolère pas bien les changements de rotation ou d'angle de vue, et surtout assez coûteuse en calcul en raison de la construction des fenêtres glissantes et du calcul des caractéristiques HOG à chaque arrêt de la fenêtre. [53]

- **DNN:** Il est facile de remplacer les cascades de Haar par des détecteurs de visages plus précis issus de l'apprentissage profond. La dernière version comprend un module DNN (Deep Neural Network), qui est livré avec un réseau de neurones convolutifs (CNN) pré-entraîné pour la détection des visages. Ce nouveau modèle améliore les performances de détection des

visages par rapport aux modèles traditionnels, tels que Haar. Le Framework utilisé pour entraîner le nouveau modèle est Caffe.

Pour utiliser le module de réseau neuronal profond avec des modèles Caffe, il est nécessaire d'avoir les deux ensembles de fichiers :

- Le(s) fichier(s) .prototxt qui définit(nt) l'architecture du modèle (c'est-à-dire les couches elles-mêmes).
- Le fichier .caffemodel qui contient les poids pour les couches réelles.

Caffe ou architecture convolutive pour l'intégration rapide de fonctionnalités, est un Framework d'apprentissage profond open source. Il se caractérise par sa vitesse, sa compatibilité, sa puissance d'expression et sa modularité. Il est écrit à l'aide des langages de programmation C++ et Python.

D'après [1000], Caffe fait partie des implémentations de convnet les plus rapides disponibles.

Parmi les avantages de la détection avec apprentissage profond nous pouvons compter qu'il est un détecteur de visage précis, qu'il utilise des algorithmes modernes d'apprentissage profond, qu'il n'a besoin d'aucun réglage de paramètre contrairement au modèle Haar-cascade. Sauf que parfois il peut être moins précis que le détecteur de visage CNN, et peut ne pas détecter les personnes à la peau foncée aussi précisément que les personnes à la peau claire.

- **CNN:** Davis King, le créateur de DLIB, a formé un détecteur de visage CNN basé sur son travail sur la détection d'objets à marge maximale [23]. La méthode est très précise, grâce à la conception de l'algorithme lui-même, ainsi qu'au soin apporté par Davis à la sélection de l'ensemble d'entraînement et à l'entraînement du modèle.

Le détecteur basé sur CNN est capable de détecter les visages dans presque tous les angles, certes il a une grande précision. Il peut également gérer les occlusions (quand une partie du visage est cachée derrière un objet). Mais ne peut pas fonctionner en temps réel sans accélération GPU, et n'arrive pas à détecter les petits visages sur les images puisqu'il est formé sur une taille de visage minimale de 80 * 80. [53]

- **MTCNN:** MultiTask Cascaded Convolutional Neural Network est un outil moderne de détection de visage, exploitant un détecteur de réseau neuronal à 3 étages.

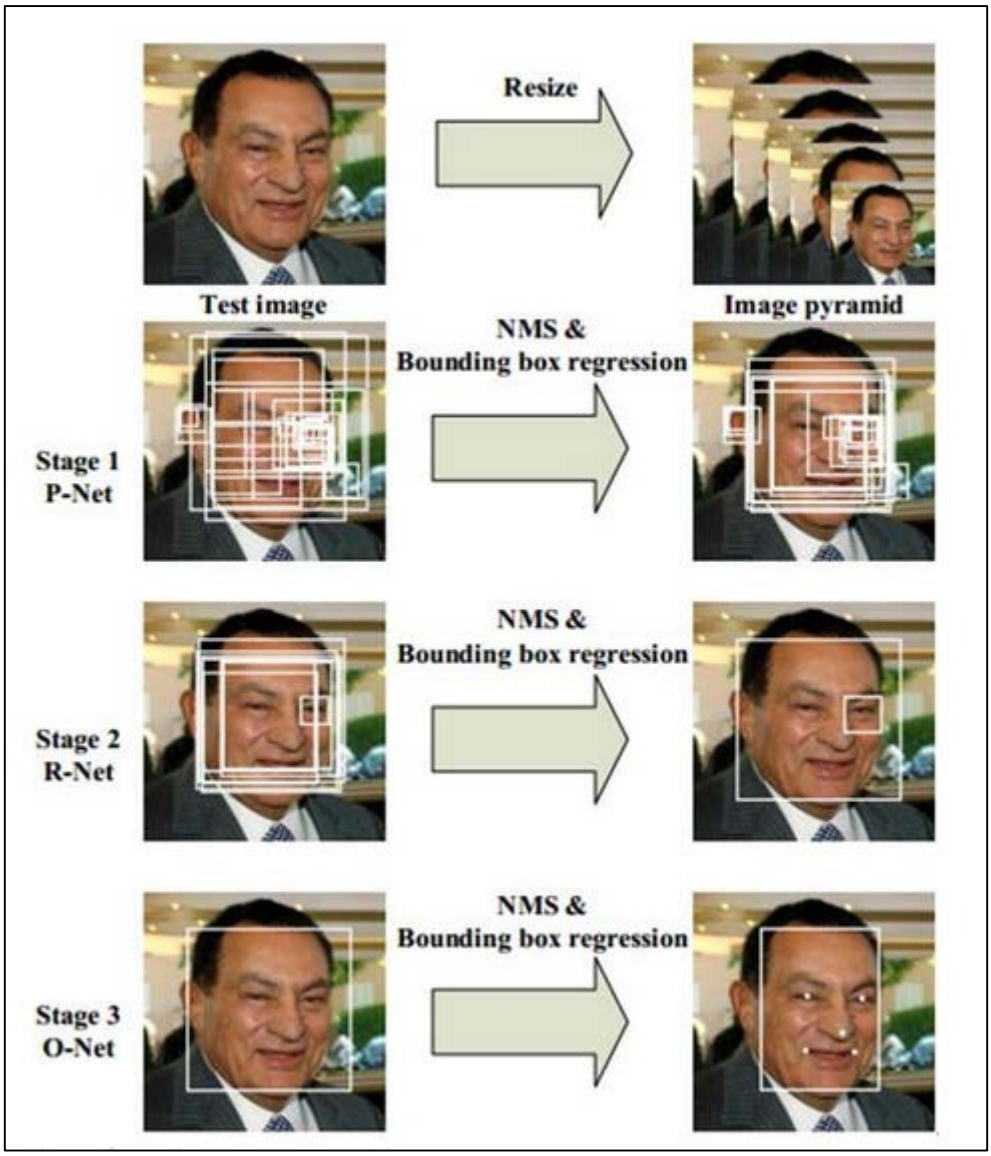

Figure 12: Différentes étapes de MTCNN [24]

D'abord, l'image est redimensionnée plusieurs fois pour détecter des visages de différentes tailles. Ensuite, le P-Net (Proposal Net) scanne les images, effectuant la première détection. Il a un seuil de détection bas et détecte donc de nombreux faux positifs, même après NMS (Non-Maximum Suppression), mais fonctionne comme ça à dessein.

Les régions proposées (contenant de nombreux faux positifs) sont entrées pour le deuxième réseau, le R-Net (Refinement Network), qui, filtre les détections (également avec NMS) pour obtenir des boîtes englobantes assez précises.

La dernière étape, l'O-Net (Output Network) effectue le raffinement final des boîtes englobantes. De cette façon, non seulement les visages sont détectés, mais les cadres de délimitation sont très justes et précis.

Il est utilisable sous implémentation TensorFlow, et aussi sous celle de PyTorch qui est plus rapide. Il est utilisable aussi pour détection en temps réel.

MTCNN est très précis et robuste. Il détecte correctement les visages même avec des tailles, un éclairage et des rotations importantes. Mais il est plus lent par rapport à quelques détecteurs comme Haar-cascade, HOG et DNN. Un autre désavantage est qu'il détecte des faux négatifs, chose à laquelle nous pouvons remédier en modifiant le seuil de MTCNN.

4.2. Extraction Des Caractéristiques

4.2.1. Définition

Après l'étape de la détection vient celle de l'extraction des caractéristiques qui est l'étape d'initialisation de base et la plus importante pour la reconnaissance faciale consistant en l'extraction des composants biologiques du visage. Il existe différentes méthodes pour extraire les diverses combinaisons de caractéristiques qui en théorie ne peuvent être identique pour deux personne à l'exception des jumeaux identiques.

Nous pouvons diviser les méthodes d'extraction en deux approches :

4.2.2. L'approche Peu Profonde

L'analyse en composantes principales (ACP) est utilisée pour simplifier les données en les transformant linéairement et en formant de nouvelles coordonnées avec une variation maximale. Elle calcule la matrice de covariance à partir de plusieurs parties d'une collection d'images de visage d'entraînement [24].

Cette méthode d'analyse est utilisée dans l'algorithme Eigenfaces qui émane de travaux considéré comme étant fondamentaux dans l'histoire de la vision par ordinateur.

Pour une image donnée de taille k x k, on concatène les rangés pour avoir une liste k^2 d'intensités de pixels en niveaux de gris.

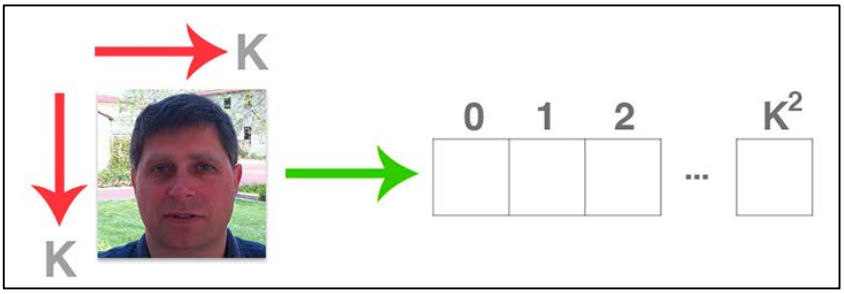

Figure 13 : Aplatir une image kxk en un seul vecteur caractéristique. [40]

Pour un ensemble de données, il faut aplatir chaque image puis en former une matrice ou Z est le nombre d'images, et K colonnes carrées est la représentation de l'image aplatie.

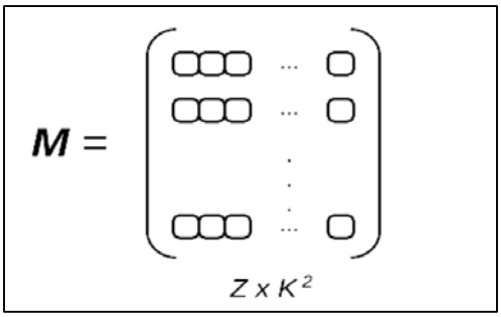

Figure 14 : Matrice représentant l'ensemble de données du visage. [40]

C'est là que la méthode ACP entre en jeu et transforme les visages en petites séries de caractéristiques centrales, les faces propres (Eigenfaces), qui sont les composantes principales de la série d'images de formation initiale [40].

Les paramètres utilisés dans l'extraction des caractéristiques des Eigenfaces comprennent la valeur moyenne de l'image, la matrice de covariance, la valeur moyenne de l'image, la valeur propre et le vecteur propre.

Etant donnée V la matrice de vecteur propre avant de procéder à l'identification des visages à l'aide de l'algorithme des faces propres, discutons des représentations des faces propres :

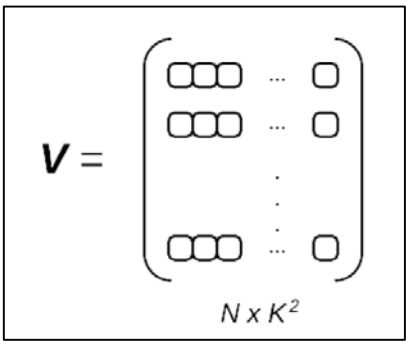

Figure 15 : Matrice V contenant N lignes (nos vecteurs propres), chacune de dimension K^{2}. [40]

Chaque ligne de la matrice ci-dessus est une face propre avec K^ {2} entrées, exactement comme l'image originale. Ceci veut dire qu'il est possible de remodeler chaque vecteur en bitmap k x k (tableau de pixels composant une image)

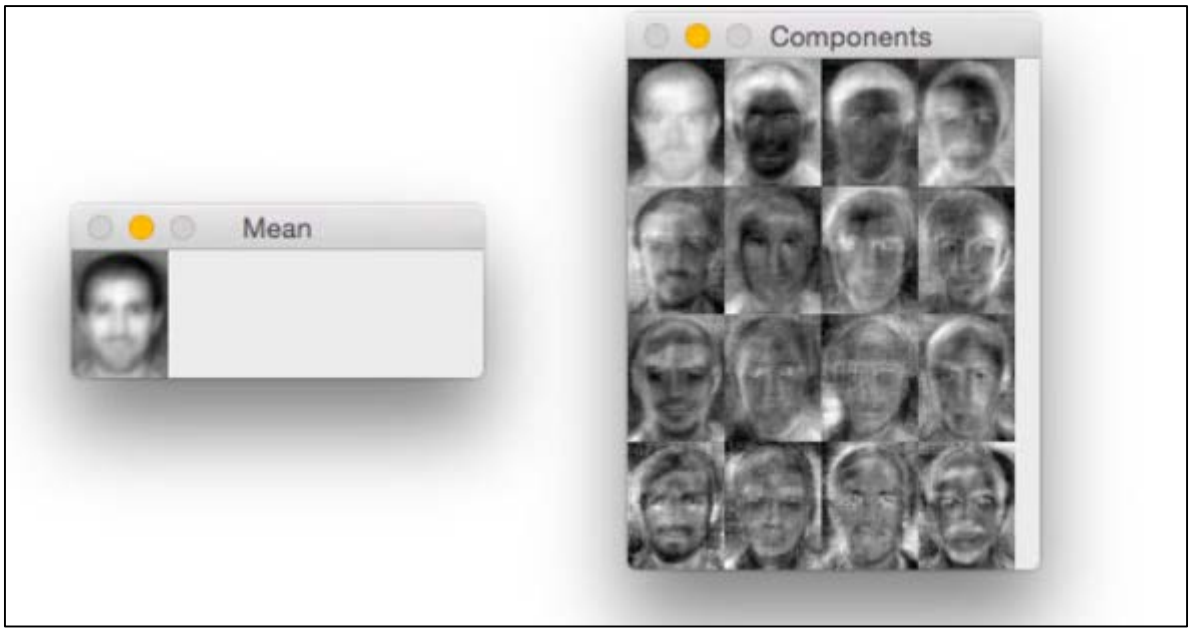

Figure 16 : À gauche : image moyenne de visage. À droite : les 16 meilleures représentations de visage. Les régions claires montrent une forte variation dans l'ensemble de données, tandis que les zones sombres suggèrent une faible variation. [40]

L'image de gauche représente simplement la moyenne de tous les visages de l'ensemble de données, tandis que les figures de droite montrent les écarts les plus importants par rapport à la moyenne dans cet ensemble.

Ceci nous permet de visualiser la dimension dans laquelle les visages des personnes varient le plus ; les régions plus claires correspondent à une variation plus importante, tandis que les régions plus sombres correspondent à une variation faible ou nulle.

L'avantage de cette approche par rapport aux autres systèmes de reconnaissance faciale réside dans sa simplicité, sa rapidité et son insensibilité aux changements mineurs ou graduels du visage. Par contre, les images doivent être des vues frontales verticales de visages humains, de plus. [40]

Pour effectuer l'identification des visages, Sirovich et Kirby [2] ont proposé de calculer la distance euclidienne entre les représentations des faces propres projetées :

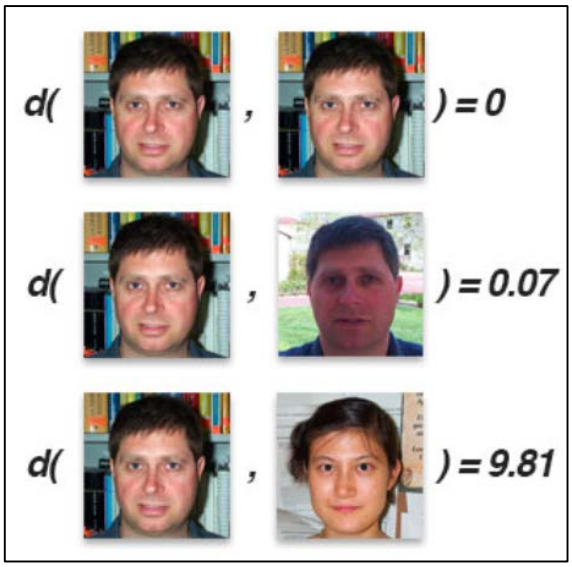

Figure 17 : Calcul de la distance euclidienne (d) entre les représentations de visages.[40]

Plus la distance euclidienne (d) est petite, plus les deux visages sont "similaires". L'identification globale est trouvée en prenant l'étiquette associée au visage ayant la plus petite distance euclidienne.

Par exemple, dans la figure 17, la paire d'images du haut a une distance de 0 car les deux images sont identiques. La paire d'images du milieu a une distance de 0,07 puisque les images contiennent le même visage bien que différente. Contrairement à la troisième paire qui présente une distance beaucoup plus grande (9,81), ce qui indique que les deux visages ne représentent pas la même personne. [40]

L'approche Profonde

Grace à la disponibilité de données massives et de réseaux pour apprentissage profond, et puisqu'on emploie les algorithmes pour apprendre automatiquement lesquelles des caractéristiques sont les plus pertinentes pour obtenir des vecteurs les plus discriminants possible en fonction des données fournies pour l'apprentissage, cette approche est devenue particulièrement intéressante et appliquée de plus en plus.

Nous allons d'abord commencer par un aperçu sur l'apprentissage profond (Deep Learning)

4.2.1.1. Deep Learning

C'est une classe du Machine Learning (apprentissage automatique) qui utilise plusieurs couches pour l'extraction progressive de caractéristiques du plus haut niveau à partir d'une entrée brute. D'ailleurs, le mot « deep » de deep learning fait référence au nombre de couches dites cachées d'un réseau de neurones par lesquelles les données sont transformées. Les modèles d'apprentissages profonds sont capables d'extraire de meilleures caractéristiques que

les modèles peu profonds, ce qui veut dire que plus il y a des couches supplémentaires plus ces dernières aident à apprendre les caractéristiques de manière efficace.

La naissance du concept du Deep Learning vient grâce à la convergence des facteurs :

- Les réseaux de neurones multicouches ;
- Les algorithmes apprenants ;
- Les algorithmes d'analyse discriminante : qui est une technique statistique visant à décrire, expliquer et prédire l'appartenance d'un ensemble observé à des groupes prédéfinis (classes, modalités de la variable à prédire…) et ce à partir d'une série de variables prédictives (descripteurs, variables exogènes…).
- Des machines capables de traiter des données massives ;
- Des bases de données suffisamment grandes pour traiter des systèmes de grandes tailles.

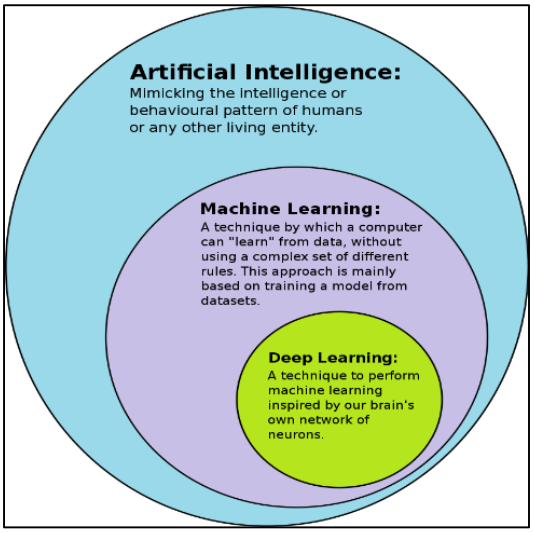

Figure 18 : Deep Learning [25]

Dans l'apprentissage profond, chaque niveau apprend à transformer ses données d'entrée en une représentation légèrement plus abstraite et composite. Dans une application de reconnaissance d'images, l'entrée brute peut être une matrice de pixels ; la première couche de représentation peut abstraire les pixels et coder les bords ; la deuxième couche peut composer et coder des arrangements de bords ; la troisième couche peut coder un nez et des yeux ; et la quatrième couche peut reconnaître que l'image contient un visage (figure 19). Il est important de noter qu'un processus d'apprentissage profond peut apprendre de lui-même quelles caractéristiques placer de manière optimale dans quel niveau. (Bien entendu, cela n'élimine pas complètement la nécessité d'un réglage manuel ; par exemple, le fait de varier le nombre de couches et la taille des couches peut fournir différents degrés d'abstraction).

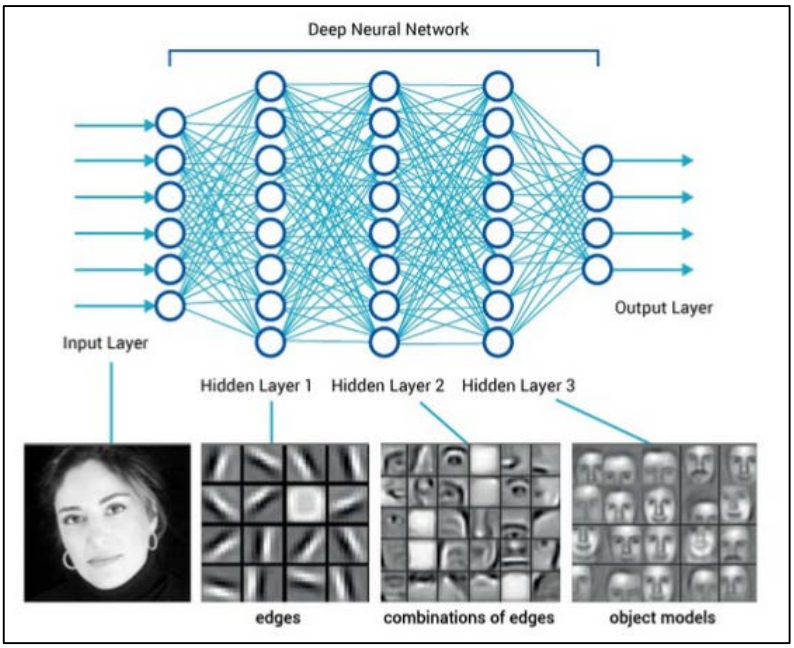

Figure 19: Deep Neural Network

Dans les années 1990, Yann Le Cun, considéré comme l'un des inventeurs de l'apprentissage profond [26] a développé la technique des réseaux convolutifs pour la reconnaissance d'image.

4.2.1.2. Réseaux convolutifs

Les CNN désignent une sous-catégorie de réseaux de neurones et sont à ce jour un des modèles de classification d'images réputés être les plus performants.

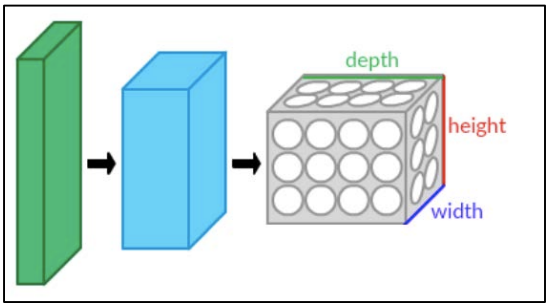

Figure 20: Architecture d'une couche de CNN [27]

(CNN) qui tirent parti de la structure de l'image d'entrée et définissent une architecture de réseau de manière plus judicieuse. Avec des couches disposées dans un volume 3D en trois dimensions : largeur, hauteur et profondeur (où la profondeur fait référence à la troisième dimension du volume, comme le nombre de canaux dans une image ou le nombre de filtres dans une couche). On peut diviser CNN en deux blocs :

Le premier bloc est celui fait sa particularité, puisqu'il fonctionne comme un extracteur de features. Pour cela, il effectue du Template matching en appliquant des opérations de filtrage par convolution. La première couche filtre l'image avec plusieurs noyaux de

convolution, et renvoie des "feature maps", qui sont ensuite normalisées (avec une fonction d'activation) et/ou redimensionnées. Ce procédé peut être réitéré plusieurs fois : on filtre les features maps obtenues avec de nouveaux noyaux, ce qui nous donne de nouvelles features maps à normaliser et redimensionner, et qu'on peut filtrer à nouveau, et ainsi de suite. Finalement, les valeurs des dernières feature maps sont concaténées dans un vecteur qui définit la sortie du premier bloc, et l'entrée du second. [28]

Le second bloc n'est pas caractéristique d'un CNN : il se retrouve à la fin de tous les réseaux de neurones utilisés pour la classification. Les valeurs du vecteur en entrée sont transformées (avec plusieurs combinaisons linéaires et fonctions d'activation) pour renvoyer un nouveau vecteur en sortie contenant autant d'éléments qu'il y a de classes : l'élément i représente la probabilité que l'image appartienne à la classe i. Chaque élément est donc compris entre 0 et 1, et la somme de tous vaut 1. Ces probabilités sont calculées par la dernière couche de ce bloc (et donc du réseau), qui utilise une fonction logistique (classification binaire) ou une fonction softmax (classification multi-classe) comme fonction d'activation. [28]

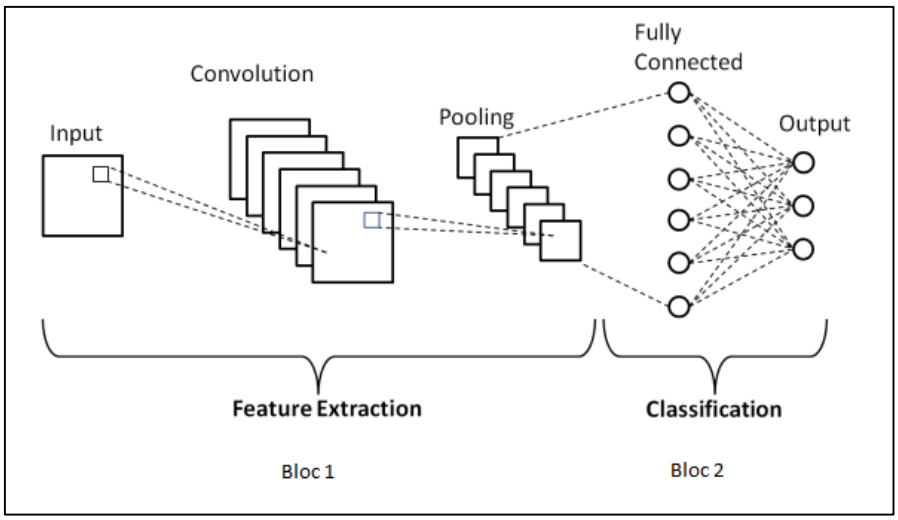

Figure 21 : Architecture d'un réseau de neurones convolutif basique.

Comme pour les réseaux de neurones ordinaires, les paramètres des couches sont déterminés par rétro propagation du gradient : l'entropie croisée est minimisée lors de la phase d'entraînement. Mais dans le cas des CNN, ces paramètres désignent en particulier les caractéristiques des images.

- **Couches CNN**

Il existe plusieurs types de couches utilisées pour construire des réseaux neuronaux convolutifs, mais celles que nous sommes le plus susceptible de rencontrer sont les suivantes :

- Convolutionnel (CONV)
- Activation (RELU)
- Pooling (POOL)

- Fully connected (FC)
- Batch normalization (BN)
- Dropout (DO)

La description d'un CNN se fait souvent par des diagrammes textuels simples :

INPUT => CONV=> POOL => RELU => FC => SOFTMAX.

Ici, nous définissons un CNN simple qui accepte une entrée, applique une couche de convolution, puis une couche d'activation, ensuite une couche entièrement connectée, et, enfin, un classificateur softmax pour obtenir les probabilités de classification en sortie. La couche d'activation SOFTMAX est souvent omise du diagramme du réseau car on suppose qu'elle suit directement le FC final. Parmi ces types de couches, CONV et FC (et dans une moindre mesure, BN) sont les seules couches qui contiennent des paramètres appris pendant le processus de formation. CONV, POOL, RELU et FC sont les plus importants lors de la définition d'une architecture réseau réelle, raison pour laquelle nous nous étendrons un peu plus dessus [29].

- **Couche de Convolution**

C'est la composante clé des CNN, et constitue toujours au moins leur première couche. Son but est de repérer la présence d'un ensemble de caractéristiques dans les images reçues en entrée. 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. Une caractéristique est alors vue comme un filtre: les deux termes sont équivalents dans ce contexte. La couche de convolution reçoit donc en entrée plusieurs images, et calcule la convolution de chacune d'entre elles avec chaque filtre. Les filtres correspondent exactement aux caractéristiques que l'on souhaite retrouver dans les images. On obtient pour chaque paire (image, filtre) une carte d'activation (feature map) qui nous indique où se situent les caractéristiques dans l'image: plus la valeur est élevée, plus l'endroit correspondant dans l'image ressemble à la caractéristique. Le choix des features contrairement aux méthodes traditionnelles, les features ne sont pas prédéfinies selon un formalisme particulier (par exemple SIFT), mais apprises par le réseau lors la phase d'entraînement ! Les noyaux des filtres désignent les poids de la couche de convolution. Ils sont initialisés puis mis à jour par rétro propagation du gradient. C'est là toute la force des réseaux de neurones convolutifs: ceux-ci sont capables de déterminer tout seul les éléments discriminants d'une image, en s'adaptant au problème posé. [30]

- **Couche de Pooling :**

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs cartes d'activation, et applique à chacune d'entre elles l'opération de pooling qui consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes. Pour cela, l'image est découpée en cellules régulières, puis on garde au sein de chaque cellule

la valeur maximale. En pratique, on utilise souvent des cellules carrées de petite taille pour ne pas perdre trop d'informations. On obtient en sortie le même nombre de *cartes d'activation* qu'en entrée, mais celles-ci sont bien plus petites. La couche de pooling permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage. [58] [30]

- **Couche de Correction Relu :**

ReLU (*Rectified Linear Units*) désigne la fonction réelle non-linéaire définie par ReLU(x)=max (0, x).

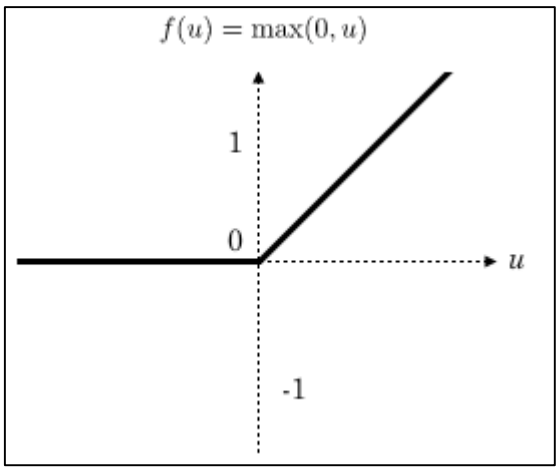

Figure 22 : Allure de la fonction RELU [58]

Après chaque couche CONV dans un CNN, nous appliquons une fonction d'activation non linéaire, telle que ReLU. Les couches d'activation ne sont pas techniquement des "couches" (du fait qu'aucun paramètre/poids n'est appris dans une couche d'activation) et sont parfois omises des diagrammes d'architecture de réseau car on suppose qu'une activation suit immédiatement une convolution. [58]

La couche de correction ReLU remplace donc toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation. [29]

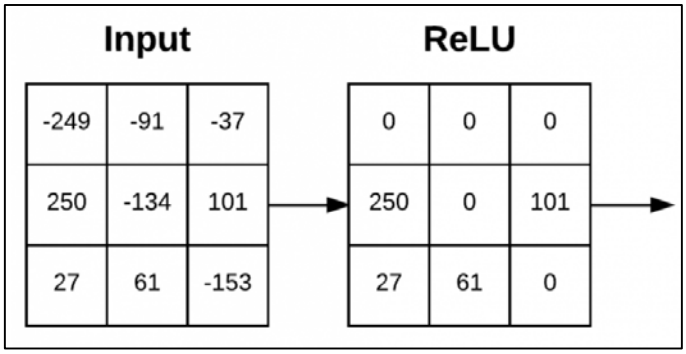

Figure 23 : Exemple d'un volume d'entrée passant par une activation ReLU, max (0, x). Les activations sont effectuées sur place, il n'est donc pas nécessaire de créer un volume de sortie distinct, même s'il est facile de visualiser le flux du réseau de cette manière. [29]

- **Couche Fully-Connected:**

La couche fully-connected constitue toujours la dernière couche d'un réseau de neurones, convolutif ou non – elle n'est donc pas caractéristique d'un CNN.

La sortie des couches de convolution et d'échantillonnage représente des caractéristiques de haut niveau de l'image d'entrée. L'objectif de la couche complètement connectée est d'utiliser ces caractéristiques pour classer l'image d'entrée du réseau en différentes classes en fonction de la base de données d'apprentissage. Le terme « fully connected » implique que chaque neurone de la couche précédente est connecté à chaque neurone de la couche suivante. [30]

Les couches entièrement connectées sont généralement suivies d'un Dropout [30]. Ce dernier agit sur les poids de ces couches afin de désactiver un certain nombre de neurones pour réduire le nombre de paramètres. Cela permet de contrôler le sur-apprentissage qui peut être causé par un nombre important de paramètres. [30]

4.2.1.3. Architectures CNN

En 1998, le chercheur français Yann LeCun a proposé avec ses collègues l'architecture Lenet-5, l'un des premiers modèles pré-entraînés [31] qui est un réseau neuronal de convolution multicouche pour la classification d'images dont la principale raison de popularité est son architecture simple et directe.

LeNet a inspiré quelques chercheurs et en particulier ceux de Toronto qui ont remportés la compétition d'ImageNet en 2012 avec leur réseau AlexNet, faisant ainsi de LeNet le pionnier, et d'AlexNet celui qui a créé l'enthousiasme autour du Deep Learning.

AlexNet : AlexNet possède huit couches avec des paramètres apprenables. Le modèle se compose de cinq couches de convolution avec une combinaison de max pooling suivi de 3 couches entièrement connectées et ils utilisent l'activation Relu dans chacune de ces couches sauf la couche de sortie. [55]

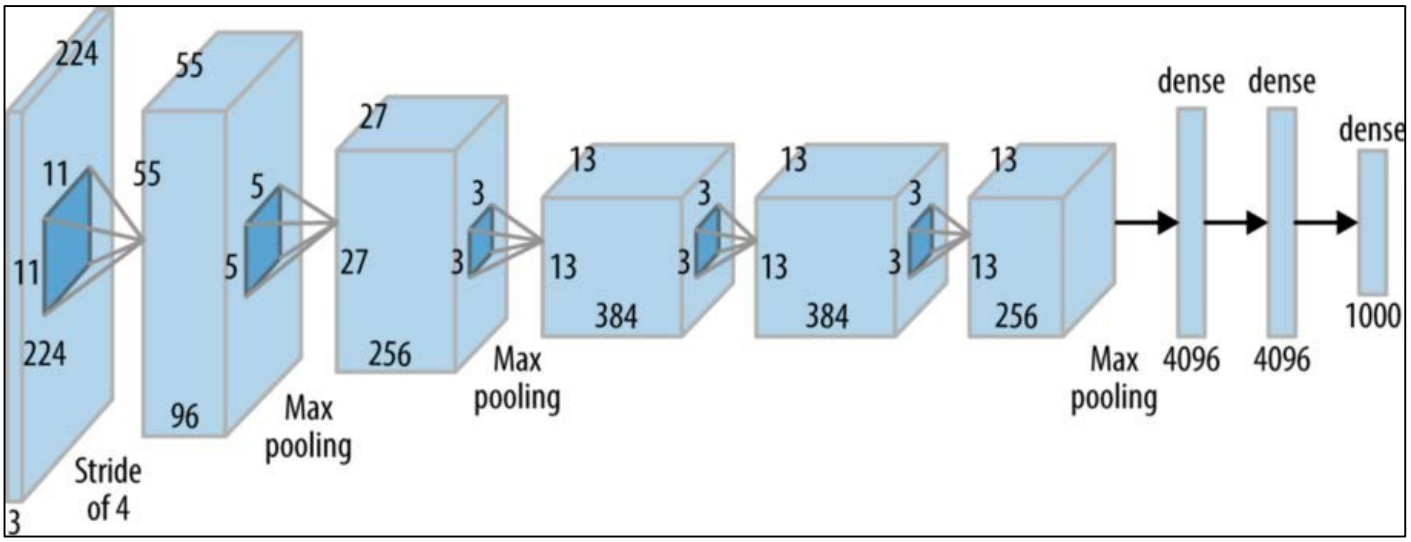

Figure 24 : Schéma fonctionnel d'Alexnet. [50]

Les développeurs de AlexNet ont découvert que l'utilisation du Relu comme fonction d'activation a accéléré la vitesse de l'apprentissage de près de six fois. Ils ont également utilisé les couches d'exclusion, qui ont empêché leur modèle de s'adapter de manière excessive. En outre, le modèle est entraîné sur le dataset Imagenet qui contient près de 14 millions d'images réparties sur un millier de classes. [55]

L'intuition derrière ce réseau est que chaque couche convolutionnelle apprend une représentation plus détaillée des images (feature map) que la précédente. Par exemple, la première couche est capable de reconnaître des formes ou des couleurs très simples, et la dernière des formes plus complexes comme des visages complets par exemple. [56]

Certains chercheurs affirment que la victoire d'ImageNet en octobre 2012 a ancré le début d'une " révolution de l'apprentissage profond " qui a transformé l'industrie de l'IA [32].

VGGNet : L'architecture de réseau VGG a été introduite par Simonyan et Zisserman dans leur article de 2014, Very Deep Convolutional Networks for Large Scale Image Recognition [33]. Elle est née de la nécessité de réduire le nombre de paramètres dans les couches CONV et d'améliorer le temps de formation [51].

Il existe quelques variantes de VGGNet (VGG16, VGG19, etc.) qui ne diffèrent que par le nombre total de couches dans le réseau. Les détails structurels d'un réseau VGG16 sont présentés ci-dessous.

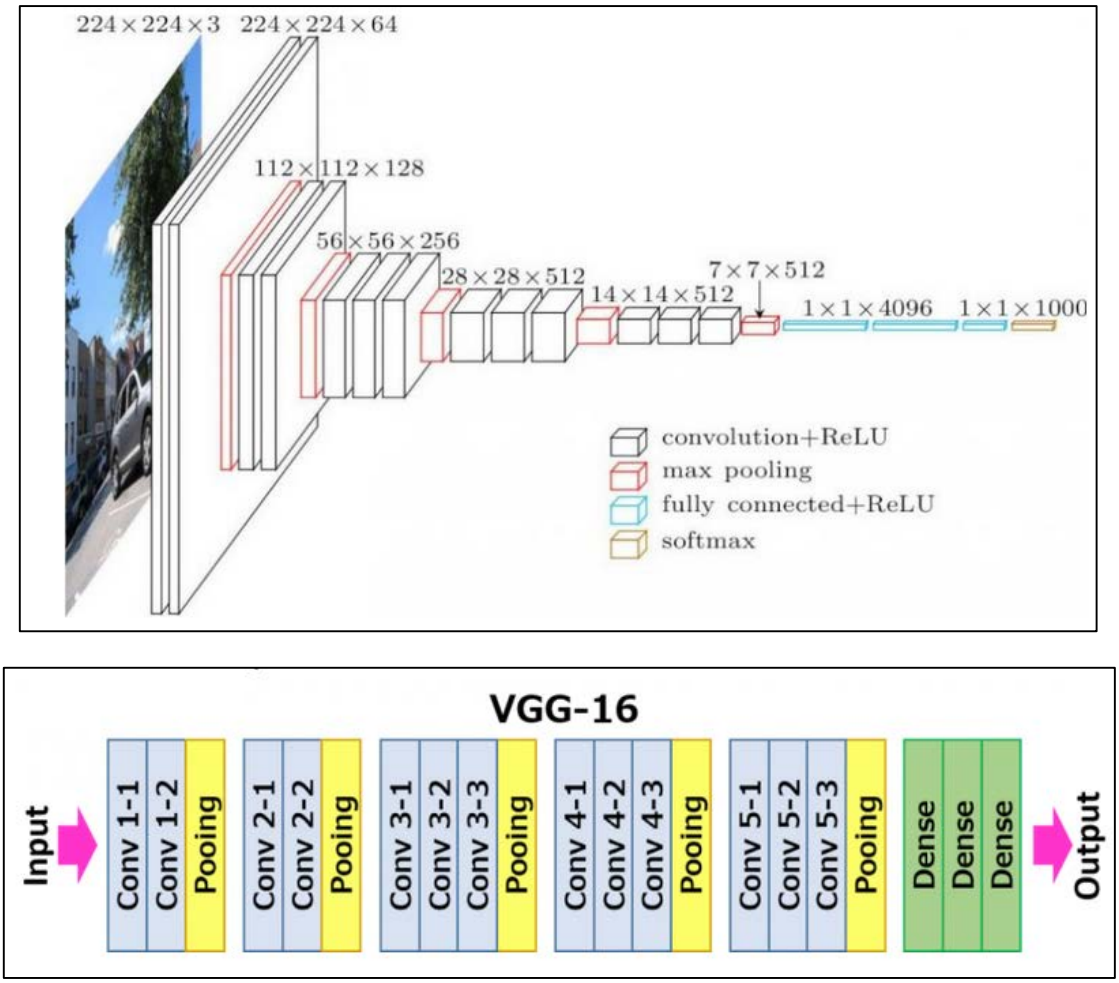

Figure 25: Architecture du réseau VGG 16

Le modèle a atteint une précision de 92.7% sur ImageNet ce qui est un des meilleurs scores obtenus. Il a marqué une progression par rapport aux modèles précédents en proposant, dans les couches de convolution, des noyaux de convolution de plus petites dimensions (3×3) que ce qui avait été fait jusque-là. [54]

Pour un filtre à 5x5 couches conv, le nombre de variables est de 25. Par contre, deux couches conv de taille de noyau 3x3 ont un total de 3x3x2=18 variables (une réduction de 28%). De même, l'effet d'une couche conv 7x7 (11x11) peut être obtenu en mettant en œuvre trois (cinq) couches conv 3x3 avec un pas de un. Cela réduit le nombre de variables entraînables de 44,9 % (62,8 %). Un nombre réduit de variables entraînables signifie un apprentissage plus rapide et plus robuste. [51]

Le modèle a été entraîné sur des semaines en utilisant des cartes graphiques de pointe. Le réseau VGG-16 présente cependant deux inconvénients majeurs : Il est lent à former, et les poids de l'architecture du réseau sont eux-mêmes assez importants. [54]

Inception : Dans une tâche de classification d'images, la taille de l'élément saillant peut varier considérablement dans l'image. Il est donc assez difficile de décider d'une taille de

noyau fixe. Les noyaux plus grands sont préférés pour les caractéristiques plus globales qui sont distribuées sur une grande surface de l'image, d'autre part, les noyaux plus petits donnent de bons résultats dans la détection des caractéristiques spécifiques à une zone qui sont distribuées dans l'image. Pour une reconnaissance efficace d'une telle caractéristique de taille variable, nous avons besoin de noyaux de tailles différentes. C'est ce que fait Inception. Au lieu de simplement aller plus loin en termes de nombre de couches, il va plus loin. Plusieurs noyaux de tailles différentes sont mis en œuvre dans la même couche. [51]

L'architecture du réseau Inception se compose de plusieurs modules de démarragedont la structure est la suivante :

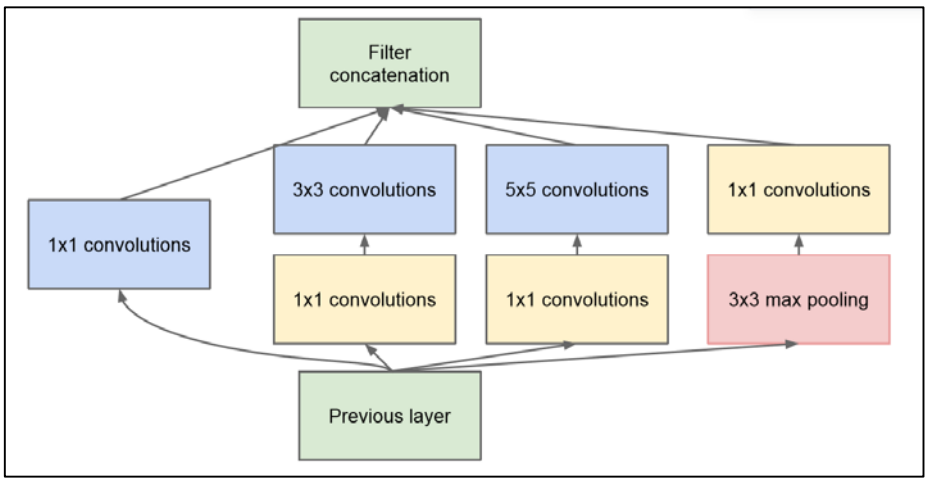

Figure 26: Module Inception [34]

GoogLeNet a proposé ce module ou cette mini-architecture dont l'objectif est d'agir comme un " extracteur de caractéristiques à plusieurs niveaux " en calculant des convolutions 1×1, 3×3 et 5×5 au sein du même module du réseau. La sortie de ces filtres est ensuite empilée le long de la dimension du canal et avant d'être alimentée dans la couche suivante du réseau. Il a été conçu pour résoudre le problème des coûts de calcul, ainsi que celui de l'overfitting, entre autres. [31]

GoogleNet : En 2014, plusieurs grands modèles ont été développés comme VGG mais le gagnant du concours ImageNet a été GoogleNet.

GoogleNet est l'incarnation originale de l'architecture Inception, et quelques-unes de ces fonctionnalités sont :

- Réseau profond de 22 couches.
- Puissance de calcul efficace et plus rapide. Coût de calcul: 2 fois moins qu'AlexNet.
- Beaucoup plus précis qu'AlexNet et VGG.
- Faible utilisation de la mémoire et faible consommation d'énergie.

Le réseau est plus grand mais le nombre de paramètres est plus petit ; 12 fois moins de paramètres par rapport à AlexNet.

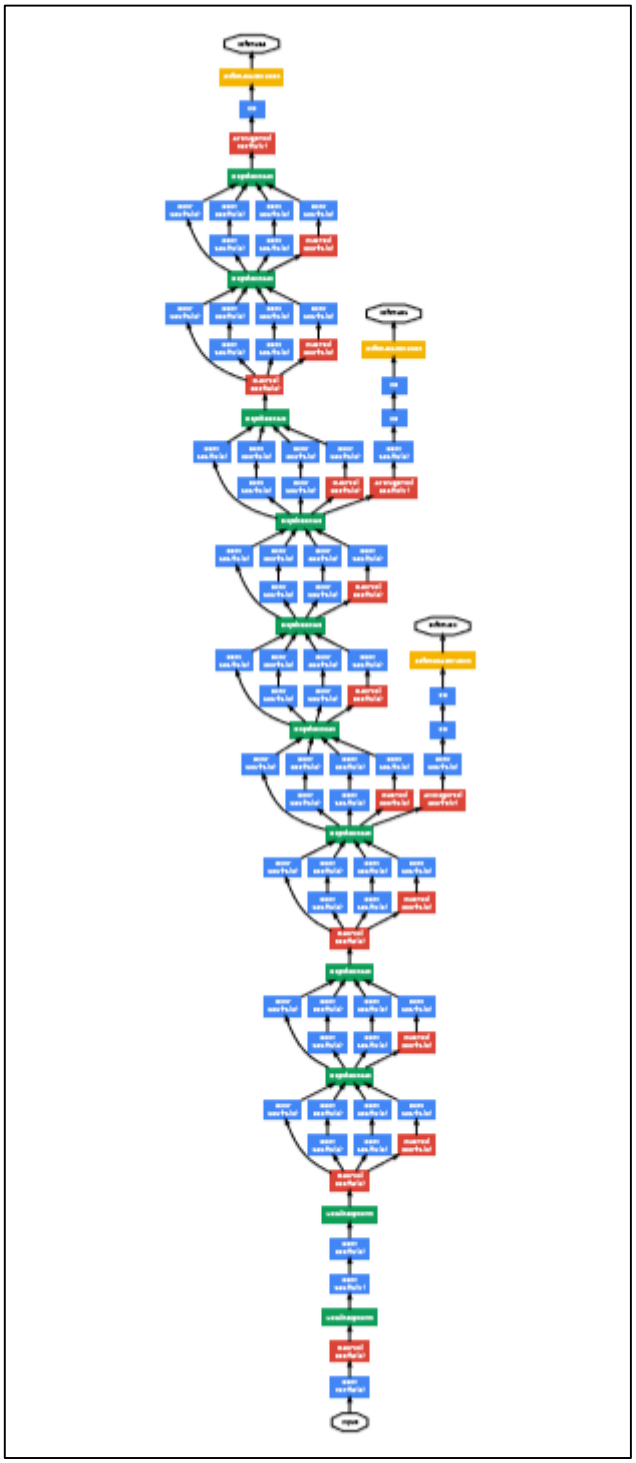

Figure 27: GoogleNet avec toutes ses caractéristiques. [34]

La plus grande amélioration de GoogLeNet par rapport au précédent réseau neuronal convolutionnel est de concevoir une structure de réseau avec des paramètres épars, mais il peut générer des données denses, ce qui augmente les performances du réseau neuronal et assure l'efficacité de l'utilisation des ressources informatiques.

FaceNet: FaceNet est un réseau neuronal de pointe pour la reconnaissance, la vérification et le regroupement des visages. Il s'agit d'un réseau neuronal profond à 22 couches, publié en 2015 par les chercheurs de Google, Schroff et al. [35].

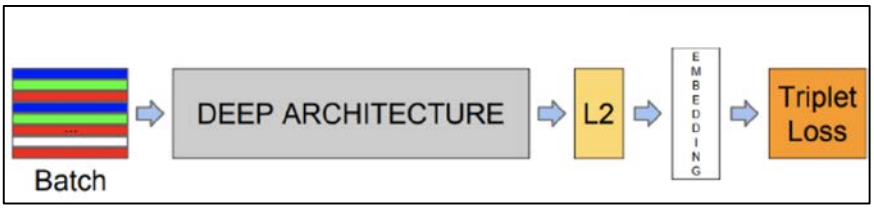

Figure 28: Structure du modèle FaceNet. [32]

Le réseau profond illustré à la figure est issu de l'architecture GoogleNet (avec de nombreuses révisions). L'article de FaceNet ne traite pas beaucoup du fonctionnement interne de l'architecture GoogleNet, et considère le réseau neuronal profond comme une boîte noire.

FaceNet prend une image d'un visage en entrée et sort un vecteur de 128 nombres qui représentent les caractéristiques les plus importantes d'un visage. En apprentissage automatique, ce vecteur est appelé embeddings parce que toutes les informations importantes d'une image sont intégrées dans ce vecteur. Fondamentalement, FaceNet prend le visage d'une personne et le compresse dans un vecteur de 128 chiffres. Idéalement, les incorporations de visages similaires sont également similaires.

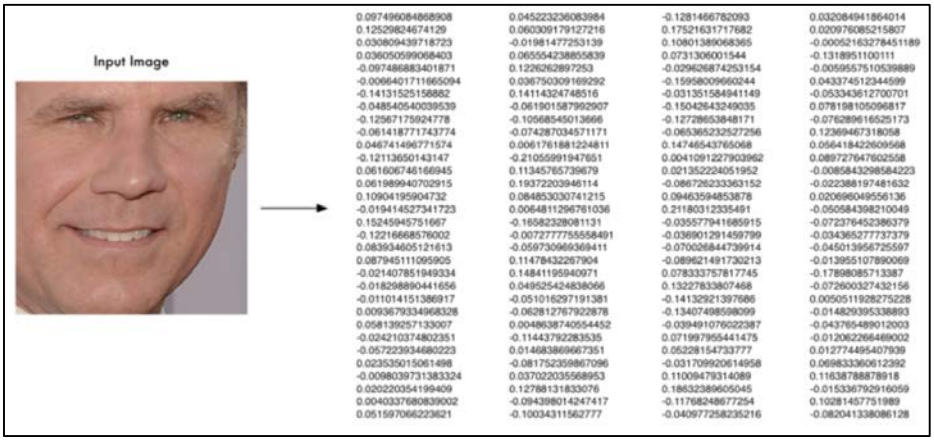

Figure 29: les 128 mesures générées par FaceNet pour un visage. [52]

Les chercheurs ont découvert que l'approche la plus précise consiste à laisser l'ordinateur déterminer lui-même les mesures à collecter. L'apprentissage profond parvient mieux que les humains à déterminer les parties du visage qu'il est important de mesurer.

L'idée de FaceNet est donc former un réseau neuronal convolutif profond et l'entraîner à générer 128 mesures pour chaque visage. Le processus d'entraînement fonctionne en examinant 3 images de visage à la fois :

- Charger une image de visage d'une personne connue

- Chargez une autre image de la même personne connue
- Charger une image d'une personne totalement différente

L'algorithme examine ensuite les mesures qu'il génère pour chacune de ces trois images. Il modifie ensuite légèrement le réseau neuronal afin de s'assurer que les mesures qu'il génère pour les images 1 et 2 sont légèrement plus proches et que les mesures pour les images 2 et 3 sont légèrement plus éloignées : Cette méthode d'apprentissage est appelée perte triplet (triplet loss). [52]

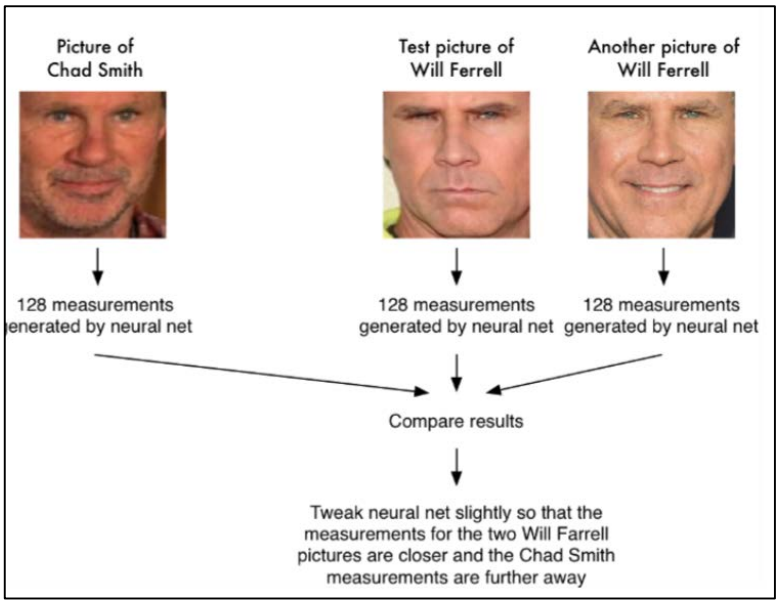

Figure 30: Triplet loss. [52]

Après avoir répété cette étape des millions de fois pour des millions d'images de milliers de personnes différentes, le réseau neuronal apprend à générer de manière fiable 128 mesures pour chaque personne. Dix photos différentes d'une même personne devraient donner à peu près les mêmes mesures. [52]

ResNet : ou deep residual networks ,développé par Kaiming He et al, est l'un des réseaux considérés comme les plus récents et les plus performants en termes d'utilisation des réseaux de neurones convolutifs pour la reconnaissance d'images. Il a remporté le concours ImageNet Large-Scale Visual Recognition Challenge en 2015 (ILSVRC-15).

Les réseaux très profonds étaient historiquement difficiles à apprendre ; lorsque les réseaux deviennent aussi profonds, ils se heurtent au problème des gradients évanescents. Les signaux sont atténués à mesure qu'ils progressent dans le réseau, ce qui entraîne une diminution de l'apprentissage. Cette atténuation peut être expliquée mathématiquement, mais l'effet est que chaque couche supplémentaire réduit de façon multiplicative la force du signal, ce qui conduit à plafonner la profondeur effective des réseaux.

Le ResNet a introduit une innovation permettant de contrôler cette atténuation : la connexion de dérivation. Ces connexions permettent à une partie du signal provenant de couches plus profondes de passer sans être atténuée, ce qui permet d'entraîner efficacement des réseaux beaucoup plus profonds. La connexion de dérivation du ResNet est illustrée à la figure 37. [50]

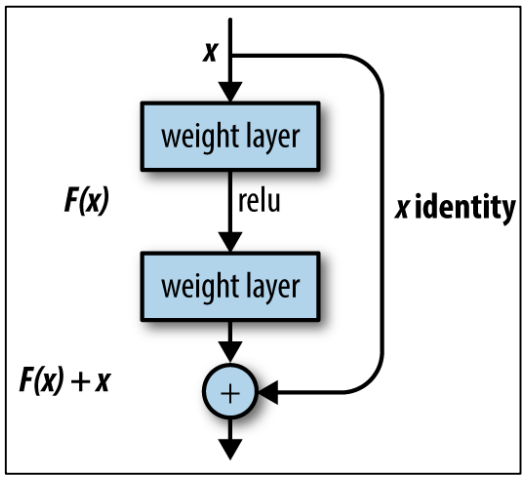

Figure 31 : La cellule ResNet. La connexion d'identité du côté droit permet à une version non modifiée de l'entrée de passer à travers la cellule. Cette modification permet l'entraînement efficace d'architectures convolutionnelles très profondes. [50]

Des preuves empiriques complètes ont été présentées montrant que ces réseaux résiduels sont plus faciles à optimiser et peuvent gagner en précision avec une profondeur considérablement accrue. Sur le jeu de données ImageNet, les réseaux résiduels ont été évalués avec une profondeur allant jusqu'à 152 couches, soit 8 fois plus que les réseaux VGG, mais toujours avec une complexité moindre.

Depuis que ResNet a époustouflé les gens en 2015, de nombreux membres de la communauté de recherche se sont plongés dans les secrets de son succès, de nombreux raffinements ont été apportés à l'architecture.

VGGFace : Il existe deux principaux modèles VGG pour la reconnaissance faciale ce sont VGGFace et VGGFace2.

Le modèle VGGFace, a été décrit par Omkar Parkhi dans l'article de 2015 intitulé "Deep Face Recognition" [18]. L'une des contributions de cet article était une description de la manière de développer un très grand ensemble de données d'entraînement, nécessaire à l'entraînement des systèmes de reconnaissance des visages basés sur des réseaux neuronaux à résolution moderne, afin de concurrencer les grands ensembles de données utilisés pour entraîner les modèles de Facebook et de Google. Le jeu de donnée a été ensuite utilisé comme base pour le développement de réseaux CNN profonds. C'est un modèle qui génère des caractéristiques généralisées à partir de visages, il utilise la fonction triplet loss.

Le modèle VGGFace2 dans l'article «VGGFace2 : A dataset for recognizing faces across pose and age" en 2017[36], est décrit comme un jeu de données beaucoup plus important qu'ils

ont collecté dans l'intention d'entraîner et d'évaluer des modèles de reconnaissance des visages plus efficaces. Des modèles sont entraînés sur ce jeu de données, en particulier les architectures ResNet-50 et SENet, et ont atteint une performance de pointe [34].

5. Correspondance des visages

Après le choix de l'architecture, son entraînement à l'aide de la fonction de perte déterminée et des données d'entraînement, l'extraction de caractéristiques profondes peut être effectuée pour les données de test, et les opérations d'identification et de vérification des visages peuvent être réalisées. La correspondance des visages peut être effectuée en utilisant la distance euclidienne ou bien la Similitude en cosinus.

5.1. Distance euclidienne

IL s'agit d'une méthode de classification des caractéristiques basée sur la distance qui calcule la distance entre les nœuds faciaux et le visage qui a la différence minimale entre ces valeurs de distance est considéré comme la correspondance. Mais il convient aux ensembles de données ayant un plus petit nombre de classes et des caractéristiques de dimensionnalité inférieures.[37]

5.2. Similitude en cosinus

Dans la similarité de cosinus, la solution que nous obtenons après avoir calculé le cosinus d'un angle est pris en compte. Ici, nous comparerons les différences entre ces résultats. Plus la valeur est proche de 1, plus la probabilité de correspondance est grande. Mais cela peut donner un faux résultat si les caractéristiques des données de test sont incomplètes (c'est-à-dire si la valeur résultante est 0, alors les caractéristiques ne correspondent pas, et si presque toutes les caractéristiques correspondent, la valeur est 1).[37]

6. Conclusion

Après une brève présentation des différentes approches de la reconnaissance faciale, nous avons présenté les étapes de cette dernière, nous avons mis la lumière sur quelques-unes des méthodes de la détection du visage, nous sommes par la suite passés à ceux de la reconnaissance faciale, sans oublier de nous intéresser à l'apprentissage profond, aux réseaux neuronaux convolutifs ainsi qu'aux différentes architectures de classification, après quoi nous sommes passés aux deux méthodes que nous estimons intéressantes pour le calcul de la correspondance des visages.

Chapitre 3 : Le SYSTEME DE RECONNAISSANCE FACIALE PROPOSE

1. Introduction

Ca chapitre contient en première partie une présentation des outils que nous avons utilisé durant notre stage. En deuxième partie, il comprend notre étude comparative entre les différents détecteurs de visages et les modèles d'extraction des caractéristiques, ainsi que notre approche proposée du système de reconnaissance faciale destinée à l'implémentation. Enfin, dans la troisième partie nous présentons notre collaboration dans le développement avec Angular.

2. Les outils d'environnement

2.1. Python

Est un langage de programmation qui s'accompagne d'un certain nombre de bibliothèques d'apprentissage automatique très attrayantes ainsi qu'un ensemble de paquets puissants pour les besoins analytiques.

Cela nous a encouragés à choisir Python plutôt que d'autres langages de programmation.

En outre, Python dispose d'une grande communauté qui fournit un soutien par le biais de forums, etc.[38]

2.2. Numpy

Est un package Python composé d'objets de type tableau multidimensionnel et d'une collection NumPy nous permettra de traiter des tableaux et d'effectuer des opérations liées à l'algèbre linéaire. [39]

2.3. Matplotlib

Est une bibliothèque Python utilisée pour créer des graphiques et des tracés 2D à l'aide de scripts Python. Bien que le concurrent le plus coriace de Python, R, soit meilleur, mais comme nous avons choisi python pour les tâches ML, nous utiliserons matplotlib pour la visualisation des données.[40]

2.4. Opencv

OpenCV a été lancé chez Intel en 1999 par **Gary Bradsky**. La première version est arrivée un peu plus tard en l'an 2000. OpenCV signifie essentiellement **Open Source Computer Vision Library**. Bien qu'il soit écrit en C/C++ optimisé, il possède des interfaces pour Python et Java

ainsi que C++. OpenCV se vante d'une base d'utilisateurs actifs dans le monde entier, son utilisation augmentant de jour en jour en raison de l'essor des applications de vision par ordinateur.

OpenCV-Python est l'API python pour OpenCV. Vous pouvez le considérer comme un wrapper python autour de l'implémentation C++ d'OpenCV. OpenCV-Python est non seulement rapide (puisque l'arrière-plan est constitué de code écrit en C/C++) mais est également facile à coder et à déployer (grâce au wrapper Python au premier plan). Cela en fait un excellent choix pour exécuter des programmes à forte intensité de calcul.[41]

2.5. TensorFlow

La principale bibliothèque Open Source pour le développement et l'entraînement de modèles de machine Learning.

TensorFlow est une plate-forme Open Source dédiée à la machine Learning. Elle propose un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs d'avancer dans le domaine du machine Learning, et aux développeurs de créer et de déployer facilement des applications qui exploitent cette technologie, aussi construisez et entraînez facilement des modèles de machine Learning à l'aide d'API intuitives de haut niveau comme Keras, ce qui permet une itération immédiate des modèles et un débogage facile.[42]

2.6. Keras

Keras est une bibliothèque de réseaux neuronaux de haut niveau, écrite en Python et capable de s'exécuter sur TensorFlow ou Theano. Il a été développé dans le but de permettre une expérimentation rapide. Pouvoir passer de l'idée au résultat avec le moins de retard possible est la clé d'une bonne recherche.

Utilisez Keras si vous avez besoin d'une bibliothèque d'apprentissage en profondeur qui :

- Permet un prototypage facile et rapide (grâce à une modularité totale, un minimalisme et une extensibilité).
- Prend en charge à la fois les réseaux convolutifs et les réseaux récurrents, ainsi que les combinaisons des deux.
- Prend en charge les schémas de connectivité arbitraires (y compris la formation multi-entrées et multi-sorties).
- Fonctionne de manière transparente sur le CPU et le GPU.[43]

2.7. Dlib

Dlib est une boîte à outils C++ moderne contenant des algorithmes d'apprentissage automatique et des outils permettant de créer des logiciels complexes en C++ pour résoudre

des problèmes du monde réel. Il est utilisé à la fois dans l'industrie et dans les universités dans un large éventail de domaines, notamment la robotique, les appareils embarqués, les téléphones mobiles et les grands environnements informatiques haute performance. La licence open source de Dlib vous permet de l'utiliser dans n'importe quelle application, gratuitement.[44]

2.8. Streamlit

Streamlit est une bibliothèque Python open source qui facilite la création et le partage de belles applications Web personnalisées pour l'apprentissage automatique et la science des données. En quelques minutes seulement, vous pouvez créer et déployer de puissantes applications de données.[45]

2.9. Angular

Angular est un Framework côté client, open source, basé sur TypeScript, et codirigé par l'équipe du projet « Angular » à Google et par une communauté de particuliers et de sociétés. Angular est une réécriture complète d'AngularJS. Il permet la création d'applications Web et plus particulièrement de ce qu'on appelle des applications web accessibles via une page web unique qui permet de fluidifier l'expérience utilisateur et d'éviter les chargements de pages à chaque nouvelle action. Le Framework est basé sur une architecture du type MVC et permet donc de séparer les données, le visuel et les actions pour une meilleure gestion des responsabilités. Un type d'architecture qui a largement fait ses preuves et qui permet une forte maintenabilité et une amélioration du travail collaboratif. [46]

3. Reconnaissance faciale

3.1. Notions utilisées

Transfert learning :

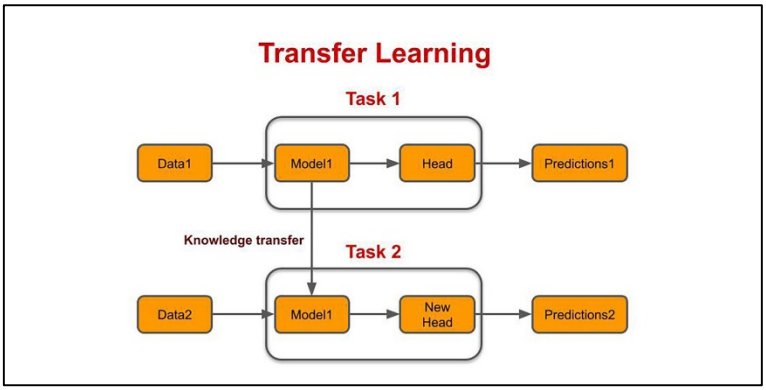

Figure 32 : Apprentissage par Transfert

L'apprentissage par transfert consiste à prendre des caractéristiques apprises sur un problème et à les exploiter sur un nouveau problème similaire.

L'apprentissage par transfert est généralement effectué pour les tâches pour lesquelles votre ensemble de données contient trop peu de données pour former un modèle à grande échelle à partir de zéro.

L'incarnation la plus courante de l'apprentissage par transfert dans le contexte de l'apprentissage en profondeur est le workflow suivant :

- Prenez des couches d'un modèle préalablement formé.
- Congelez-les, afin d'éviter de détruire les informations qu'ils contiennent lors des futurs tours d'entraînement.
- Ajoutez de nouvelles couches pouvant être entraînées au-dessus des couches gelées. Ils apprendront à transformer les anciennes caractéristiques en prédictions sur un nouvel ensemble de données.
- Entraînez les nouvelles couches sur votre jeu de données.[47]

Augmentation Des Données :

Figure 33 : Exemple D'augmentation d'image

L'augmentation d'image est une technique consistant à appliquer différentes transformations aux images originales, ce qui donne lieu à plusieurs copies transformées de la même image. Chaque copie, cependant, est différente de l'autre dans certains aspects selon les techniques d'augmentation que vous appliquez comme le décalage, la rotation, le retournement, etc.

L'application de ces petites quantités de variations sur l'image d'origine ne change pas sa classe cible mais fournit seulement une nouvelle perspective de capture de l'objet dans la vie réelle. Et donc, nous l'utilisons assez souvent pour construire des modèles d'apprentissage en profondeur.[48]

Alignement de visage :

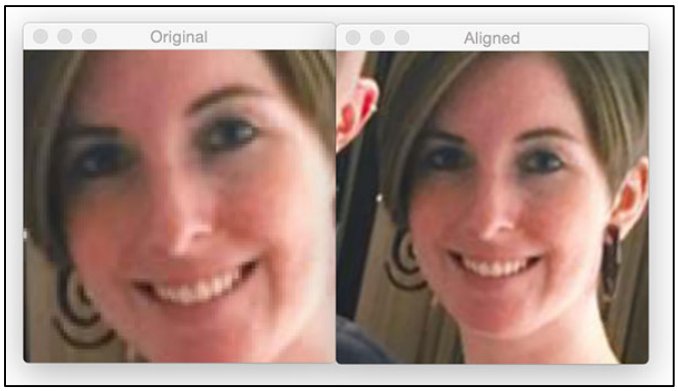

Figure 34: Exemple d'alignement de visage

Alignement de visage est une technique de normalisation, souvent utilisée pour améliorer la précision des algorithmes de reconnaissance faciale, y compris les modèles d'apprentissage en profondeur.

Le processus d'alignement facial modifie les visages de la base de données de manière à être :

- centrés dans l'image.
- tournés de sorte que les yeux se trouvent sur une ligne horizontale (c'est-à-dire que le visage est tourné de telle sorte que les yeux se trouvent le long des mêmes coordonnées y).
- mis à l'échelle de telle sorte que la taille des visages soit approximativement identique.

3.2. La Détection de visage

3.2.1. Comparaison entre les détecteurs de visage sur plusieurs niveaux

Afin de choisir le détecteur de visage convenable nous avons fait une étude comparative entre les détecteurs de visage les plus utilisés Haar Cascade, DNN, HOG, CNN et MTCNN.

3.3. Ensemble de données

La comparaison a été faite par rapport à une base de données d'images que nous avons construit en utilisant des images de différents types (images frontales et non frontales, images de visages avec bavette, avec lunettes, etc.). La base de données contient 500 images.

3.4. Précision de Détection

HOG a une précision inférieure à celle de Haar, bien que les résultats visuellement soient meilleurs dans HOG. CNN a eu une meilleure performance par rapport à HOG. MTCNN a également obtenu une bonne précision (83.8%), bien meilleure que celle de CNN (73.8%). DNN trône la course avec 98% de précision, dépassant ainsi le modèle MTCNN de plus de 14%.

```
MTCNN Accuracy: 0.838
DLIB CNN Accuracy: 0.738
DLIB HOG Accuracy: 0.562
OPENCV DNN Accuracy: 0.980
OPENCV HAAR CASCADE Accuracy: 0.694
```

Figure 35 : Précision pour chaque détecteur de visage

3.5. Détection à l'échelle

Nous notons que le DNN et MTCNN détectent tous les visages, tandis que HOG et CNN ne détecte que les visages plus grands. Nous indiquons à la fois la taille et la boîte englobante du visage détecté. CNN et HOG peuvent être utilisées pour détecter des faces d'échelle allant jusqu'à 80x80, quand le visage est en dessous de cette taille, ils ont du mal à le détecté.

*Figure 36 : Exemple de Détection de visage de petite taille (25*50)*

*Figure 37 : Exemple de Détection de visage de petite taille (55*60)*

3.6. Détection de visage Non Frontale

La détection d'images non frontales a connue plusieurs difficultés, et dans la littérature, plusieurs méthodes de détection ont été destinées aux visages frontaux, vu le défi existant pour pouvoir détecter des images de visages non frontales.

Nous tenons à préciser que nous parlons de visage non frontal quand la personne ne regarde pas droit devant elle.

Figure 38 : Exemple de Détection de visage non frontal, regard vers la gauche

Figure 39 : Exemple de Détection de visage non frontal, regard vers le bas

Figure 40 : Exemple de Détection de visage non frontal, regard vers la droite

L'une des limites de la méthode HaarCascade est qu'elle n'arrive pas à détecter des visages frontaux, ce qui explique qu'elle n'arrive à détecter aucune des images dans cette section. Et bien que HOG ne tolère pas bien les changements de rotation ou d'angle de vue, il arrive parfois à faire des exceptions, comme dans la figure 36. Entre CNN, MTCNN et DNN, le meilleur détecteur est sans surprise DNN avec le meilleur rapport précision/vitesse.

3.7. Détection sous Occlusion

Le port des masques en raison de la crise sanitaire nous a poussé à vouloir tester la robustesse de nos différents modèles de détection en cas d'occlusion. Nous avons donc nous avons calculé la précision des cinq techniques sur un ensemble de donnés qui contient 500 images de visages avec occlusion, à savoir des visages avec masque.

Figure 41: Détection de visage sous une occlusion

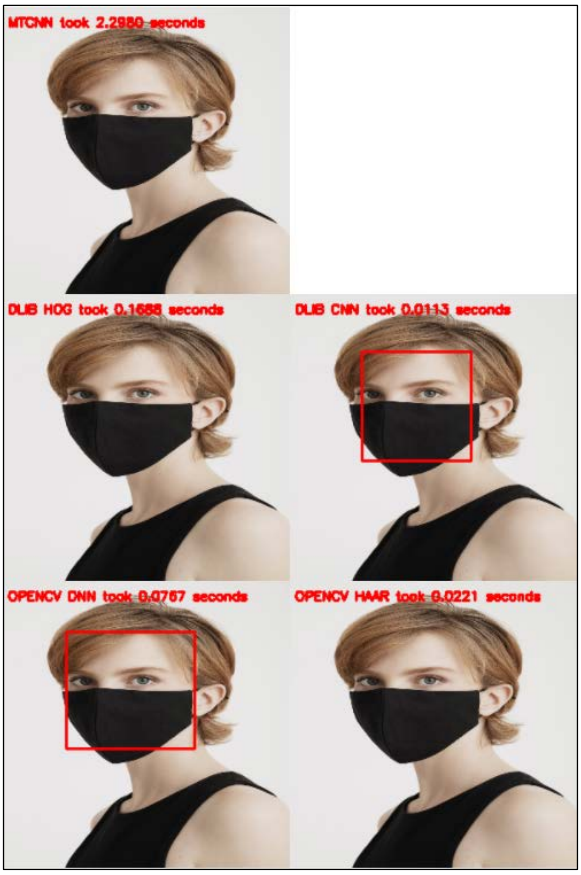

Figure 42 : Détection de visage avec port du masque

Ici nous remarquons que HOG l'emporte sur la méthode HaarCascade avec une faible différence de 1.6%, CNN vient encore en 3ème place avec une précision de 20.4%, suivie de la méthode MTCNN avec une précision de 44.6%. Et encore une fois le trône va à la méthode DNN avec la meilleure précision de 96.6% dépassant ainsi MTCNN avec 52%.

```
MTCNN Accuracy: 0.446
DLIB CNN Accuracy: 0.204
DLIB HOG Accuracy: 0.048
OPENCV DNN Accuracy: 0.966
OPENCV HAAR CASCADE Accuracy: 0.032
```

Figure 43 : Comparaison des différentes méthodes de détection en cas de détection de visage avec occlusion

3.8. Vitesse de Détection

Nous avons comparé nos méthodes en terme de précision, mais également en terme de rapidité, nous avons donc calculé le temps de performance de chacune des méthodes durant la détection d'une image sur CPU et sur GPU. La méthode la plus rapide était cascade de Haar suivi DNN, HOG et CNN quand on travaille sur un environnement avec GPU et le plus lent c'est MTCNN.

```
MTCNN took 1.6285 seconds
DLIB HOG took 0.1193 seconds
DLIB CNN took 0.0078 seconds
OPENCV DNN took 0.0784 seconds
OPENCV HAAR took 0.0380 seconds
```

Figure 44 : le temps de détection pour chaque détecteur de visage sur GPU

```
MTCNN took 1.7117 seconds
DLIB HOG took 0.2360 seconds
DLIB CNN took 3.1264 seconds
OPENCV DNN took 0.0759 seconds
OPENCV HAAR took 0.0440 seconds
```

Figure 45 : le temps de détection pour chaque détecteur de visage sur CPU

3.2.2. Discussion des résultats

La méthode de détection DNN a eu les meilleurs résultats en rapport précision/vitesse, puisque dans les différentes conditions elle a eu la meilleure précision, et une vitesse acceptable dans l'ensemble. Il est certes plus lent que la cascade Haar, mais il est beaucoup plus précis et fonctionne bien dans un large éventail de conditions. Concernant la méthode CNN, elle a également une meilleure précision que Haar Cascade, mais est beaucoup trop lente, pas autant que MTCNN qui s'est aussi démarqué par rapport à la précision de sa détection.

Dans la littérature, les précisions des deux méthodes CNN et HOG sont meilleurs que ceux sur dans notre cas, il faut préciser donc que pour qu'ils aient une meilleure performance, la taille du visage doit être égale ou supérieure à 80 × 80.

Ci-dessous, un tableau contenant une comparaison des cinq méthodes en allant de la première jusqu'à la cinquième par rapport aux critères évoqués ci-dessous.

Critère	Premier	Deuxième	Troisième	Quatrième	Cinquième
Précision	DNN	MTCNN	CNN	Haar Cascade	HOG
Détection sur différentes échelles	DNN	MTCNN	Haar Cascade	HOG	CNN
Détection de visage non frontal	DNN	MTCNN	CNN	HOG	Haar Cascade
Détection sous une occlusion	DNN	MTCNN	CNN	HOG	Haar Cascade
Vitesse	Haar Cascade	DNN	HOG	MTCNN	CNN

Tableau 1: Conclusion de notre étude comparative entre les détecteurs de visage

3.3. L'Extraction Des Caractéristiques

3.3.1. Comparaison entre les modèles de reconnaissance faciale

Dans cette étape, nous avons mis en œuvre de nombreux modèles d'apprentissage profond tels que : CNN, AlexNet, VGG16, FaceNet, VGGFace. Afin de comparer les différences entre eux en termes de performance et de stabilité et de choisir le meilleur de ces modèles comme modèle principal de prédiction pour notre système proposé.

Ensemble de données

Ensemble de données que nous avons utilisé dans notre étude est un jeu de données que nous avons reçus de notre organisme d'accueil, il se constitue de 2950 images.

La perte

Une fonction de perte est utilisée pour optimiser un algorithme d'apprentissage automatique . La perte est calculée sur la formation et la validation et son interprétation est basée sur la performance du modèle dans ces deux ensembles. Il s'agit de la somme des erreurs commises pour chaque exemple dans les ensembles d'apprentissage ou de validation. La valeur de perte implique à quel point un modèle se comporte mal ou bien après chaque itération d'optimisation.

La précision

Une métrique de précision est utilisée pour mesurer les performances de l'algorithme de manière interprétable. La précision d'un modèle est généralement déterminée après les paramètres du modèle et est calculée sous la forme d'un pourcentage. C'est la mesure de la précision de la prédiction de votre modèle par rapport aux données réelles.

Par exemple supposons que nous avons 1000 échantillons de test et que notre modèle soit capable d'en classer 990 correctement, la précision du modèle sera de 99,0 %.

CNN

Nous avons utilisé un modèle CNN avec 5 couches de convolutions après chaque couche de convolution on a une couche de max_pooling et à la fin le réseau fully connected avec une couche cachée.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 96, 96, 16)	160
max_pooling2d (MaxPooling2D)	(None, 48, 48, 16)	0
conv2d_1 (Conv2D)	(None, 48, 48, 32)	4640
max_pooling2d_1 (MaxPooling2	(None, 24, 24, 32)	0
conv2d_2 (Conv2D)	(None, 24, 24, 64)	18496
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 64)	0
conv2d_3 (Conv2D)	(None, 12, 12, 128)	73856
max_pooling2d_3 (MaxPooling2	(None, 6, 6, 128)	0
conv2d_4 (Conv2D)	(None, 6, 6, 256)	295168
max_pooling2d_4 (MaxPooling2	(None, 3, 3, 256)	0
flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 512)	1180160
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 30)	15390

Figure 46 : Architecture CNN Utilisée

Nous avons entrainé ce modèle sur notre ensemble de données et nous avons obtenu comme résultats Figure 45 :

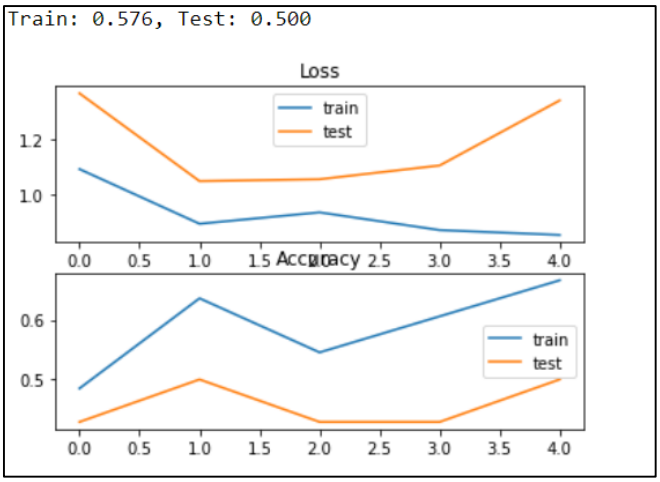

Figure 47 : Précision de CNN

La perte (loss) s'agit d'une somme des erreurs commises pour chaque exemple dans les ensembles d'apprentissage ou de validation donc pour CNN nous avons trouvé que pour l'apprentissage 57% de l'ensemble de données qui sont bien entrainé et pour le test ou bien la validation le modèle a pu savoir ou bien prédire juste 50% de l'ensemble de données.

AlexNet

Le modèle AlexNet se constitue de 5 couches de convolutions et une combinaison de couche de max_pooling et à la fin le réseau fully connected avec deux couches cachées.

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 96, 96, 96)	11712
max_pooling2d (MaxPooling2D)	(None, 48, 48, 96)	0
conv2d_1 (Conv2D)	(None, 48, 48, 256)	614656
max_pooling2d_1 (MaxPooling2	(None, 24, 24, 256)	0
conv2d_2 (Conv2D)	(None, 24, 24, 384)	885120
conv2d_3 (Conv2D)	(None, 24, 24, 384)	1327488
conv2d_4 (Conv2D)	(None, 24, 24, 256)	884992
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 256)	0
flatten (Flatten)	(None, 36864)	0
dense (Dense)	(None, 512)	18874880
dropout (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 512)	262656
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 30)	15390
dropout_2 (Dropout)	(None, 30)	0
dense_3 (Dense)	(None, 30)	930

Figure 48 : Architecture d'AlexNet

Nous avons entrainé ce modèle sur notre ensemble de données et nous avons obtenu comme résultats Figure 47 :

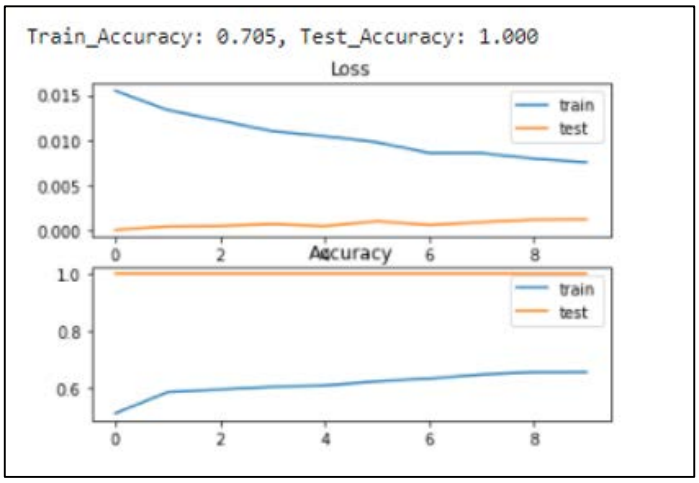

Figure 49 : Précision d'AlexNet

Pour AlexNet nous avons trouvé que pour l'apprentissage 70% de l'ensemble de données qui sont bien entrainé et pour le test le modèle a pu prédire 100% de l'ensemble de données.

VGG16

Le modèle VGG16 se constitue d'une combinaison de couches de convolution et de max_pooling sous forme de blocs.

Model: "vgg16"

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0

Figure 50: Architecture de VGG 16

Le modèle VGG-16 que nous avons importé est celui de Keras, c'est un modèle pré-entrainé, les poids sont donc ceux de la base de données ImageNet

```
from keras.applications import vgg16

img_rows, img_cols = 224, 224
model = vgg16.VGG16(weights = 'imagenet',
                    include_top = False,
                    input_shape = (img_rows, img_cols, 3))
```

Figure 51

Les images de notre base de données sont tous frontales, pour mieux entrainer notre modèle, nous appliquons la DataAugmentation, ceci a pour but qu'il soit prêt à reconnaitre une

personne dans différentes positions. La figure 50 montre la DataAugmentation que nous avons effectué pour les images de test et ceux de validation.

```
train_datagen = ImageDataGenerator(
    rescale = 1./255,
    rotation_range = 45,
    width_shift_range = 0.3,
    height_shift_range = 0.3,
    horizontal_flip = True,
    fill_mode = 'nearest')

validation_datagen = ImageDataGenerator(rescale = 1./255)
```

Figure 52 : Représentation de Data Augmentation

Après ceci, nous créons un générateur de données avec l'augmentation que nous avons effectué.

```
batch_size = 16

train_generator = train_datagen.flow_from_directory(
    train_data_dir,
    target_size = (img_rows, img_cols),
    batch_size = batch_size,
    class_mode = 'categorical')

validation_generator = validation_datagen.flow_from_directory(
    validation_data_dir,
    target_size = (img_rows, img_cols),
    batch_size = batch_size,
    class_mode = 'categorical')
```

Figure 53 : Générateur de données.

- Batch_size définit le nombre d'échantillon qui vont être à la fois propagés dans le réseau.
- Class_mode = 'categorical' veut dire une sortie 2D, ce qui signifie que sortie multi-étiquette est supportée.

Et pour finir nous créons un modèle et nous executons model.fit_generator, étant une bibliothèque d'apprentissage profond utilisée pour entraîner notre modèle.

```
history = model.fit_generator(
    train_generator,
    epochs = epochs,
    validation_data = validation_generator)
```

Figure 54 : Le modèle créé pour exécuter model.fil_generator

Epoch : définit le nombre de fois que l'algorithme d'apprentissage travaillera sur l'ensemble des données d'apprentissage. Un epoch signifie que chaque échantillon de l'ensemble de données d'apprentissage a eu l'occasion de mettre à jour les paramètres du modèle interne.[43]

Nous avons entrainé ce modèle sur notre ensemble de données et nous avons obtenu comme résultats Figure 53 :

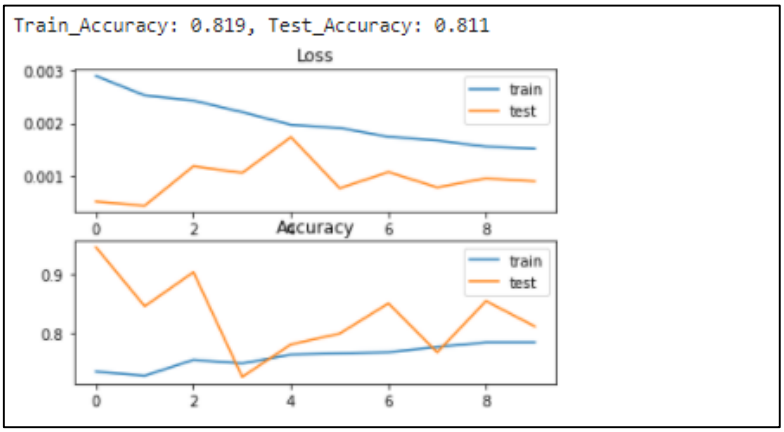

Figure 55 : Précision de VGG16

Pour VGG16 nous avons trouvé que pour l'apprentissage 82% de l'ensemble de données qui sont bien entrainé et pour le test le modèle a pu prédire 81% de l'ensemble de données.

Les résultats que ce soit pour AlexNet ou VGG-16 paraissent plutôt satisfaisants, sauf que le type d'image dans la base de données ne nécessite pas un grand effort, nous avons donc essayé de faire des tests sur nos propres images, les modèles n'arrivaient pas toujours à nous reconnaitre dans des images différentes, et parfois nous confondaient.

Figure 56 : images de notre base de données

FACENET et VGGFACE

FaceNet et VGGFace en une similarité, c'est qu'ils « encodent » le visage, le premier modèle en un vecteur de 128 chiffres, et le deuxième en une présentation de 2048 chiffres.

Chacun de ses modèle utilise une architecture, pour VGGFace, nous avons optez pour l'architecture ResNET-50, et pour FaceNet c'est l'architecture Inception.

Nous avons utilisé ces deux modèles pré-entrainés, et avons eu les résultats suivants :

Modèle	Précision
FaceNet	98.96 %
VGGFACE	99.63 %

Tableau 2 : Précision de FaceNet et VGGFACE

3.3.2. Discussion des résultats

Nous avons commencé par former un réseau CNN de 5 couches pour d'abord comprendre comment nous pouvons entrainer une simple architecture CNN, et par curiosité afin de voir à quel point il pourra être performant, tout en sachant qu'il était non mesurable face aux autres modèles et architectures.

FaceNet et VGGFace utilisent des architectures de loin plus performantes que VGG-16 et AlexNet. Ceci a été d'ailleurs expliqué dans la littérature et nous avons pu le constater en faisant notre étude.

En ce qui concerne la comparaison entre le modèle FaceNet et VGGFace, nos résultats n'ont fait qu'appuyer ceux de la littérature. Le modèle VGGFace l'emporte alors avec une meilleure précision de 99,63%.

3.4. Approche Proposée

Nous nous sommes basées sur ce qui précède, et nous avons sélectionnés les méthodes avec les meilleures performances pour le système que nous proposons

- La détection de visage en utilisant le détecteur de visage DNN
- Alignement de visage pour augmenter la précision du modèle
- L'extraction des caractéristiques en utilisant le modèle de VGGFACE

Nous pouvons résumer notre système de reconnaissance faciale sur le schéma suivant :

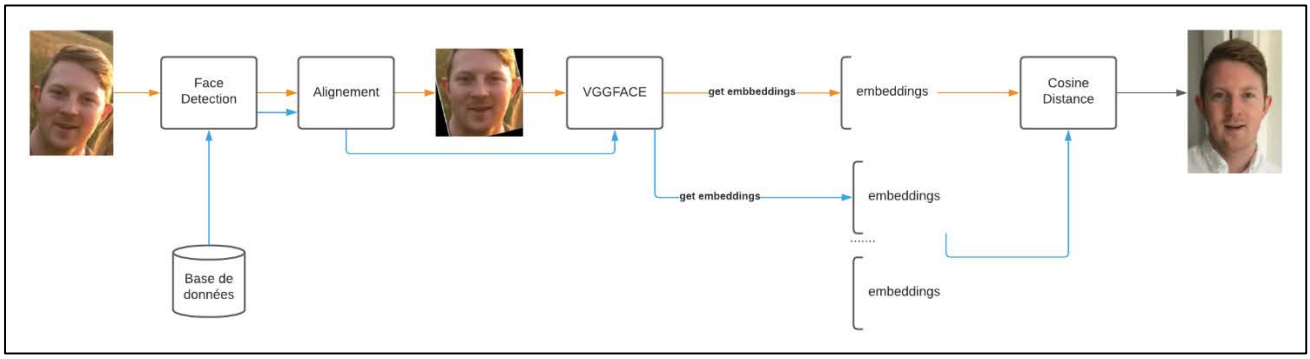

Figure 57 : Schéma de l'approche proposée

L'alignement de visage est une étape cruciale dans la plupart des systèmes d'analyse de visages puisqu'il facilite la détection des caractéristiques et du coup facilite la reconnaissance du visage.

Pour ce faire, il faut d'abord déterminer les points clés appelés points de repère du visage (68 points) avec le prédicteur de la forme de Dlib : '*shape_predictor_68_face_landmarks.dat*' qui permet de prédire la forme ou bien la pose de visage

Figure 58 : Les 68 points clés de visage

Une fois les 68 points de repère sont extraits, chaque partie du visage a été marquée, après quoi nous avons utilisé la fonction d'alignement dans package « imutils » pour aligner le visage avant de l'enregistrer.

3.5. Application

Afin de tester notre approche et de visualiser les résultats, et en attendant de pouvoir intégrer notre code dans l'application « Doccerts », nous avons créé une interface pour faciliter l'implémentation avec la bibliothèque Python Streamlit.

L'interface qui s'affiche après le démarrage de l'application contient un lien pour charger l'image, un select box pour choisir l'opération souhaitée à savoir la détection du visage ou l'identification du visage et un bouton Process pour appliquer l'opération souhaitée.

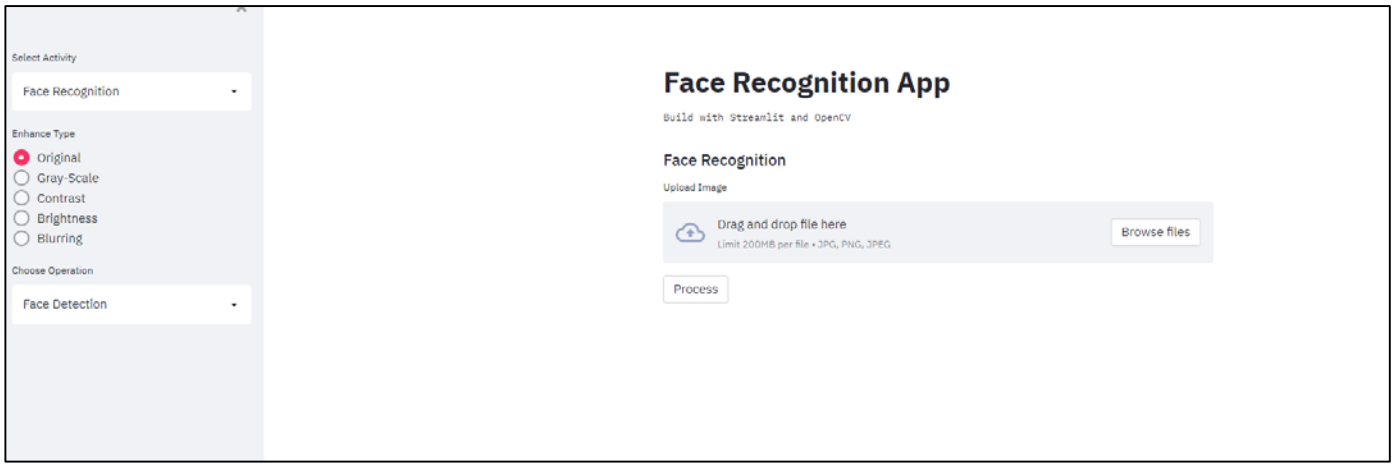

Figure 59 : structure de l'application

Après le chargement de l'image, nous avons le choix entre procéder à une détection ou une vérification du visage.

En cas de vérification, si l'image entrée correspond à une déjà existante dans la base de donnée, le système propose sa modification avec la nouvelle image entrée, sinon, si elle ne correspond à aucune image existante, elle s'enregistre dans la base de données.

La détection se fait avec le détecteur DNN. Nous obtenons l'image résultante de cette opération et un message de confirmation, comme le montre la figure 61.

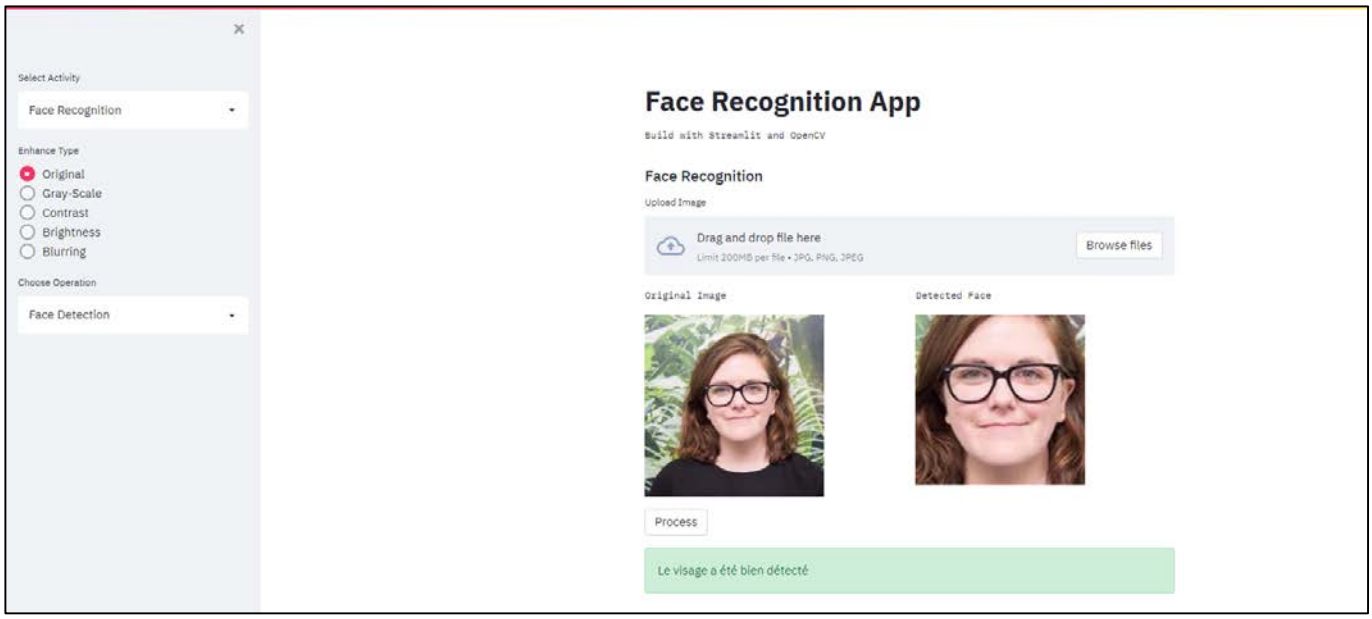

Figure 60: la détection de visage

Dans la figure 62, après le chargement de l'image, le système nous montre l'image qui lui correspond, et procède à sa modification.

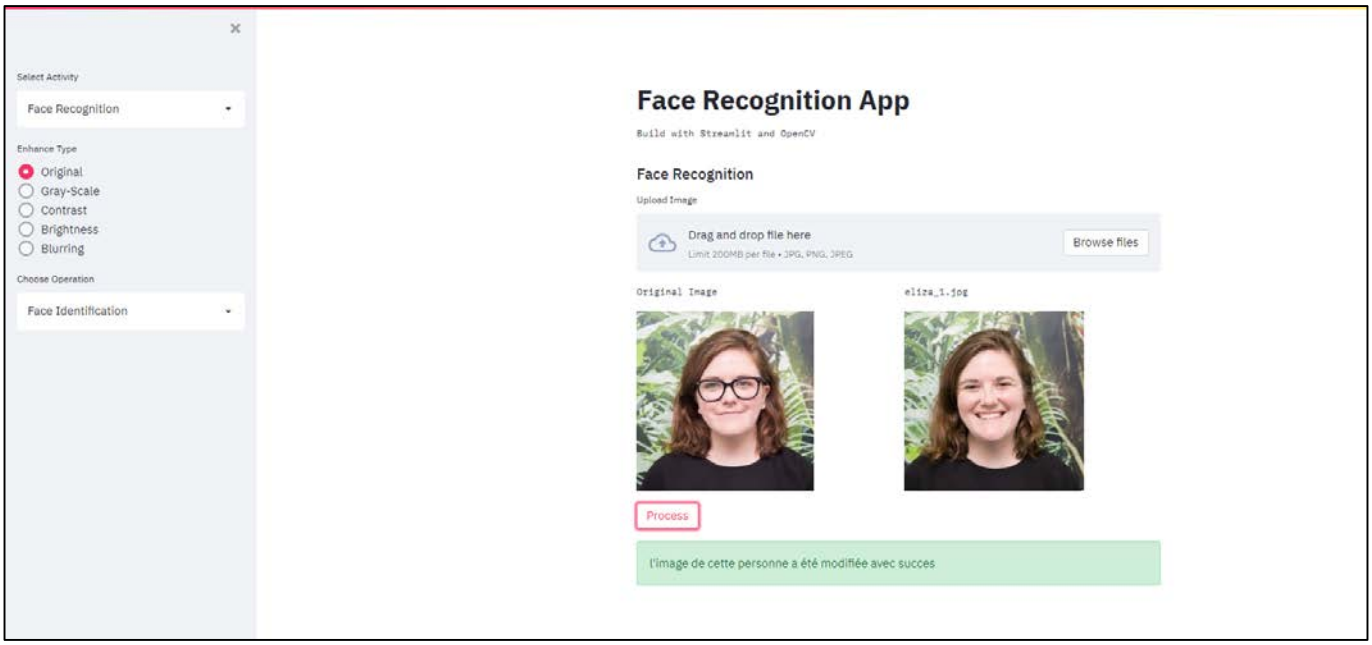

Figure 61 : L'identification du visage existant

L'image entrée dans la figure 63 ne correspond à aucune autre existante dans la base de données, dans ce cas l'image y est enregistrée.

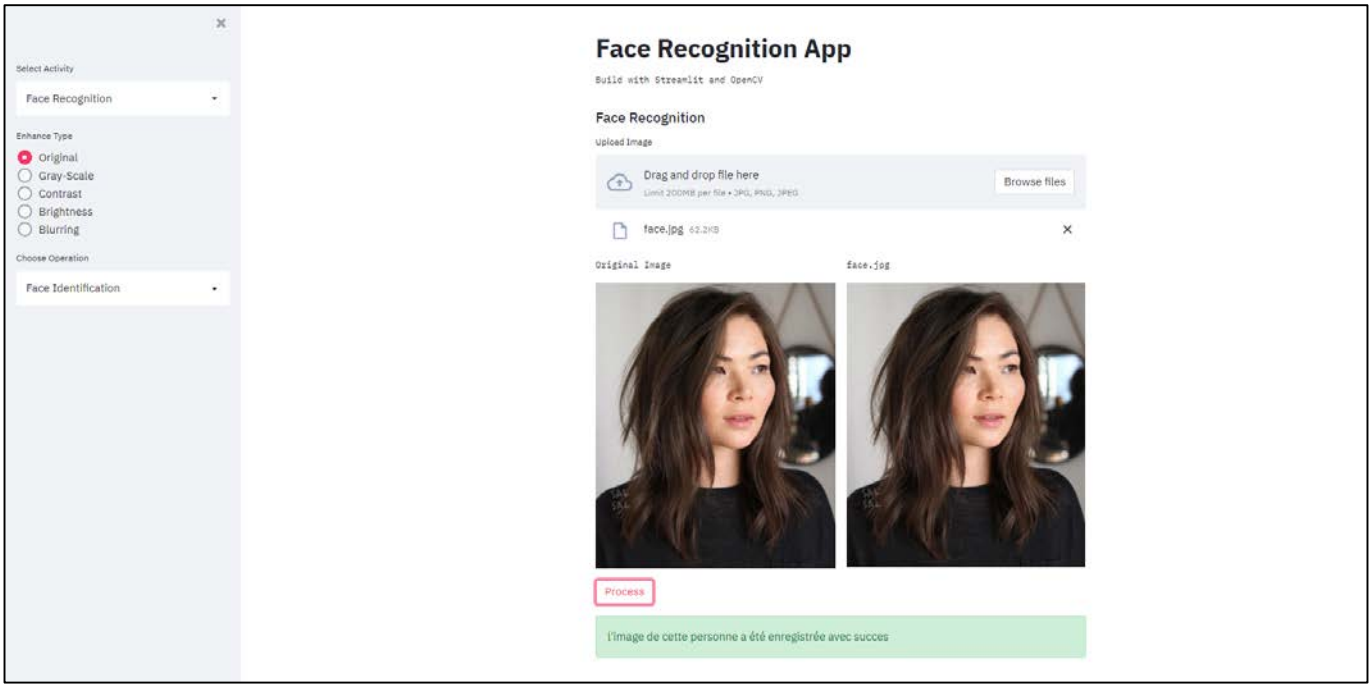

Figure 62 : L'identification du visage non existant

4. Application « Doccerts »

Comme nous l'avons mentionné, en plus de ce qui précède nous avons également participé au développement de la partie front-end l'application avec le framework Angular. Nous avons eu comme tâches d'abord la création d'un système de messagerie, puis la récupération des

informations de l'utilisateur, et leurs affichage dans son profile tout en lui donnant la possibilité de les modifier.

4.1. Système de messagerie

Afin d'assurer une communication entre les acteurs de l'application « Doccerts » nous avons créé une boite email ou un système de messagerie qui répond à ce besoin. Chaque utilisateur peut utiliser ce système pour envoyer, recevoir des messages à d'autres utilisateurs de l'application, la figure 63 représente le diagramme de cas d'utilisation de ce système.

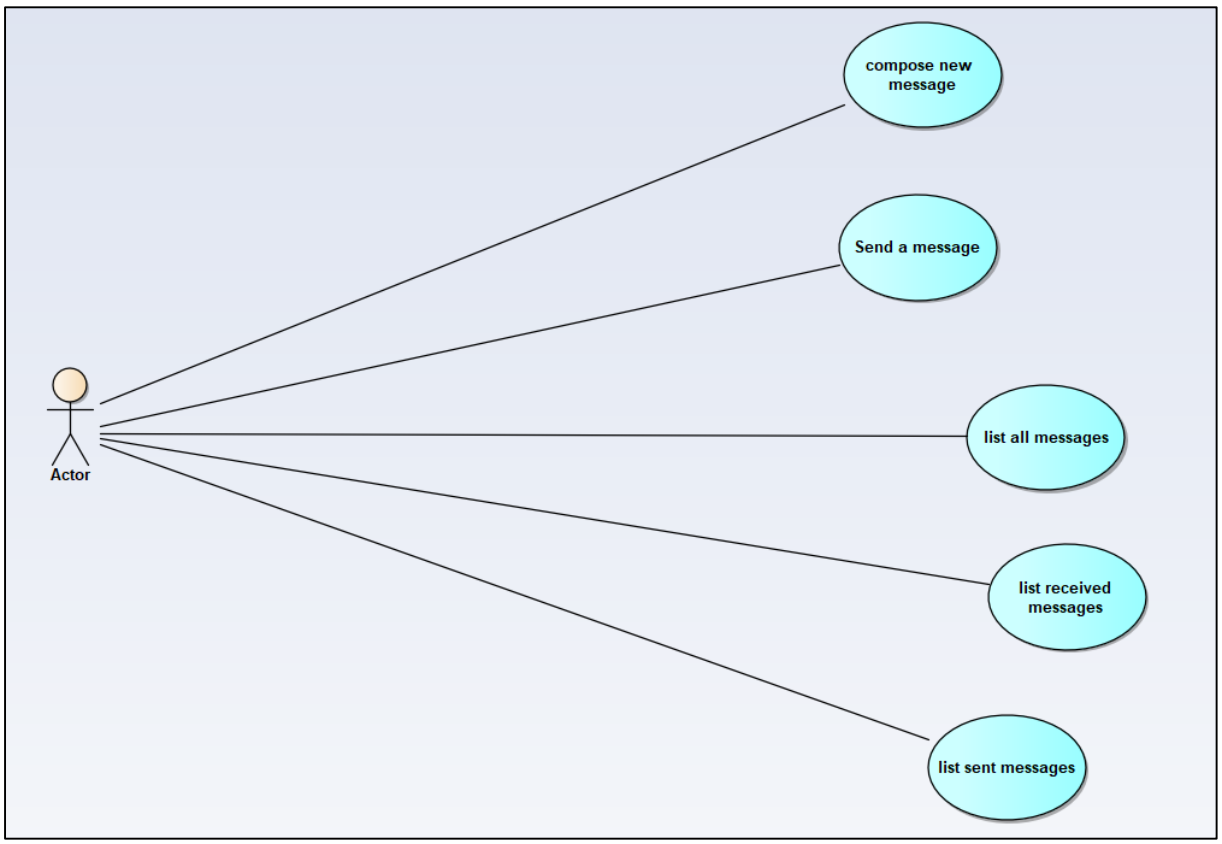

Figure 63 : Diagramme de cas d'utilisation du système de messagerie.

Les interfaces suivantes montrent les fonctionnalités de ce système

Dans cette interface l'utilisateur peut lister ses emails reçus, envoyés ou bien les deux avec un ordre tel que le premier email qui s'affiche est le plus récent est peut aussi créer un nouveau email.

Figure 64 : Interface de système de messagerie.

Après que l'utilisateur clique sur le bouton *new message* la boite de création de l'email s'ouvre comme il est montré dans la figure suivante avec la vérification des champs de saisie.

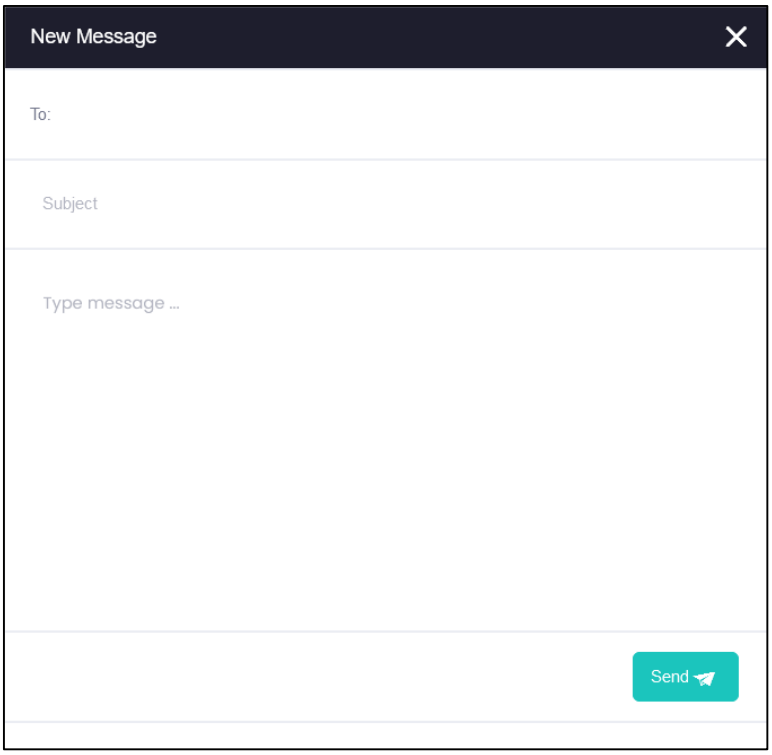

Figure 65 : La boite d'envoi de message

4.2. Profil de l'utilisateur

Concernant le profil de l'utilisateur, ce dernier doit être capable d'accéder à ses informations et de les modifier, à savoir ses informations personnelles (nom, prénom, date de naissance, etc.), son mot de passe, le mode de réception des notifications et quelques paramètres de sécurité.

Il existe différents types d'utilisateur qui dans ce cas de figure ont les même options, sauf pour le « issuer » qui en plus a une identité blockchain. La figure 66 présente le diagramme de cas d'utilisation correspondant.

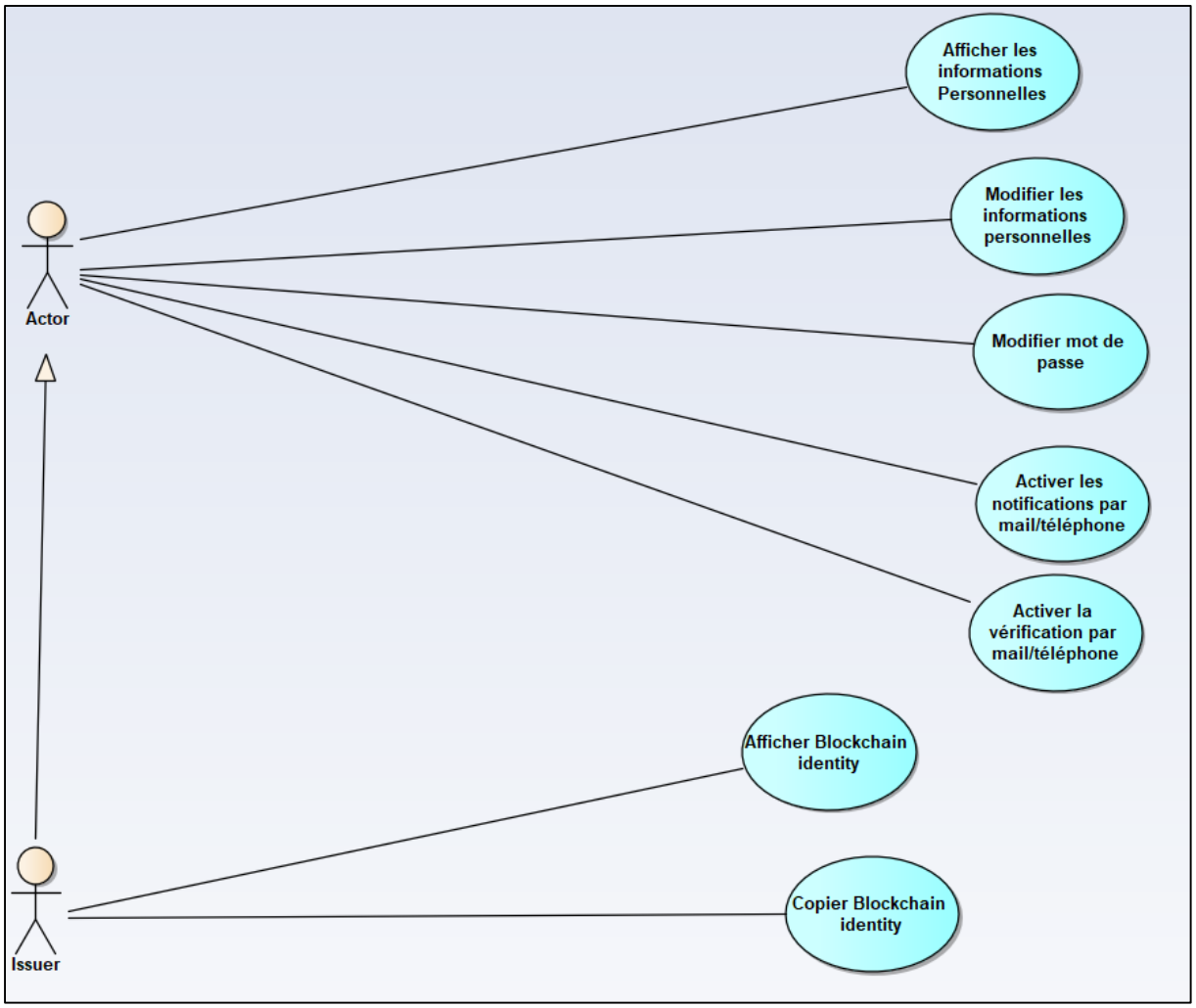

Figure 66 : Diagramme de cas d'utilisation de gestion du profil

En ce qui suit nous présentons les différentes figures d'interfaces du profil.

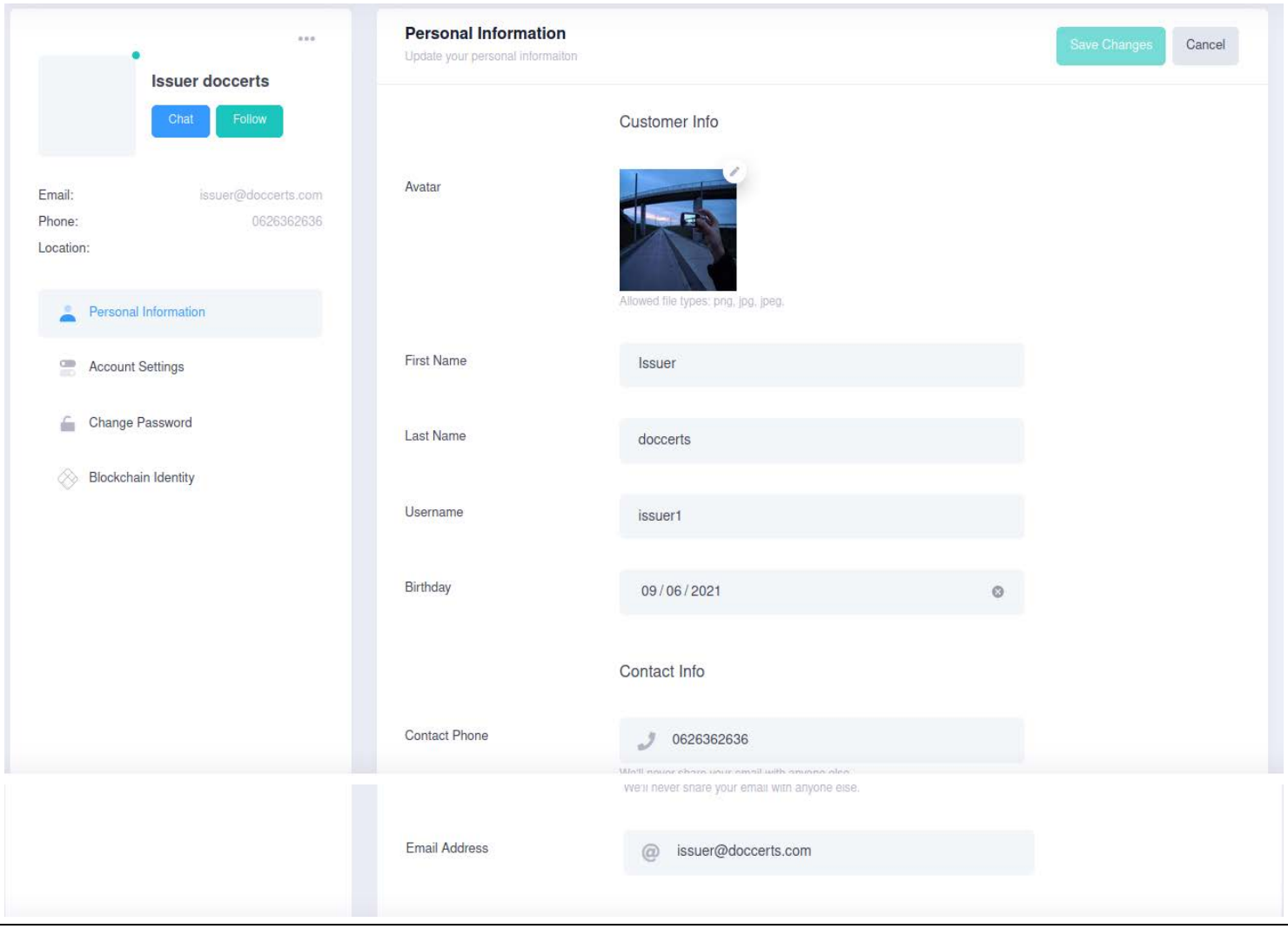

Figure 67 : Informations personnelles

Ici (figure 67) nous récupérons les informations personnelles de l'utilisateur à savoir son nom et prénom, sa date de naissance, son numéro de téléphone et son adresse e-mail.

L'utilisateur a le droit de modifier ces informations, il y a cependant un contrôle de saisi sur les champs. Il peut également changer sa photo de profil, et notre système de vérification du visage entre en jeu pour s'assurer que la nouvelle photo entrée appartient bien à la même personne sur l'image initiale.

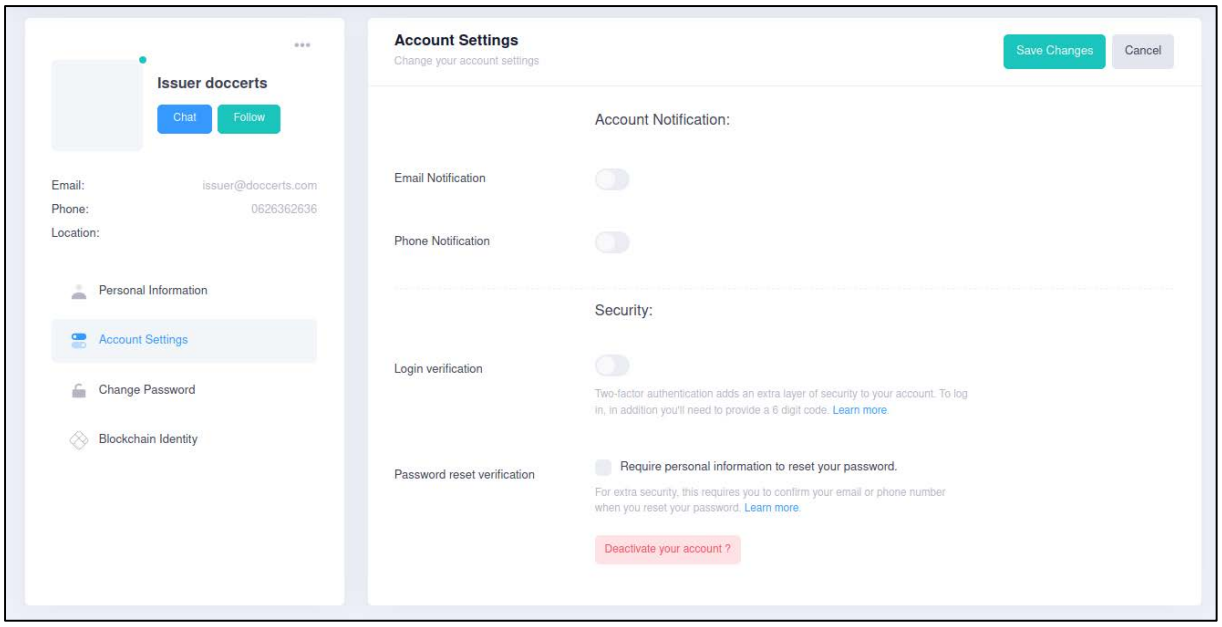

Figure 68 : Paramètres du compte

Les paramètres du compte se divisent en deux parties, les notifications du compte et la sécurité.

Concernant les notifications, l'utilisateur peut choisir recevoir les notifications de son compte sur email, sur téléphone ou peut activer les deux à la fois.

Pour ce qui l'en ai de la sécurité, il peut activer la vérification du Login, et la vérification de l'adresse e-mail ou du numéro de téléphone lors de la modification du mot de passe, et ce pour assurer plus de sécurité. L'utilisateur peut également désactiver son compte.

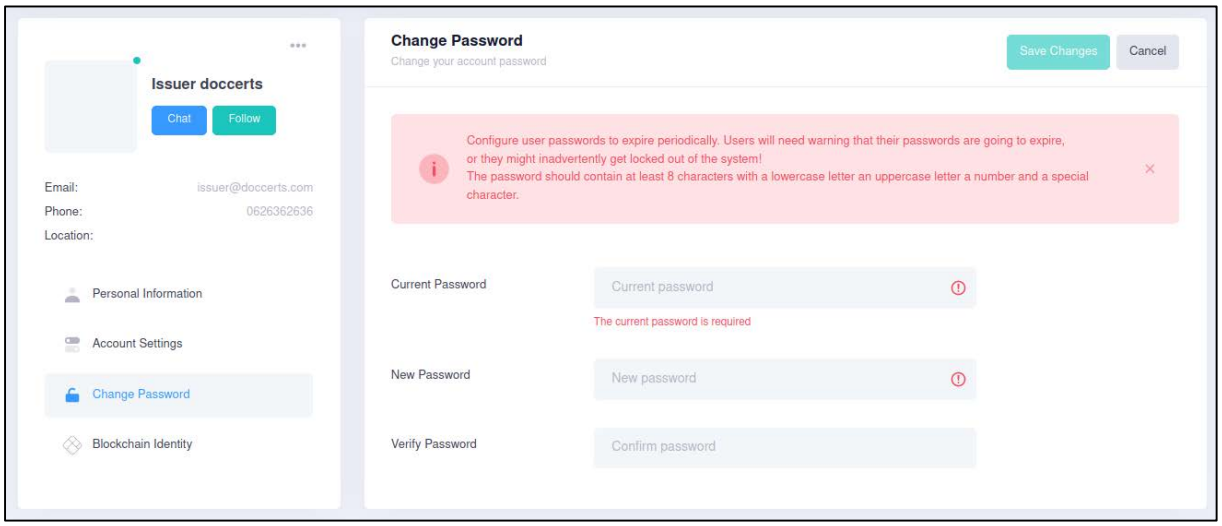

Figure 69 : Changement de mot de passe.

L'utilisateur peut naturellement modifier son mot de passe (figure 69), il y a des avertissements ; l'un l'avise de devoir changer son mot de passe périodiquement pour ne pas risquer le verrouillage du compte, et l'autre concerne les règles d'élaboration du mot de passe pour qu'il soit accepté par le système. Il y a d'ailleurs un contrôle de saisi sur les trois champs, celui qui vérifie le mot de passe actuel, celui qui contrôle la viabilité du nouveau mot de passe et pour finir celui de la vérification du nouveau mot de passe.

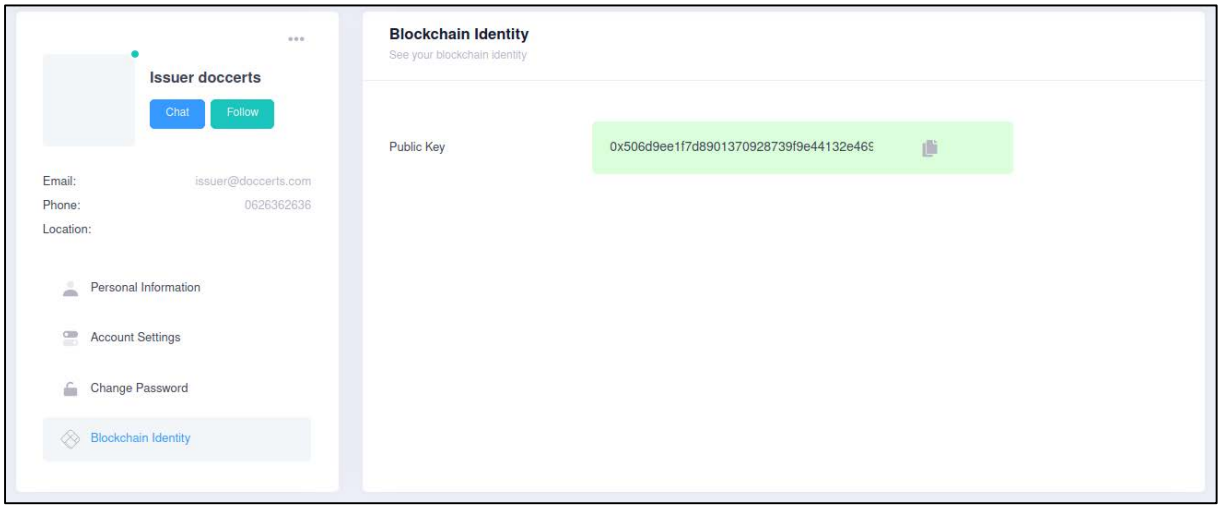

Figure 70 : Interface de l'identité blockchain.

La figure 70 nous montre une clé privée qui sert d'identité de Blockchain. Le champ de la clé publique n'est pas modifiable, nous pouvons cependant le copier en cliquant dessus. Quand nous passons la sourie sur le champ de la clé, sa couleur change en vert comme le montre la figure.

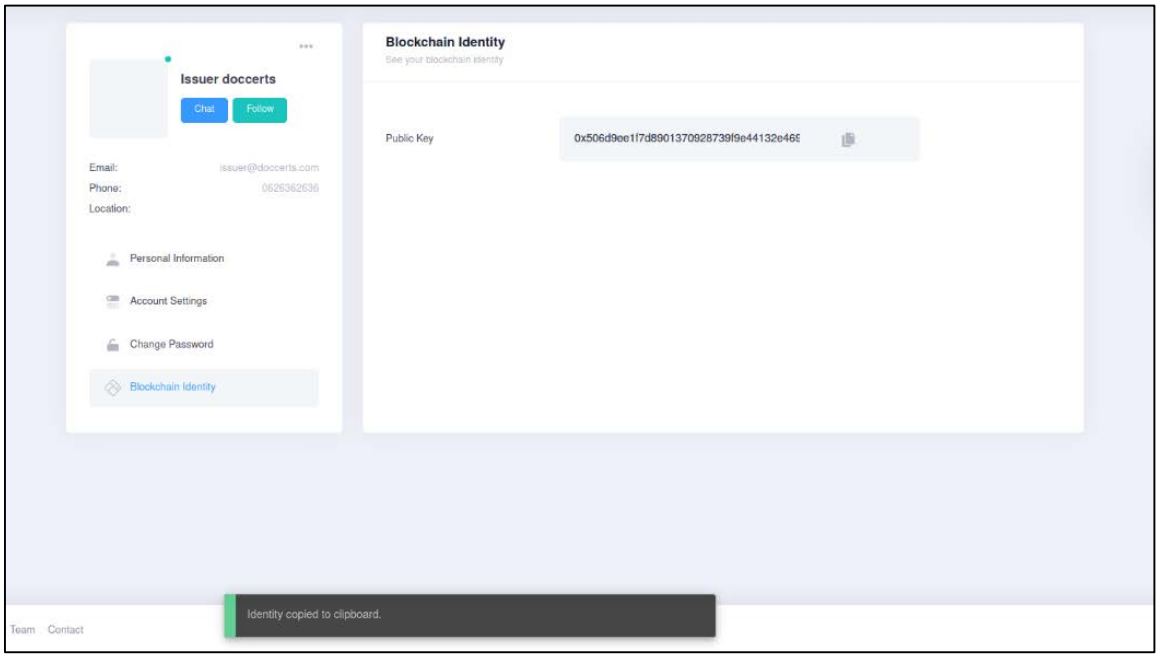

Figure 71 : Clé publique copiée.

En cliquant sur le champ contenant la clé publique, une notification de type « succès » nous informe que la clé a été copiée au presse-papier.

« Issuer » est le type de l'utilisateur auquel appartient le profil. Cet utilisateur a plusieurs autorité et privilège, entre autre avoir une identité blockchain. Ce qui veut dire que seuls les « issuers » ont la visibilité de cette interface.

« Verifier » est un autre type d'utilisateur, d'après la figure 6 nous pouvons voir que cet utilisateur n'a pas d'identité blockchain.

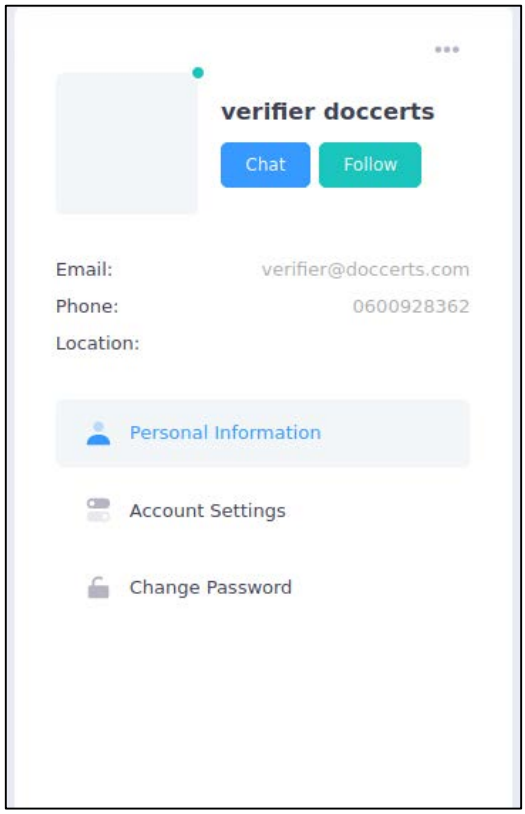

Figure 72 : Profil du Verifier.

5. Conclusion

Après avoir testé différentes méthodes et modèles de détection et de reconnaissance du visage, nous avons choisi les meilleurs pour les implémenter dans notre application. Nous avons également fait une démonstration en utilisant une interface que nous avons créée, dans l'attente de l'intégration du code dans l'application Doccerts.

CONCLUSION

En guise de conclusion, nous pouvons affirmer que notre travail enrichi la littérature par deux contributions majeurs. D'abord sur le plan théorique avec une revue de la littérature actualisée basée sur des travaux imminents, sachant que la littérature est d'une immense richesse nous avons essayé de nous intéresser à ce qui nous importe sans trop nous perdre dans le flux d'informations existant. Et ensuite sur le plan pratique avec l'implémentant d'un algorithme fiable de reconnaissance faciale capable d'assurer une bonne précision après l'accomplissement de quelques tests et l'entrainement de quelques modèles, et surtout en profitant de recherches et d'études déjà faites, de modèles pré-entrainés avec des bases données immenses.[49]

Dans le cadre de notre stage, et prenant en considération les conditions d'exécution de notre code, nous sommes optimiste quant à l'implémentation qui d'ailleurs est la tâche que nous nous apprêtons à accomplir, après quoi nous comptons essayer d'appliquer la notion de l'apprentissage par renforcement qui consiste à laisser les ordinateurs apprendre de leurs expériences grâce à un système de récompense ou de pénalité. D'après un article, cette notion pourrait même s'agir de la clé permettant l'avènement d'une intelligence artificielle générale comparable à celle de l'humain. [50]

Durant notre période de stage, en plus du travail théorique et pratique en rapport avec la reconnaissance faciale, nous avons eu la chance de nous initier au Framework Angular en réalisant quelques tâches qui nous ont été confiées.

Grâce aux résultats spectaculaires de la reconnaissance faciale elle se répand de plus en plus et se développe dans le monde entier, il est donc normal qu'elle soit un sujet de débat, mais est-il normal que les individus ressentent de l'insécurité et le manque de respect à leurs droits et libertés ? La reconnaissance faciale divise la population terrestre en deux camps, le premier qui la supporte la considérant comme outils de sécurité, et le deuxième qui craint ces éventuels mauvais usages. Des filtres sont d'ailleurs appliquer sur des images pour nuire voire empêché le processus de la reconnaissance faciale. D'ailleurs depuis le 11 mars 2021, 243.198 photos où figurent des personnes ont été altérées. Plus exactement, les visages ont été floutés, comme l'indiquent une mise à jour et un article de recherche mis en ligne sur le site d'ImageNet. [51]

Avec ces conditions, est-il possible de trouver un moyen d'assurer la reconnaissance faciale avec les mêmes précisions et beaucoup moins de données ? Allant ainsi vers une intelligence artificielle qui requière moins de données, mais davantage de diversité, comme publié dans le nouveau rapport Gartner de 2021 [52].

REFERENCES

[1] « Blockchain: the answer to life, the universe and everything? », *the Guardian*, juill. 07, 2016. http://www.theguardian.com/world/2016/jul/07/blockchain-answer-life-universe-everything-bitcoin-technology (consulté le juill. 07, 2021).

[2] L. Sirovich et M. Kirby, « Low-Dimensional Procedure for the Characterization of Human Faces », *Journal of the Optical Society of America. A, Optics and image science*, vol. 4, p. 519-24, avr. 1987, doi: 10.1364/JOSAA.4.000519.

[3] M. A. Turk et A. P. Pentland, « Face recognition using eigenfaces », in *1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Proceedings*, juin 1991, p. 586-591. doi: 10.1109/CVPR.1991.139758.

[4] M. Wang et W. Deng, « Deep Face Recognition: A Survey », *Neurocomputing*, vol. 429, p. 215-244, mars 2021, doi: 10.1016/j.neucom.2020.10.081.

[5] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, et Y. Ma, « Robust face recognition via sparse representation », *IEEE Trans Pattern Anal Mach Intell*, vol. 31, n° 2, p. 210-227, févr. 2009, doi: 10.1109/TPAMI.2008.79.

[6] P. N. Belhumeur, J. P. Hespanha, et D. J. Kriegman, « Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection », in *Computer Vision — ECCV '96*, Berlin, Heidelberg, 1996, p. 43-58. doi: 10.1007/BFb0015522.

[7] « Beyond eigenfaces: probabilistic matching for face recognition | IEEE Conference Publication | IEEE Xplore ». https://ieeexplore.ieee.org/document/670921 (consulté le juill. 05, 2021).

[8] W. Deng, J. Hu, J. Lu, et J. Guo, « Transform-Invariant PCA: A Unified Approach to Fully Automatic FaceAlignment, Representation, and Recognition », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, n° 36, p. 1275-1284, 2014, doi: 10.1109/TPAMI.2013.194.

[9] X. He, S. Yan, Y. Hu, P. Niyogi, et H.-J. Zhang, « Face recognition using Laplacianfaces », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, n° 3, p. 328-340, mars 2005, doi: 10.1109/TPAMI.2005.55.

[10] « [PDF] Graph embedding: a general framework for dimensionality reduction | Semantic Scholar ». https://www.semanticscholar.org/paper/Graph-embedding%3A-a-general-framework-for-reduction-Yan-Xu/bc3c2d97e967e18e1eafd9f2b1b887bf79c9d545 (consulté le juill. 05, 2021).

[11] W. Deng, J. Hu, J. Guo, H. Zhang, et C. Zhang, « Comments on "globally maximizing, locally minimizing: unsupervised discriminant projection with application to face and palm biometrics" », *IEEE Trans Pattern Anal Mach Intell*, vol. 30, n° 8, p. 1503-1504, août 2008, doi: 10.1109/tpami.2007.70783.

[12] L. Zhang, M. Yang, et X. Feng, « Sparse Representation or Collaborative Representation: Which Helps Face Recognition? », nov. 2011, vol. 2011, p. 471-478. doi: 10.1109/ICCV.2011.6126277.

[13] W. Deng, J. Hu, et J. Guo, « Extended SRC: Undersampled Face Recognition via Intraclass Variant Dictionary », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, n° 9, p. 1864-1870, sept. 2012, doi: 10.1109/TPAMI.2012.30.

[14] W. Deng, J. Hu, et J. Guo, « Face Recognition via Collaborative Representation: Its Discriminant Nature and Superposed Representation », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, n° 10, p. 2513-2521, oct. 2018, doi: 10.1109/TPAMI.2017.2757923.

[15] T. Ahonen, A. Hadid, et M. Pietikainen, « Face Description with Local Binary Patterns: Application to Face Recognition », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, nº 12, p. 2037-2041, déc. 2006, doi: 10.1109/TPAMI.2006.244.

[16] W. Deng, J. Hu, et J. Guo, « Compressive Binary Patterns: Designing a Robust Binary Face Descriptor with Random-Field Eigenfilters », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, nº 3, p. 758-767, mars 2019, doi: 10.1109/TPAMI.2018.2800008.

[17] A. Krizhevsky, I. Sutskever, et G. E. Hinton, « ImageNet classification with deep convolutional neural networks », *Commun. ACM*, vol. 60, nº 6, p. 84-90, mai 2017, doi: 10.1145/3065386.

[18] O. M. Parkhi, A. Vedaldi, et A. Zisserman, « Deep Face Recognition », in *Procedings of the British Machine Vision Conference 2015*, Swansea, 2015, p. 41.1-41.12. doi: 10.5244/C.29.41.

[19] « OpenCV Haar Cascades - PyImageSearch ». https://www.pyimagesearch.com/2021/04/12/opencv-haar-cascades/ (consulté le juill. 05, 2021).

[20] « Fig. 1. Haar features used for Viola Jones face detection method », *ResearchGate*. https://www.researchgate.net/figure/Haar-features-used-for-Viola-Jones-face-detection-method_fig1_268348020 (consulté le juill. 05, 2021).

[21] N. Dalal et B. Triggs, « Histograms of oriented gradients for human detection », in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, juin 2005, vol. 1, p. 886-893 vol. 1. doi: 10.1109/CVPR.2005.177.

[22] « Histograms of oriented gradients for human detection | IEEE Conference Publication | IEEE Xplore ». https://ieeexplore.ieee.org/document/1467360 (consulté le juill. 05, 2021).

[23] D. E. King, « Max-Margin Object Detection », *arXiv:1502.00046 [cs]*, janv. 2015, Consulté le: juill. 05, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1502.00046

[24] R. Kaur et E. Himanshi, « Face recognition using Principal Component Analysis », *Souvenir of the 2015 IEEE International Advance Computing Conference, IACC 2015*, p. 585-589, juill. 2015, doi: 10.1109/IADCC.2015.7154774.

[25] « Artificial Intelligence and Radiotherapy », *Mission Search*, mai 26, 2021. https://missionsearch.com/artificial-intelligence-and-radiotherapy/ (consulté le juill. 09, 2021).

[26] « Yann LeCun, l'intelligence en réseaux », *Le Monde.fr*, juin 08, 2015. Consulté le: juill. 06, 2021. [En ligne]. Disponible sur: https://www.lemonde.fr/sciences/article/2015/06/08/yann-lecun-l-intelligence-en-reseaux_4649831_1650684.html

[27] « Réseau neuronal convolutif », *Wikipédia*. avr. 15, 2021. Consulté le: juill. 12, 2021. [En ligne]. Disponible sur: https://fr.wikipedia.org/w/index.php?title=R%C3%A9seau_neuronal_convolutif&oldid=181914105

[28] « Qu'est ce qu'un réseau de neurones convolutif (ou CNN) ? », *OpenClassrooms*. https://openclassrooms.com/fr/courses/4470531-classez-et-segmentez-des-donnees-visuelles/5082166-quest-ce-quun-reseau-de-neurones-convolutif-ou-cnn (consulté le juill. 12, 2021).

[29] « Convolutional Neural Networks (CNNs) and Layer Types - PyImageSearch ». https://www.pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/ (consulté le juill. 12, 2021).

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, et R. Salakhutdinov, « Dropout: A Simple Way to Prevent Neural Networks from Overfitting », *Journal of Machine Learning Research*, vol. 15, nº 56, p. 1929-1958, 2014.

[31] « Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, “Gradient-based learning applied to document recognition” - Recherche Google ». https://www.google.com/search?q=Y.+Lecun%2C+L.+Bottou%2C+Y.+Bengio%2C+P.+Haffner%2C+%E2%80%9CGradient-based+learning+applied+to+document+recognition%E2%80%9D&rlz=1C1GCEA_enMA875MA875&oq=Y.+Lecun%2C+L.+Bottou%2C+Y.+Bengio%2C+P.+Haffner%2C+%E2%80%9CGradient-based+learning+applied+to+document+recognition%E2%80%9D&aqs=chrome..69i57.549j0j4&sourceid=chrome&ie=UTF-8 (consulté le juill. 06, 2021).

[32] « Bilinear deep learning for image classification — The Hong Kong Polytechnic University ». https://research.polyu.edu.hk/en/publications/bilinear-deep-learning-for-image-classification (consulté le juill. 06, 2021).

[33] K. Simonyan et A. Zisserman, « Very Deep Convolutional Networks for Large-Scale Image Recognition », *arXiv:1409.1556 [cs]*, avr. 2015, Consulté le: juill. 06, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1409.1556

[34] C. Szegedy *et al.*, « Going deeper with convolutions », in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, juin 2015, p. 1-9. doi: 10.1109/CVPR.2015.7298594.

[35] F. Schroff, D. Kalenichenko, et J. Philbin, « FaceNet: A Unified Embedding for Face Recognition and Clustering », *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 815-823, juin 2015, doi: 10.1109/CVPR.2015.7298682.

[36] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, et A. Zisserman, « VGGFace2: A dataset for recognising faces across pose and age », *arXiv:1710.08092 [cs]*, mai 2018, Consulté le: juill. 09, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1710.08092

[37] S. Ghosh, « How to create a Face Recognition Model using FaceNet Keras? », *Medium*, juill. 10, 2020. https://medium.com/clique-org/how-to-create-a-face-recognition-model-using-facenet-keras-fd65c0b092f1 (consulté le juill. 06, 2021).

[38] « 3.9.6 Documentation ». https://docs.python.org/3/ (consulté le juill. 12, 2021).

[39] « NumPy Documentation ». https://numpy.org/doc/ (consulté le juill. 12, 2021).

[40] « Matplotlib: Python plotting — Matplotlib 3.4.2 documentation ». https://matplotlib.org/ (consulté le juill. 12, 2021).

[41] « OpenCV: Introduction to OpenCV-Python Tutorials ». https://docs.opencv.org/master/d0/de3/tutorial_py_intro.html (consulté le juill. 06, 2021).

[42] « TensorFlow », *TensorFlow*. https://www.tensorflow.org/?hl=fr (consulté le juill. 06, 2021).

[43] « Accueil - Keras Documentation ». https://faroit.com/keras-docs/1.2.0/ (consulté le juill. 06, 2021).

[44] « Bibliothèque C++ dlib ». http://dlib.net/ (consulté le juill. 06, 2021).

[45] « Bienvenue dans Streamlit — Documentation Streamlit 0.84.0 ». https://docs.streamlit.io/en/stable/ (consulté le juill. 08, 2021).

[46] « Angular/ Versionn Français (Consulté le 09 juil. 2021) — Wikipédia ». https://fr.wikipedia.org/wiki/Angular/_Versionn_Fran%C3%A7ais_(Consult%C3%A9_le_09_juil._2021) (consulté le juill. 09, 2021).

[47] K. Team, « Keras documentation: Transfer learning & fine-tuning ». https://keras.io/guides/transfer_learning/ (consulté le juill. 09, 2021).

[48] « Keras ImageDataGenerator for Image Augmentation | Python Use Case », *Analytics Vidhya*, août 11, 2020. https://www.analyticsvidhya.com/blog/2020/08/image-augmentation-on-the-fly-using-keras-imagedatagenerator/ (consulté le juill. 09, 2021).

[49] J. Brownlee, « Difference Between a Batch and an Epoch in a Neural Network », *Machine Learning Mastery*, juill. 19, 2018. https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/ (consulté le juill. 10, 2021).

[50] « Reinforcement Learning : qu'est-ce que l'apprentissage par renforcement ? » https://www.lebigdata.fr/reinforcement-learning-definition (consulté le juill. 09, 2021).

[51] « ImageNet, la base de données inquiétante pour le respect de la vie privée », *Sciences et Avenir*, avr. 08, 2021. https://www.sciencesetavenir.fr/high-tech/intelligence-artificielle/imagenet-la-base-de-donnees-inquiete-pour-la-vie-privee_153293 (consulté le juill. 09, 2021).

[52] I. for Business, « Vers la fin du Big Data : 10 tendances Data & Analytics en 2021 », *IT for Business*, févr. 23, 2021. https://www.itforbusiness.fr/vers-la-fin-du-big-data-10-tendances-data-analytics-en-2021-42569 (consulté le juill. 09, 2021).

Système de vérification d'IMAGE

Résumé

Eviter la redondance d'images, éviter qu'une personne s'inscrive dans l'application avec l'image d'une autre; ce que l'on peut qualifier d'usurpation de l'identité et s'assurer qu'en cas de modification, la nouvelle photo entrée par un utilisateur lui appartient sont les tâches ou les conditions que nous avons dû respecter lors de notre période de stage, et que nous présentons dans ce rapport. Nous avons implémenté et entrainé différentes méthodes et architectures de la reconnaissance faciale pour pouvoir arriver à une meilleure précision compte tenu de la difficulté principale de cette tâche, qui est en même temps la richesse et la pauvreté au niveau des données. Une situation qui a nécessité l'utilisation de quelques techniques pour arriver à un meilleur compromis.

Mots clés : Reconnaissance faciale – détection de visage – apprentissage profond – réseau neuronal convolutif - apprentissage par transfert.

IMAGE verification system

Abstract

Avoiding image redundancy, avoiding that a person registers in the application with the image of another; what can be qualified as identity theft and ensuring that in case of modification, the new photo entered by a user belongs to him are the tasks or conditions that we had to respect during our internship period, and that we present in this report. We have implemented and trained different methods and architectures of facial recognition in order to achieve a better accuracy considering the main difficulty of this task, which is at the same time the richness and poverty of the data. A situation that required the use of some techniques to achieve a better compromise.

Keywords: Face recognition - face detection - deep learning - convolutional neural network - transfer learning.

Master Systèmes Intelligents & Réseaux

Département d'Informatique

Faculté des Sciences et Techniques de Fès

A.U. 2020 - 2021