Université Sultan Moulay Slimane
Ecole Nationale des Sciences Appliquées
ENSA Khouribga

Master Professionnel

Ingénierie en Intelligence Artificiel, Business Analytics et Big Data

MÉMOIRE DE FIN D'ÉTUDES

Application de Reconnaissance Faciale

Présenté par : YOUSSOUF HASSANI BEN ALI
benalben525@gmail.com

Professeurs :

- Mr SOUFIANE LYAQINI (IA) lyaqini.soufiane@gmail.com
- Mr ABDELGHANI GHAZDALI (ML) a.ghazdali@usms.ma
- Mme HIND HAFSI (DL) hind.hafsi@usms.ac.ma

Année Universitaire : 2023/2024

Adresse : Ecole Nationale des Sciences Appliquées Bd Béni Amir, BP 77 Khouribga - Maroc

Site web : http ://ensak.usms.ac.ma/

Table des matières

Table des figures

Dédicaces et Remerciements

Tout d'abord je tiens à remercier **ALLAH** le tout puissant de m'avoir donné la santé, la volonté, le courage et la patience pour pouvoir réaliser ce projet de fin d'études.

Je tiens à remercier les membres du jury pour leur lecture attentive de ce mémoire, ainsi que pour les remarques qu'ils m'adresseront lors de cette soutenance afin d'améliorer mon travail. Mes professeurs, en espérant que vous découvrirez, dans ce manuscrit, les résultats du dévouement que vous avez démontré lors des cours que vous nous avez donnés.

Je tiens à exprimer ma gratitude envers mon père **BACAR SOILIH** et ma mère **AMINA ALI** pour leur amour précieux, leurs sacrifices, leur confiance, leur soutien et toutes les valeurs qu'ils ont su me transmettre.

Je tiens à remercier ma **grand-mère** pour toute l'affection qu'elle m'a donnée et pour son précieux encouragement.

Je dédie ce mémoire à mes soeurs et mon frère, **BAHIA**, **AICHA** et **BILAL** pour leurs tendresses, leurs complicités et leurs présences malgré la distance qui nous sépare.

Je dédie ce mémoire à mes **tantes** et mes **oncles** de la famille, pour leurs mots d'encouragement et leurs gentillesses.

Je souhaite remercier tous ceux qui ont contribué de près ou de loin à l'élaboration de ce mémoire ainsi qu'à la réussite de ce parcours universitaire.

Enfin, je remercie affectueusement tous ceux qui m'ont soutenu dans mes études.

Introduction Générale

Il est devenu crucial de pouvoir identifier l'identité d'un individu de manière efficace et précise, car l'accès sécurisé et la surveillance sont devenus des sujets d'une grande importance aujourd'hui. En effet, malgré notre ignorance, notre identité est vérifiée chaque jour par de nombreuses organisations : lors de l'utilisation de notre carte bancaire, lors de notre connexion à un réseau informatique, etc.

La technologie émergente de la **reconnaissance faciale**, qui se situe à la croisée de l'**informatique** et de la **vision par ordinateur**[3], a bouleversé notre environnement numérique en proposant des méthodes novatrices pour identifier et vérifier l'identité des individus en se basant sur leurs **caractéristiques faciales** uniques. Cette technologie a été très appréciée depuis ses débuts pour ses applications potentielles dans différents secteurs, tels que la sécurité, la surveillance, le commerce et même la santé. Toutefois, elle suscite également des inquiétudes éthiques et juridiques majeures concernant la vie privée, la discrimination et la surveillance de masse. Les **systèmes de reconnaissance faciale**[4] peuvent être utilisés pour repérer des individus sur des photos, des vidéos ou en direct.

La **reconnaissance faciale** fait partie d'un domaine de sécurité biométrique[2]. D'autres applications biométriques comprennent la reconnaissance vocale[5] et les empreintes digitales[6]. Cette technologie est principalement employée dans le domaine de la sécurité, même si d'autres secteurs s'y intéressent de plus en plus. Chaque personne a en effet des particularités qui lui sont propres : sa voix, ses empreintes digitales, les traits de son visage, la forme de sa main, sa signature et même son ADN. Ces informations appelées "**biométriques**" peuvent donc servir à l'identifier. Un des principaux atouts de ce que l'on nomme mot de passe biométrique[7] réside dans le fait qu'il ne peut pas être oublié ou transmis à quelqu'un d'autre.

La **reconnaissance faciale** est bien connue par de nombreuses personnes grâce à l'utilisation de FaceID, qui permet de déverrouiller les iPhone (cependant, ce n'est qu'une utilisation parmi d'autres). En règle générale, la **reconnaissance faciale** n'est pas basée sur une vaste base de données de photos pour déterminer l'identité d'une personne : elle ne fait qu'identifier et reconnaître une personne comme propriétaire unique de l'appareil, afin d'empêcher autrui d'y accéder.

En plus de déverrouiller les téléphones, la reconnaissance faciale fonctionne en comparant le visage des individus qui passent devant des caméras spéciales à des photos de personnes surveillées. Ces listes de surveillés peuvent inclure des photos de toutes les personnes, y compris celles qui n'ont jamais commis de crime, et ces photos peuvent être prises de n'importe quel endroit, y compris des comptes de réseaux sociaux.

Il est indéniable que le visage est la **caractéristique biométrique**[8] la plus naturelle que les individus utilisent pour s'identifier entre eux. Ainsi, la **reconnaissance de visages** est devenue l'une des branches de la **vision par ordinateur** qui connaît un succès considérable et qui est en constante évolution.

Le projet présenté dans ce rapport consiste à créer une **application** capable de détecter et de reconnaître les visages humains à l'aide de l'**intelligence artificiel**, l'**apprentissage automatique** et l'**apprentissage profond** en utilisant Python.

Ce présent rapport est scindé en trois chapitres, le premier comportera des notions de l'**intelligence artificiel** et des outils nécessaires pour l'élaboration de notre travail. Le deuxième chapitre, il vise la présentation des méthodes existantes pour résoudre les différentes problématiques du projet. Dans le troisième chapitre sera sous forme de présentation de l'environnement de travail, des données utilisées, des différentes méthodes et l'API.

Chapitre 1
Rappels et Préliminaires

1.1 Intelligence Artificielle (IA)

L'**intelligence artificielle**, désigne la capacité des machines à effectuer des tâches qui nécessitent normalement l'**intelligence humaine**. Ces tâches peuvent inclure la résolution de problèmes, la reconnaissance de formes, la prise de décision, et bien plus encore.

Il y a diverses catégories et niveaux d'**IA** :

1. **IA faible** (ou étroite) : ce type d'IA est conçu pour effectuer une tâche spécifique. Par exemple, un système de recommandation[9] sur un site de streaming ou un programme de jeu d'échecs. Elle n'a pas de capacité à généraliser ou à effectuer des tâches en dehors de son domaine spécifique.
2. **IA forte** (ou générale) : ce type d'IA serait capable de comprendre, d'apprendre, de raisonner et de résoudre des problèmes de manière similaire à un être humain dans une grande variété de domaines. Nous n'en sommes pas encore là et c'est un objectif de recherche et de développement ambitieux.
3. **IA super-intelligente** : il s'agit d'une **intelligence artificielle** hypothétique qui surpasserait de loin l'**intelligence humaine** dans tous les aspects. C'est un concept souvent exploré dans la science-fiction et la philosophie, mais qui pose également des questions éthiques et existentielles importantes.

Les méthodes couramment utilisées pour construire des systèmes d'**IA** comprennent l'**apprentissage supervisé**, l'**apprentissage non supervisé**, et l'**apprentissage par renforcement**. L'**apprentissage supervisé** implique de fournir à l'IA des données étiquetées pour qu'elle puisse apprendre à faire des prédictions ou à prendre des décisions. L'**apprentissage non supervisé** consiste à laisser l'IA trouver des modèles ou des structures dans des ensembles de données non étiquetées. L'**apprentissage par renforcement** implique de récompenser ou de punir l'IA en fonction de ses actions, afin qu'elle puisse apprendre à maximiser une certaine récompense.

L'IA est utilisée dans de nombreux domaines, tels que la santé, les finances, la logistique, l'automobile, les jeux, la reconnaissance vocale[5] et faciale, les recommandations en ligne[9], et bien plus encore. Son développement soulève également des questions éthiques, notamment en ce qui concerne la confidentialité des données, la discrimination algorithmique et l'impact potentiel sur le marché du travail.

1.2 Apprentissage Automatique (ML)

Le **machine learning**, ou **apprentissage automatique** en français, est une branche de l'intelligence artificielle qui se concentre sur le développement de techniques permettant aux ordinateurs d'apprendre à partir de données et d'améliorer leurs performances dans des tâches spécifiques sans être explicitement programmés pour chaque situation. En d'autres termes, au lieu de programmer des règles spécifiques pour résoudre un problème, on fournit à l'ordinateur des données et des algorithmes qui lui permettent d'apprendre par lui-même à partir de ces données.

Le **machine learning** est largement utilisé dans de nombreux domaines, notamment la reconnaissance d'images et de voix, la traduction automatique, la recommandation de produits, la détection de fraude, la prédiction de séries temporelles, la biologie computationnelle, et bien d'autres. Son utilisation est en constante expansion à mesure que de nouvelles techniques et applications émergent.

1.3 Apprentissage Profond (DL)

Le **deep learning**, ou **apprentissage profond** en français, est une sous-catégorie du **machine learning** qui utilise des **réseaux de neurones artificiels** composés de multiples couches pour modéliser et extraire des motifs complexes à partir de données. Ce type d'approche est appelé "**profond**" car il implique généralement des **réseaux de neurones** avec plusieurs couches cachées, permettant ainsi de capturer des représentations hiérarchiques des données.

Voici quelques éléments clés du deep learning :

- **Réseaux de neurones artificiels** : Les réseaux de neurones artificiels sont des **modèles mathématiques** inspirés par le fonctionnement du cerveau humain. Ils sont composés de plusieurs couches de neurones, chaque neurone étant connecté à ceux des couches précédentes et suivantes. Ces neurones transmettent et traitent des informations qui, lorsqu'elles sont combinées, permettent au réseau de prendre des décisions ou de faire des prédictions.

- **Architecture de réseau profond** : Les réseaux de neurones profonds sont caractérisés par leur profondeur, c'est-à-dire le nombre de couches cachées qu'ils contiennent. Ces couches permettent au réseau de capturer des caractéristiques complexes et abstraites des données en les transformant progressivement à travers les différentes couches.

- **Apprentissage par rétropropagation** : L'apprentissage dans les réseaux de neurones profonds se fait généralement par rétropropagation du gradient. Cela consiste à calculer la dérivée de la fonction de perte par rapport aux poids du réseau, puis à ajuster ces poids pour minimiser la perte. Ce processus est réalisé de manière itérative sur l'ensemble des données d'entraînement jusqu'à ce que le modèle atteigne une performance satisfaisante.

- **Applications** : Le **deep learning** a révolutionné de nombreux domaines, notamment la vision par ordinateur (reconnaissance d'images et de vidéos), le traitement du langage naturel (traduction automatique, compréhension du langage), la reconnaissance vocale, la biologie computationnelle, la robotique, la santé, les jeux, et bien d'autres.

En résumé, le **deep learning** a permis des avancées significatives dans de nombreux domaines en permettant aux machines de comprendre et de traiter des données complexes d'une manière qui ressemble de plus en plus à la façon dont les humains le font. Son potentiel continue d'être exploré et son utilisation devrait continuer à croître avec le développement de nouvelles techniques et applications.

1.4 Réseau de Neurones Convolutif (CNN)

Un **CNN**, ou **Convolutional Neural Network** en anglais, est un type spécifique de **réseau de neurones profonds** qui est particulièrement efficace pour traiter des données structurées en grille telles que des images ou des séquences temporelles. Les **CNN** ont révolutionné le domaine de la vision par ordinateur et sont largement utilisés dans de nombreuses applications de reconnaissance d'images et de vidéos.

Les **CNN** sont largement utilisés dans de nombreuses applications telles que la reconnaissance d'objets, la détection de visages, la segmentation d'images, la classification d'images médicales, la reconnaissance de caractères, et bien d'autres. Leur architecture spécifique les rend particulièrement adaptés à la capture et à l'analyse de structures spatiales dans des données en grille comme les images.

1.5 Architectures CNN

Les **architectures de réseaux de neurones convolutifs** (**CNN**), également connues sous le nom de **ConvNets**, sont des modèles de **réseaux de neurones** spécialement conçus pour le traitement des données structurées en grille, telles que les images.

Voici quelques-unes des architectures **CNN** les plus populaires :

1. **LeNet-5**
2. **AlexNet**
3. **VGGNet**
4. **ResNet**
5. **FaceNet**

Ces architectures **CNN** ont été développées pour résoudre divers problèmes de vision par ordinateur, tels que la classification d'images, la détection d'objets, la segmentation sémantique, etc. Chacune de ces architectures a ses propres caractéristiques et avantages, et le choix dépend souvent des exigences spécifiques de la tâche à accomplir. Plus de détaille dans le chapitre 2.

1.6 Réseau de Neurones Profonds (DNN)

Un **DNN**, ou **Deep Neural Network** en anglais, est un type de **réseau de neurones artificiels** composé de plusieurs couches cachées. Ces réseaux sont caractérisés par leur profondeur, c'est-à-dire par le nombre de couches intermédiaires qu'ils contiennent. Les **DNN** peuvent être utilisés pour effectuer une variété de tâches d'**apprentissage automatique**, notamment la classification, la régression, la génération de texte, la détection d'anomalies, et bien d'autres.

Les **DNN** ont été à la base de nombreuses avancées récentes en **intelligence artificielle** et en **apprentissage automatique**, et leur utilisation continue d'évoluer avec le développement de nouvelles architectures, techniques d'entraînement et applications.

1.7 Traitement d'image

Le **traitement d'image** est un domaine de l'informatique qui se concentre sur l'acquisition, le traitement, l'analyse et l'interprétation des images numériques. Il s'agit de transformer une image numérique en une forme plus significative et plus facile à analyser pour les humains ou pour les machines. Le **traitement d'image** est largement utilisé dans de nombreux domaines, notamment la vision par ordinateur, la médecine, la surveillance, la reconnaissance de formes, la photographie numérique, et bien d'autres.

Voici quelques-unes des principales techniques et opérations utilisées dans le **traitement d'image** :

- **Pré-traitement** : Cette étape comprend des opérations telles que la correction des couleurs, l'amélioration du contraste, le filtrage du bruit, le redimensionnement et la normalisation. L'objectif est de préparer l'image pour des analyses ultérieures en améliorant sa qualité et en éliminant les imperfections indésirables.

- **Filtrage** : Les filtres sont des opérations qui permettent de mettre en évidence ou de supprimer certaines caractéristiques de l'image. Les filtres les plus couramment utilisés comprennent les filtres de lissage (pour réduire le bruit), les filtres de détection de bord (pour détecter les contours), et les filtres de mise en relief (pour mettre en évidence les détails).

- **Segmentation** : La segmentation consiste à diviser une image en régions ou en objets significatifs. Cela peut inclure la détection de contours, la séparation des objets les uns des autres, ou la classification des pixels en fonction de certaines caractéristiques.

- **Extraction de caractéristiques** : Cette étape consiste à extraire des informations importantes ou des caractéristiques de l'image qui peuvent être utilisées pour la classification, la reconnaissance ou d'autres tâches d'analyse. Les caractéristiques peuvent inclure des textures, des formes, des couleurs, des motifs, et bien d'autres.

- **Reconnaissance d'objets et de motifs** : Cette étape consiste à identifier et à reconnaître des objets spécifiques ou des motifs dans l'image. Cela peut inclure la détection de visages, la reconnaissance de caractères, la classification d'objets, ou la détection de zones d'intérêt.

- **Interprétation et analyse** : Cette étape consiste à interpréter les résultats du traitement d'image et à en tirer des conclusions ou des décisions. Cela peut inclure la mesure de certaines propriétés de l'image, la détection d'anomalies, ou la prise de décisions basées sur les informations extraites de l'image.

Le **traitement d'image** est un domaine vaste et en évolution constante, avec de nombreuses applications et techniques différentes. Il joue un rôle crucial dans de nombreux aspects de notre vie quotidienne, de la médecine à l'industrie en passant par la science et la technologie.

Chapitre 2
État De L'Art

2.1 Introduction

Ce travail se focalise sur deux tâches principales : la **détection** et l'**identification** de visages. La littérature présente de nombreuses techniques de **détection de visage** et d'**identification**. Dans ce chapitre, nous examinerons les méthodes les plus couramment employées. Le problème majeur dans la **détection** et la **reconnaissance** d'un objet est celui des différentes représentations qu'il peut avoir. Il existe plusieurs facteurs qui influencent la **détection** et la **reconnaissance du visage**, dont les plus étudiés sont :

- La **position** : sur une photographie, on peut observer un visage de face ou d'un angle donné.
- L'**expression du visage** : l'aspect d'un visage est également influencé par son expression.
- La **présence d'attributs** : Une personne peut posséder un chapeau, des lunettes, une moustache, une barbe, une cicatrice, etc.
- Les **conditions extérieures** : Chaque image présente des variations dans les conditions extérieures telles que la couleur, l'intensité de l'éclairage, la taille et la texture.
- L'**occultation** : une partie du visage peut être cachée par un autre objet ou par une autre personne.
- La **couleur** : les individus possèdent des teintes de peau distinctes, ce qui explique la disparité de la valeur du pixel qui représente la peau de chaque individu.

2.2 Historique

La **reconnaissance faciale** a une histoire qui remonte à plusieurs décennies, mais son développement significatif a eu lieu au cours des dernières années grâce aux progrès de la technologie informatique et des algorithmes d'**apprentissage automatique**. Les premiers travaux sur la **reconnaissance faciale** ont débuté dans les années **1960** et **1970** [1], principalement sous forme d'analyses basées sur des **caractéristiques faciales** spécifiques.

En **1980** Les premiers systèmes automatisés de reconnaissance faciale ont commencé à émerger, bien que leurs performances étaient limitées en raison de la technologie disponible à l'époque[10]. Des avancées significatives ont été réalisées dans les techniques de reconnaissance faciale, notamment avec l'introduction de méthodes basées sur la géométrie faciale et les modèles statistiques en **1990**.

L'année **2000**, La reconnaissance faciale est devenue plus répandue dans les applications de sécurité, comme le contrôle d'accès aux bâtiments et les systèmes de surveillance. Cependant, la précision des systèmes restait parfois un défi, en particulier dans des conditions variables telles que l'éclairage et l'angle de vue.

En **2010** La technologie de reconnaissance faciale a connu une croissance exponentielle, en grande partie grâce aux progrès de l'**apprentissage automatique**, en particulier du **deep learning**. Des entreprises comme **Facebook**, **Google** et **Apple** ont intégré la reconnaissance faciale dans leurs produits, notamment pour la gestion des photos et la sécurité des appareils.

Ces dernières années La **reconnaissance faciale** est devenue omniprésente dans de nombreux domaines, y compris la sécurité, la publicité, le commerce de détail et même la gestion des émotions. Cependant, elle suscite également des préoccupations croissantes en matière de vie privée et de sécurité, en raison de son potentiel d'utilisation abusive et de ses implications sur les libertés individuelles.

La **reconnaissance faciale** a connu une évolution au fil des années pour devenir une technologie puissante, mais son utilisation pose également des interrogations éthiques et juridiques cruciales qui doivent être prises en considération afin d'assurer une utilisation responsable et éthique.

2.3 Fonctionnement

Malgré la complexité du processus, les algorithmes de **détection de visage** commencent généralement par chercher des yeux humains ou un visage frontal. Les yeux forment ce que l'on nomme une vallée et sont l'une des caractéristiques les plus simples à repérer. Après avoir repéré les yeux, l'algorithme pourrait ensuite essayer de repérer les différentes parties du visage, telles que les sourcils, la bouche, le nez, les narines et l'iris. Après avoir supposé qu'il a repéré une zone faciale, l'algorithme peut alors effectuer des tests supplémentaires afin de vérifier s'il a réellement repéré un visage. Les principales étapes d'un **système de reconnaissance faciale**[4] sont les suivantes :

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

2.4 Détection

Les visages sont détectés en estimant la boîte de délimitation du visage dans une image spécifique. Le but est d'extraire le visage afin de pouvoir l'utiliser de manière plus efficace dans la reconnaissance. Nous exposons ci-dessous quelques techniques de détection :

2.4.1 Haar Cascade

Cette méthode est un algorithme de détection d'objets qui permet d'identifier des visages dans une image ou une vidéo en temps réel. Un classificateur Haar est basé sur un cadre de détection d'objets proposé par **Paul Viola** et **Michael Jones** dans leur article "**Détection rapide d'objets utilisant une cascade améliorée de fonctionnalités simples**"[11].

Chaque fonctionnalité est utilisée pour former un seul classificateur, comme illustré dans l'illustration ci-dessous. Toutefois, un seul classificateur n'est pas très précis, et plusieurs de ces classificateurs sont donc combinés. Le classificateur final obtenu est une combinaison pondérée de faibles classificateurs. Le classificateur offre une précision de classification supérieure à 95% grâce à cette méthode.

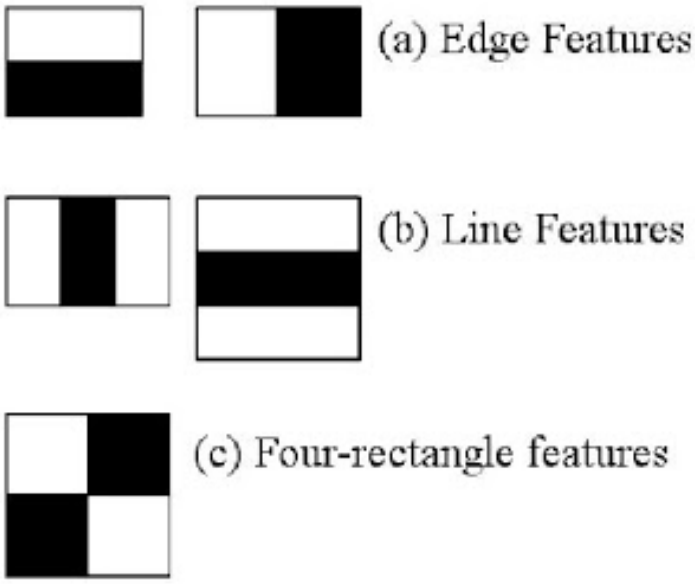


FIGURE 2.1 – 5 différents types de caractéristiques de type Haar

Dans l'exemple suivant, les images servent de noyaux convolutifs pour **extraire des caractéristiques**, où chaque caractéristique est une valeur obtenue en soustrayant la somme des pixels sous le rectangle noir de la somme des pixels sous le rectangle blanc.

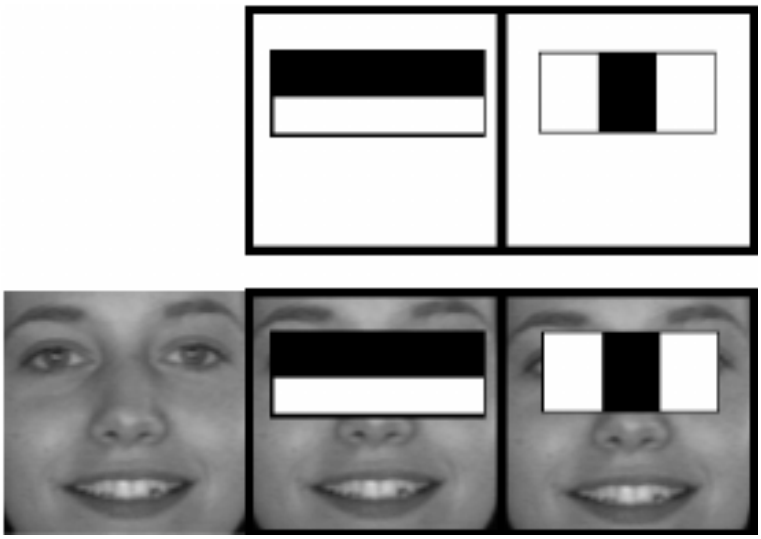


FIGURE 2.2 – Exemples de fonctionnalités Haar

L'illustration ci-dessus illustre deux caractéristiques : le bord et la ligne. Les **caractéristiques Edge** synchronisent de manière efficace les **caractéristiques du visage**, ce qui signifie que la zone des yeux est plus foncée que toute autre partie du visage. La ligne verticale du nez sur le visage est plus claire que les côtés, ce qui souligne l'attribut linéaire. Comme aucune de ces caractéristiques ne permet de classer le modèle de manière précise de manière individuelle, elles sont utilisées en cascade, c'est pourquoi on les appelle des **classificateurs en cascade** basés sur les fonctionnalités Haar.

2.4.2 Histogramme de gradients orientés

On utilise un histogramme de gradient orienté (**HOG**) en vision par ordinateur pour détecter des objets. Il a été introduit pour la première fois par **Dalal** et **Triggs** dans leur livre fondamental de 2005 intitulé "**Histogrammes of Oriented Gradients for Human Detection**"[12].

Le **HOG + SVM** linéaire, similaire aux **cascades de Haar**, utilise des fenêtres glissantes pour repérer des objets/faces dans une image. Le processus implique de séparer l'image en petites régions adjacentes, connues sous le nom de cellules, et de calculer l'histogramme des orientations des contours pour les pixels à l'intérieur de chaque cellule. Ainsi, l'association des histogrammes crée le descripteur **HOG**.

Afin d'accomplir cela, il est nécessaire de mettre l'image en noir et blanc. Ensuite, il est nécessaire d'analyser chaque pixel de l'image, et pour chaque pixel, il est nécessaire d'analyser chaque pixel qui l'entoure directement.

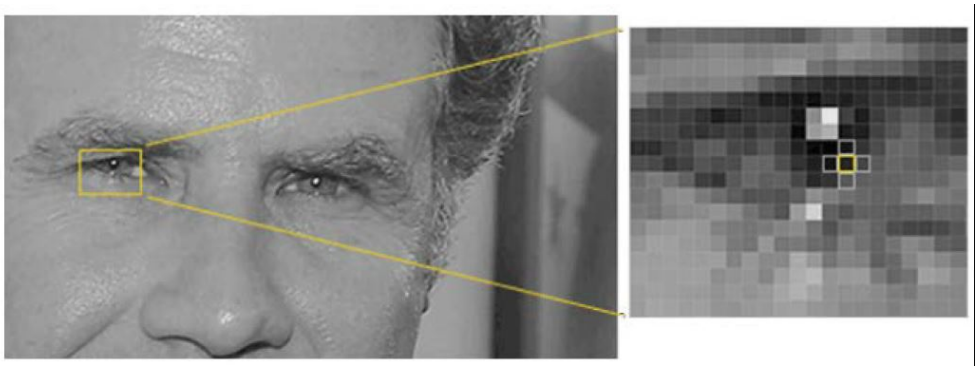


FIGURE 2.3 – Un pixel et ceux qui l'entoure directement

L'objectif est d'abord de mesurer la luminosité d'un pixel par rapport à ceux qui l'entourent directement, puis de tracer une flèche dans le sens de l'assombrité de l'image. En reproduisant cette procédure, les pixels seront remplacés par des flèches connues sous le nom de gradients, qui illustrent le changement du clair au foncé dans toute l'image.

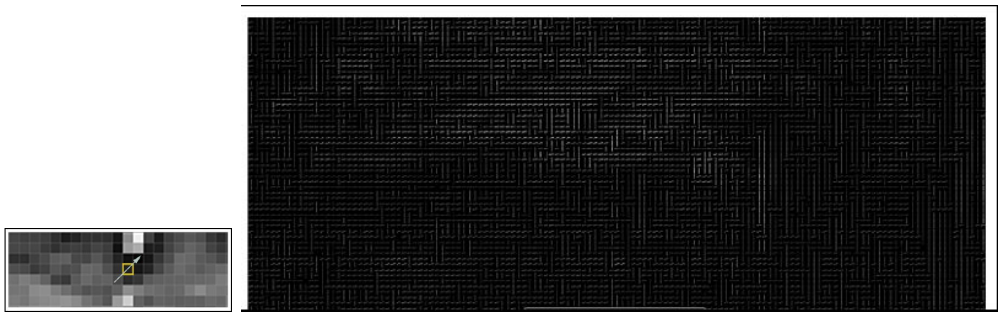


FIGURE 2.4 – Ensemble de flèches montrant le passage du clair au foncé de l'oeil

Cependant, cela nous fournit beaucoup trop d'informations. Pour voir le modèle de base de l'image, il serait préférable de voir le flux de base de la clarté et de l'obscurité à un niveau supérieur. Dans cette optique, l'image est séparée en petits carrés de 16x16 pixels. Il est nécessaire de mesurer le nombre de dégradés dans chaque direction principale (combien pointent vers le haut, vers le haut-droit, vers la droite, etc.) dans chaque carré. Par la suite, ce carré dans l'image doit être remplacé par les flèches de direction qui étaient les plus puissantes.

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

FIGURE 2.5 – Image originale

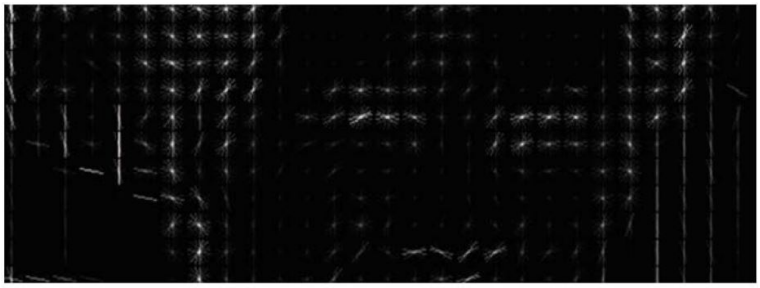


FIGURE 2.6 – Transformée en une représentation HOG

Tout ce qui a été mentionné précédemment illustre les efforts déployés pour entraîner le système. Ainsi, afin de trouver des visages dans cette forme **HOG**, il suffit de trouver la partie de notre image qui ressemble le plus à un modèle **HOG** connu qui a été extrait d'un grand nombre d'autres visages **HOG**.

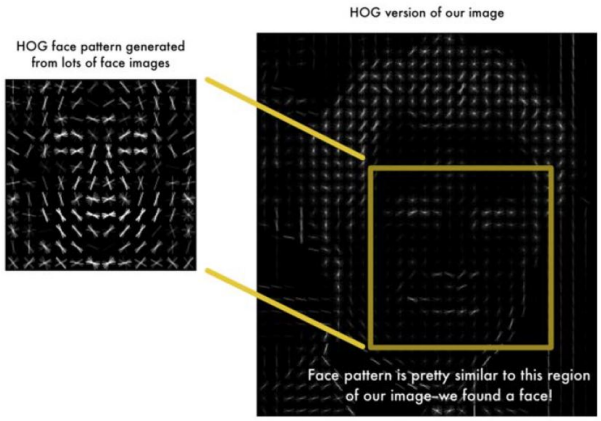


FIGURE 2.7 – Comparaison

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.

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. 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.

2.4.3 DNN

Les **cascades de Haar** peuvent facilement être remplacés par des détecteurs de visages plus précis provenant 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. Les performances de détection des visages de ce nouveau modèle sont améliorées par rapport aux modèles classiques, comme Haar. Le nouveau modèle est entraîné à l'aide du Framework **Caffe**[13].

Il est essentiel d'avoir les deux ensembles de fichiers pour utiliser le module de **réseau neuronal profond** avec des modèles **Caffe**[13] :

- Le(s) fichier(s) **.prototxt** qui définit l'architecture du modèle (c'est-à-dire les couches elles-mêmes).

- Le fichier **.caffemodel** qui contient les poids pour les couches réelles.

Caffe, également connu sous le nom d'architecture convolutive pour faciliter l'intégration rapide de fonctionnalités, est un Framework open source d'**apprentissage profond**. Sa particularité réside dans sa rapidité, sa compatibilité, sa capacité d'expression et sa flexibilité. Il utilise les langages de programmation **C++** et **Python** pour être écrit. Selon les informations, **Caffe** est l'une des mises en œuvre de connexion les plus rapides disponibles.

Parmi les bénéfices de la détection avec **apprentissage profond**, on peut citer la précision du détecteur de visage, l'utilisation d'algorithmes d'**apprentissage profond** modernes et le fait qu'il n'est pas nécessaire de régler aucun paramètre, contrairement au modèle **Haar-cascade**.

Cependant, il peut parfois être moins précis que le détecteur de visage **CNN** et ne pas détecter aussi bien les individus à la peau foncée que ceux à la peau claire.

2.4.4 CNN

Le fondateur de **DLIB**[14], **Davis King**, a développé un détecteur de visage **CNN** en se basant sur son expérience dans la détection d'objets à marge maximale. L'approche est extrêmement précise, grâce à la conception de l'algorithme lui-même et à la précision de **Davis** dans le choix de l'ensemble d'entraînement et dans l'entraînement du modèle.

Le détecteur **CNN** peut repérer les visages dans pratiquement tous les angles, bien qu'il soit très précis. Il est aussi capable de gérer les occlusions (quand une partie du visage est dissimulée derrière un objet). Cependant, il est impossible de travailler en temps réel sans accélération GPU, et il ne parvient pas à repérer les petits visages sur les images car il est créé avec une taille minimale de visage de 80 * 80.

2.4.5 MTCNN

Un outil de détection de visage moderne, le **MultiTask Cascaded Convolutional Neural Network** (**MTCNN**), utilise un détecteur de réseau neuronal à 3 étages.

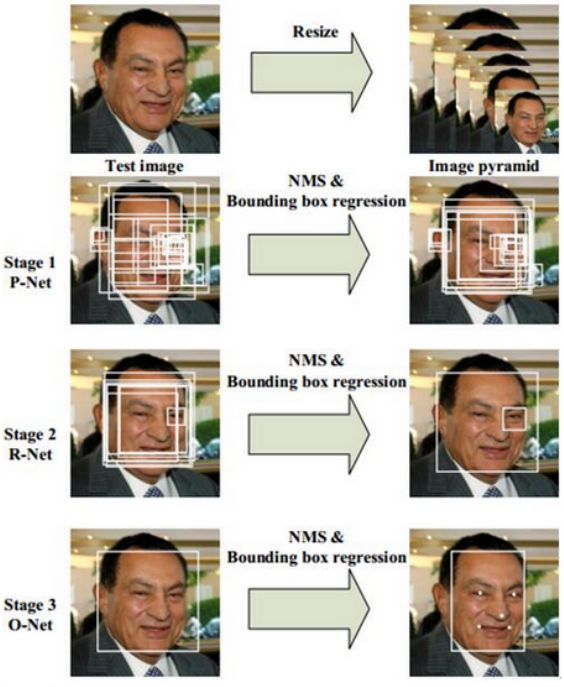


FIGURE 2.8 – Différentes étapes de MTCNN

Tout d'abord, l'image est ajustée à plusieurs reprises afin de repérer des visages de diverses tailles. Le **P-Net** (Proposal Net) procède ensuite au scanner des images, réalisant ainsi la première détection. Il possède un faible seuil de détection et donc détecte de nombreux faux positifs, même après une suppression non-maximale(**NMS**), mais fonctionne de manière intentionnelle.

Le deuxième réseau, le **R-Net** (Refinement Network), utilise les régions proposées (avec de nombreux faux positifs) pour filtrer les détections (également avec **NMS**) afin d'obtenir des boîtes englobantes assez précises.

Enfin, l'**O-Net** (Output Network) réalise la finalisation des boîtes englobantes. Ainsi, les visages sont non seulement repérés, mais les cadres de délimitation sont très exacts et précis.

Il est compatible avec la mise en uvre **TensorFlow**, ainsi qu'avec celle de **PyTorch**, qui est plus rapide. Il peut également servir à détecter en temps réel. **MTCNN** est extrêmement précis et solide. Il perçoit de manière adéquate les visages, même avec des dimensions, un éclairage et des rotations considérables. Cependant, il est plus lent que certains détecteurs tels que **Haar-cascade**, **HOG** et **DNN**. Un autre inconvénient réside dans sa capacité à repérer des fausses négatives, ce qui peut être résolu en ajustant le seuil de **MTCNN**.

2.5 Extraction des caractéristiques

Les différentes combinaisons de caractéristiques qui ne peuvent être identiques pour deux individus, à l'exception des jumeaux identiques, sont extraites de différentes manières. Ci-dessous nous présentons quelques méthodes d'extraction des caractéristique.

2.5.1 ACP

La méthode **Eigenface**[15] est une méthode très répandue, qui repose sur la technique **ACP** (**Analyse en composantes principales**)[16]. Elle repose sur le principe suivant : étant donné un ensemble d'images de visages exemples, il s'agit d'abord de déterminer les principales composantes de ces visages. Il s'agit d'identifier les vecteurs spécifiques de la matrice de covariance constituée par toutes les images exemples. On peut donc décrire chaque visage exemple en utilisant une combinaison linéaire de ces vecteurs spécifiques. Afin de créer la matrice de covariance, chaque représentation visuelle est convertie en vecteur. L'intensité lumineuse d'un pixel est représentée par chaque élément du vecteur.

L'**ACP** est une méthode rapide, facile et répandue pour identifier des modèles, elle est considérée comme l'une des meilleures méthodes. Les hypothèses de l'**ACP** sont idéales pour la réalisation d'une base de petite taille.

2.5.2 AlexNet

En **1998**, l'architecture **Lenet-5**[17] a été présentée par le chercheur français **Yann LeCun** et ses collègues, l'un des premiers modèles pré-entraînés, un **réseau neuronal de convolution multicouche** pour la classification d'images, dont la principale raison de popularité est sa structure simple et directe. Quelques chercheurs, notamment ceux de Toronto, ont été inspirés par **LeNet** et leur réseau **AlexNet**, qui ont remporté la compétition d'ImageNet en **2012**, ce qui en fait le pionnier, et **AlexNet** celui qui a suscité l'enthousiasme autour du **Deep Learning**.

AlexNet est composé de huit couches qui sont accessibles à l'apprentissage. Le modèle comprend cinq couches de convolution combinées avec une combinaison de max pooling, suivies de trois couches entièrement connectées. Chacune de ces couches est activée par Relu, à l'exception de la couche de sortie.

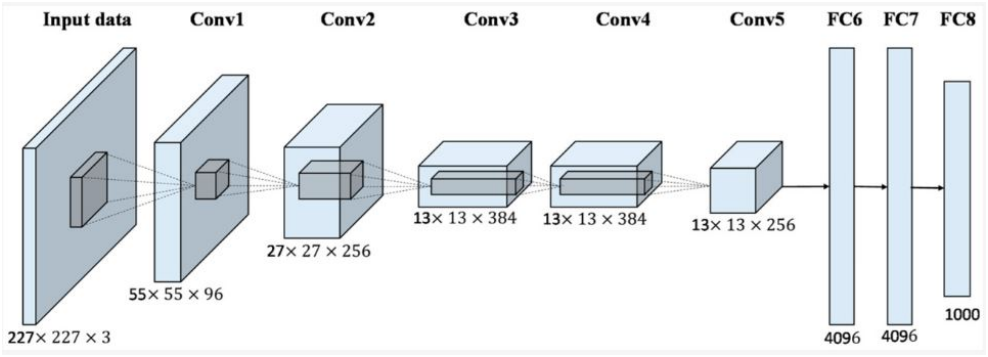


FIGURE 2.9 – AlexNet architecture

Selon les développeurs d'**AlexNet**, l'emploi du Relu comme fonction d'activation a permis d'accélérer l'apprentissage de près de six fois. Les couches d'exclusion ont également été employées par eux, ce qui a empêché leur modèle de s'adapter de manière excessive. De plus, le modèle est formé à l'aide du dataset Imagenet qui regroupe près de 14 millions d'images réparties sur un millier de catégories.

La raison pour laquelle ce réseau est intuitif est que chaque couche convolutionnelle acquiert une représentation plus précise des images (feature map) que la précédente. Par exemple, la première couche peut distinguer des formes ou des couleurs très simples, tandis que la dernière peut distinguer des formes plus complexes telles que des visages complets, par exemple.

Selon certains experts, la victoire d'ImageNet en octobre **2012** a marqué le commencement d'une "**révolution de l'apprentissage profond**" qui a profondément bouleversé le secteur de l'**IA**.

2.5.3 VGGNet

Simonyan et **Zisserman** ont présenté l'architecture de réseau **VGG** dans leur article de **2014** intitulé "**Very Deep Convolutional Networks for Large Scale Image Recognition**"[18]. La raison de sa création est de diminuer le nombre de paramètres dans les couches CONV et d'améliorer le temps de formation.

VGGNet est composé de plusieurs versions (**VGG16**, **VGG19**, etc.) qui ne varient que par le nombre total de couches dans le réseau. Les spécificités structurales d'un réseau **VGG16** sont exposées ci-après.

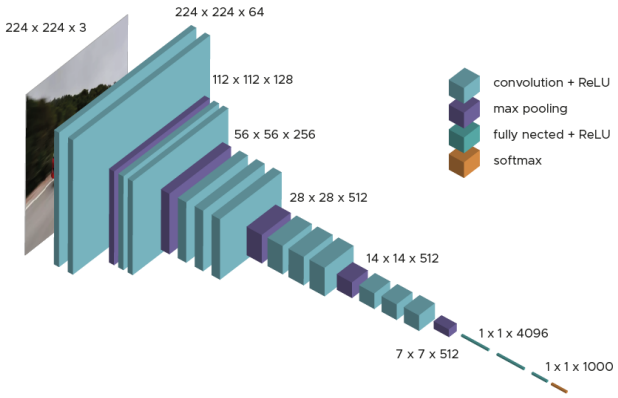
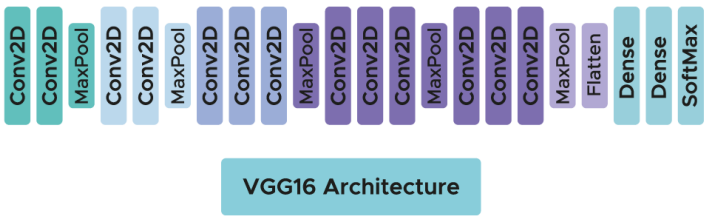


FIGURE 2.10 – VGGNet architecture



Le modèle a obtenu un score de précision de 92.7% sur ImageNet, ce qui en fait l'un des meilleurs. Par rapport aux modèles précédents, il a réalisé une avancée en proposant, dans les couches de convolution, des noyaux de convolution de plus petites dimensions (3x3) que ce qui avait été fait jusqu'alors. Le nombre de variables pour un filtre à 5x5 couches conv est de 25. En revanche, deux couches conv ayant une taille de noyau de 3x3 ont un total de 3x3x2=18 variables (une diminution de 28%). De la même manière, on peut obtenir l'effet d'une couche conv de 7x7 (11x11) en utilisant trois (cinq) couches convexes de 3x3 avec un pas de un. Cela diminue de 44.9% le pourcentage de variables entraînables (62.8%). Les variables entraînables sont plus nombreuses, ce qui permet un apprentissage plus rapide et plus solide.

Il a fallu des semaines pour entraîner le modèle en utilisant des cartes graphiques de pointe. Cependant, le réseau **VGG-16** comporte deux désavantages importants : La formation est lente et les charges de l'architecture du réseau sont elles-mêmes assez considérables.

2.5.4 FaceNet

FaceNet est une architecture de réseau de neurones convolutifs (**CNN**) conçu pour extraire des caractéristiques et reconnaître les visages en utilisant des méthodes d'apprentissage en profondeur. **Google Research** a suggéré cette architecture en **2015** dans un article intitulé "**FaceNet : A Unified Embedding for Face Recognition and Clustering**" par **Florian Schroff**, **Dmitry Kalenichenko** et **James Philbin**[19].

FaceNet a pour but principal de produire une image vectorielle compacte (dite "**embedding**") de chaque visage dans un espace euclidien, de manière à ce que les visages similaires soient proches les uns des autres et les visages différents soient éloignés les uns des autres dans cet espace. Le réseau neuronal acquiert cette représentation vectorielle à partir des images de visages.

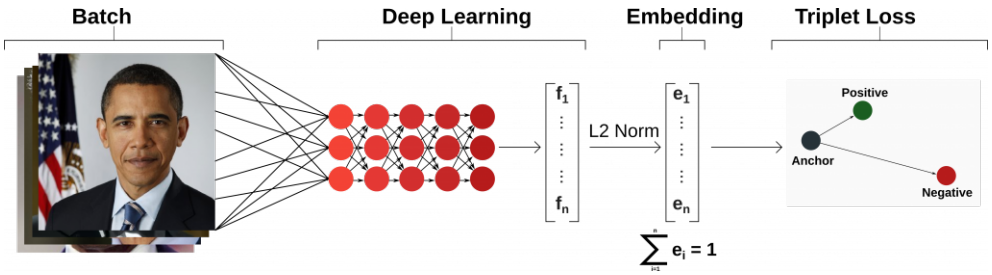


FIGURE 2.11 – Structure du modèle FaceNet

FaceNet collecte une représentation graphique d'un visage en entrée et produit un vecteur de **128 nombres** qui représentent les principales caractéristiques d'un visage. Dans le domaine de l'**apprentissage automatique**, ce vecteur est désigné sous le nom d'**embeddings** car il renferme toutes les informations essentielles d'une image. En principe, **FaceNet** extrait le visage d'un individu et le comprime dans un vecteur de **128 chiffres**. Dans l'idéal, les inclusions de visages similaires sont aussi similaires.

Selon les chercheurs, la méthode la plus précise est de laisser l'ordinateur sélectionner lui-même les mesures à recueillir. L'**apprentissage profond** est plus efficace que l'**intelligence humaine** pour identifier les parties du visage qu'il convient de mesurer.

FaceNet a pour objectif de créer un **réseau neuronal convolutif profond** et de l'entraîner à produire **128 mesures** pour chaque visage.

2.5.5 ResNet

ResNet ou **Residual Network**, est une architecture de réseau de neurones convolutifs (**CNN**) créé par **Kaiming He**, **Xiangyu Zhang**, **Shaoqing Ren** et **Jian Sun** de **Microsoft Research**. Dans l'article intitulé "**Deep Residual Learning for Image Recognition**" en 2015[20], cette architecture a été exposée et a bouleversé le domaine du **deep learning** en offrant la possibilité d'entraîner des réseaux beaucoup plus profonds que ce qui était auparavant envisageable.

La disparition du gradient est le principal obstacle à l'entraînement de **réseaux de neurones profonds**, car les gradients deviennent très petits à mesure qu'ils sont propagés vers l'arrière à travers de nombreuses couches, ce qui rend l'apprentissage difficile. Une solution novatrice à ce problème est proposée par **ResNet** en introduisant des connexions résiduelles. Grâce à ces liens, une partie du signal provenant de couches plus profondes peut être transmise sans être réduite, ce qui permet d'entraîner de manière efficace des réseaux beaucoup plus profonds.

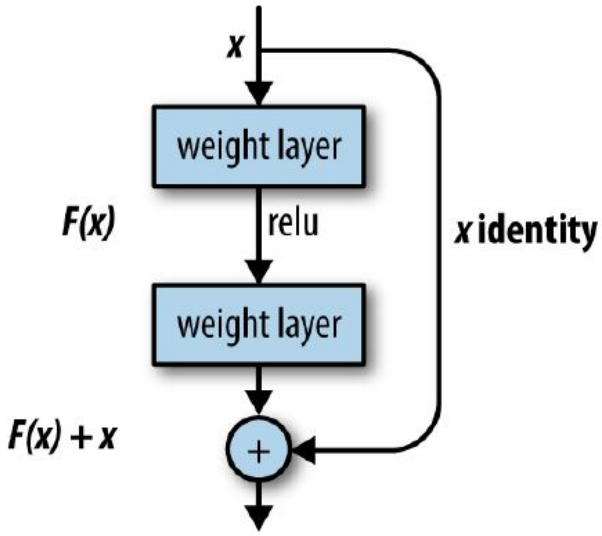


FIGURE 2.12 – Cellule ResNet

Ces connexions résiduelles permettent à **ResNet** d'acquérir des représentations plus approfondies et plus expressives des données, ce qui lui permet d'améliorer ses performances dans diverses tâches de vision par ordinateur, telles que la classification d'images, la détection d'objets et la segmentation sémantique.

2.5.6 VGGFace

L'architecture de réseau de neurones convolutifs **VGGFace** a été spécialement développée pour la **reconnaissance faciale**. Son architecture repose sur l'architecture **VGGNet**, créée par le **Visual Geometry Group** (**VGG**) à l'université d'Oxford.

L'architecture **VGGFace** a été créée dans le but d'acquérir des connaissances sur les représentations faciales discriminantes à partir d'un vaste inventaire de visages. Les couches convolutives profondes sont utilisées pour extraire des caractéristiques à partir des images de visages, tandis que les couches entièrement connectées sont utilisées pour la classification.

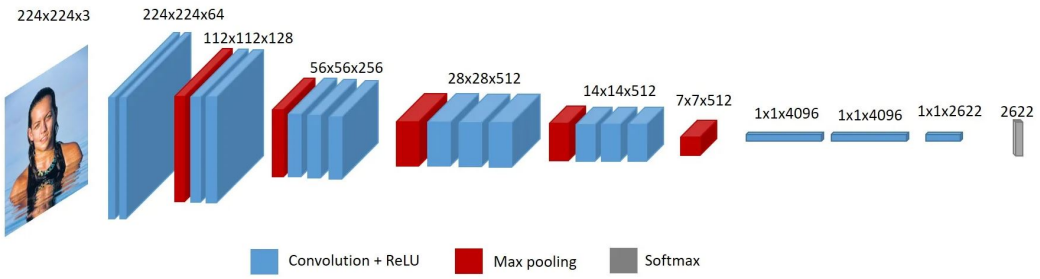


FIGURE 2.13 – VGGFace architecture

Deux modèles **VGG** principaux sont disponibles pour la **reconnaissance faciale** : **VGGFace** et **VGGFace2**.

Omkar Parkhi, dans son article de **2015** intitulé "**Deep Face Recognition**"[21], a expliqué le modèle **VGGFace**. Cet article présentait une description de la façon de créer un ensemble de données d'entraînement très vaste, indispensable à l'entraînement des systèmes de reconnaissance des visages basés sur des **réseaux neuronaux** à résolution moderne, afin de rivaliser avec les ensembles de données très étendues utilisés pour entraîner les modèles de **Facebook** et de **Google**. Par la suite, le jeu de données a servi de fondement au développement de réseaux **CNN** profonds. Il s'agit d'un modèle qui produit des propriétés généralisées à partir de visages en utilisant la fonction de triplet loss.

Dans l'article intitulé "**VGGFace2 : A dataset for recognizing faces across pose and age**", publié en **2017**[22], le modèle **VGGFace2** est présenté comme un jeu de données bien plus vaste qu'ils ont recueilli dans le but d'entraîner et d'évaluer des modèles de reconnaissance des visages plus performants. On entraîne des modèles sur ce jeu de données, notamment les architectures **ResNet-50** et **SENet**, et ils ont obtenu des performances exceptionnelles.

En bref, **VGGFace** est une architecture **CNN** dédiée à la **reconnaissance faciale**, qui exploite des méthodes de **deep learning** afin d'extraire des caractéristiques discriminantes à partir d'images de visages.

2.6 Reconnaissance des visages

Il est possible de réaliser la correspondance des visages en utilisant la **distance euclidienne**, la **similitude en cosinus** ou les **SVM** (**Support Vector Machine**).

2.6.1 Distance euclidienne

La **distance euclidienne** peut être employée dans le domaine de la **reconnaissance faciale** afin de mesurer la ressemblance entre les caractéristiques faciales extraites des images. Après avoir extrait les traits faciaux pour deux visages à comparer, on calcule la **distance euclidienne** entre les vecteurs de caractéristiques correspondants.

Une fois que les vecteurs de caractéristiques de deux visages ont été calculés, un seuil de décision est employé afin de déterminer si les visages sont à la même personne ou non. Dans le cas où la **distance euclidienne** est inférieure à un seuil préétabli, les visages sont considérés comme semblables et peuvent être attribués à celle-ci.

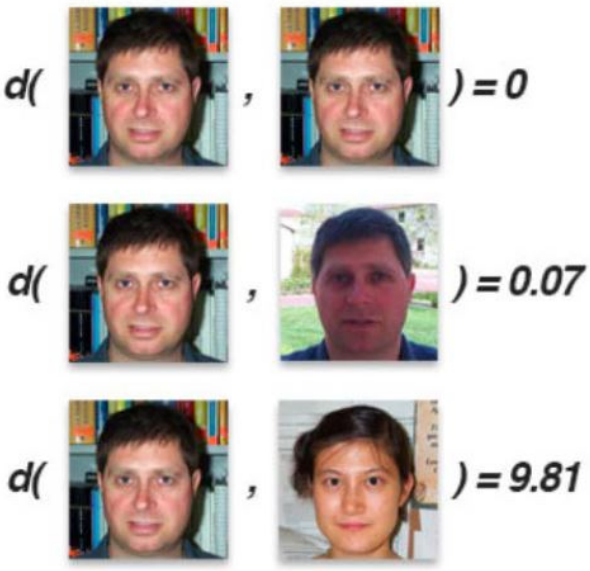


FIGURE 2.14 – Distance Euclidienne

Malgré sa simplicité et son utilisation répandue pour évaluer la similarité entre les **caractéristiques faciales**, la **distance euclidienne** n'est pas toujours la plus efficace, car elle peut être influencée par des facteurs tels que l'éclairage, l'orientation du visage, etc. On étudie également des méthodes plus avancées, telles que l'emploi de réseaux de neurones siamois ou de métriques d'apprentissage de distances, pour la reconnaissance faciale.

2.6.2 Similitude en cosinus

La **similitude en cosinus** est une mesure de la similarité entre deux vecteurs dans un espace vectoriel. Cette mesure est souvent utilisée pour comparer les caractéristiques faciales extraites des images.

La **similitude en cosinus** est souvent préférée dans certains contextes de **reconnaissance faciale** car elle est relativement insensible à la magnitude des vecteurs de caractéristiques, ce qui peut être un avantage lorsque les caractéristiques ont été normalisées.

Cependant, comme pour toute méthode de comparaison de caractéristiques, il est important de choisir judicieusement les caractéristiques à extraire et de définir des seuils appropriés pour la prise de décision afin d'obtenir des performances optimales.

2.6.3 SVM

Les **SVM** (**Support Vector Machines**) sont des algorithmes d'**apprentissage supervisé** utilisés dans divers domaines, y compris la reconnaissance faciale. Voici une procédure générale pour utiliser les **SVM** dans le contexte de la reconnaissance faciale :

1. **Collecte des données d'entraînement** : La première étape consiste à collecter un ensemble de données d'images faciales. Ces images doivent être étiquetées avec les identités des personnes représentées.

2. **Pré-traitement des données** : Les images peuvent subir diverses transformations pour normaliser l'éclairage, la rotation, la taille, etc. Cela garantit que les **SVM** peuvent apprendre des caractéristiques pertinentes indépendamment des variations dans les images.

3. **Extraction de caractéristiques** : Les **SVM** fonctionnent mieux avec des données numériques. Par conséquent, les images faciales doivent être converties en vecteurs de caractéristiques numériques. Des techniques telles que la méthode de Haar, ou les réseaux de neurones convolutifs (**CNN**) peuvent être utilisées pour extraire ces caractéristiques.

4. **Entraînement du SVM** : Une fois les caractéristiques extraites, le **SVM** est entraîné sur ces données. L'objectif est de trouver la meilleure séparation entre les différentes classes d'images faciales (par exemple, différentes personnes) dans l'espace des caractéristiques.

5. **Évaluation du modèle** : Une fois que le **SVM** est entraîné, il est évalué sur un ensemble de données de test distinct pour mesurer ses performances en termes de reconnaissance faciale. Des métriques telles que la **précision**, le **rappel** et la **F-score** peuvent être utilisées pour évaluer la qualité du modèle.

6. **Utilisation du modèle** : Une fois que le modèle a été évalué et jugé satisfaisant, il peut être utilisé pour la reconnaissance faciale en temps réel ou sur des images statiques. Lorsqu'une nouvelle image est présentée au modèle, il prédit l'identité de la personne représentée en fonction de son apprentissage antérieur.

Chapitre 3
API de Reconnaissance Faciale

3.1 Outils et Technologies

3.1.1 Anaconda

Anaconda est une distribution open source des langages de programmation **Python** et **R**, utilisée principalement dans les domaines du **calcul scientifique**, de l'**analyse de données** et de l'**apprentissage automatique**. Cette distribution est populaire car elle comprend une large gamme de bibliothèques et d'outils pré-installés, ce qui facilite le démarrage des projets de développement dans ces domaines.

Grâce à Anaconda, les utilisateurs peuvent créer des environnements virtuels isolés pour leurs projets, ce qui leur permet de gérer facilement les dépendances logicielles et de maintenir la cohérence des versions entre différents projets. De plus, **Anaconda** propose un gestionnaire de paquets appelé **conda**, qui permet d'installer, de mettre à jour et de gérer les packages **Python** et **R**, ainsi que leurs dépendances.

3.1.2 JupyterLab

JupyterLab est une interface utilisateur interactive pour le **Jupyter Notebook**, qui est un environnement de programmation interactif open source largement utilisé dans le domaine de la **science des données**, de l'**analyse de données** et de l'**apprentissage automatique**. Contrairement au **Jupyter Notebook** classique, qui offre une seule page de blocs-notes avec un ensemble limité de fonctionnalités d'édition, **JupyterLab** fournit une interface utilisateur plus avancée et modulaire.

Dans **JupyterLab**, les utilisateurs peuvent organiser leurs travaux dans une disposition flexible avec des onglets, des fenêtres et des panneaux déplaçables. Il prend en charge l'édition de texte, de code, de graphiques, de fichiers et bien plus encore, le tout dans une seule interface intégrée. De plus, il offre un large éventail d'extensions qui permettent aux utilisateurs de personnaliser leur environnement de développement selon leurs besoins spécifiques.

3.1.3 Python

Python est un langage de programmation de haut niveau, polyvalent et interprété. Il est largement utilisé dans de nombreux domaines, notamment le **développement web**, l'**automatisation de tâches**, le **calcul scientifique**, l'**analyse de données**, l'**intelligence artificielle** et bien plus encore.

En raison de sa simplicité, de sa polyvalence et de sa robustesse, **Python** est devenu l'un des langages de programmation les plus populaires et les plus largement utilisés dans le monde entier.

3.1.4 Keras

Keras est une bibliothèque open source haut niveau pour l'**apprentissage en profondeur**, écrite en **Python**. Elle fournit une interface conviviale et intuitive pour construire, entraîner, évaluer et déployer des modèles d'**apprentissage en profondeur**, en mettant l'accent sur la rapidité de prototypage.

3.1.5 OpenCV

OpenCV (**Open Source Computer Vision Library**) est une bibliothèque open source spécialisée dans le **traitement d'images** et la **vision par ordinateur**. Elle offre un ensemble d'outils et de fonctionnalités pour effectuer diverses tâches liées au traitement d'images, telles que la manipulation d'images, la détection et la reconnaissance d'objets, le suivi de mouvement, la segmentation d'images, la stéréovision, et bien plus encore.

3.1.6 Scikit-learn

Scikit-learn est une bibliothèque open source de **machine learning** pour le langage de programmation **Python**. Elle offre une vaste gamme d'outils et d'algorithmes pour effectuer des tâches d'**apprentissage automatique** telles que la classification, la régression, le clustering, la réduction de dimensionnalité, la sélection de modèle, et bien plus encore.

3.1.7 Numpy

NumPy, abréviation de "**Numerical Python**", est une bibliothèque open source pour **Python**, principalement utilisée pour effectuer des calculs numériques et des opérations sur des tableaux multidimensionnels. Elle fournit des fonctionnalités pour la manipulation efficace de données, la génération de nombres aléatoires, le traitement de signaux, la transformation de Fourier, et bien plus encore.

3.1.8 Matplotlib

Matplotlib est une bibliothèque de visualisation de données en **Python**, conçue pour produire des graphiques de haute qualité pour une grande variété de données et de formats de sortie. Elle offre une grande flexibilité pour créer une large gamme de graphiques, notamment des graphiques linéaires, des graphiques à barres, des histogrammes, des nuages de points, des diagrammes circulaires, des graphiques en boîte, des graphiques 3D, et bien plus encore.

3.1.9 Pickle

En **Python**, **pickle** est un module intégré qui permet de sérialiser et de désérialiser des objets **Python**. La **sérialisation** est le processus de conversion d'un objet Python en une représentation de flux d'octets, tandis que la **désérialisation** est le processus inverse, c'est-à-dire la conversion de la représentation de flux d'octets en un objet **Python**.

3.1.10 Streamlit

Streamlit est une bibliothèque open source en **Python** qui permet aux développeurs de créer des applications web interactives pour l'analyse de données et la visualisation sans avoir à écrire de code HTML, CSS ou JavaScript. Il simplifie le processus de création d'applications web en transformant les scripts **Python** en applications web interactives.

3.2 Méthodes utilisés

3.2.1 Détection de visage

MTCNN, ou **Multi-Task Cascaded Convolutional Neural Network**, est un algorithme utilisé pour la détection de visages dans des images. Contrairement aux approches traditionnelles qui nécessitent plusieurs étapes distinctes pour détecter un visage (par exemple, la détection de contours suivie de la localisation des visages), **MTCNN** combine ces étapes en une seule **architecture de réseau neuronal**.

MTCNN est efficace pour détecter des visages dans une grande variété de conditions, y compris des images de résolution variable, des angles de vue différents, et même des visages partiellement cachés ou obscurcis. Son architecture en cascade lui permet d'être à la fois précis et rapide, en faisant de lui un choix populaire pour les applications de reconnaissance faciale, de surveillance vidéo, et d'analyse d'images.

3.2.2 Extraction des caractéristiques

FaceNet est un **système de reconnaissance faciale** basé sur l'**apprentissage en profondeur** (**deep learning**), développé par **Google**. Son objectif principal est de représenter les visages de manière à ce que les distances entre les représentations correspondent à des distances significatives dans l'espace des caractéristiques faciales.

Cette représentation permet ensuite d'effectuer des tâches de **reconnaissance faciale** telles que la **vérification** ou l'**identification**.

FaceNet est utilisé pour extraire des caractéristiques faciales significatives à partir d'images de visages. Ces caractéristiques sont représentées sous forme de vecteurs numériques compacts et représentent les attributs distinctifs du visage qui sont importants pour la reconnaissance faciale.

FaceNet est capable de générer des vecteurs de caractéristiques hautement discriminants et invariants, ce qui le rend extrêmement précis pour la **reconnaissance faciale**, même dans des conditions variables telles que les changements d'éclairage, les expressions faciales et les angles de vue différents. C'est pourquoi il est largement utilisé dans de nombreuses applications de reconnaissance faciale, de la sécurité aux réseaux sociaux en passant par la recherche médicale.

3.2.3 Identification

On va utilise **SVM** (**Support Vector Machine**) pour identifier les visages des personnes. **SVM** est souvent utilisé en combinaison avec des techniques d'extraction de caractéristiques comme **FaceNet**. **FaceNet** extrait les caractéristiques faciales et produit des vecteurs de caractéristiques, puis **SVM** est utilisé pour classifier ces vecteurs en fonction des identités associées.

SVM peut être un choix efficace pour l'identification dans la **reconnaissance faciale** en utilisant des vecteurs de caractéristiques pour représenter les visages et en apprenant à classifier ces vecteurs en fonction des identités correspondantes lors de l'entraînement.

3.3 API

Afin de déployer notre model, nous avons crée une **application web** avec la bibliothèque **Python Streamlit**. L'interface qui s'affiche après le démarrage de l'application est l'**objectif du projet**. Dans la partie de **reconnaissance faciale** contient un lien pour charger l'image et un bouton "**run**" pour appliquer l'opération.

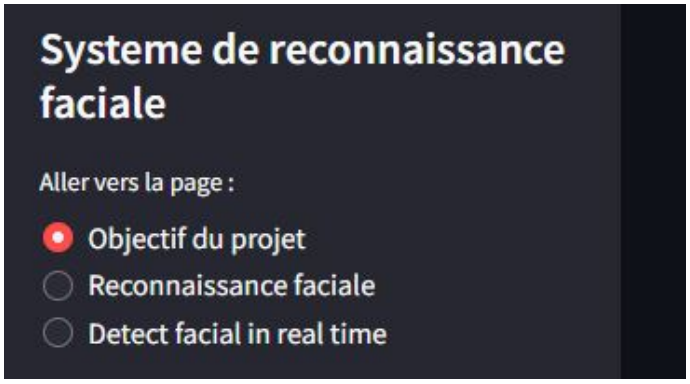


Figure 3.1 – API - Objectif

On charge l'image pour faire la reconnaissance faciale.

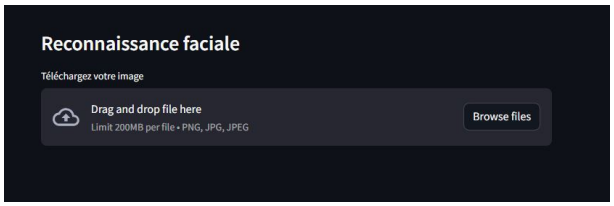


FIGURE 3.2 – API - Chargement

on appuie sur le bouton "**run**" pour lancer l'opération.

FIGURE 3.3 – API - bouton run

On affiche le résultat trouvé.

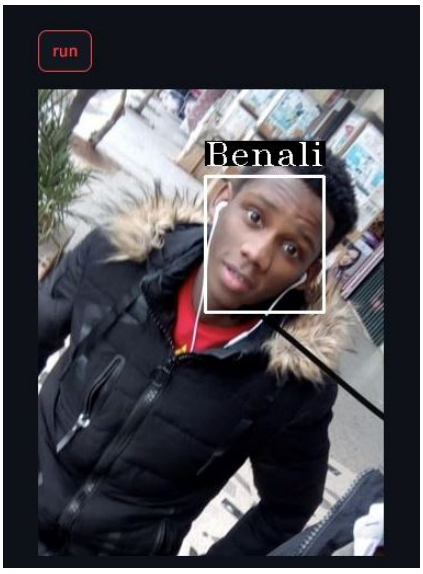


FIGURE 3.4 – API - Résultat

Résultat d'une autre visage inconnu

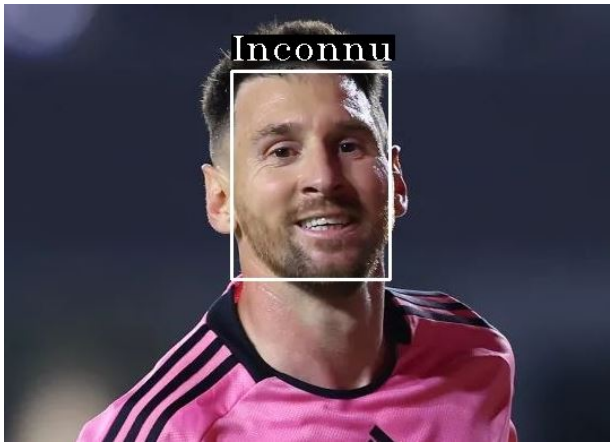


FIGURE 3.5 – API - Résultat

La dernière partie est la **reconnaissance faciale en temps réel**. Cliquer sur le bouton "**run**" pour lancer le **webcam** et pour **fermer** le **webcam** appuyer sur le bouton "**Q**" du clavier.

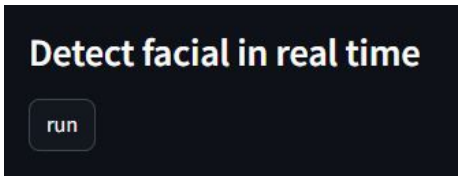


FIGURE 3.6 – API - Detect

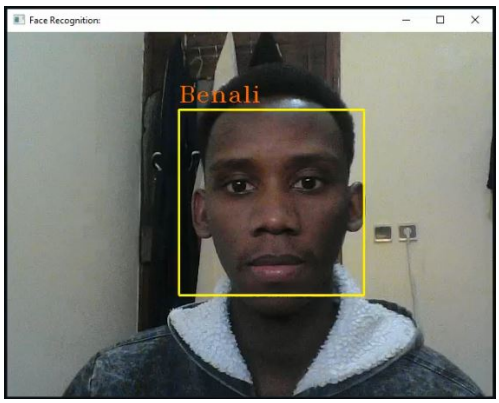


FIGURE 3.7 – Face - Benali

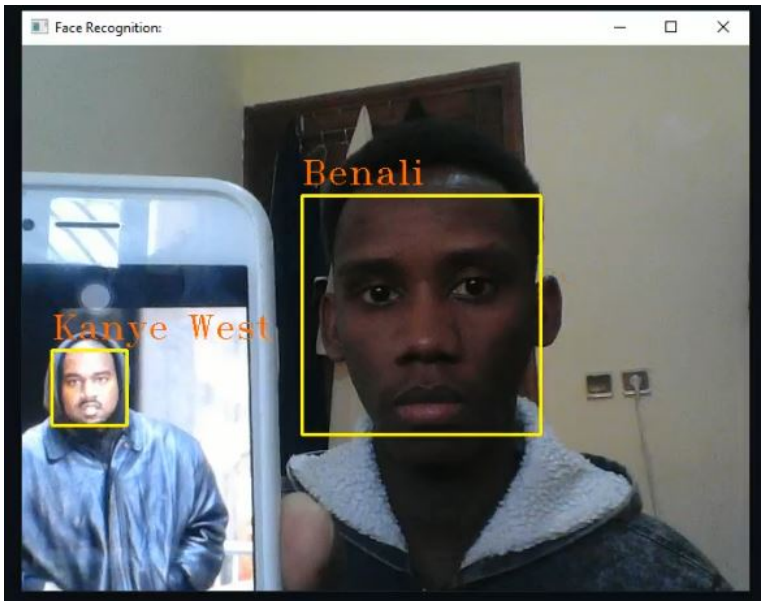


FIGURE 3.8 – Face - Kanye West, Benali

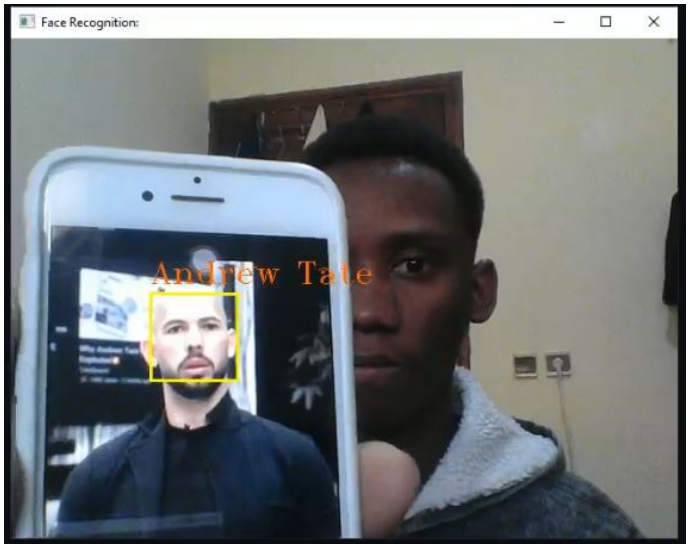


FIGURE 3.9 – Face - Andrew Tate

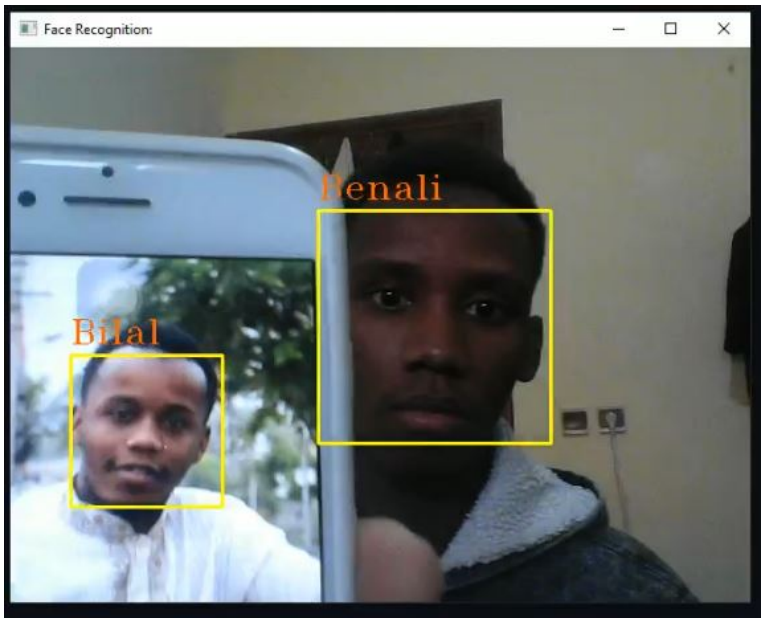


FIGURE 3.10 – Face - Bilal, Benali

Conclusion Générale

La technologie en plein essor de la **reconnaissance faciale** présente des opportunités incroyables dans des domaines aussi divers que la sécurité, la gestion des identités et même les interactions quotidiennes. En offrant la possibilité d'identifier automatiquement les individus en se basant sur leurs **caractéristiques faciales uniques**, elle assure une amélioration de l'efficacité, une sécurité renforcée et une expérience utilisateur améliorée. Toutefois, son utilisation pose des interrogations essentielles concernant la vie privée, le consentement et le risque de discrimination. Afin de tirer le meilleur parti de cette technologie émergente tout en réduisant les risques, il est essentiel d'adopter des politiques de gouvernance solides, de mettre en place des mesures de sécurité solides et de favoriser une prise de conscience accrue des enjeux éthiques liés à cette technologie.

Le problème reste complexe et non encore pleinement résolu, même si tous les travaux sont réalisés ces dernières années. Différents défis se posent lors de cette tâche d'**identification** et chacun d'entre eux n'est pas banal. La performance d'un système est influencée par de nombreuses conditions réelles, mais la détection automatique des visages a un impact considérable sur les performances du module d'**identification**.

L'objectif de ce mémoire de Master était de concevoir une **API** qui permettrait de **reconnaître** et d'**identifier** les visages humains en utilisant l'**intelligence artificielle**, l'**apprentissage automatique**, l'**apprentissage profond** et les **réseaux de neurones**.

La **détection des visages** a été réalisée en utilisant **MTCNN**, puis **FaceNet** pour extraire les **caractéristiques faciales**, et un modèle **SVM** pour **identifier** les visages.

Grâce à ce projet, nous avons aussi pu développer des compétences solides en programmation et en **traitement d'image**, tout en nous familiarisant avec les domaines de l'**IA**, du **ML** et du **DL**.

Références

[1] A Facial Recognition Project Report, Bledsoe, Woodrow Wilson and others, 1963
https ://archive.org/details/firstfacialrecognitionresearch/FirstReport/mode/2up

[2] Security of Biometric Systems, Milan Adámek, Miroslav Matýsek, Petr Neumann, 2025
https ://www.sciencedirect.com/science/article/pii/S1877705815003823

[3] Vision par ordinateur
https ://www.ibm.com/fr-fr/topics/computer-vision

[4] Système de reconnaissance faciale
https ://www.kaspersky.fr/resource-center/definitions/what-is-facial-recognition

[5] Reconnaissance vocale
https ://www.ibm.com/fr-fr/topics/speech-recognition

[6] Empreintes digitales
https ://www.interpol.int/fr/Notre-action/Police-scientifique/Empreintes-digitales

[7] Authentification biométrique
https ://www.onespan.com/fr/topics/authentification-biometrique

[8] Caractéristiques de la biométrie
https ://www.preventica.com/dossier-surete-biometrie-caracteristiques.php

[9] Système de recommandation
https ://www.smalsresearch.be/introduction-aux-systemes-de-recommandation/

[10] Low-dimensional procedure for the characterization of human faces
https ://opg.optica.org/josaa/abstract.cfm ?uri=josaa-4-3-519

[11] Rapid object detection using a boosted cascade of simple features
https ://ieeexplore.ieee.org/document/990517

[12] Histograms of oriented gradients for human detection
https ://ieeexplore.ieee.org/abstract/document/1467360

[13] Caffe | Deep Learning Framework
https ://caffe.berkeleyvision.org/

[14] Dlib
http ://dlib.net/

[15] Eigenfaces for Face Recognition
https ://pyimagesearch.com/2021/05/10/opencv-eigenfaces-for-face-recognition/

[16] Analyse en Composantes principales
https ://datascientest.com/acp

[17] GradientBased Learning Applied to Document Recognition
https ://ieeexplore.ieee.org/abstract/document/726791

[18] Very Deep Convolutional Networks for Large Scale Image Recognition
https ://arxiv.org/pdf/1409.1556

[19] FaceNet : A Unified Embedding for Face Recognition and Clustering
https ://arxiv.org/pdf/1503.03832

[20] Deep Residual Learning for Image Recognition
https ://arxiv.org/pdf/1512.03385

[21] Deep Face Recognition
https ://www.robots.ox.ac.uk/ vgg/publications/2015/Parkhi15/parkhi15.pdf

[22] VGGFace2 : A Dataset for Recognising Faces across Pose and Age
https ://ieeexplore.ieee.org/abstract/document/8373813