

**Université de Gafsa**  
**Institut Supérieur des Sciences Appliquées et de Technologie de GAFSA**  
**Département d'Automatique Des Systèmes Industriels**



**Développement d'un système de sécurité et supervision par la  
reconnaissance faciale**

Présenté et soutenu par :

**Gouader Abir**

En vue de l'obtention de

**Licence en Technologies de l'Information et de la Communication**

Sous la Direction de :

**Mr. Ben Hessine Moez**

Soutenu le .....

Devant le jury composé de :

**Président :**

**Rapporteur :**

**Membres :**

**2022/2023**

## **Remerciement :**

Avant d'entamer la rédaction de me projet fin d'étude, je tiens à présenter mes respects, mes remerciements à tous ceux qui par leurs collaborations, leurs aides qui ont permis d'achever l'étude de ce projet.

Tous d'abord, je tiens de remercier le Directeur de l'ISSAT Gafsa ainsi que le chef de département : « Hamdi Ilyes »

Je présente également mes respects et profonds remerciement à mon encadreur : « Hadj Abdallah Saber » pour l'aide précieux qui m'a apporté tout au long de ce travail ; notamment les conseils de recherche et l'aide alloués pour l'achèvement de ce travail. Je remercie aussi Monsieur « Ben Hessine Moez » pour sa confiance et son professionnalisme.

Enfin, mes remerciement à tous les enseignements de l'ISSAT qui ont été ma source des connaissances et aussi avec fidèles dans leurs nobles missions durant notre études universitaires.

## **Dédicace :**

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

Je dédie ce travail aussi à **mon frère et mes sœurs** pour ses encouragements.

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

# Liste des abréviations

**RIO** : Région d'intérêt connue

**TAR** : True Acceptance Rate

**FAR** : False Acceptance Rate

**FRR** : False Reject Rate

**IA** : Intelligence Artificielle

**ML** : Machine Learning

**DL** : Deep Learning

**HAAR** : Repense aigue à haute attitude

**HDMI** : High Definition Multimedia Interface

**GPIO** : General-Purpose Input/Output

**RAM** : Random Access Memory

**USB** : Universal Serial Bus

**SMSC** : Short Message Service Center

**LAN** : Local Area Network

**OpenCV** : Open source Computer Vision

## Table de figures

Figure 1 Présentation d'un système de sécurité par la reconnaissance faciale .....	2
Figure 2 Les catégories de la biométrie .....	4
Figure 3 L'intelligence artificielle .....	9
Figure 4 L'apprentissage supervisé et l'apprentissage non supervisé .....	10
Figure 6 Représentation de perceptron .....	15
Figure 5 Neurone artificiels .....	15
Figure 7 Exemple de fonctionnement similaire de Haar-Cascade .....	17
Figure 8 Exemple de calcul d'image intégrale.....	18
Figure 9 Cascades classifiées .....	19
Figure 10 Raspberry Pi .....	20
Figure 11 Les composants standard d'un Raspberry Pi 4 modèle B .....	23
Figure 12 Alimentation type C .....	23
Figure 13 MicroSD.....	24
Figure 14 Formatage de la carte MicroSD .....	29
Figure 16 Logiciel Win32 DiskImage .....	31
Figure 15 Décompression du fichier Zip .....	30
Figure 17 Copie le fichier Raspbian sur la carte SD .....	31
Figure 18 bureau de Raspberry Pi .....	32
Figure 19 Le terminal.....	32
Figure 20 La transformation de Keyboard de gb vers fr .....	33
Figure 21 Menu des applications .....	33
Figure 22 Réglage de WiFi.....	34
Figure 24 L'activation d'interfaces utilisées .....	35
Figure 23 Modifier le mot de passe.....	34
Figure 25 Schéma fonctionnel du système de reconnaissance faciale.....	37
Figure 26 Création d'un répertoire de travail.....	38
Figure 27 Fichier HaarCascade .....	38
Figure 28 Création de dossier d'image datab.....	39
Figure 29 Les images enregistrées dans le répertoire datab .....	41
Figure 30 Création de dossier codage .....	42
Figure 31 Fichier d'entraînement.....	45

## Table de tableaux

Tableau 1 Les différences entre ML et DL .....	14
Tableau 2 Le modèle A .....	20
Tableau 3 Le modèle B.....	22
Tableau 4 Le modèle Zéro.....	22

## Table de matières

Introduction Générale :	1
Chapitre 1 : La Reconnaissance Faciale	2
1. Introduction :	2
2. La sécurité :	3
3. La biométrie	3
3.1. Fonctionnement de la biométrie	3
3.1.1. Application d'identification	3
3.1.2. Application d'authentification	4
3.2. Types de la biométrie	4
4. Système de reconnaissance faciale	4
4.1. Historique	5
4.2. Les phases d'un système de reconnaissance faciale	5
4.2.1. L'acquisition	5
4.2.2. Le prétraitement	5
4.2.3. Détection de visage	5
4.2.4. L'extraction des caractéristiques	6
4.2.5. La classification	6
4.2.6. La décision	6
4.3. Les avantages et les inconvénients de la reconnaissance faciale	6
4.3.1. Les avantages	6
4.3.2. Les inconvénients	7
4.4. Etats de l'art des techniques de reconnaissance faciale	7
4.4.1. Approches globales	7
4.4.2. Approches locales	8
4.4.3. Approche hybride	8
4.5. Mesure de performance d'un système de reconnaissance faciale	8
5. L'intelligence artificielle	9

5.1. L'apprentissage automatique .....	9
5.1.1. Les différents types de l'apprentissage automatique .....	10
5.1.2. Le fonctionnement de l'apprentissage automatique .....	11
5.2. L'apprentissage profond.....	12
5.2.1. Le fonctionnement de Deep Learning .....	12
5.3. Les réseaux de neurones.....	15
6. Conclusion .....	16
Chapitre 2 : Etude des systèmes de reconnaissance faciale .....	17
1. Introduction.....	17
2. Procédé de détection de Haar-Cascade .....	17
2.1. Haar comme caractéristique.....	17
2.2. Image intégrale .....	18
2.3. AdaBoost.....	18
2.4. Classificateur en Cascade.....	18
3. Environnement de développement : .....	19
3.1. Environnement Hardware : .....	19
3.1.2. Les composants du Raspberry Pi 4 modèle B (11) .....	23
3.1.3. Connexion des composants d'un Raspberry Pi.....	23
4. Environnement software : .....	24
4.1. Raspbian .....	24
4.2. Langage Python .....	25
4.3. Bibliothèques utilisées.....	25
4.4. Firebase.....	26
5. Conclusion .....	27
Chapitre 3 : Conception et Réalisation .....	28
1. Introduction.....	28
2. Le système de sécurité par la reconnaissance faciale.....	29
2.1. Préparation du système d'exploitation sur Micro SD .....	29
2.1.1. Formater la carte SD .....	29



2.1.2. Installer Rasbian sur la carte SD .....	30
2.2. Premier démarrage du système .....	31
2.2.1. Le clavier en français.....	32
2.2.2. Configuration de la localisation .....	33
2.2.3. Modification du mot de passe .....	34
2.2.4. Activation des interfaces utilisées.....	34
2.2.5. Mettre à jour le système de Raspberry Pi.....	35
2.3. Installation d'OpenCV sur le Raspberry Pi .....	36
2.4. Système de reconnaissance faciale.....	36
3. La supervision par l'application mobile.....	50
4. Conclusion .....	50
Conclusion générale.....	51

# **Introduction Générale :**

De nos jours, il existe de plus en plus d'insécurité dans divers secteurs, pour cela plusieurs moyens informatiques sont mis en œuvre pour contrer cette tendance. Un de ces moyens moderne et le système de reconnaissance faciale ce système et très utile dans les entreprises, les établissements publics et les résidences pour contrôler l'accès à une zone sécurisée ou pour surveiller l'entrée et la sortie de personnes.

Un système de reconnaissance faciale est une application logicielle visant à reconnaître une personne grâce à son visage de manière automatique. C'est un domaine de la vision par ordinateur consistant à reconnaître automatiquement une personne à partir d'une image de son visage à partir d'un ensemble de visage stockés dans une base de données.

Les systèmes à base de reconnaissance de visage présentent de nombreuses applications en vidéosurveillance, biométrie, robotique, indexation d'image et de vidéo, recherche d'images par le contenu, ect.

Ils sont appréciés car considérés comme peu invasifs, en comparaison avec les autres systèmes biométrique (empreintes digitales, reconnaissance de l'iris...). Le fonctionnement de ces systèmes se base sur une ou plusieurs caméras pour reconnaître l'utilisateur

Il s'agit d'un sujet particulièrement étudié en vision par ordinateur, avec de très nombreuses publications et brevets et de conférences spécialisées.

Ce projet de fin d'étude est formé de trois chapitres, le premier est une présentation d'un état de l'art sur la technologie de reconnaissance faciale, et comme solution on propose et on décrit notre système à faible coût dans un deuxième chapitre ainsi les différentes composantes utilisées dans ce projet. Le troisième chapitre est dédié à la présentation des étapes de réalisation pratique du système, ses codes sources, ses modes de fonctionnements, ainsi que la mise en place d'une maquette présentant notre système.

# Chapitre 1 : La Reconnaissance Faciale

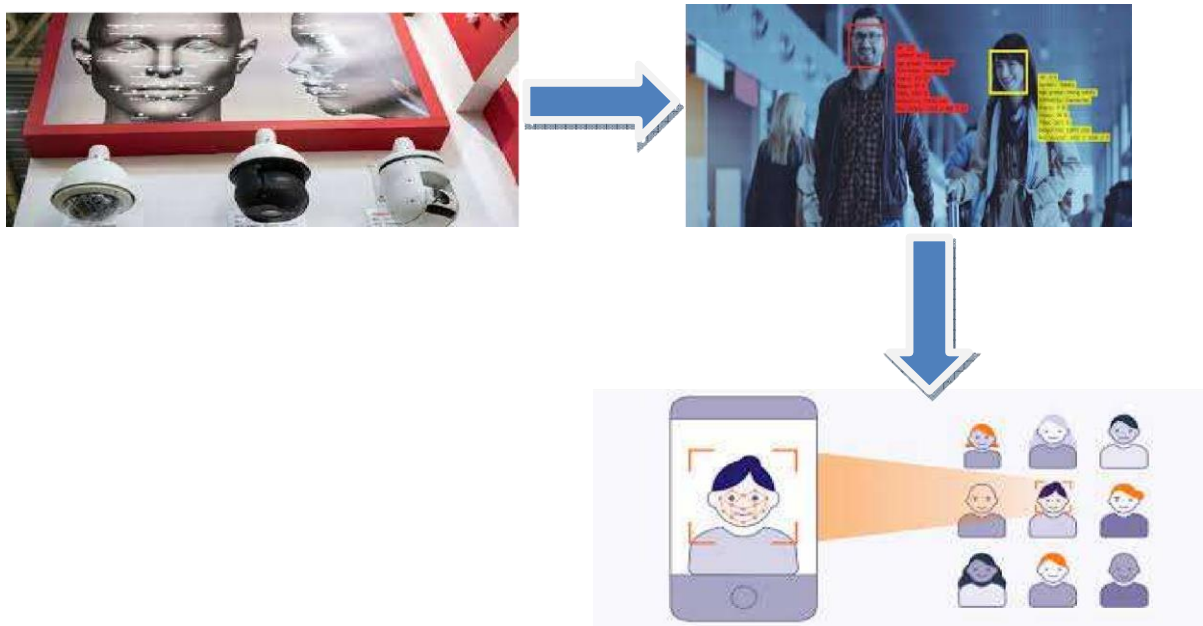
## 1. Introduction :

La biométrie est une technologie qui permet d'identifier des individus en se basant sur des caractéristiques physiques ou comportementales unique.

La reconnaissance faciale est l'une des applications les plus courantes de la biométrie. Elle utilise des algorithmes de l'intelligence pour identifier les individus à partir de leur visage. Pour cela, elle analyse les traits du visage (forme de nez, de la bouche, des yeux...) et les compare avec une base de données d'images préalablement enregistrées.

L'intelligence artificielle joue un rôle clé dans la reconnaissance faciale, car elle permet d'optimiser la précision de l'identification en s'adaptant aux variations de l'environnement (luminosité, angle de vue...) et on apprenant à reconnaître de nouveaux visages.

Le système de sécurité intègre une technologie avancée de reconnaissance faciale avec une application mobile pour offrir une solution complète et pratique (figure1).



**Figure 1 :** Présentation d'un système de sécurité par la reconnaissance faciale

## **2. La sécurité :**

La sécurité par la reconnaissance fait référence à l'utilisation de la reconnaissance des personnes, des voix, des visages ou d'autres caractéristiques uniques pour garantir la sécurité. Il s'agit d'une forme de biométrie qui utilise des caractéristiques physiques ou comportementales pour identifier les individus et leur donner accès à des systèmes ou des zones sécurisées.

La sécurité par la reconnaissance puisse offrir de nombreux avantages, elle peut également présenter des risques pour la vie privée et la sécurité. Par exemple, la reconnaissance faciale peut être utilisée pour suivre les mouvements des personnes sans leur consentement, et les données biométrique stockées peuvent être volées ou piratées.

Il est donc important de mettre en place des mesures de sécurité appropriées pour protéger la donnée biométrique et de s'assurer que l'utilisation de la reconnaissance est conforme aux réglementations en matière de confidentialité et de protection des données.

## **3. La biométrie**

La biométrie regroupe l'ensemble des techniques informatique visant de reconnaître automatiquement un individu à partir de ces caractéristiques physique, biologique, voire comportementales (1).

Les données biométrique sont des données à caractère personnel car elles permettent d'identifier une personne (1).

### **3.1. Fonctionnement de la biométrie**

La biométrie comporte deux fonctionnements l'application d'identification et l'application d'authentification :

#### **3.1.1. Application d'identification**

L'identification consiste à déterminer l'identité d'une personne.

Il s'agit de saisir une donnée biométrique de cette personne, en prenant par exemple une photo de son visage, en enregistrant sa voix, ou en captant l'image de son empreinte digitale. Ces données son ensuite comparées aux données biométriques de plusieurs autres personnes qui figurent dans une base (2).

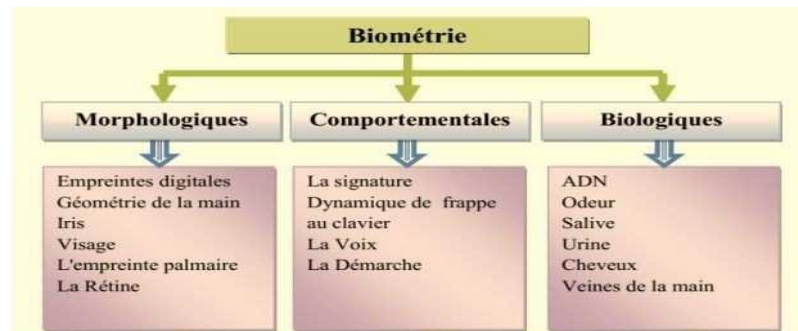
### 3.1.2. Application d'authentification

L'authentification, appelée également vérification, est le processus qui consiste à comparer les données caractéristiques provenant d'une personne, au modèle de référence biométrique de cette dernière, afin de déterminer la ressemblance. Le modèle de référence est probablement enregistré et stocké dans une base de données, dans un équipement ou objet personnel sécurisé (2).

### 3.2. Types de la biométrie

Les techniques biométriques peuvent être classées en trois catégories (figure 2).

La première repose sur les techniques d'analyse des comportements. Cette dernière concerne l'étude des actions répétitives et usuelles des personnes. La seconde catégorie est basée sur les techniques d'analyse de la morphologie humaine. Ces dernières utilisent comme moyen d'analyse les empreintes digitales, la forme géométrique de la main, les traits du visage, le dessin du réseau veineux. La troisième catégorie est basée sur tout ce qui est relié à la biologie humaine et qui est stable et permanent pendant toute la durée de la vie.



**Figure 2** Les catégories de la biométrie

## 4. Système de reconnaissance faciale

La reconnaissance faciale est un moyen d'identifier ou de confirmer l'identité d'un individu grâce à son visage. Les systèmes de reconnaissance faciale peuvent servir à l'identification de personnes sur des photos, dans les vidéos ou en temps réel (3).

#### **4.1. Historique**

La reconnaissance faciale est une technique biométrique relativement récente. Si l'empreinte digitale est la technique biométrique la plus ancienne inventée en 1903 pour rechercher les criminels, la reconnaissance des visages a été développée par «Benton et Van Allen » en 1968 pour évaluer la capacité d'identification des visages non familiers. Il ne s'agit pas d'un test de reconnaissance ménisque de visage familiers ou non familiers, mais d'une épreuve consistant à appairer des photographies de visage non familiers présentés sous différents éclairages et selon des angles différents et nécessitant une bonne capacité d'intégration Visio-spatiale (4).

L'utilisation des techniques de reconnaissance faciale a connu un développement à grande échelle depuis le milieu des années90, avec l'utilisation efficace de nouvelles technologies, notamment l'ordinateur et sa capacité de traitement d'images. L'utilisation de ces technique existe depuis qu'une machine est capable de comprendre ce qu'elle « voit » lorsqu'on le connecte à une ou plusieurs caméras, c.-à-d. que les premiers essais datent du début des années70 (Benton et Van Allen en 1968), et sont basés sur des méthodes à bases d'heuristiques, basés sur des attributs faciaux mesurables comme l'écartement des yeux, des sourcils, des lèvres, la position du menton, la forme, ect (5).

Ces méthodes sont très peu robustes, car elles font de nombreuses suppositions en se plaçant dans des cas très simples (visage de face, bonnes condition d'illuminations, ect) . L'une des premières tentatives de reconnaissance de visage est faite par Takeo Kanade en 1973.

#### **4.2. Les phases d'un système de reconnaissance faciale**

Il existe 6 phases de la reconnaissance faciale :

##### **4.2.1. L'acquisition**

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

##### **4.2.2. Le prétraitement**

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

#### **4.2.3. Détection de visage**

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

#### **4.2.4. L'extraction des caractéristiques**

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

#### **4.2.5. La classification**

Elle consiste à concevoir un des modèles à partir des caractéristiques d'un visage ou d'un ensemble de visages extraites dans la phase précédente en se basant sur les caractéristiques qu'ils ont en commun. Un modèle est un ensemble d'information unique un ou plusieurs individus possédant des caractères communs, ces individus sont groupés dans une même classe.

#### **4.2.6. La décision**

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

### **4.3. Les avantages et les inconvénients de la reconnaissance faciale**

#### **4.3.1. Les avantages**

La reconnaissance faciale présente plusieurs avantages lors du processus de vérification d'identité pour une entrée en relation à distance.

**La rapidité :** la reconnaissance faciale permet une vérification d'identité à distance rapide et fluide.

**L'expérience utilisateur :** les systèmes de reconnaissance faciale offrent une expérience utilisateur unique, fluide et rapide, évitant les déplacements dans les bureaux ou les visioconférences qui peuvent être parfois longues ainsi qu'avec des attentes.

**La sécurité :** comme les empreintes, la voix, chaque visage est unique et présente des caractéristiques inimitables. Les systèmes, programmes ou logiciels de reconnaissance faciale comparent à travers la biométrie faciale et les algorithmes de reconnaissance faciale.

**La conformité :** la reconnaissance faciale à travers une vidéo identification et la seule méthode reconnue comme standard pour la vérification d'identité à distance pour les opérations à hauts risques.

#### **4.3.2. Les inconvénients**

La reconnaissance faciale présente des inconvénients qui influent sur la qualité de la reconnaissance. On peut citer les aspects suivant (6).

- Changement d'illumination.
- Expressions faciales.
- Présence ou absence des composantes structurales, telles que (la barbe, la moustache et les lunettes).

#### **4.4. Etats de l'art des techniques de reconnaissance faciale**

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

Plusieurs méthodes de reconnaissance de visages ont été proposées durant les vingt dernières années. La reconnaissance de visage est un axe de recherche ouvert attirant des chercheurs venants de disciplines différentes : psychologie, reconnaissance de formes, réseaux neuraux, vision artificielle et infographie. Dans ce paragraphe, nous présenterons les approches de la reconnaissance faciale les plus connues. Ces dernières peuvent être subdivisées en trois catégories : les approches globales, les approches locales et les approches hybrides.

##### **4.4.1. Approches globales**

Ces approches sont également appelées méthodes basées sur l'apparence. Ces méthodes identifient un visage en utilisant l'image entière de ce dernier comme entrée du système de reconnaissance. Le principe est comme suit : chaque image de visage de dimension  $(n,m)$  est



représentée par un vecteur simple de dimension  $n,m$ , en concaténant les valeurs du niveau de gris de tous les pixels de l'image du visage.

#### **4.4.2. Approches locales**

Les approches locales de la reconnaissance faciale sont basées sur des modèles et reposent sur un traitement séparé appliqué sur les différentes régions de l'image contenant un visage. Ce processus conduit à un vecteur caractéristique pour chaque région du visage.

#### **4.4.3. Approche hybride**

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

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

### **4.5. Mesure de performance d'un système de reconnaissance faciale**

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

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

$$TRA = \text{nombre de personnes correctement reconnue} / T$$

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

$$FAR = \text{nombre de personnes faussement acceptées} / T$$

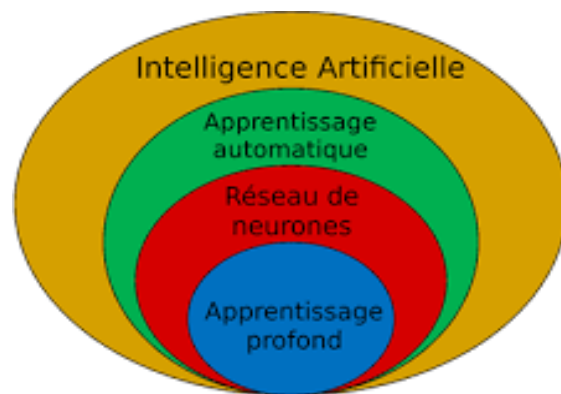
- **FRR (False Reject Rate):** Il représente le taux de faux rejet noté par le pourcentage de personnes censées être reconnues mais qui sont rejetées par le système. Il est calculé selon l'équation suivante :

$$FRR = \text{nombre de personnes faussement rejetées} / T$$

## 5. L'intelligence artificielle

L'intelligence artificielle (IA) est une branche de l'informatique qui vise à créer des systèmes et des machines capables de réaliser des tâches normalement associées à l'intelligence humaine, telles que la reconnaissance faciale, l'apprentissage vocal, la vision par ordinateur, la planification, le raisonnement et l'apprentissage automatique. Les technologies de l'IA incluent des algorithmes d'apprentissage automatique, de réseaux de neurones artificiels et l'apprentissage profond (figure 3).

L'IA est utilisée dans de nombreux domaines, notamment la médecine, la finance, les transports, l'industrie, la sécurité...



**Figure 3** L'intelligence artificielle

### 5.1. L'apprentissage automatique

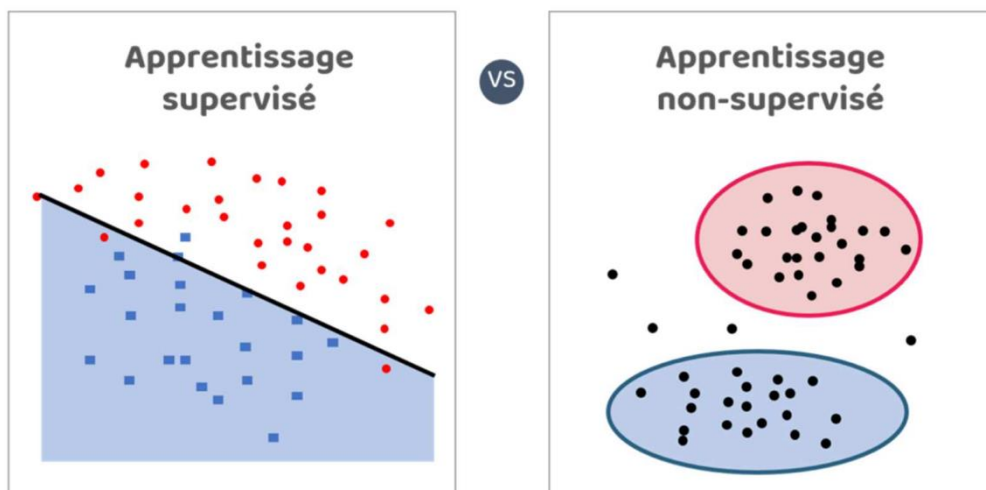
L'apprentissage automatique (Machine Learning en anglais) est un champ d'étude de l'intelligence artificielle qui vise à donner aux machines la capacité d'apprendre à partir de données, des modèles mathématiques. Plus précisément, il s'agit de procédé par lequel les informations pertinentes sont tirées d'un ensemble de données d'entraînement.

Le but de cette phase est l'obtention des paramètres d'un modèle qui atteindront les meilleures performances, notamment de la réalisation de la tâche attribuée au modèle. Une fois l'apprentissage réalisé, le modèle pourra ensuite être déployé en production.

### 5.1.1. Les différents types de l'apprentissage automatique

On distingue trois types de l'apprentissage automatique :

- **L'apprentissage supervisé** : un apprentissage est dit supervisé si les classes sont prédéterminées et les exemples connus dont le système apprend à classer selon un modèle de classification ou de classement. Le processus passe en deux phases, la première phase est hors ligne c.-à-d. les données sont statiques pendant l'apprentissage. Il s'agit de déterminer un modèle à partir des données étiquetées. La deuxième phase qui en ligne c.-à-d. les données sont présentés les un après les autres au fur et à mesure de leur disponibilité, cette phase consiste à prédire une étiquette d'une nouvelle donnée, cette phase est généralement connue sous le nom 'test'.
- **L'apprentissage non supervisé** : un apprentissage est dit non supervisé si le système dispose d'aucun étiquetage préalable des données si le système ne dispose aucun étiquetage préalable des données, et que le nombre de classes et leur nature n'ont pas été prédéterminés. Aucun expert n'est requis. L'algorithme doit découvrir par lui-même la structure plus ou moins cachée des données. Le partitionnement des données (en anglais : data clustering) est un exemple d'algorithme de l'apprentissage non supervisé. Il y a une différence entre l'apprentissage supervisé et non supervisé (figure 4) (7).
- **L'apprentissage par renforcement** : c'est une méthode d'apprentissage où la machine se comporte comme un agent qui apprend de son environnement d'une manière interactive jusqu'à ce qu'elle découvre les comportements qui produisent des récompenses.



**Figure 4** L'apprentissage supervisé et l'apprentissage non supervisé

### **5.1.2. Le fonctionnement de l'apprentissage automatique**

Le développement d'un modèle de l'apprentissage automatique repose sur quatre étapes principales. En règle générale, c'est un Data Scientist qui gère et supervise ce procédé.

La première étape consiste à sélectionner et à préparer un ensemble de données d'entraînement. Ces données seront utilisées pour nourrir le modèle de Machine Learning pour apprendre à résoudre le problème pour lequel il est conçu.

Les données peuvent être étiquetées, afin d'indiquer au modèle les caractéristiques qu'il devra identifier. Elles peuvent aussi être non étiquetées, et le modèle devra réparer et extraire les caractéristiques récurrentes de lui-même.

Dans les deux cas, les données doivent être soigneusement préparées, organisées et nettoyées. Dans le cas contraire, l'entraînement du modèle de Machine Learning risque d'être biaisé. Les résultats de ses futures prédictions seront directement impactés.

La deuxième étape consiste à sélectionner un algorithme à exécuter sur l'ensemble de données d'entraînement. Le type d'algorithme à utiliser dépend du type et du volume de données d'entraînement et du type de problème à résoudre.

La troisième étape est l'entraînement de l'algorithme. Il s'agit d'un processus itératif. Des variables sont exécutées à travers l'algorithme, et les résultats sont comparés avec ceux qu'il aurait dû produire.

On exécute ensuite de nouveau les variables jusqu'à ce que l'algorithme produise le résultat correct la plupart du temps. L'algorithme, ainsi entraîné, est le modèle de Machine Learning.

La quatrième et dernière étape est l'utilisation et l'amélioration de modèle. On utilise le modèle sur de nouvelles données, dont la provenance dépend du problème à résoudre par exemple, un modèle d'apprentissage automatique conçu pour détecter les spams sera utilisé sur des emails.

De son côté, le modèle de Machine Learning d'un aspirateur robot ingère des données résultant de l'interaction avec le monde réel comme le déplacement de meubles ou l'ajout de nouveaux objets dans la pièce. L'efficacité et la précision peuvent également s'accroître au fil du temps.

## **5.2. L'apprentissage profond**

L'apprentissage profond ou le Deep Learning est un type d'intelligence artificielle dérivé du Machine Learning (apprentissage automatique) où la machine est capable d'apprendre par elle-même, contrairement à la programmation où elle se contente d'exécuter à la lettre des règles prédéterminées.

Le Deep Learning est d'une grande utilité dans l'univers des technologies de l'information et de la communication.

Il est employé dans les systèmes de reconnaissance faciale et vocale qu'embarquent certains Smartphones, et en robotique pour que les équipements intelligents puissent avoir la réaction attendue dans une situation donnée.

### **5.2.1. Le fonctionnement de Deep Learning**

Le Deep Learning s'appuie sur un réseau de neurones artificiels s'inspirant de cerveau humain. Ce réseau est composé de dizaines voire de centaines de « couches » de neurones, chacune recevant et interprétant les informations de la couche précédente.

Comme à l'intérieur du cerveau humain, les signaux voyagent entre les neurones du cerveau artificiel. Le secret de cette prouesse repose en grande partie sur les algorithmes. Dans le cas de la reconnaissance visuelle, pour être performant, l'algorithme du Deep Learning doit être capable d'identifier toutes les formes existantes et dans tous les angles.

Il y a plusieurs différences entre Machine Learning et Deep Learning (Tableau 1) (8)

<b>Paramètre</b>	<b>Machine Learning (ML)</b>	<b>Deep Learning (DL)</b>
<b>Dépendance vis-à-vis des données</b>	Bien que les deux apprentissages dépendent d'une grande quantité de données, l'apprentissage automatique peut fonctionner avec une plus petite quantité de	Les algorithmes d'apprentissage profond reposent en grande partie sur une vaste quantité de données, les moteurs doivent donc être alimentés avec d'énormes quantités de

	données.	données et éventuellement être réentraînées pour obtenir de bonnes performances.
<b>Temps d'exécution</b>	L'algorithme de ML prend moins de temps pour entraîner le modèle que celui de l'apprentissage profond, mais tester le modèle peut prendre beaucoup de temps.	Le DL nécessite un certain temps d'exécution pour entraîner le modèle. Tester et obtenir les résultats est généralement plus rapide.
<b>Dépendance vis-à-vis du matériel</b>	Comme les modèles d'apprentissage automatique n'ont pas besoin de beaucoup de données, ils peuvent fonctionner sur des machines bas de gamme.	Les modèles d'apprentissage profond ont besoin d'une énorme quantité de données pour fonctionner efficacement et donc de machine haut de gamme
<b>Feature engineering</b>	les modèles d'apprentissage automatique ont besoin d'une étape d'extraction de caractéristique, qui doit être réalisée par un expert, avant de pouvoir poursuivre leur apprentissage.	Le DL est une version améliorée du ML. Il n'est donc pas nécessaire de développer un extracteur de caractéristiques pour chaque problème. Au lieu de cela, il essaie d'apprendre par lui-même des caractéristiques de haut niveau à partir des données.
<b>Approche par résolution de</b>	Pour résoudre un problème donné, le	L'approche de résolution de problèmes d'un modèle DL

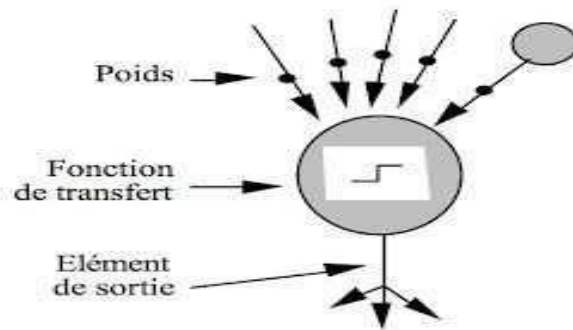
<b>problèmes</b>	modèle ML traditionnel divise le problème en sous-parties, et après avoir résolu chaque partie, il produit le résultat final.	est différente de celle du modèle ML traditionnel, car il prend part à un problème donné et produit le résultat final. Il suit donc une approche de bout en bout.
<b>Interprétation du résultat</b>	Le résultat pour un problème donné peut être facilement interprété. Comme c'est souvent le cas lorsque nous travaillons avec l'apprentissage automatique, nous pouvons facilement interpréter le résultat.	L'interprétation du résultat pour un problème donné peut être complexe. Nous pouvons obtenir un meilleur résultat pour un problème donné qu'avec le modèle ML, mais le réseau agit comme une « boîte noire » ; nous ne pouvons pas savoir pourquoi il a produit un résultat particulier ni comprendre la logique sous-jacente.
<b>Type de données</b>	Les modèles ML nécessitent principalement des données sous une forme structurée.	Les modèles DL peuvent fonctionner avec des données structurées et non structurées, car ils dépendent des couches du réseau neuronal artificiel.
<b>A quoi ces modèles peuvent-ils servir ?</b>	Les modèles ML sont utilisés pour résoudre des problèmes aussi bien simples que complexes.	Les modèles DL sont principalement utilisés pour résoudre des problèmes complexes.

**Tableau 1** Les différences entre ML et DL

### 5.3. Les réseaux de neurones

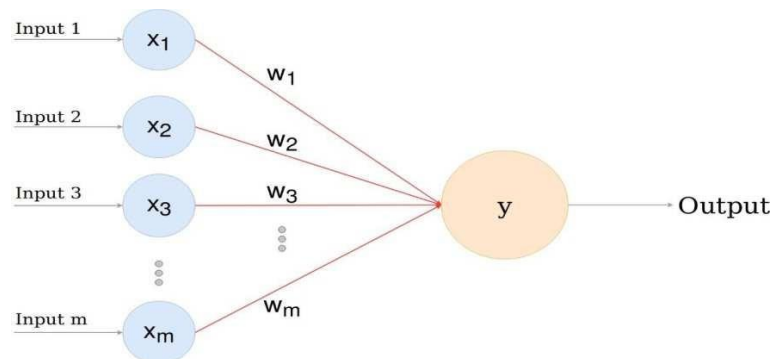
Un réseau de neurones artificiels (figure 5) est un système dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologique, et qui par la suite s'est rapproché des méthodes statistique.

Les réseaux neurones sont généralement optimisés par des méthodes d'apprentissage de type probabiliste.



**Figure 5** Neurone artificiels

L'un des neurones artificiels simplifié c'est le perceptron (figure 6) qui a été introduit par Frank Rosenblatt en 1957 (9). Il contient plusieurs entrées ou chacune possède un poids. Les entrées sont toutes connectées à une seule sortie en passant par une fonction de transfert qui est appelée aussi fonction d'activation.



**Figure 6** Représentation de perceptron



Où :

$X_i$  : représente les entrées de perceptron.

$W_i$  : représente les poids associés à ces entrées.

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

## **6. Conclusion**

Ce chapitre est composé de trois sections. Dans la première, nous avons abordé les concepts de base de la biométrie en évoquant les différentes modalités ainsi que les applications des systèmes biométriques. Dans la deuxième section, nous avons introduit le fonctionnement d'un système de reconnaissance faciale, les différentes techniques répandues dans la littérature pour la reconnaissance visage. Une synthèse des avantages et inconvénients des différentes catégories de techniques de reconnaissance de visage a été effectuée, et la troisième section, nous avons présenté en détails le principe de l'apprentissage automatique. La conception de système de reconnaissance faciale sera à l'aide de Raspberry Pi. Dans le chapitre 2, nous allons définir les grandes lignes de notre travail telles que les techniques utilisées pour la détection de visage ainsi que les différents modules nécessaires pour la réalisation de notre système de reconnaissance faciale.

# Chapitre 2 : Etude des systèmes de reconnaissance faciale

## 1. Introduction

Avec la croissance continue de la société dans tous les domaines, la sécurité personnelle est devenue une préoccupation majeure, entraînant une augmentation de la demande pour des moyens de protection. L'identification des individus par leurs caractéristiques physiques et la reconnaissance faciale sont considérées comme d'excellents moyens de sécurité.

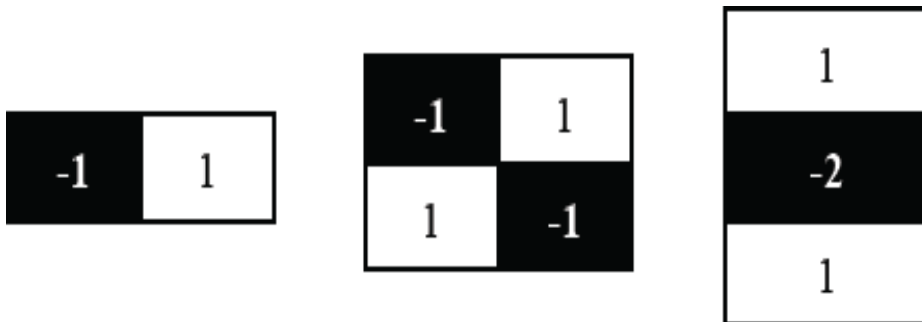
Dans ce chapitre, nous avons examiné les principales composantes. Nous avons choisi le programmer notre projet en utilisant le langage Python et sa bibliothèque principales, OpenCV, qui facilite grandement les procédures de détection.

## 2. Procédé de détection de Haar-Cascade

La méthode de Haar-Cascade, également connue sous le nom de méthode Viola-Jones, est la méthode la plus largement utilisée pour détecter les objets. Les processus de détection d'objets par cette méthode comprennent quatre parties principales, à savoir des caractéristiques Haar, l'image intégrale, le boost adaptatif ou l'AdaBoost et la combinaison de classificateur cascade.

### 2.1. Haar comme caractéristique

Haar comme caractéristique est une fonction haar bidimensionnelle utilisée pour coder l'apparence de l'objet. Une caractéristique Haar se compose de deux rectangles ou plus côte à côte pour former un modèle, (figure 7).



**Figure 7** Exemple de fonctionnement similaire de Haar-Cascade

## 2.2. Image intégrale

Une image intégrale est une technique de calcul rapide de la valeur de la fonction haar en changeant la valeur de chaque pixel en une nouvelle représentation d'image appelée image intégrale.

Chaque pixel de l'image intégrale à une valeur qui est accumulée à partir du haut et à la gauche de la position du pixel de l'image d'entrée, (figure 8)

1	2	3
4	5	6
7	8	9

image originale

1	3	6
5	12	21
12	27	45

image intégrale

**Figure 8** Exemple de calcul d'image intégrale

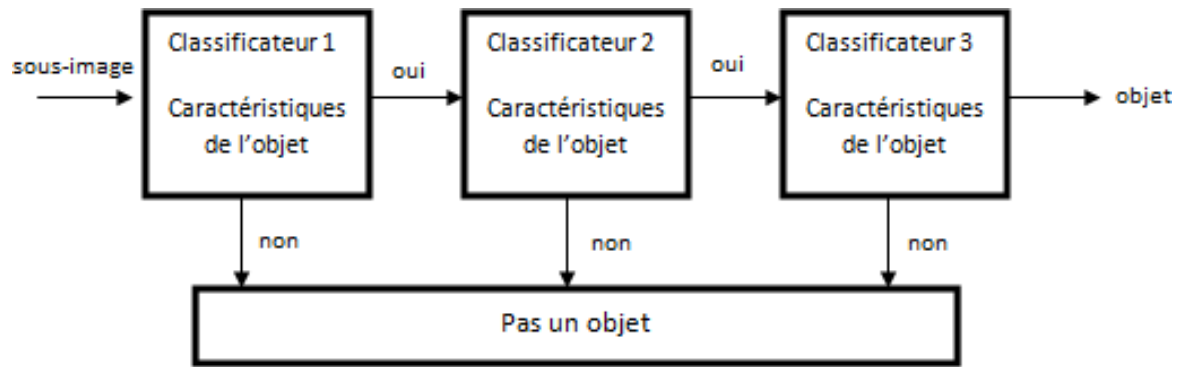
## 2.3. AdaBoost

Le processus de détection n'est pas fait avec un seul type de fonctionnalités. Au lieu de cela, différents types de fonctionnalisés avec différentes échelles et positions sont utilisés. Par conséquent, les calculs vont ralentir le processus de détection. AdaBoost est un algorithme exécuté par Viola-Jones pour réduire le nombre de fonctionnalités non pertinentes dans une détection. AdaBoost trouve des fonctionnalités qui ont un degré élevé de différenciation en évaluant chaque fonctionnalité par rapport aux données de fonction en utilisant la valeur de fonctionnalité. Les entités qui ont la limite la plus élevée entre l'objet et le non-objet sont considérées comme les meilleures fonctionnalités. AdaBoost est utilisé pour améliorer la performance de classification avec une formation simple pour combiner beaucoup de classificateurs faibles dans un classificateur fort. Un classificateur faible est une bonne réponse avec un degré de vérité qui est moins précis, tandis qu'un classificateur fort est une réponse correcte avec un degré précis de vérité.

## 2.4. Classificateur en Cascade

Le classificateur cascade est une méthode permettant de combiner un classificateur complexe dans une structure à plusieurs niveaux qui peut augmenter la vitesse du processus de détection. La sélection de l'existence d'objets se fait en évaluant les sous-images en classificateur. Si une sous-image a réussi à travers l'ensemble des classificateurs, il peut donc

être conclu qu'il y a un objet dans la sous-image. Si non, le processus d'évaluation ne passera pas au prochain classificateur et a conclu qu'il n'y a pas d'objet détecté dans la sous- image. Cette combinaison de classificateurs est regroupée en plusieurs classificateurs en cascade, (figure 9).



**Figure 9** Cascades classifiées

### 3. Environnement de développement :

L'environnement de développement se divise par deux parties Hardware et Software, ils sont essentiels pour concevoir et optimiser un système de reconnaissance faciale performante et fiable.

#### 3.1. Environnement Hardware :

La Raspberry Pi est une nano-ordinateur mono carte à processeur ARM conçu par des professeurs du département informatique de l'université de Cambridge dans le cadre de la fonction Raspberry Pi (figure 10).

Cet ordinateur, de la taille d'une carte de crédit, est destiné à encourager l'apprentissage de la programmation informatique ; il permet l'exécution de plusieurs variantes du système d'exploitation libre GNU/Linux, notamment Debian et des logiciels compatibles. Mais il fonction également avec le système d'exploitation Microsoft Windows.

Il est fourni nu, c.-à-d. la carte mère seule, sans boîtier, alimentation, clavier, souris ni écran, dans l'objectif de diminuer les coûts et de permettre l'utilisation de matériel de récupération. Néanmoins des « kits » regroupant le « tout en un » sont disponibles sur le web (10).



Figure 10 Raspberry Pi

### 3.1.1. Les modèles de Raspberry Pi :

La carte Raspberry Pi est très polyvalentes et peuvent être utilisées pour une grande variété de projets pour cela elle existe en plusieurs modèles tels que le modèle A (tableau2), le Modèle B (tableau3) et le modèle Zéro (tableau4).

Modèle A	Modèle A+
Le Raspberry Pi A est lancé en février 2013, il est équipé de 256 RAM et ne dispose pas de port Ethernet, 2 sortie vidéo : composite et HDMI1 sortie audio stéréo Jack 3.5 mm et 1 port USB 2.0.	Le Raspberry Pi A+est lancé en novembre 2014, plus petit que le Raspberry Pi A, lecteur de carte micro SD au lieu et place du lecteur SD et consommation électronique moindre, GPIO 40 broches et nouveau chipset audio.

Tableau 2 Le modèle A

<b>Le modèle 2 B</b>	Le 2 février 2015, la fondation Raspberry Pi annonce la sortie du Raspberry Pi2, plus puissant, il est équipé d'un processeur BroadcomBCM2836, quatre cœurs ARMv7 à 900 MHz, accompagné de 1 GO de RAM.
<b>Le modèle 3B</b>	Le 29 février 2016, pour le quatrième anniversaire de la commercialisation du premier modèle, la fondation Raspberry Pi annonce la sortie du Raspberry Pi 3. Il est à présent équipé nativement du Wifi b/g/n et du Bluetooth 4.1. processeur Broadcom BCM2837 64 bit Quatre cœur ARM Cortex-A53 à 1,2GHz, Puce Wifi 802.11n et Bluetooth 4.1 intégrée
<b>Le modèle 3 B+</b>	Le 14 mars 2018, la fondation Raspberry Pi annonce la mise à jour du Raspberry Pi 3 vers le modèle B on y trouve une mise à jour du processeur Broadcom BCM2837B0 64bit à quatre cœurs ARM Cortex-A53 cadencé à 1,4GHz au lieu du 1,2 GHz. La puce Cypress CYW43438 est remplacée par une nouvelle puce CYW43455 supportant le Wifi Dual-band 802.11ac et la version 4.2 du Bluetooth et d'une prise en charge du Power over Ethernet grâce à un élément supplémentaire.
<b>Le modèle 4 B</b>	Le 24 juin 2019, la fondation Raspberry Pi annonce la sortie du Raspberry Pi 4 le 10 juillet 2019, la fondation annonce des problèmes de conception, notamment l'absence d'une résistance, qui ne permet pas à certains chargeurs USB-C d'alimenter le Raspberry Pi. La fondation annonce que le problème sera corrigé dans une révision future du produit. Dans les faits, seuls les câbles « e-marked » posent problème parce qu'ils détectent le Raspberry Pi comme étant un équipement audio.

<b>Le modèle 4B 8GB</b>	<p>Le 28 mai 2020 est annoncée une version à 8Go de RAM Bien que Raspbian ne soit que 32 bits, les 8 Go peuvent être exploités via des processus indépendants lancés sous ce système. Une version 64 bits est annoncée, elle aussi branche de Debian, qui ne se nommera plus Raspbian, mais Raspberry Pi OS pour éviter les confusions, et identique en interface.</p>
-------------------------	--

**Tableau 3** Le modèle B

<b>Le modèle Zéro</b>	<p>Le 26 novembre 2015, la fondation Raspberry Pi annonce la sortie du Raspberry Pi annonce la sortie du Raspberry Pi zéro. Il reprend les spécifications du modèle A/B avec un processeur cadencé à 1GHz au lieu de 700 MHz, il est par contre plus petit, disposant d'une connectique minimale. Son prix de 5 \$ US est largement revu à la baisse par rapport aux autres Raspberry Pi</p>
<b>Modèle Zéro W</b>	<p>Lancé le 28 février 2017 le Raspberry Pi zéro W étend la famille Pi zéro avec une connectivité Bluetooth et LAN sans fil supplémentaire.</p> <p>CPU : 1GHz</p> <p>Mémoire vive : 512 Mb</p>
<b>Modèle Zéro WH</b>	<p>Lancé le 15 janvier 2018, même carte que Raspberry Pi zéro W mais avec un connecteur 40 pins GPIO déjà soudé SPI + I2C Camera+MicroSD</p>

**Tableau 4** Le modèle Zéro





**Une carte MicroSD :** Le Raspberry Pi est livré sans disque dur. La carte MicroSD fait office de stockage permanent du Raspberry Pi ; tous les fichiers créés et les logiciels installés, ainsi que le système d'exploitation lui-même, sont stockés sur la carte MicroSD (figure 13).

Il est donc obligatoire d'en avoir une carte de bonne qualité et très performante pour ne pas ralentir le Pi, l'idéal est d'avoir une carte au standard SDHC pouvant atteindre au maximum 8Go pour permettre le démarrage, bien qu'une carte de 16Go ou bien 32Go offre d'espace pour ce développer.



**Figure 13** MicroSD

**Camera :** Une webcam, parfois cybercaméra ou webcaméra, est une caméra conçue pour être utilisée comme un périphérique d'ordinateur, et qui produit une vidéo dont la finalité n'est pas d'atteindre une haute qualité, mais de pouvoir être transmise en direct au travers d'un réseau, typiquement Internet. Cela dit, la norme HD (haute définition) tend à se généraliser avec l'Internet haut débit.

## **4. Environnement software :**

Dans l'environnement software on a les logiciels et les bibliothèques de la carte Raspberry Pi et aussi les logiciels de l'application mobile.

### **4.1. Raspbian**

Après savoir quel Raspberry Pi choisir les différents accessoires importants pour son fonctionnement et son utilisation, la chose de plus concert est le choix du système d'exploitation. Les distributions les plus utiles et les plus populaires sont présentées comme suite :

- Les distributions sur Linux
- Windows 10
- Les distributions pour Media Center
- Les distributions pour les jeux vidéo

Le système d'exploitation de référence pour Raspberry Pi et qui est basé sur Linux est le Rasbian qui présente une distribution polyvalente optimisée spécialement pour Raspberry Pi, et qui permettra la familiarisation très facilement avec le matériel.

## **4.2. Langage Python**

Python est un langage orienté objet qui à été développé à la fin des années 80 (le nom est dérivé de la série télévisée britannique, Monty python's Flying Circus). Cependant, python n'est pas aussi bien connu dans les cercles d'ingénierie que certains autres langages suivant une considération dans la communauté de programmation (12).

Python peut être considéré comme un langage émergent car il est encore au cours de développement et de recherche. C'est un excellent langage pour développer une application d'ingénierie.

Python présente de nombreux avantages par rapport au langage traditionnel, ce qui est important dans un environnement d'apprentissage :

Les programmes python ne sont compilés en code machine, mais sont exécutés par un interpréteur, ce qui signifie que les programmes peuvent être testés et débogués rapidement, permettant à l'utilisateur de se concentrer d'avantage sur les principes du programme et moins sur la programmation elle-même car il n'est pas nécessaire de compiler, de lier et d'exécuter après chaque correction (13).

Python peut être développé en un temps beaucoup plus court que d'autres programmes tels que c++. Python est une source de style, ce qui signifie que sans identifiant il est inclus dans la plupart des distributions LINUX.

## **4.3. Bibliothèques utilisées**

### **4.3.1. Numpy**

Numpy est une bibliothèque Python populaire utilisée pour effectuer des calculs numériques basés sur des tableaux. L'implémentation canonique de numpy utilisée par la plupart des programmeurs s'exécute sur un seul cœur de processeur et est parallélisée pour utiliser plusieurs cœurs pour certaines opérations.

Cette restriction à une exécution à processeur unique à nœud unique limite à la fois, la taille des données pouvant être traitées et la vitesse potentielle du code numpy.

### **4.3.2. OpenCV**

OpenCV (Open Source Computer Vision Library) cette bibliothèque est utilisée pour développer des applications de traitement d'images et de vidéos en temps réel.

Elle fournit des fonctions avancées pour la vision par ordinateur, telles que la reconnaissance faciale, la détection d'objets, le suivi de mouvement, la segmentation d'image, ect.

#### **Pourquoi utiliser OpenCV ?**

OpenCV est très utilisée dans de nombreux domaines, tels que l'automobile, la robotique, la surveillance, la sécurité, ect. Cette bibliothèque est disponible sous une licence libre et gratuite, ce qui la rend accessible à tous les développeurs. Elle est compatible avec de nombreux langage de programmation, tels que C++, Python, Java, ect.

#### **Les fonctionnalités d'OpenCV**

OpenCV propose une vaste gamme de fonctionnalités pour le traitement d'images et de vidéos. Ces fonctionnalités comprennent :

- La lecture et l'écriture de fichiers d'images et des vidéos.
- La manipulation d'images, telles que la redimensionne, la rotation, la translation, la modification de la couleur...
- La détection de la reconnaissance d'objets, tels que les visages, les yeux, les souries, les véhicules...
- La segmentation d'image, qui consiste à diviser une image en plusieurs parties.
- La détection de contours, qui consiste à détecter les bords d'un objet dans une image.
- La stéréovision, qui permet de reconstruire la géométrie 3D à partir de deux images.

### **4.4. Firebase**

Firebase est une plateforme de développement d'application mobiles et web développée par Google. Elle offre une multitude de services qui facilitent le développement, la gestion et le déploiement d'application, en fournissant une infrastructure Cloud robuste et évolutive.

Dans notre cas on utilise le Firebase Cloud Messaging car il s'agit d'un service de messagerie push multiplateforme qui permet d'envoyer des notifications à vos utilisateurs. Vous pouvez envoyer des notifications ciblées, programmées ou à diffusion générale pour engager vos utilisateurs et les informer.

## **5. Conclusion**

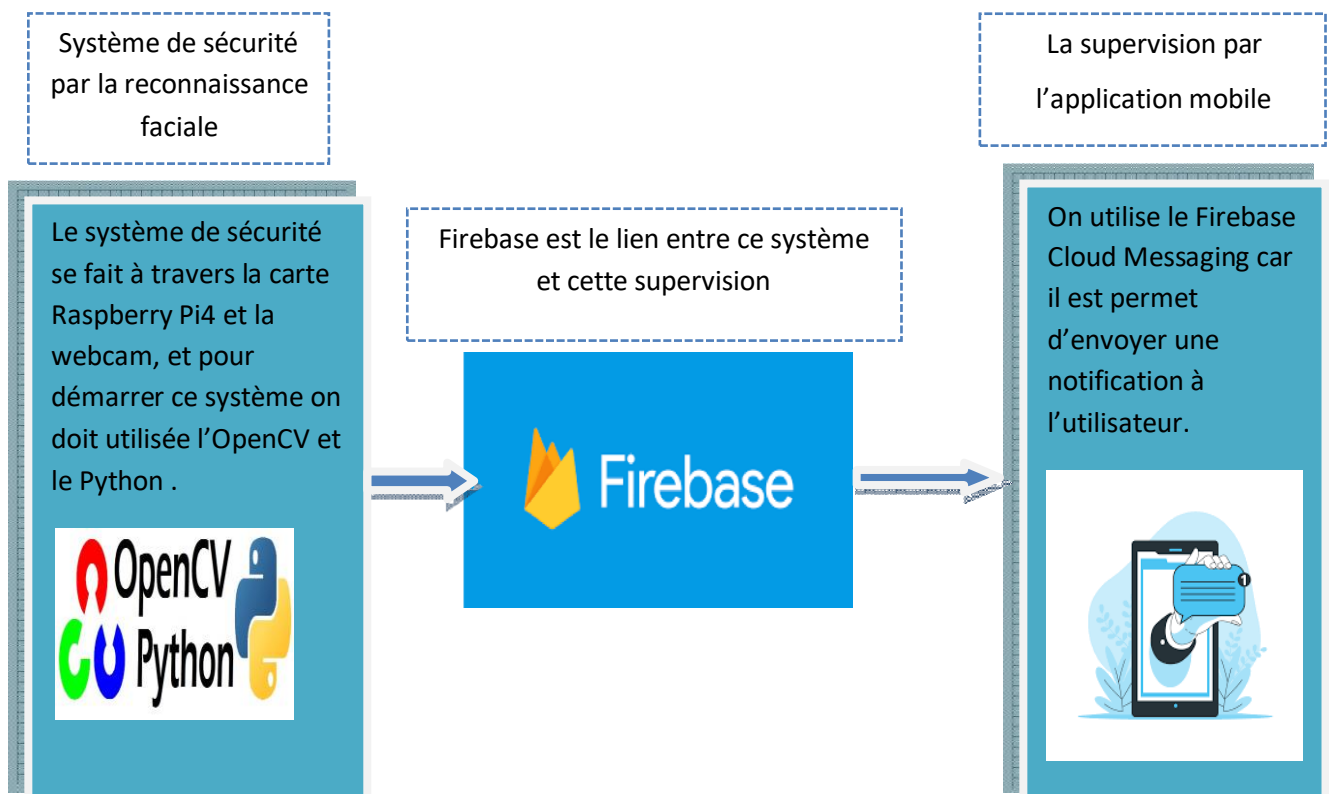
Dans ce chapitre nous avons vu le principe du procédé de détection HAAR-CASCADE, nous avons présenté la bibliothèque OpenCV qui réduire le temps de programmation de la détection des visages ainsi les différentes composants utiliser dans ce projet de conception.

Dans le dernier chapitre nous allons présenter la préparation de système d'exploitation, l'installation du Rasbian, citer les procédures de réalisation de notre système de la reconnaissance faciale.

# Chapitre 3 : Conception et Réalisation

## 1. Introduction

Ce dernier chapitre est consacré à la conception de l'application qui permettra d'identifier des personnes par reconnaissance faciale. Le programme de reconnaissance faciale sera codé en python. Il utilisera les bibliothèques OpenCV pour pouvoir prendre des photos à l'aide d'une webcam. Son rôle est d'identifier, parmi un ensemble de visage connus et stockés dans une base de données. Il devra également indiquer à quel point il s'en rapproche, afin de pouvoir valider ou infirmer l'identification. Il utilisera la technique dite cascade de Haar, l'une des première et des lus connues des méthodes de détection d'objet et de visage, elle est 15 fois plus rapide que les autres méthodes de détection de visage. Elle affichera dans une fenêtre le visage reconnu ainsi que diverses informations comme le nom de la personne et le score de ressemblance, ce dernier est comparé à un seuil déjà fixé pour vérifier l'existence de la personne recherchée dans la base de données. Plusieurs étapes sont nécessaires, l'étape d'extraction des caractéristiques est la plus importante car les performances du système en dépendent. On peut divisée se chapitre en deux parties la partie de sécurité de système de reconnaissance faciale et la partie de l'application mobile :



## 2. Le système de sécurité par la reconnaissance faciale

### Matériels de conception :

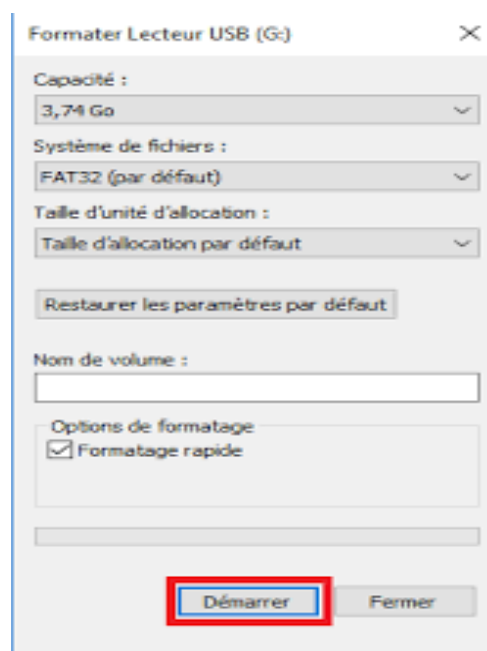
- ✓ Raspberry pi 4 modèle B
- ✓ Une alimentation
- ✓ Une carte micro SD « 16 GO »
- ✓ Un clavier et une souris
- ✓ Un câble HDMI
- ✓ Un adaptateur micro SD
- ✓ Webcaméra
- ✓ Un écran

### 2.1. Préparation du système d'exploitation sur Micro SD

#### 2.1.1. Formater la carte SD

La carte SD est ce qui fera office de mémoire morte sur le Raspberry Pi, tout comme un disque dur d'ordinateur. C'est donc là-dessus que sera stocké le système d'exploitation, et à priori, nos documents, photos, musique, vidéos...

Même si la carte SD est neuve, le mieux est de la formater avant de copier les fichiers d'installation. Nous avons utilisé le logiciel de formatage SD Card Formatter disponible pour Windows.



**Figure 14** Formatage de la carte MicroSD

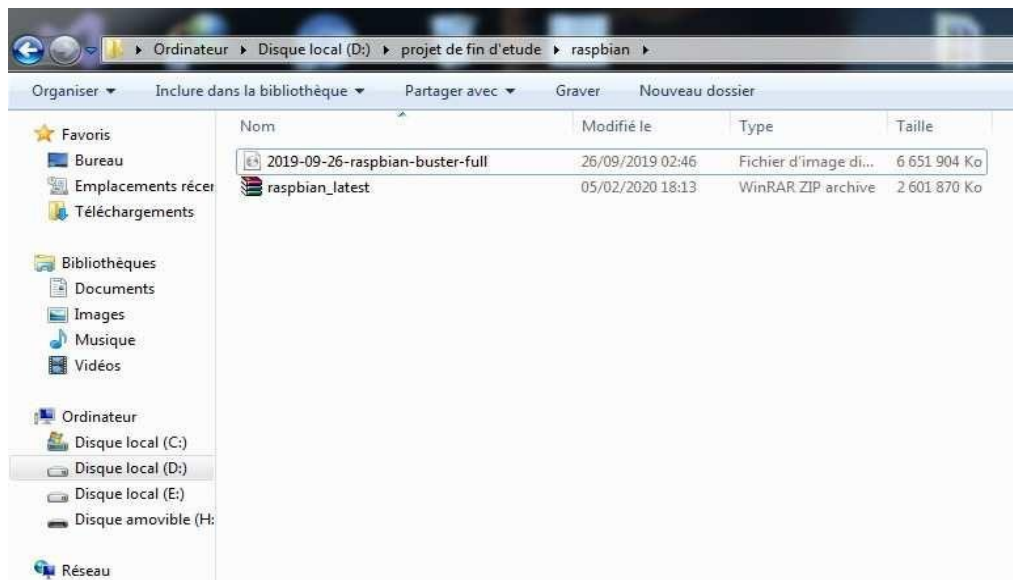
### 2.1.2. Installer Raspbian sur la carte SD

Nous allons donc commencer par télécharger le système d'exploitation, qui est en fait une distribution de Linux qui s'appelle Raspbian. Cette distribution est une version modifiée (adaptée pour le Raspberry Pi) de Debian, qui est utilisée sur la grande majorité des serveurs de sites Internet.

Pour obtenir le système d'exploitation Raspbian Buster il faut accéder à cette page du site officiel du Raspberry Pi :

➤ <https://www.raspberrypi.org/downloads/raspbian/>  
Et télécharger Raspberry Pi OS  
(32bits) with desktop

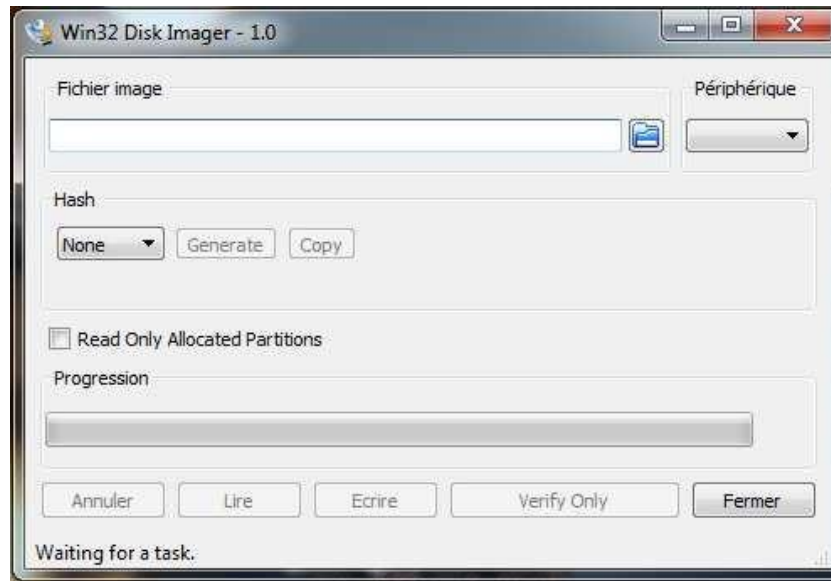
Une fois téléchargé, il faut décompresser le fichier .Zip pour obtenir un fichier .img



**Figure 15** Décompression du fichier Zip

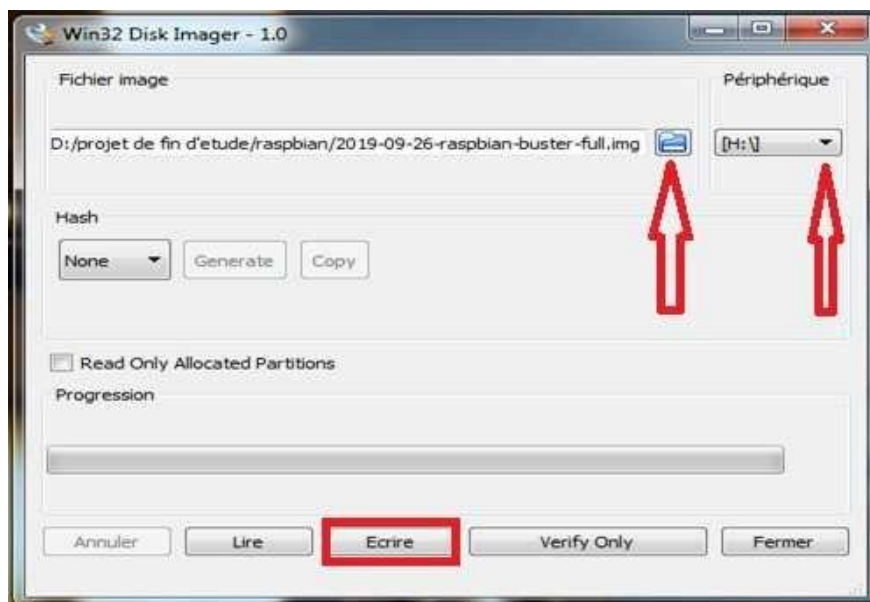
Nous avons besoin maintenant d'installer le logiciel spécifique Win32DiskImager, qui va nous permettre correctement Raspbian sur la carte SD, lien direct pour le téléchargement :

➤ <https://sourceforge.net/projects/win32diskimager/>



**Figure 16** Logiciel Win32 DiskImage

Pour copier l'image sur la carte SD, c'est très simple : nous avons sélectionné l'image de Raspbian, et choisi la lettre de notre lecteur de carte SD, puis cliqué sur écrire. Une fois l'écriture terminée, la fenêtre affiche « terminer » sous la barre de progression, la carte Raspbian est prête, il ne nous reste plus qu'à l'insérer dans notre Raspberry Pi et à brancher cette dernière.

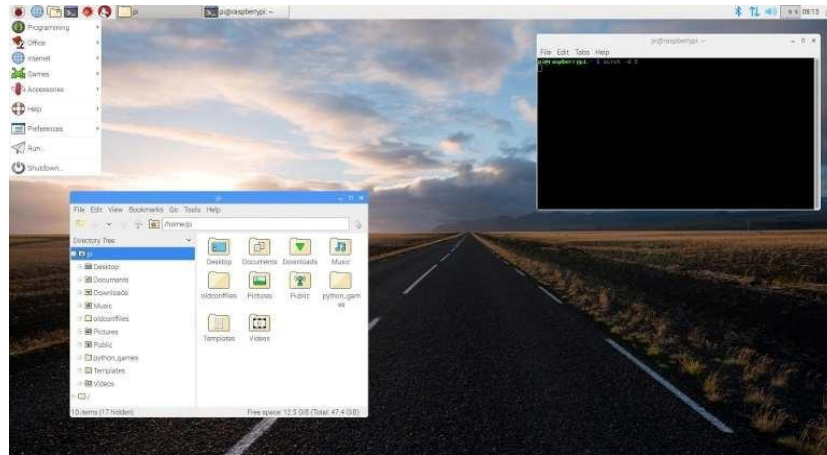


**Figure 17** Copie le fichier Raspbian sur la carte SD

## 2.2. Premier démarrage du système

Après le branchement de tout l'équipement, notre Raspberry Pi4 est allumé



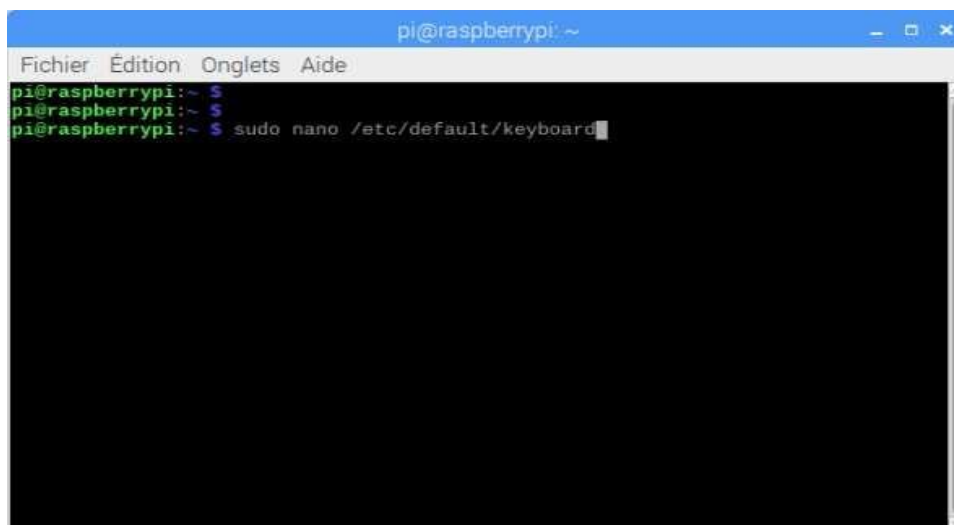


**Figure 18** bureau de Raspberry Pi

une fois que notre Raspberry est allumé et a fini de démarrer, il faut faire quelques réglages et configuration avant le commencer à l'utiliser.

### 2.2.1. Le clavier en français

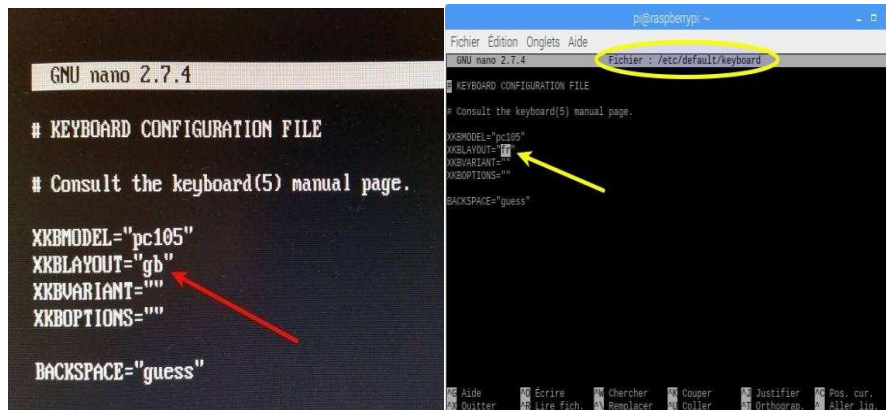
Nous avons ouvrir le terminal et écrire la commande `sudo nano /etc/default/keyboard` et valider par Entrée :



**Figure 19** Le terminal

Le fichier s'ouvre alors Keyboard dans l'éditeur nano directement dans le terminal, puis déplacer le curseur avec les flèches du clavier dans cette fenêtre pour aller modifier **gb** et remplacer par **fr**, on doit lire maintenant `XKBLAYOUT=fr` (figure 20).

Enregistre la modification avec la combinaison de touche **Ctrl+O** au clavier et valide par la touche Entrée. Quitter l'éditeur nano avec la combinaison de touches **Ctrl+X**



**Figure 20** La transformation de Keyboard de gb vers fr

Dernière étape: nous faisons redémarrer le Raspberry Pi pour prendre compte les modifications effectuées avec la commande **sudo reboot**.

### 2.2.2. Configuration de la localisation

Nous avons accéder au menu des applications → **préférence** → **configuration du Raspberry Pi** (figure21).



**Figure 21** Menu des applications

Dans la fenêtre qui s'ouvre, nous avons accéder au l'onglet « Localisation » → cliquer sur « régler le clavier » → cliquer sur « régler le WiFi »



**Figure 22** Réglage de WiFi

### 2.2.3. Modification du mot de passe

Nous avons accéder au menu des applications → Préférences → Configuration du Raspberry Pi. Dans la fenêtre qui s'ouvre, nous accédons au 1<sup>er</sup> onglet « Système » et nous avons cliqué sur le bouton « Changer le mot de passe » et saisissez notre nouveau mot de passe.



**Figure 23** Modifier le mot de passe

Sur cette même fenêtre, profitez-en pour décocher le case « se connecter en tant que raspberry pi » : l'ouverture de session ne se fera plus de manière automatique.

### 2.2.4. Activation des interfaces utilisées

On active le protocole SSH (Secure Shell) est un protocole de communication sécurisé, dans lequel toutes les données échangées entre deux machines distantes sont chiffrées, puis on active la caméra.

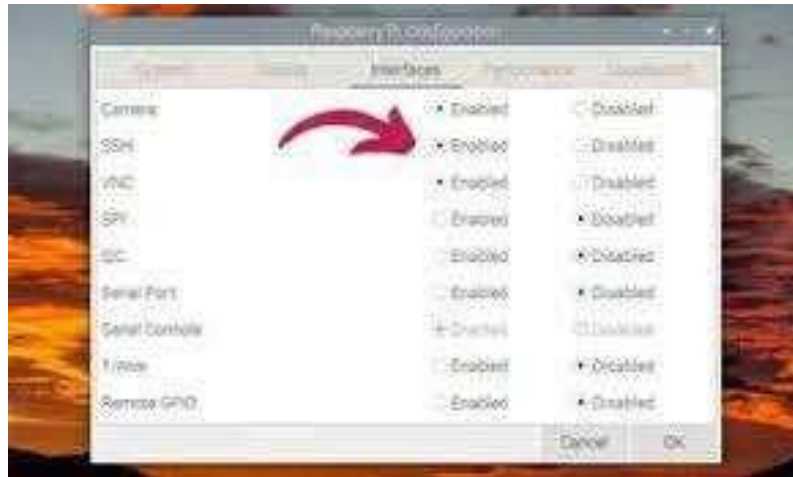


Figure 24 L'activation d'interfaces utilisées

### 2.2.5. Mettre à jour le système de Raspberry Pi

Mettre à jour le Raspberry Pi est important puisque cela nous apporte des correctifs de sécurité, des nouvelles versions de logiciels, de Raspbian, ect. Pour mettre à jours la liste des paquets en écrire la commande **sudo apt-get update**

```

individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Jan 27 23:02:03 2015 from 192.168.1.11
pi@pi ~$ sudo apt-get update
Hit http://archive.raspberrypi.org wheezy Release.gpg
Get:1 http://mirrordirector.raspbian.org wheezy Release.gpg [490 B]
Hit http://raspberrypi.collabora.com wheezy Release.gpg
Hit http://raspberrypi.collabora.com wheezy Release
Get:2 http://mirrordirector.raspbian.org wheezy Release [14.4 kB]
Hit http://archive.raspberrypi.org wheezy Release
Hit http://raspberrypi.collabora.com wheezy/rpi armhf Packages
Get:3 http://mirrordirector.raspbian.org wheezy/main armhf Packages [6,894 kB]
Hit http://archive.raspberrypi.org wheezy/main armhf Packages
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en_GB
Ign http://raspberrypi.collabora.com wheezy/rpi Translation-en
7% [Packages 467 kB] [3 Packages 443 kB/6,894 kB 6%] [Waiting for headers]

```

Télécharger et installer les mises à jour par la commande **sudo apt-get upgrade**

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
1 not fully installed or removed.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Setting up libpam-chkssshpwd:armhf (1.1.8-3.1+deb8u2+rp12) ...
mkdir: cannot create directory '/var/lib/chkssshpwd/': File exists
dpkg: error processing package libpam-chkssshpwd:armhf (--configure):
 subprocess installed post-installation script returned error exit status 1
Errors were encountered while processing:
 libpam-chkssshpwd:armhf
E: Sub-process /usr/bin/dpkg returned an error code (1)
pi@raspberrypi:~$

```

### 2.3. Installation d'OpenCV sur le Raspberry Pi

On commence par l'installation des dépendances, pour cela nous utilisons les commandes suivantes :

```
sudo apt-get install build-essential cmake pkg-config
sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
sudo apt-get install libxvidcore-dev libx264-dev
sudo apt-get install libgtk2.0-dev libgtk-3-dev
dev sudo apt-get install libatlas-base-dev gfortran
```

Puis on va faire l'installation de Python3 et pip 3 à travers les commandes suivantes:

```
Sudo apt-get install python3-dev
Sudo apt-get install python3-pip
```

Finalement l'installation d'OpenCV par les deux commandes suivantes :

```
Sudo apt-get install python3-opencv
```

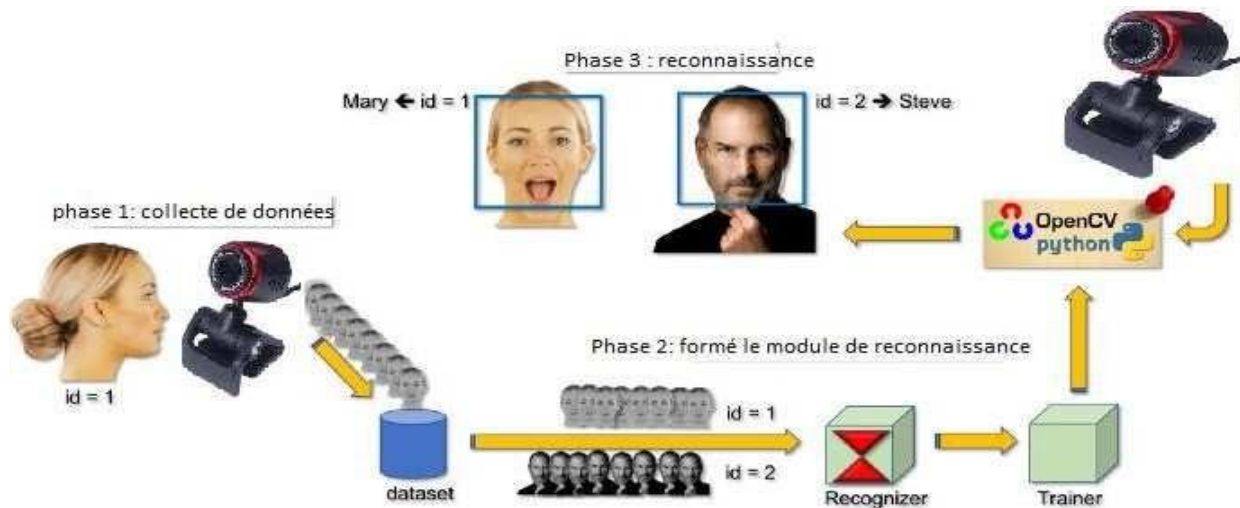
```
Pip3 install opencv-python
```

### 2.4. Système de reconnaissance faciale

Pour obtenir un système de reconnaissance faciale il faut travailler sur 3 phases bien distinctes :

- ✓ Détection des visages et collecte de données
- ✓ Former le système de reconnaissance
- ✓ Reconnaissance de visage

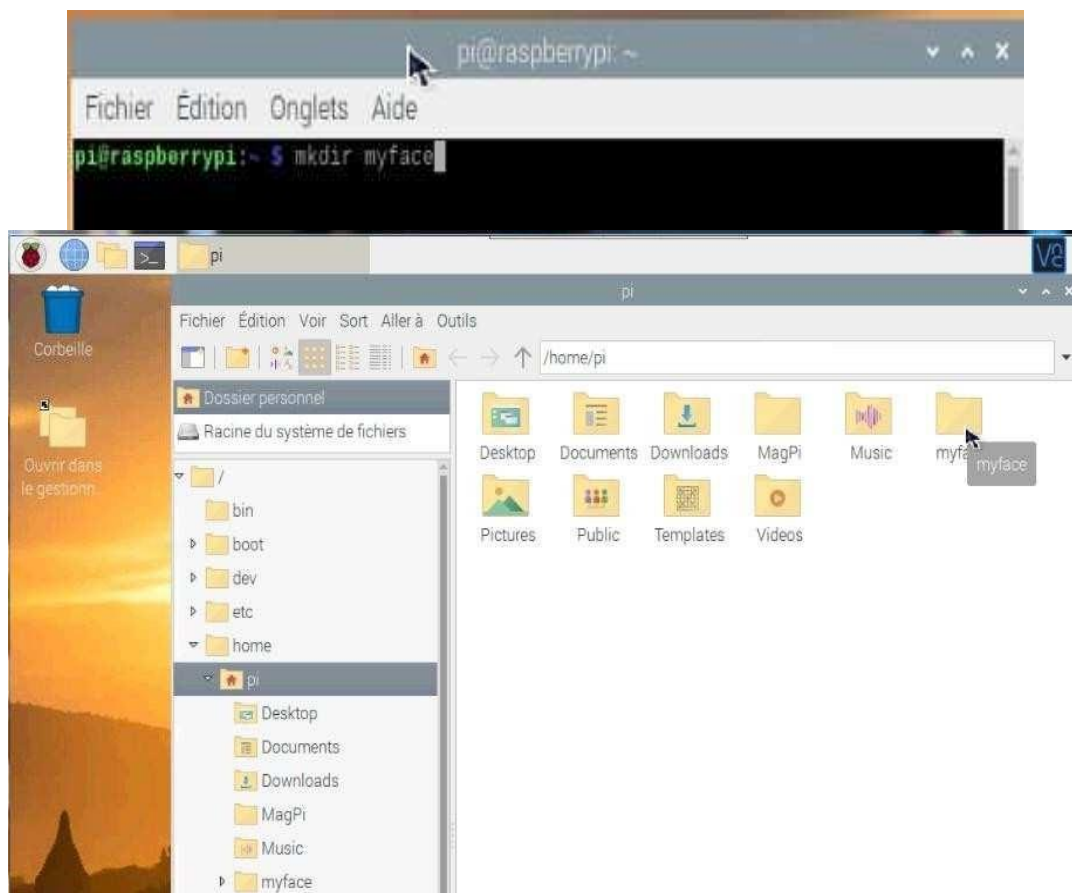




**Figure 25** Schéma fonctionnel du système de reconnaissance faciale

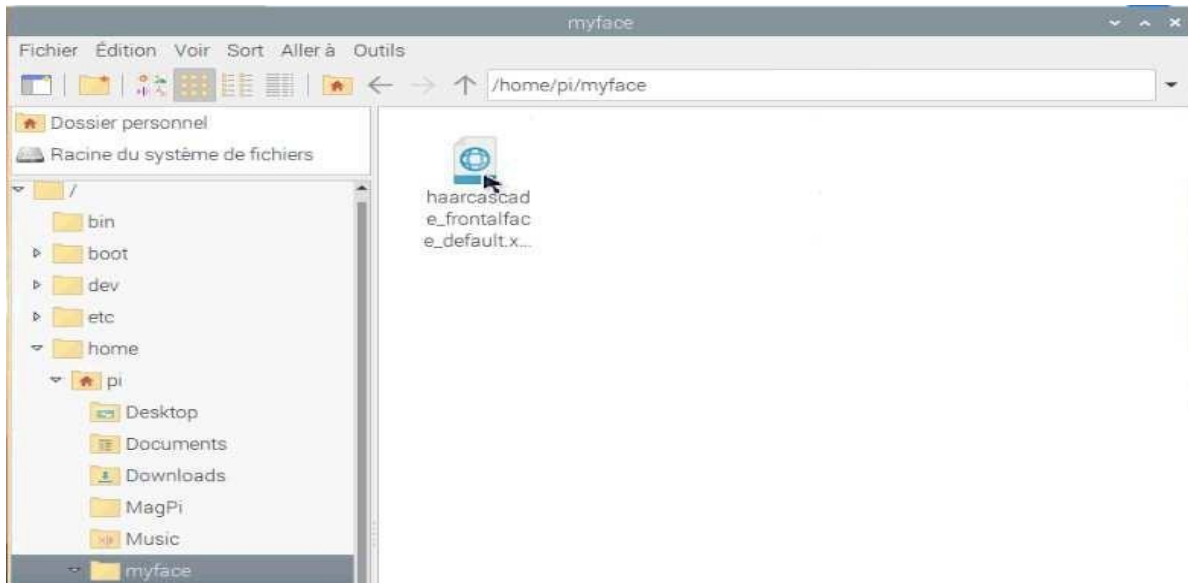
### 2.4.1. Collecte des données et détection du visage

Avant de commencer, nous allons créer un répertoire de travail dans lequel nous enregistrerons tous les fichiers et les programmes nécessaires à notre travail, nous l'avons appelé "myface". Pour cela, nous avons utilisé la commande suivante:



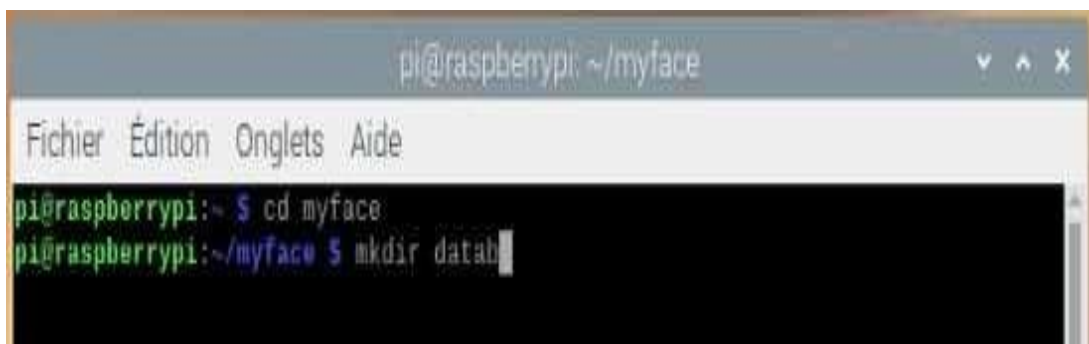
La première tâche est de recueillir les données pour lesquelles nous allons former notre classificateur. Nous avons écrit un code python « insertion.py » qui prendra 50 visages de chaque personne à l'aide d'OpenCV classificateur préformé.

OpenCV contient déjà de nombreux classificateurs préformés pour le visage, les yeux, le sourire, etc. Le classificateur que nous avons utilisé détectera les visages et le fichier cascade est disponible sur GitHub. Nous avons enregistré ce fichier dans notre répertoire de travail sous la forme de "haarcascade\_frontalface\_default.xml ».

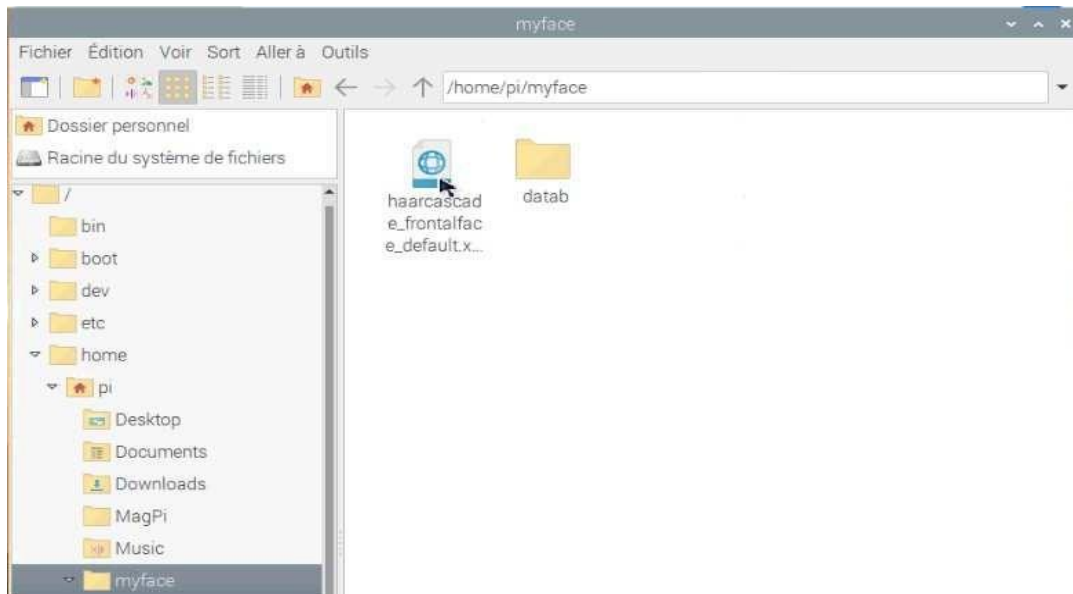


**Figure 26** Fichier HaarCascade

Ensuite, nous avons créé un sous-dossier nommé "datab" où nous enregistrerons les photos que nous prendrons avec le code écrit dans le script "insertion.py".



**Figure 27** Création d'un répertoire de travail



**Figure 28** Création de dossier d'image datab

Maintenant, nous écrivons le code dans le script « insertion.py », pour cela nous allons créer ce script avec la commande suivante :

```
pi@raspberrypi:~/myface $ sudo nano insertion.py
```

```
import cv2

import numpy as numpy
```

```
 facedetect=cv2.CascadeClassifier ('haarcascade_frontalface_default.xml')
```

```
cam = cv2.VideoCapture(0)
```

```
id=1 ; #changer pour chaque nouvelle personne
Num=0
```



```
while True:
```

```
    ret, img = cam.read()
```

```
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

Ici, nous définissons une nouvelle variable, le gris, comme img, converti en gris. Notons que cela indique BGR2GRAY. Il est important de noter qu'OpenCV lit les couleurs comme BGR (bleu vert rouge), où la plupart des applications informatiques lisent comme RVB (rouge vert bleu).

```
face = facedetect.detectMultiScale (gray, 1.3, 5)
```

```
for (x,y,w,h) in face:
```

```
    Num=Num+1
```

```
    #sauvegarder les photos dans le dossier datab
```

```
    cv2.imwrite("datab/User."+str(id)+'.'+str(Num) + ".jpg", gray[y:y+h,x:x+w])
```

```
    cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)
```

```
    cv2.waitKey(100) & 0xff
```

```
cv2.imshow("reconnaissance faciale",img)
```

```
cv2.waitKey(100) & 0xff
```

```
#Arreter si Num est superieur a 50
```

```
if(Num>50):
```

```
    break
```

Ensuite, nous appelons notre fonction de classification pour détecter les visages dans l'image. Le premier argument que nous passons est l'image en niveaux de gris. Le deuxième argument est le paramètre spécifiant dans quelle mesure la taille de l'image est réduite à chaque échelle

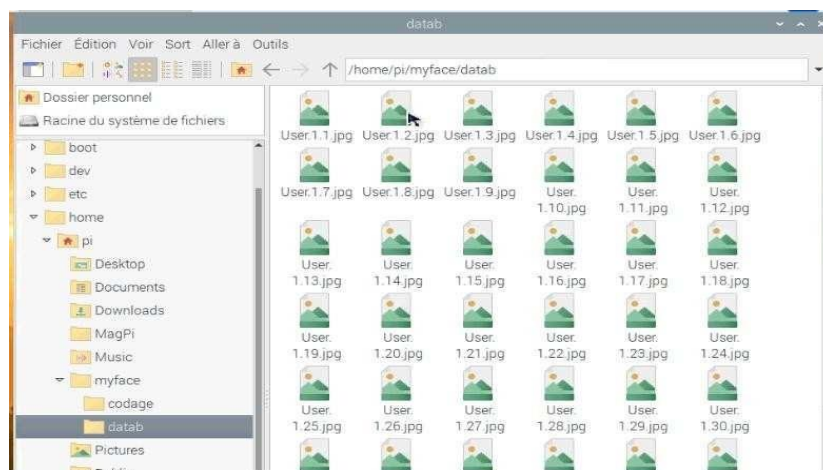
d'image. Le troisième argument est un paramètre spécifiant le nombre de voisins que chaque rectangle candidat doit avoir pour le conserver. Un nombre plus élevé donne moins de faux positifs.

```
cam.release()
cv2.destroyAllWindows()
```

Nous exécutons le script « insertion.py » par la commande suivante :

```
pi@raspberrypi:~/myface $ sudo python3 insertion.py
```

La détection des visages et la collecte de données sont maintenant terminées. Nous devrions avoir 50 images dans le répertoire nouvellement créé.



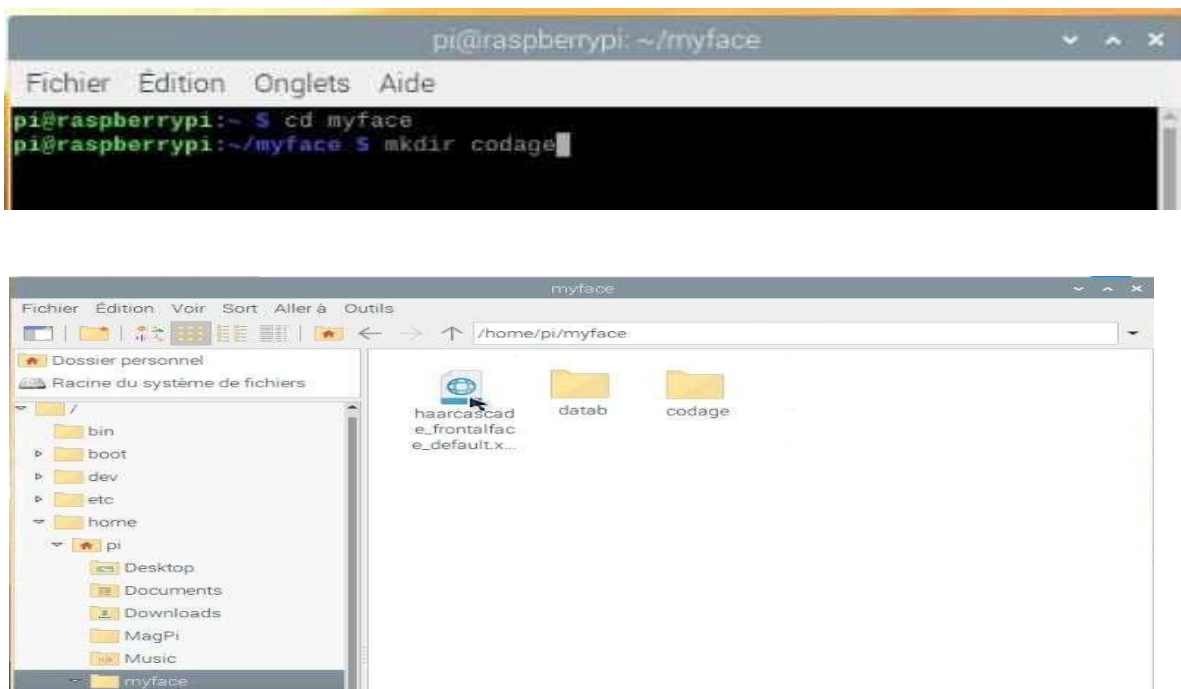
**Figure 29** Les images enregistrées dans le répertoire datab

#### 2.4.2. Former le système de reconnaissance

Nous pouvons maintenant entraîner le module de reconnaissance en fonction des données que nous avons collectées à l'étape précédente.

Dans cette partie, nous allons créer un formateur qui générera un fichier trainingData.yml pour notre modèle de reconnaissance faciale. Donc, fondamentalement, nous allons former notre reconnaissance faciale.

Premièrement nous avons créé le dossier «codage» dans le même dossier de notre travail.



**Figure 30** Création de dossier codage

Ensuite, nous allons créer un script `codage_photo.py` dans le dossier du projet pour écrire le code qui va faire le codage des photos prendre de l'étape précédente.

Pour créer le script nous utilisons la commande suivante :

```
pi@raspberrypi:~/myface $ sudo nano codage_photos.py
```

Nous sommes besoin d'importer les bibliothèques suivantes :

```
import cv2,os
import numpy as np
```

Nous utiliserons le logiciel de reconnaissance de visage LBPH (LOCAL BINARY PATTERNS HISTOGRAMS), inclus dans le package OpenCV. Pour initialiser le reconnaisseur et le détecteur de visage :

```
recognizer=cv2.face.createLBPHFaceRecognizer()
facedetect=cv2.CascadeClassifier ('haarcascade_frontalface_default.xml')
path= "datab"
```

Ensuite, nous allons charger les images d'entraînement à partir de datab.

Nous allons donc écrire une fonction qui prendra des images du dossier «datab», et obtiendra également les identifiants de son nom de fichier.

Disons que la fonction sera `getImagesAndLabels (path)`.

```
imagePaths = [os.path.join(path,f) for f in os.listdir(path)]
```

Nous avons créé le chemin de l'image.

Cela obtiendra le chemin de chaque image dans le dossier.

Maintenant nous devons créer deux listes de visages et d'identifiants pour stocker les visages et identifiants.

```
face = [ ]
IDs = [ ]
```

Permet de mettre cela dans une boucle.

Et une fois que nous aurons l'image, nous nous entraînerons sur les images. La fonction complète ressemblera donc à ceci.

```
for imagePath in imagePaths :
```

```
    #chargement de l'image et conversion en échelle de
```

```
    gris facImg=Image.open(imagePath).convert("L")
```

```
    #Maintenant, nous convertissons l'image PIL en tableau
```

```
    numpy faceNp=np.array(facImg,"uint8")
```

```
    #obtenir l'identifiant de l'image
```

```
    ID=int (os.path.split(imagePath) [-1].split(".")[1])
```

```
    #extraire le visage de l'échantillon d'image
```

```
    d'entraînement
```

```
    faces=facedetect.detectMultiScale(faceNp)
```

```
    print (ID)
```

```
        face.append(faceNp[y:y+h,x:x+
```

```
        w])IDs.append(ID)
```

```
cv2.imshow("training",faceNp)
```

```
cv2.waitKey(10)
```

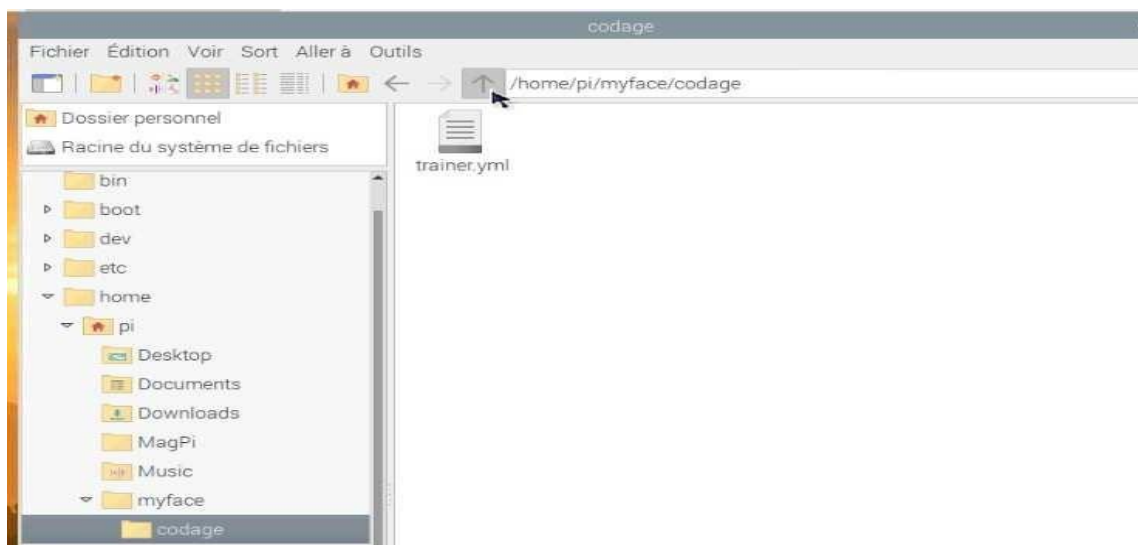
Nous avons terminé ici, appelons la fonction et fournissons les données, et nous donnerons également le chemin pour sauvegarder notre fichier qui sera généré après l'entraînement.

```
IDs,faces=getImagesWithID(path)
recognizer.train(faces,np.array(IDs))
recognizer.save('/codage/trainingData.yml')
cv2.destroyAllWindows()
```

Nous enregistrons le fichier de `codage_photo.py` et une fois que nous l'exécutons par cette commande :

```
pi@raspberrypi:~/myface $ sudo python3 codage_photo.py
```

Et nous obtiendrons ce fichier dans notre dossier «codage».



**Figure 31** Fichier d'entrainement

#### 2.4.3. Reconnaissance faciale

Le module de reconnaissance que nous avons configuré dans la section précédente peut maintenant être utilisé pour reconnaître les visages. Cela nous donnera la confiance et l'identifiant de l'étiquette (dans quelle mesure le module de reconnaissance est-il par rapport à cette correspondance).

Premièrement, nous allons créer un script `reconnaissance.py` dans le dossier du projet pour écrire le code qui va faire la reconnaissance des visages, nous utilisons la commande suivante :

```
pi@raspberrypi:~/myface $ sudo nano reconnaissance.py
```

Importation de bibliothèques

```
import cv2  
import numpy as np  
import os
```

Ensuite, nous chargeons le classificateur qui détectera les visages et le module de reconnaissance qui prédira les visages et les données entraînées.

```
recognizer = cv2.face.createLBPHFaceRecognizer()  
recognizer.load('/home/pi/myface/codage/trainer.yml')  
  
faceCascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

Choisissons la police, car nous afficherons le nom de la personne

```
font = cv2.FONT_HERSHEY_SIMPLEX
```

Nous incluons un nouveau tableau, donc nous afficherons des «noms», au lieu des identifiants numérotés:

```
# noms liés aux identifiants: exemple ==> sarra: id=1, etc  
names = ['None','steve']
```

Nous allons maintenant créer l'objet de capture vidéo

```
cam = cv2.VideoCapture(0)
```

Ici, nous arrivons à la boucle et la fonction principale

- Capture d'images à partir de l'objet de capture vidéo.
- Convertissez les images en échelle de gris.
- Détectez le visage dans l'objet.
- Reconnaissez l'ID à partir des données disponibles dans le fichier `trainer.yml`.
- Créez un rectangle sur le visage et imprimez l'ID de la personne détectée sur le rectangle.

```
while True:
    ret, img =cam.read()

    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)

    faces =
    faceCascade.detectMultiScale(
        gray,

        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (int(minW), int(minH)),
```

Ensuite, nous allons détecter un visage, comme nous l'avons fait auparavant avec le classificateur `haarCascade`. Ayant un visage détecté, nous pouvons appeler la fonction la plus importante dans le code ci-dessus:



```

for(x,y,w,h) in faces:

    cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)

    id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

    # Vérifiez si la confiance est inférieure à 100 ==> "0" correspond
    parfaitement
    if (confidence < 100):

        id = names[id]

        confidence = " {0}%".format(round(100 - confidence))
    else:

        id = "unknown"

        confidence = " {0}%".format(round(100 - confidence))

    cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)

    cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)

```

```

cv2.imshow('camera',img)

k = cv2.waitKey(10) & 0xff # Appuyez sur 'ESC' pour quitter la
vidéo  if k == 27:

    break

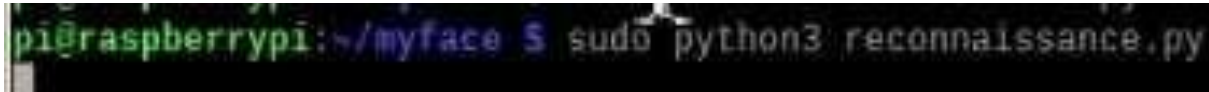
```

La fonction de `recognizer.predict ()` prendra comme paramètre une partie capturée du visage à analyser et retournera son propriétaire probable, indiquant son identifiant et le degré de confiance du dispositif de reconnaissance par rapport à cette correspondance.

Notons que l'indice de confiance renverra «zéro» s'il est considéré comme une correspondance parfaite

Et enfin, si le dispositif de reconnaissance pouvait prédire un visage, nous plaçons un texte sur l'image avec l'identifiant probable et combien est la «probabilité» en% que la correspondance soit correcte («probabilité» = 100 - indice de confiance). Sinon, une étiquette «Inconnu» est apposé sur le visage.

Nous exécutons le script « reconnaissance.py par la commande suivante :



```
pi@raspberrypi:~/myface $ sudo python3 reconnaissance.py
```

### **3. La supervision par l'application mobile**

### **4. Conclusion**

Le projet a conçu et mis en œuvre un système de reconnaissance facial basé sur le Raspberry Pi. Les aspects du système sont les suivants : détection de visage à l'aide de module OpenCV, faire la reconnaissance de visage à partir d'une vidéo à l'aide d'une caméra USB et finalement la supervision de ce système par l'application mobile. Dans ce dernier chapitre nous avons présenté ce système avec ses différentes parties et leur principe de fonctionnement ainsi que sa réalisation pratique.

## **Conclusion générale**

En conclusion, l'utilisation d'un système de reconnaissance faciale basé sur Raspberry Pi 4, avec une supervision via une application mobile, offre de nombreuses possibilités et avantages. Ce système permet une automatisation efficace de la reconnaissance des visages, offrant une solution pratique et sécurisée pour diverses applications.

Le Raspberry Pi 4 est une ordinateur mono-carte puissante et abordable, qui peut être utilisé pour exécuter des algorithmes de reconnaissance faciale en temps réel. Il dispose de suffisamment de puissance de calcul pour traiter les données d'image et exécuter des modèles d'apprentissage automatique. Cette capacité permet au système de reconnaître rapidement les visages et d'effectuer des tâches de vérification ou d'identification.

L'utilisation d'une application mobile pour superviser le système de reconnaissance faciale apporte une facilité d'utilisation et une flexibilité accrues. L'application mobile peut être utilisée pour configurer et gérer les paramètres du système, visualiser les résultats de la reconnaissance en temps réel et recevoir des notifications. Cela permet aux utilisateurs de surveiller et de contrôler le système à distance, ce qui est particulièrement pratique dans les environnements où la mobilité est requise.

De plus, l'utilisation du Raspberry Pi 4 et d'une application mobile permet une implémentation à faible coût par rapport à d'autres solutions commerciales. Cela le rend accessible aux petites entreprises, aux projets personnels ou à des environnements où les ressources financières sont limitées.

En résumé, l'utilisation d'un système de reconnaissance faciale basé sur Raspberry Pi 4, supervisé par une application mobile, offre une solution puissante, flexible et abordable pour l'automatisation de la reconnaissance des visages. Ce système peut être utilisé dans de nombreux domaines et offre de nombreux avantages, bien qu'il soit important de prendre en compte les aspects liés à la confidentialité et à l'éthique lors de sa mise en œuvre.

## Bibliographie

1. CNIL. [En ligne] <https://www.cnil.fr/en/node/24658>.
2. THALES. [En ligne] <https://www.thalesgroup.com/fr/europe/france/dis/gouvernement/inspiration/biometrie?fbclid=IwAR22EchCwRzT9t4R2jVzb1FwPafxdYwQ2lUTnNzrbHZHFSac4FE0YIdRUi4>.
3. Kaspersky. [En ligne] <https://www.kaspersky.fr/resource-center>.
4. clicours. [En ligne] [https://www.clicours.com/4-secrets-pour-gerer-avec-succes-un-projet/?fbclid=IwAR1rAS8\\_Vhv2LaJoZ-iNLQ5PmwK-PsI2ehOyhIJfH4V3DFLflsO3qIvUBTg](https://www.clicours.com/4-secrets-pour-gerer-avec-succes-un-projet/?fbclid=IwAR1rAS8_Vhv2LaJoZ-iNLQ5PmwK-PsI2ehOyhIJfH4V3DFLflsO3qIvUBTg).
5. talend. [En ligne] [https://www.talend.com/fr/resources/?fbclid=IwAR1dDNSxwgitRl\\_9wLDIdbnstAF8YMCiQGMOXUvBtntZKE3DGbc3dTqJSAw](https://www.talend.com/fr/resources/?fbclid=IwAR1dDNSxwgitRl_9wLDIdbnstAF8YMCiQGMOXUvBtntZKE3DGbc3dTqJSAw).
6. **M.Fedias**. « *Combinaisons de Données d'Espaces Couleurs et de Méthodes de Vérification d'Identité pour l'Authentification de Visages* ». 21 Février 2013. thèse.
7. **S.Madene, I.Chaibeddra et**. *Chaibeddra et S. Madene « Détection visuelle d'objets statiques et dynamiques dans un environnement de type route et classification en exploitant l'apprentissage profond*.
8. pangeanic. [En ligne] [https://blog.pangeanic.com/fr/quelle-est-la-diff%C3%A9rence-entre-l'apprentissage-automatique-et-l'apprentissage-profond\[8?fbclid=IwAR1bzdA8q\\_XZedeQHztCP1HBVkwEloUZQizYJfZv7e96\\_Pd7E99xMPgELBc](https://blog.pangeanic.com/fr/quelle-est-la-diff%C3%A9rence-entre-l'apprentissage-automatique-et-l'apprentissage-profond[8?fbclid=IwAR1bzdA8q_XZedeQHztCP1HBVkwEloUZQizYJfZv7e96_Pd7E99xMPgELBc).
9. **Rosenblatt, F**. *The perceptron: a probabilistic model for information storage and organization in the brain* ». 1988.
10. wikipedia. [En ligne] [https://fr.wikipedia.org/wiki/Raspberry\\_Pi](https://fr.wikipedia.org/wiki/Raspberry_Pi) .
11. framboise. [En ligne] <https://www.framboise314.fr/raspberry-pi-4-4-nouveautes-qui-vont-vous-faire-craquer/>.
12. *Le Monde*. pp. [https://www.lemonde.fr/pixels/article/2018/07/25/je-n-imaginai-pas-que-python-connaissait-un-tel-succes\\_5335917\\_4408996.html](https://www.lemonde.fr/pixels/article/2018/07/25/je-n-imaginai-pas-que-python-connaissait-un-tel-succes_5335917_4408996.html).
13. openclassrooms. [En ligne] <https://openclassrooms.com/fr/courses/235344-apprenez-a-programmer-en-python/230722-faites-vos-premiers-pas-avec-linterpreteur-de-commandes-python>.
14. towardsdatascience. [En ligne] <https://towardsdatascience.com/what-is-a-perceptron-210a50190c3b>.

## **Résumé**

Un système de reconnaissance faciale est une application logicielle visant à reconnaître une personne grâce à son visage de manière automatique. C'est un domaine de la vision par ordinateur consistant à reconnaître automatiquement une personne à partir d'une image de son visage. Il s'agit d'un sujet particulièrement étudié en vision par ordinateur, avec de très nombreuses publications et brevets, et des conférences spécialisées.

La reconnaissance de visage a de nombreuses applications en vidéosurveillance, biométrie, robotique. Ces systèmes sont généralement utilisés à des fins de sécurité pour déverrouiller ordinateur/mobile/console. Ils sont appréciés car considérés comme peu invasifs, en comparaison avec les autres systèmes biométriques (empreintes digitales, reconnaissance de l'iris...). Le fonctionnement de ces systèmes se base sur une ou plusieurs caméras pour reconnaître l'utilisateur.

Ils peuvent également être utilisés afin de faciliter la vie de l'utilisateur, pour identifier des visages sur des images. Ces systèmes se basent alors sur des photos/vidéos (base de données) d'une ou plusieurs personnes.

L'objectif de notre projet sera de mettre en œuvre un système d'identification de personnes, cette identification s'appuie sur des séquences vidéo. Ces séquences sont analysées et on extrait l'information visuelle liée au visage, on construit alors notre base de données et on présente la technique permettant de reconnaître une personne parmi l'ensemble. Le système de reconnaissance faciale est basé sur le principe de filtre de Haar et nous utilisons la bibliothèque OpenCV et son implémentation de l'algorithme de Viola et Jones pour la détection de visage.

Mots-clés: Reconnaissance faciale, filtre de Haar, Viola et Jones, Prétraitement, Base de données, OpenCV.

## **Abstract**

A facial recognition system is a software application designed to automatically recognize a person through their face. It is an area of computer vision that automatically recognizes a person from an image of their face. It is a subject particularly studied in computer vision, with numerous publications and patents, and specialized conferences.

Face recognition has many applications in video surveillance, biometrics, robotics. These systems are generally used for security purposes to unlock computer/mobile/console. They are appreciated as considered not invasive, compared to other biometric systems (finger prints, iris recognition...). The operation of these systems is based on one or more cameras to recognize the user.

They can also be used to make life easier for the user, to identify faces in images. These systems are then based on photos/videos (database) of one or more people.

The objective of our project will be to implement a system of identification of people, this identification is based on video footage. These sequences are analyzed and visual information related to the face is extracted, our database is built and the technique is presented to recognize a person among the whole. The facial recognition system is based on Haar's filter principle and we use the OpenCV library and its implementation of the Viola and Jones algorithm for face detection.

Keywords: Facial recognition, Haar filter, Viola and Jones, Pre-processing, Database, OpenCV.