

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي و البحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة أبي بكر بلقايد- تلمسان

Université Aboubakr Belkaïd-Tlemcen

كلية التكنولوجيا

Faculté de Technologie

Département de Génie Electrique et Electronique (GEE)

Filière : Electronique



MASTER INSTRUMENTATION
PROJET DE FIN D'ETUDES

Présenté par : MEHDAOUI SARRA & MAATI IMANE

Intitulé du Sujet

**Conception d'un système de sécurité a reconnaissance facial à
l'aide d'un Raspberry pi**

Soutenu en 2020, devant le jury composé de :

Mr BENAHMED Nasreddine

Prof à l'Université de Tlemcen

Président

Mr BENHABIB Mohamed Choukri

MCA à l'Université de Tlemcen

Examineur

Mr MOULAI KHATIR AHMED NASSIM

MCB à l'université de tlemcen

Encadreur

Année Universitaire 2019-2020

REMERCIEMENTS



Nous remercions et glorifions Allah le tout puissant et miséricordieux d'avoir guidé nos pas pour accomplir ce modeste travail et de le mené à terme. Ce dernier, a été effectué au sein de l'université (Abou Bekr Belkaid) de Tlemcen et plus précisément au département de génie électrique et électronique de la faculté de technologie.

Nous tenons à remercier vivement notre encadreur, Mr MOULAI KHATIR AHMED NASSIM, d'avoir accepté de nous guider tous le long de ce travail. Et cela, avec leur grande disponibilité, leur rigueur scientifique et leurs précieux conseils qui nous ont permis de travailler dans les meilleures conditions, et ce, depuis le début de la préparation de notre projet.

Nous tenons à remercier, aussi, Mr BENAHMED NASREDDINE l'honneur, de bien vouloir accepter de présider les jurys de notre soutenance.

Les remerciements sont, également, adressés à monsieur BENHABIB MOHAMMED CHOUKRI, nous sommes très sensibles à l'honneur que vous nous faites en acceptant d'examiner le présent travail que nous souhaitons est d'un bon niveau,

Par ailleurs, trouverons ici, nos sincères reconnaissances à tous les ingénieurs de laboratoires du département de génie électrique et électronique et à toute personne ayant contribué de près ou de loin à la réalisation de ce projet de fin d'étude.

Dédicace

Je tiens tout d'abord à remercier ALLAH tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce modeste travail.

Ce projet d'étude dédié à la mémoire de mon cher père qui a contribué toutes les années de sa vie à me soutenir et à renforcer ma confiance en moi, qui restera dans mon cœur

AU joyau le plus précieux de ma vie, à ma chère mère qui a souffert sans me laisser souffrir, qui n'a jamais cessé de me soutenir et de m'épauler pour que je puisse atteindre mon objectif. Qui n'a jamais dit non à mes exigences mon adorable mère que le dieu la protège et la garde

A ma chère sœur pour me soutenir et pour me donner ses conseils précieux durant toutes mes études

C'est un moment de plaisir de dédier ce projet à mes frères, en signe d'amour et de gratitude pour les sacrifices qu'ils m'ont toujours faits

A toutes la famille Maati et Saci

A Mr. MOULAI KHATIR AHMED NASSIM qui est toujours à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'ils ont bien voulu me consacrer et sans leur contribution ce mémoire n'aurait vu le jour

Et enfin à mes amis Hadjer, Ferial, Sarra qui n'ont jamais cessé de me soutenir pour faire tout ce que je pouvais dans tous les domaines de ma vie et surtout le domaine de mes études

Ce projet marque la fin de mes études comme l'aboutissement du soutien et de l'encouragement qui m'a été donnés tout au long de mes études, je les remercie infiniment pour ce dévouement très modeste.

Maati Imane

Dédicace

Tout d'abord, je rends grâce à Dieu le tout puissant de m'avoir donné la force de mener à terme ce modeste travail que je dédie particulièrement:

A mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, mon père que Dieu lui prête longue vie, santé et joie.

A la lumière de mes jours, la source de mes efforts, ma mère que j'adore.

A ma sœur Soumia qui m'a conseillée, aidée et encouragée durant toute la période de préparation de ce mémoire. A mes frères Dussama, Mohamed, Mehdi et Abderezzak qui m'ont été d'un grand soutien aussi bien moral que matériel.

Aux familles Mehdaoui et Mesaoudi, du plus grand au plus petit pour leurs encouragements.

A mon encadreur Dr MOULAI KHATIR AHMED NASSIM pour la qualité de ses orientations, de ses conseils, pour sa patience et sa disponibilité durant toute la période de ce travail.

A mes amies et sœurs notamment Rawida, Soumia, Rania, Imene pour leur soutien et leur encouragement. A tous qui m'ont inculqué le savoir du primaire jusqu'à l'université.

Mehdaoui Farra

Liste des abréviations

- LDR** : Linear discriminant analysis(Analyse discriminante linéaire)
- HAAR** : High Altitude Acute Response (Réponse aiguë à haute altitude)
- XML** : Extensible Markup Language(Extensible Markup (langue))
- CSV** : Comma-separated values (valeur séparée par des virgules)
- LBP** : Local binary patterns (modèle binaire locaux)
- SVM** : Vecteur de support machine
- ROI** : Region d'intérêt connue
- HOG**: Histogramme caractéristique orientées des gradients
- HDMI**: High definition multimedia interface (interface multimédia haute définition)
- USB**: Universal Serial Bus (Autobus de série universel)
- GPIO** : General - Purpose Input/Output (Généralités - Entrée/sortie d'objet)
- CPU** : Unité central de traitement
- GPU** : Unité de traitement graphique
- LCD**: Liquid Crystal Display (Affichage à cristaux liquides)
- SMSC**: Short Message Service Center (Centre de service de messages courts)
- SOC**: System On a Chip
- HD**: Haute définition
- OPENCV**: Open source computer vision
- SSH** : Secure Shell (Ligne de commande sécurisée)
- GITHUB**: A contraction of the words Git and Hub (Une contraction des mots Git et Hub)
- HTML**: HyperText Markup Language (HyperText Markup (langue))
- PHP**: Hypertext Preprocessor(Préprocesseur hypertexte)

TABLE DES FIGURES

Chapitre 1 :

Figure 1.1.Schéma général d'un système de reconnaissance des visages

Chapitre 2 :

Figure 2.1.Exemples de fonctionnalité similaires à Haar

Figure 2.2.Exemple de calcul d'image intégrale

Figure 2.3.Cascades Classifiées

Figure 2.4.Le Raspberry pi

Figure 2.5.Le Raspberry Pi 3 Modèle B

Figure 2.6.Les composants standards d'un Raspberry Pi 3 Modèle B

Figure 2.7.GPIO

Figure 2.8 .Alimentation 2.5A –5

Figure 2.9.Carte Micro SD

Figure 2.10.Un écran TFT

Figure 2.11.Caméra web

Figure 2.12.Un relais

Figure 2.13. Une serrure solénoïde

Chapitre 3:

Figure 3.1.Matériels de conception

Figure 3.2.Formatage de la carte microSD

Figure 3.3.Décompression du fichier Zip

Figure 3.4.Le logiciel Win32DiskImager

Figure 3.5.Copier le fichier image de raspbian sur la carte SD

Figure 3.6.Bureau du Raspberry Pi 3

Figure 3.7.Le terminal

Figure 3.8.Fichier keyboard en « gb »

Figure 3.9.Fichier keyboard en « fr »

Figure 3.10.La commande du redémarrage du raspberry Pi 3

Figure 3.11.Menu des applications

Figure 3.12.Réglage du wifi

Figure 3.13.Modification du mot de passe

Figure 3.14.L'activation d'interfaces utilisées

Figure 3.15.Caméra USB (Webcam)

Figure 3.16.Schéma fonctionnel du système de reconnaissance facial
Figure 3.17.Création d'un répertoire de travail
Figure 3.18.Phase 1 collecte de donnée
Figure 3.19.Fichier haarcascade
Figure 3.20.Création de dossier d'images datab
Figure 3.21.Prendre des photos avec le fichier insertin.py
Figure 3.22.Nos 50 images enregistrées dans le répertoire datab
Figure 3.23.Phase 2 former le module de reconnaissance
Figure 3.24.Création de dossier codage
Figure 3.25.Formation des photos
Figure 3.26.Fichier d'entraînement
Figure 3.27.Phase 3 reconnaissance
Figure 3.28.Exécution de script reconnaissance.py
Figure 3.29.Interface graphique du système de reconnaissance facial
Figure 3.30.La base de données database.csv
Figure 3.31.Câblage du Relais et Serrure électrique avec le Raspberry Pi
Figure 3.32.Tableau généré par la commande phpinfo sur une Raspberry
Figure 3.33.Obtention de l'adresse IP avec ifconfig
Figure 3.34.La page web de notre projet
Figure 3.35.Organigramme du système
Figure 3.36.Circuit global du système
Figure 3.37.Montage expérimental global

Sommaire

Introduction générale	1
Chapitre I : La reconnaissance faciale	
Introduction.....	3
I.1.La reconnaissance faciale.....	3
I.2.Historique	4
I.3.Domaines d’application de la reconnaissance faciale.....	4
I.4.Techniques de détection et de reconnaissance faciale	5
I.5.Problématique.....	7
I.6.Avantages et inconvénients de la reconnaissance de visage.....	8
I.7.Conclusion.....	9
Chapitre II : Etude et réalisation du système de reconnaissance faciale	
Introduction.....	10
II.1.Spécification de la technique et langage utilisé	10
II.2.Techniques de détection.....	10
II.3.Procédure de détection de Haar-cascade	11
II.3.1.Haar comme caractéristique.....	11
II.3.2.Image intégrale.....	11
II.3.3.AdaBoost.....	12
II.3.4.Classificateur en cascade	12
II.4.Type de fichier utilisé	13
II.4.1.Fichier XML	13
II.4.2.Fichier CSV	13
II.4.3.Format de gris	13
II.5.Environnement de développement.....	14
II.5.1.Environnement hardware	14
II.5.1.1. Un ordinateur	14
II.5.1.2. Un Raspberry Pi.....	14
II.5.1.2.1. Présentation du Raspberry pi	14
II.5.1.2.2. Les modèles de Raspberry Pi.....	15
Les modèles A	15
Les modèles B	16

Les modèles Zero	19
II.5.1.2.3.Le Raspberry Pi 3 modèle B	20
II.5.1.2.4.Spécification technique du microcontrôleur Raspberry Pi	21
II.5.1.2.5.Les composants du Raspberry Pi modèle B	22
II.5.1.2.6.Connexion des composants de Raspberry Pi	26
II.5.1.3.Un Ecran TFT LCD 5’’	27
II.5.1.4.Une camera Web	28
II.5.1.5.Un relais	29
II.5.1.6.Une serrure solénoïde	29
II.5.2.Environnement software	30
II.5.2.1.Raspbian.....	30
II.5.2.2. Langage python.....	30
II.5.3.Les bibliothèques utilisées	31
II.5.3.1.Numpy	31
II.5.3.2.Tkinter	31
II.5.3.3.OpenCv	31
II.6.Présentation de la bibliographe OpenCv	32
II.6.1.Historique.....	32
II.6.2.Qu’est ce que OpenCv	32
II.6.3.Fonctionnalités	32
II.6.1.Traitement d’images	33
II.6.1.Traitement vidéo	33
II.7. Conclusion	33
Chapitre III: Conception et analyse	
Introduction.....	35
III.1.Matériel de conception	35
III.2. Préparation du système d’exploitation sur micro SD	36
III.2.1.Formater la carte SD	36
III.2.2.Installer Raspbian sur la carte SD	36
III.2.3.Premier démarrage du système.....	39
III.3. Première configuration	39
III.3.1.Les premiers réglages	39
III.3.2.Le clavier en français.....	39

III.3.3.	Configuration de la localisation.....	41
III.3.4.	Modification du mot de passe.....	42
III.3.5.	L'activation des interfaces utilisées.....	42
III.4.	Mettre à jour le système de la Raspberry pi	43
III.4.1.	Mettre à jour le système de la liste des paquets	43
III.5.	Activer la camera USB (webcam)	44
III.6.	Installation opencv sur le Raspberry Pi	45
III.6.1.	Mettre à jours le système d'exploitation	45
III.6.2.	Installer les dépendances	46
III.6.3.	Installation python 3 et pip 3	46
III.6.4.	Installation opencv.....	46
III.6.5.	Dépendances supplémentaire pour OpenCv et la caméra	47
III.7.	Système de reconnaissance facial.....	47
III.7.1.	Collecte de données pour détection du visage.....	48
III.7.2.	Procédure pas à pas du code	50
III.7.3.	Entraînement de la reconnaissance.....	55
III.7.4.	Utilisation de l'outil de reconnaissance pour la reconnaissance faciale	60
III.8.	Création d'une base de données avec csv file	66
III.9.	Pilotage de relais avec les GPIO de raspberry	68
III.10.	Diagramme de circuit et explication.....	69
III.11.	Création de la page HTML.....	70
III.12.	Conception du système	75
III.12.1.	Organigramme du système de reconnaissance faciale à base de Raspberry	75
III.13.	Conclusion	78
	Conclusion générale	80
	Références Bibliographique	

Introduction générale

Introduction générale

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 modernes est la reconnaissance faciale.

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 visages stockés dans une base de données. Il s'agit d'un sujet particulièrement étudié en vision par ordinateur, avec de très nombreuses publications et brevets, et des conférences spécialisées.

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

Dans ce mémoire de Master, nous allons proposer une architecture d'un système de verrouillage intelligent sous la forme d'un box domotique à faible coût, basé sur la reconnaissance faciale et qui présente comme éléments intelligents : une carte Raspberry Pi 3, une simple webcam, et un écran tactile, un relais et une serrure électrique.

A partir d'une recherche dans une base de données le système va identifier chaque personne présenté en face de la webcam, pour les visages connus on a l'affichage d'un nouveau cadre qui présente un bouton de déverrouillage ainsi que la date et l'heure de l'identification de cet individu, le déverrouillage provoque aussi l'introduction de ces informations d'identification dans une base de données csv qui peut être visualisé via une page web sous réseau local

Ce mémoire 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 que 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 I :

La reconnaissance faciale

Chapitre I : La reconnaissance faciale

Introduction

Durant la vie quotidienne chacun de nous identifie tout au long de la journée différents visages. Ainsi lorsque nous rencontrons une personne, notre cerveau va chercher dans notre mémoire et vérifie si cette personne est répertoriée ou non, c'est une tâche aisée pour les humains.

En est-il de même pour une machine ? Dans ce premier chapitre, nous allons justement présenter les grandes lignes de notre travail, nous allons ainsi expliquer des notions générales sur la reconnaissance faciale, le fonctionnement d'un système de reconnaissance faciale, On va aussi aborder quelques techniques utilisées, des exemples et les domaines d'application.

I.1.La reconnaissance faciale

La reconnaissance faciale est une technique qui permet à partir des traits de visage :

- **D'authentifier une personne** : c'est-à-dire, vérifier qu'une personne est bien celle qu'elle prétend être (dans le cadre d'un contrôle d'accès)

Ou

- **D'identifier une personne** : c'est-à-dire, de retrouver une personne au sein d'un groupe d'individus, dans un lieu, une image ou une base de données.

En pratique, la reconnaissance peut être réalisée à partir d'images fixes (photos) ou animées (enregistrements vidéo) et se déroule en deux phases :

1. A partir de l'image, un modèle ou « gabarit » qui représente, d'un point de vue informatique, les caractéristiques de ce visage sont réalisées. Les données extraites pour constituer ce gabarit sont des données biométriques.
2. La phase de reconnaissance est ensuite réalisée par la comparaison de ces modèles préalablement réalisés avec les modèles calculés en direct sur des visages présents sur l'image candidate.

Dans le cas de l'authentification, le système vérifie si l'identité prétendue est bien la bonne en comparant le modèle du visage présenté au modèle préalablement enregistré correspondant à l'identité prétendue.

Dans le cas de l'identification, le système vérifie si le modèle du visage présenté correspond à l'un des modèles contenus dans la base de données. Les résultats de la comparaison correspondent à celui ou ceux présentant le score de similarité le plus élevé parmi ceux dépassant un certain seuil prédéterminé.

La reconnaissance faciale ne doit pas être confondue avec la détection de visage qui caractérise la présence ou non d'un visage dans une image indépendamment de la personne à qui il appartient [1].

Chapitre I : La reconnaissance faciale

I.2.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 visages familiers ou non familiers, mais d'une épreuve consistant à apparier des photographies de visages non familiers présentés sous différents éclairages et selon des angles différents et nécessitant une bonne capacité d'intégration Visio-spatiale [2].

L'utilisation des techniques de reconnaissance faciale a connu un développement à grande échelle de puis le milieu des années 90, avec l'utilisation efficace de nouvelles technologies, notamment l'ordinateur et sa capacité de traitement d'images. L'utilisation de ces techniques existe depuis qu'une machine est capable de comprendre ce qu'elle « voit » lorsqu'on la connecte à une ou plusieurs caméras, c'est à dire que les premiers essais datent du début des années 70 (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, etc [3].

Ces méthodes sont très peu robustes, car elles font de nombreuses suppositions en se plaçant dans des cas très simples (visage de face, bonnes conditions d'illuminations, etc. L'une des premières tentatives de reconnaissance de visage est faite par Takeo Kanade en 1973 lors de sa thèse de doctorat à l'Université de Kyoto [4] [5].

I.3.Domains d'application de la reconnaissance faciale

La sécurité reste le domaine d'application principal. Dans ce domaine, la reconnaissance des visages est nécessaire à l'identification et à l'authentification. Un bon exemple de cet usage est En 2017, Gemalto (maintenant Thales) signe l'équipement de nouveaux cas de contrôle automatisés du système PARAFE (Passage Automatisé Rapide aux Frontières Extérieures) au sein des aéroports de Paris. La solution a évolué de la reconnaissance des empreintes digitales à la reconnaissance faciale en 2018 et a réduit le temps moyen de passage de 30-45 secondes à 10-15 secondes [6].

Le contrôle policier peut y avoir recours bien qu'il reste très encadré en Europe. En 2016, « l'homme au chapeau » des attentats de Bruxelles aurait été identifié grâce à un logiciel de reconnaissance faciale du FBI [7].

Chapitre I : La reconnaissance faciale

Les drones équipés de caméra offrent une combinaison intéressante pour la reconnaissance faciale lors de grands événements, par exemple. Les drones, en position fixe jusqu'à 100 mètres de haut, peuvent porter des optiques lourdes (10kg) et identifier une personne à 800 mètres. De plus comme ils sont reliés par câble, l'autonomie électrique est illimitée et les communications (par câble aussi) sont sécurisées [7].

Google et Facebook ont implémenté des algorithmes pour identifier les personnes dans leur base de données de photos. En 2014 Facebook annonçait le lancement de son programme baptisé DeepFace, capable de déterminer si deux visages photographiés appartiennent à la même personne, avec une précision de 97,25%. Soumis au même test, les humains répondent correctement dans 97,53% des cas, soit à peine 0,28% de mieux que le programme de Facebook. En juin 2015, Google renchérisait avec FaceNet, nouveau système de reconnaissance faciale aux scores jamais égalés : 99,63% de précision au test de référence Labeled Facebooks in The Wild ; 95 % sur la base YouTube Faces DB. En utilisant un réseau neuronal artificiel et un nouvel algorithme, la firme de Mountain View est parvenue à rapprocher un visage et son propriétaire à la quasi-perfection. Une technologie intégrée dans Google Photos pour trier les clichés et les tagger automatiquement en fonction des personnes reconnues et, preuve de sa pertinence, devenue rapidement disponible en ligne dans une version open-source officielle, OpenFace [7].

La santé a aussi connu des avancements considérables dans ce domaine.

Grâce au Deep Learning et à la reconnaissance faciale et des émotions, il est déjà possible :

De suivre plus précisément la consommation de médicaments d'un patient ;

De détecter une maladie génétique telle que le syndrome de DiGeorge avec un taux de réussite de 96,6% ;

D'accompagner les procédures de prise en charge de la douleur [7].

Il existe plein d'autres domaines d'application de la reconnaissance faciale tel que :

Le commerce, les systèmes de déverrouillage, les filtres de snapchat.... etc.

I.4. Techniques de détection et de reconnaissance faciale

Il existe une multitude de techniques consacrées à la détection du visage. Mais pour la plupart il est d'intérêt que ces techniques se basent sur des éléments du visage qui sont le moins susceptibles aux changements :

Chapitre I : La reconnaissance faciale

Les grands traits supérieurs des orbites, les secteurs entourant les pommettes, les côtés de la bouche et d'autres caractéristiques similaires de façon à ignorer les changements comme la coupe de cheveux [3]. Les méthodes de base les plus utilisées sont :

- *Le traitement automatique du visage*

C'est une technologie rudimentaire, elle caractérise les visages par des distances et des proportions entre des points particuliers comme les deux yeux, le nez, les coins de la bouche. Aussi éprouvé que les autres technologies, le traitement automatique du visage est le plus efficace dans des situations de capture d'image avec peu d'éclairage [8].

- *Eigenface*

Les « Eigenfaces » furent le premier type de caractérisation utilisé avec succès dans des traitements faciaux tels que la détection et la reconnaissance du visage [9].

Elle utilise une représentation des éléments caractéristiques d'une image de visage à partir d'images modèles en niveau de gris. Des variantes d'Eigenface sont fréquemment utilisées comme base pour d'autres méthodes de reconnaissance [10].

- *L'analyse des points particuliers*

Elle est la technique d'identification faciale la plus utilisée. Cette dernière se rapproche d'Eigenface, mais elle est capable de s'adapter à des changements d'aspect facial (sourire, froncement de sourcils, ...). Les ingénieurs numériques l'utilisent souvent [11].

- *LDA (Linear discriminant analysis) fisher*

Elle fait partie des techniques d'analyse discriminante prédictive. Il s'agit d'expliquer et de prédire l'appartenance d'un individu à une classe (groupe) prédéfinie à partir de ses caractéristiques mesurées à l'aide de variables prédictives [12].

- *Filtre de Haar*

La détection du visage dans ce filtre est réalisée par un filtre multi-échelles de Haar. Les propriétés d'un visage sont décrites dans un fichier XML. Elles ne sont pas choisies au hasard et reposent sur un échantillon de quelques centaines d'images tests [13]. Nous reviendrons plus en détails sur cette méthode.

Chapitre I : La reconnaissance faciale

- *Méthode LBP (Local Binary Patterns)*

Les LBP (Modèles binaires locaux) ont également été utilisés pour la détection du visage ou ce dernier est subdivisé en sous-régions carrées de taille égale sur lesquelles sont calculées les caractéristiques LBP. Les vecteurs obtenus sont ensuite concaténés pour obtenir le vecteur de caractéristiques final

I.5.Problématique

Le problème de la reconnaissance de visages peut être formulé comme suit : étant données une ou plusieurs images d'un visage, la tâche est de trouver ou de vérifier l'identité d'une personne par comparaison de son visage à l'ensemble des images de visage stockées dans une base de données. En général, un système de biométrie faciale est constitué de deux modules : un module de segmentation de visages (détection ou localisation de visage), et un module de reconnaissance qui se déroule en trois étapes : normalisation ou prétraitement, extraction de caractéristiques faciales, classification (Figure 1.1).

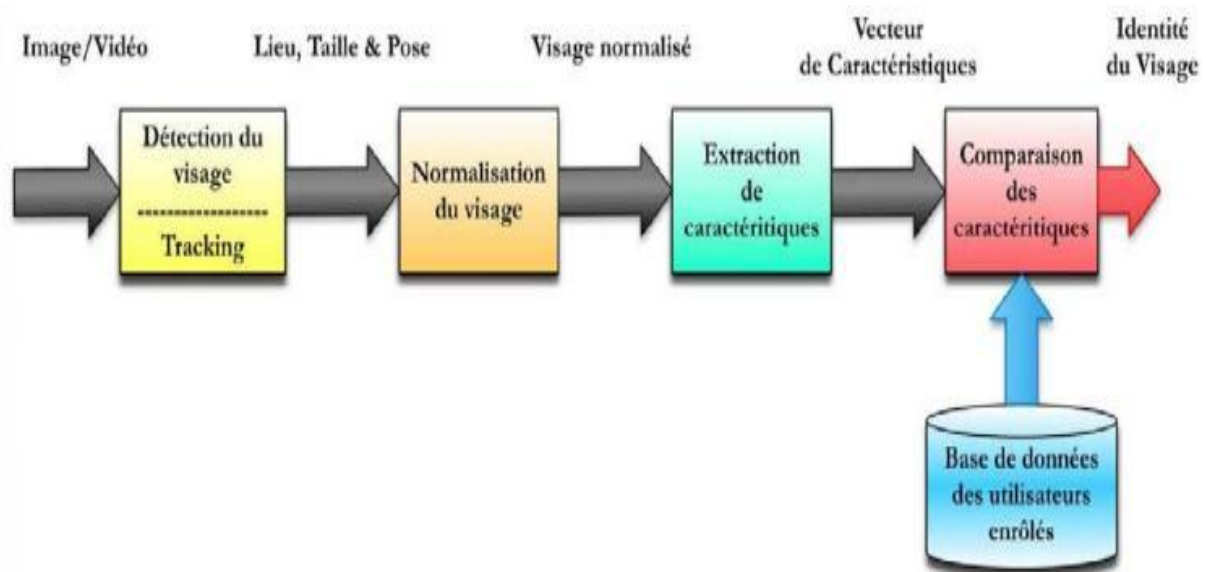


Figure 1.1. Schéma général d'un système de reconnaissance des visages [15]

- **Capture :** image ou vidéo C'est la première étape dans le processus. C'est aussi l'acquisition des informations et leur transfert vers l'unité de traitement. C'est est une étape très importante dans les systèmes de reconnaissance. En effet, avoir des images de bonne qualité en référence améliore les performances de reconnaissance. Il faut réussir à capter l'information pertinente sans bruit [15].

Chapitre I : La reconnaissance faciale

•Détection de visage : La détection de visage dans l'image est un traitement indispensable avant la phase de reconnaissance. En effet, le processus de reconnaissance de visages ne pourra jamais devenir intégralement automatique s'il n'est pas précédé par une étape de détection efficace [14]. Cela peut se faire par détection de la couleur de la peau, la forme de la tête, l'apparence faciale, ou par des méthodes détectant les différentes caractéristiques du visage par des descripteurs locaux (Adaboost). Après la segmentation du visage, on peut filtrer ou améliorer la qualité par des prétraitements qui sont appliqués au visage extrait. On peut effectuer des normalisations géométriques et photométriques. Ces prétraitements sont nécessaires pour éliminer ou limiter les variations de pose, uniformiser l'éclairage, minimiser l'influence de l'illumination, ainsi d'ajuster le visage pour qu'il ait une dimension particulière et qu'il soit horizontal [15].

•Extraction de caractéristiques : Le but est d'extraire les caractéristiques du visage qui peuvent le rendre à la fois différent de celui des autres personnes et robuste aux variations de la personne elle-même. C'est l'information nécessaire pour que le visage d'une personne ne ressemble pas à celui d'une autre personne et en même temps qu'il ressemble à lui-même dans d'autres conditions d'acquisition [15].

•Comparaison : Selon les caractéristiques extraites précédemment, les algorithmes de comparaison diffèrent. On trouve dans la littérature plusieurs approches : calcul de distance, calcul de similarité. D'autres méthodes se basent sur la classification des caractéristiques par un seul classifieur (SVM, classifieur bayésien, etc.) ou par plusieurs classifieurs (Adaboost) [15].

I.6. Avantages et inconvénients de la reconnaissance de visages

Plusieurs facteurs rendent la modalité visage attractive pour une utilisation à grande échelle, elle est acceptable par les personnes vu que c'est une partie apparente du corps, facilement vérifiable par n'importe quel opérateur pour avoir une décision, intrusive en raison de la vérification aisée. De plus, les capteurs d'images sont moins chers sur le marché [16].

En dépit de ses avantages, elle présente des inconvénients qui influent sur la qualité de la reconnaissance. On peut citer les aspects suivants [16]:

- Changement d'illumination.
- Expressions faciales.
- Présence ou absence des composantes structurales, telles que (la barbe, la moustache et les lunettes).
- Les vrais jumeaux qui ont le même indicatif d'ADN.

Chapitre I : La reconnaissance faciale

I.7.Conclusion

Notre travail s'intéresse au problème général de la reconnaissance facial en contexte de vidéosurveillance. Ainsi, dans ce mémoire, nous aurons à concevoir et à implémenter une application permettant la reconnaissance des visages des personnes à partir d'une séquence vidéo, la sortie est une valeur qui représente le score de comparaison des caractéristiques des visages extraits de la base de données avec celle qui existe dans la séquence vidéo ,la conception de cette application sera à l'aide du Raspberry Pi . Dans le chapitre 2, nous allons définir les grandes lignes de notre travail tell que les technique utilisé pour la détection de visage ainsi les différents modules nécessaire pour la réalisation de notre système de reconnaissance facial.

CHAPITRE II :
Etude et réalisation du
 système de reconnaissance
 faciale

Introduction

Dans un monde où la sécurité des individus est devenue un souci majeur, le besoin de se protéger augmente jour après jour. En effet, vu le développement permanent et important de la société dans tous ces aspects, l'utilisation des caractéristiques physiques des individus pour les identifier et la reconnaissance faciale est un excellent moyen de sécurité.

Dans notre chapitre nous allons bien détailler le procédé de détection HAAR CASCADE et ces parties principales. Pour faire la programmation de notre projet nous utilisons le langage Python et une de ces bibliothèques principale le open cv qui permet de faciliter les procédures de détection.

II.1.Spécification de la technique et langage utilisés

L'objectif de notre travail consiste à développer une application qui détecte des visages des personnes à partir d'une séquence vidéo. L'entrée de notre application est une séquence vidéo, il est facile de détecter un visage dans une séquence vidéo en utilisant son détecteur de visage Haar Cascade à partir d'une vidéo en direct, le détecteur de visage examine chaque emplacement d'image et classifie comme visage connu ou non connu à l'aide d'un classifieur utilisant des données stockées dans un fichier XML.

II.2.Techniques de détection utilisées

Plusieurs méthodes de détection humaine pour les caméras thermiques ont dans certaines études. La plupart d'entre eux ont utilisé presque la même procédure de détection: pour la région d'intérêt connue (ROI), d'abord l'extraction des fonctionnalités sera effectuée, puis les caractéristiques obtenues sont envoyées à un classificateur qui a été formé. L'histogramme des caractéristiques orientées des gradients (HOG) est le plus souvent trouvé dans le système de détection humaine, avec les méthodes de classification telles que le vecteur de support machines (SVM), le regroupement et le regroupement flous des réseaux neuronaux convolutions, ou AdaBoost .

La fonction haar-cascade est largement utilisée dans la détection de l'objet à l'aide d'images. La fonction haar-cascade offre un processus d'extraction rapide qui est plus rapide que la fonction HOG et peut représenter des images à basse résolution. Viola et Jones ont développé une méthode de détection d'objets bien connue basée sur des caractéristiques semblables à Haar, combinée à un classificateur en cascade boosté, qui est devenu connu sous le nom de classificateur Haar-Cascade. Dans cette méthode, la fonctionnalité de forme Haar est calculée efficacement en utilisant l'image intégrale et AdaBoost est utilisé pour sélectionner un petit nombre des fonctionnalités les plus importantes pour représenter l'objet.

Cette méthode a été appliquée avec succès dans plusieurs objets systèmes de détection tels que la détection du visage.

II.3.Procédé de détection de HAAR-CASCADE

La méthode 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 dire des caractéristiques haar, l'image intégrale, le boost adaptatif ou l'AdaBoost et la combinaison de classificateur cascade.

II.3.1.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, comme indiqué dans la figure 2.1. Valeur de fonctionnalité (f) de la fonction Haar qui a k rectangles est donnée par l'équation suivante :

$$f = \sum_{i=1}^k W^{(i)} \cdot \mu^{(i)} \quad (1)$$

Avec $\mu^{(i)}$ est le rectangle moyen, qui est l'intensité moyenne des pixels dans le rectangle $i^{\text{ème}}$, $w^{(i)}$ est le poids de chaque rectangle. En général, la valeur de poids est définie sur une certaine entière valeur qui satisfait l'équation:

$$\sum_{i=1}^k W^{(i)} = 0 \quad (2)$$

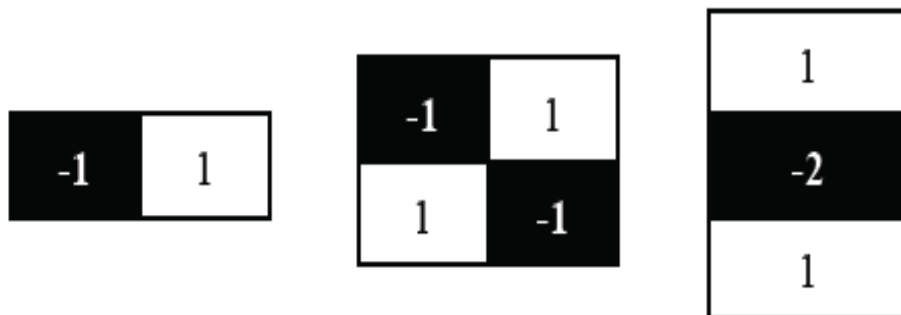


Figure 2.1. Exemples de fonctionnalité similaires à Haar [17]

II.3.2.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 a une valeur qui est accumulée à partir du haut et à la gauche de la position du pixel de l'image d'entrée, comme elle est indiquée dans la Fig2.2

1	2	3
4	5	6
7	8	9

image originale

1	3	6
5	12	21
12	27	45

image intégrale

Figure 2.2. Exemple de calcul d'image intégrale [17]

II.3.3. AdaBoost

Le processus de détection n'est pas fait avec un seul type de fonctionnalité. Au lieu de cela, différents types de fonctionnalités 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 formation 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.

II.3.4. 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 un potentiel pour contenir les caractéristiques de l'objet, le processus est poursuivi à l'aide de classificateurs ultérieurs. Si la 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 [17]. 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, comme le montre la fig2.3.

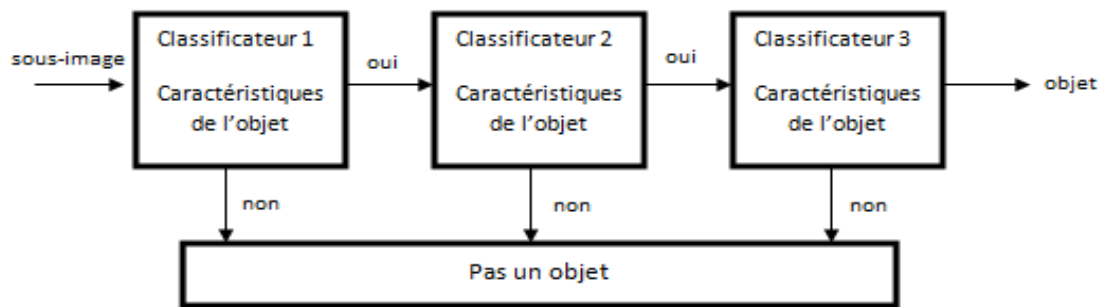


Figure 2.3 .Cascades Classifiées [17]

II.4.Types de fichier utilisé

II.4.1.Fichier XML

Le XML, un parmi les langages de description les plus connus, ils permettent de décrire et structurer un ensemble de données selon un jeu de règles et des contraintes définies .le XML permet de décrire des données à l'aide de balises et de règles que l'on peut personnaliser [18].

Le XML est utile puisqu'il facilite les échanges de données entre les machines. À cela s'ajoute un autre objectif important : décrire les données de manière aussi bien compréhensible par les hommes qui écrivent les documents XML que par les machines qui les exploitent, il est aussi compatible avec le web afin que les échanges de données puissent se faire facilement à travers le réseau Internet [18], enfin il est standardiser, simple, mais surtout extensible et configurable afin que n'importe quel type de donnée puisse être décrit [19].

II.4.2.Fichier CSV

Le CSV est un format de texte simple qui est utilisé dans de nombreux contextes. Lorsque de grandes quantités de données doivent être fusionnées sans être directement connectées les unes aux autres.

L'extension de ce type de fichiers est .csv, et ils peuvent être utilisés entre différents outils informatiques et bases de données, lorsqu'on souhaite déployer le contenu d'une base de données sur une feuille de calcul [20].

II.4.3.Format de gris

Dans notre projet nous avons utilisé le niveau de gris (image en gris). La raison pour différencier ces images de toute autre sorte d'image couleur est que l'informations fournies pour chaque pixel est réduite [21].Une couleur «grise» est une couleur dans laquelle les composants rouge, vert et bleu ont tous une intensité égale dans l'espace RVB, et il est donc nécessaire de spécifier qu'une seule valeur d'intensité pour chaque pixel, par opposition aux trois intensités nécessaires pour spécifier chaque pixel dans une image en couleur.

Souvent, l'intensité des niveaux de gris est stockée sous la forme d'un entier de 8 bits donnant 256 différentes nuances de gris possibles du noir au blanc. Si les niveaux sont également

espacés, la différence entre les niveaux de gris successifs est nettement meilleure que le pouvoir de résolution des niveaux de gris de l'œil humain.

Les images en niveaux de gris sont très courantes, en partie parce qu'une grande partie du matériel d'affichage et de capture d'image d'aujourd'hui ne peut prendre en charge que des images 8 bits. De plus, les images en niveaux de gris sont entièrement suffisantes pour de nombreuses tâches et il n'est donc pas nécessaire d'utiliser des images couleur plus compliquées et plus difficiles à traiter.

II.5. Environnement de développement

II.5.1. Environnement hardware

II.5.1.1. Un ordinateur

- Processeur : i5-7200u
- Mémoire installée (RAM) : 4,00 Go
- Type du système : système d'exploitation 32 bits, processeur x64

II.5.1.2. Un Raspberry Pi

II.5.1.2.1. Présentation du Raspberry pi

Le Raspberry Pi est un nano-ordinateur mono carte à processeur ARM conçu par des professeurs du département informatique de l'université de Cambridge dans le cadre de la fondation Raspberry Pi.

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 fonctionne également avec le système d'exploitation Microsoft Windows.

Il est fourni nu, c'est-à-dire 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 à partir de quelques dizaines d'euros seulement.



Figure 2.4. Le Raspberry pi

Son prix de vente était estimé à 25 \$ américains, soit 1 776.23 DA, début mai 2011. Les premiers exemplaires ont été mis en vente le 29 février 2012 pour environ 2 496.14 DA. En septembre 2016, plus de dix millions de Raspberry Pi ont été vendus. Par la suite différents modèles Raspberry Pi sont apparues et ils sont présentés comme suit :

II.5.1.2.2. Les modèles de Raspberry Pi

Le modèle A

Il y a deux types A et 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 Sorties vidéo : Composite et HDMI
1 Sortie audio stéréo Jack 3,5 mm
1 Port USB 2.0 ;



Modèle A

Modèle A+ :

Le Raspberry Pi A+ est lancé en novembre 2014, Plus petit que le Raspberry Pi A, Lecteur de carte microSD au lieu et place du lecteur SD et consommation électronique moindre.
GPIO 40 broches.
Nouveau chipset audio



Modèle A+

Le modèle B

Modèle 1 B :

2 révisions de modèle B en lancés, la rev1 en Avril 2012 puis la rev2 en juin 2012.

2 ports USB 2.0 au lieu de l'unique port du modèle A, mais sur un seul bus, *via* le composant SMSC LAN9512, 1 port réseau Fast Ethernet (10/100 Mbit/s) *via* le même composant SMSC

La RAM passe à 512 Mo (au lieu de 256 Mo sur les modèles précédents)



Modèle 1 B

Modèle 1 B+ :

Ce modèle est annoncé en juillet 2014

Différences par rapport au modèle initial :

- GPIO 40 broches
- 4 ports USB 2.0 et meilleur comportement en cas de surcharge
- micro SD
- réduction de consommation de 3,5 W à 3 W
- meilleur circuit audio
- suppression de la prise RCA au profit d'une prise mini-jack 4 points, comprenant une sortie sonore et vidéo



Modèle 1 B+

Chapitre II : Etude et réalisation du système de reconnaissance faciale

Modèle 2 B (Raspberry Pi 2) :

Le 2 février 2015, la fondation Raspberry Pi annonce la sortie du Raspberry Pi 2, plus puissant, il est équipé d'un processeur Broadcom BCM2836, quatre cœurs ARMv7 à 900 MHz, accompagné de 1 Go de RAM.



Raspberry Pi 2 Modèle B

Modèle 3 B (Raspberry Pi 3) :

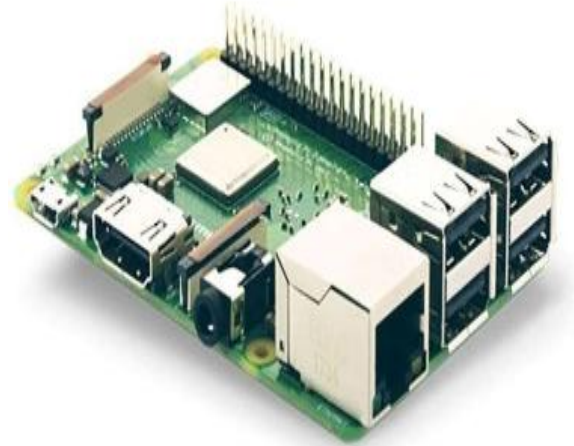
Le 29 février 2016, pour le quatrième anniversaire de la commercialisation du premier modèle, la fondation Raspberry Pi annonce la sortie du Raspberry Pi 3. Il est à présent équipé nativement du Wifi b/g/n et du Bluetooth 4.1. Processeur Broadcom BCM2837 64 bit Quatre cœurs ARM Cortex-A53 à 1,2 GHz, Puce Wifi 802.11n et Bluetooth 4.1 intégrée



Raspberry Pi 3 Modèle B

Modèle 3 B+ (Raspberry Pi 3+) :

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 64 bit à quatre cœurs ARM Cortex-A53 cadencé à 1,4 GHz 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



Raspberry Pi 3 Modèle B+

Modèle 4 B (Raspberry Pi 4) :

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 permettent 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.



Raspberry Pi 4 Modèle B

Modèle 4 B (Raspberry Pi 4) 8GB :

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.



Raspberry Pi 4 Modèle B 8GB

Le modèle Zero

Modèle Zero :

Le 26 novembre 2015, la fondation Raspberry Pi annonce la sortie du Raspberry Pi Zero. Il reprend les spécifications du modèle A/B avec un processeur cadencé à 1 GHz 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



Le Modèle Zero

Modèle Zero W:

Lancé le 28 février 2017 Le Raspberry Pi Zéro W étend la famille Pi Zéro avec une connectivité Bluetooth et LAN sans fil supplémentaire.

CPU : 1 GHz

Mémoire vive : 512 Mb



Le Modèle Zero W

Modèle Zero WH:

Lancé Le 15 janvier 2018, Meme carte que Raspberry Pi Zero W mais avec un connecteur 40 pins GPIO déjà soudé. SPI + I2CCamera + MicroSD



Le Modèle Zero WH

II.5.1.2.3. Le Raspberry Pi 3 modèle B

Pour notre projet, nous avons choisi le Raspberry Pi 3 modèle B pour beaucoup de raisons notamment sa puissance et sa rapidité par rapport au autre Type de Raspberry.

Raspberry Pi 3 Type B est la carte mère idéale pour un système d'architecture ARM. équipée d'un puissant processeur Quad-Core ARM Cortex-A53 1.2 GHz (Broadcom BCM2837), de 1024 Mo de RAM et d'un GPU Dual-Core *VideoCore IV* capable de décoder les flux vidéo HD 1080p. La carte mère Raspberry Pi 3 modèle B est une solution performante et abordable pour tous types de systèmes compacts ou embarqués (MediaCenter, domotique, affichage dynamique, robotique...etc) [22].



Figure 2.5.Le Raspberry Pi 3 Modèle B

II.5.1.2.4 Spécification technique du microcontrôleur Raspberry Pi 3

- Cadencement : 1,2 GHz
- Puce (SoC) : Broadcom BCM2837
- Processeur : ARM Cortex-A53 64 bits quatre cœurs
- Processeur graphique : Broadcom VideoCore IV double coeur (OpenGL ES 2.0, H.264 Full HD à 30 ips)
- Mémoire (SDRAM) : 1GB LPDDR2
- Nombre de ports USB 2.0 : 4
- Port extension : GPIO 40 pins
- Sorties vidéos : HDMI et RCA, plus 1 connecteur de caméra CSI
- Sorties audio : Stéréo Jack 3,5mm ou HDMI
- Sauvegarde des données : Carte MicroSD
- Connexion réseau : 10/100 Ethernet, WiFi 802.11n et Bluetooth 4.1 (BLE - Low Energy)
- Périphériques : 17 × GPIO
- Alimentation : 5v 2.5A via micro-USB
- Dimensions : 85,60 mm × 56 mm × 21 mm
- Poids : 45 g

II.5.1.2 .5. Les composants du Raspberry Pi 3 modèle B

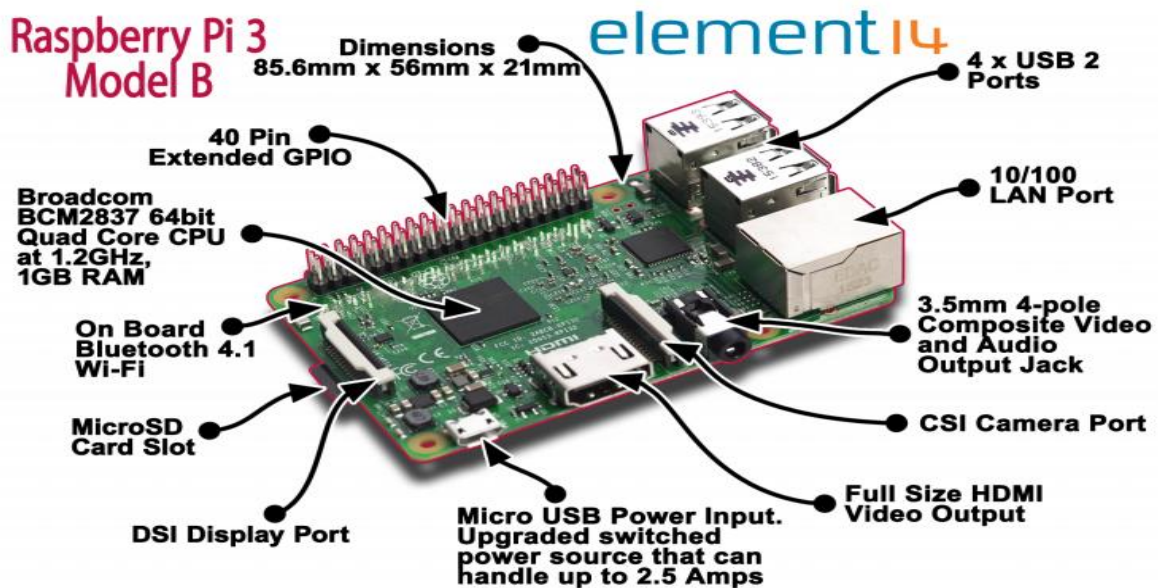


Figure 2.6. Les composants standard d'un Raspberry Pi 3 Modèle B

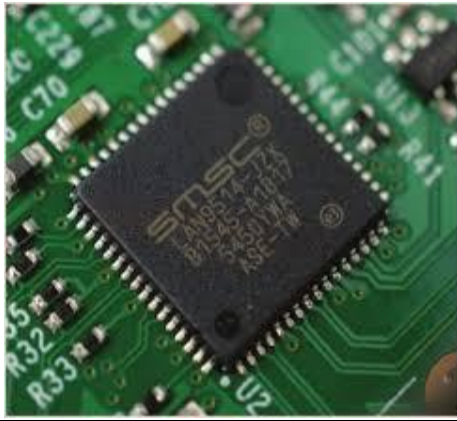
Comme tout ordinateur, le Pi est composé de différents composants, chacun ayant un rôle à jouer pour le faire fonctionner:

✓ le système on-chip (SoC) :



Le premier composant, et sans doute le plus important, se trouve juste au-dessus du point central sur le côté supérieur de la carte (figure II -15), il contient la majeure partie du système du Raspberry Pi. Cela comprend l'unité centrale de traitement (CPU), communément considérée comme le «cerveau» d'un ordinateur, et l'unité de traitement graphique (GPU), qui gère le côté visuel des choses.

✓ **Puce USB:**



Le Raspberry Pi 3 partage la même puce SMSC LAN9514 que son prédécesseur, le Raspberry Pi 2, ajoutant une connectivité Ethernet 10/100 et quatre canaux USB à la carte. Comme précédemment,

La puce SMSC se connecte au SoC via un seul canal USB, agissant comme un adaptateur USB vers Ethernet et concentrateur USB.

✓ **Antenne:**



Il n'est pas nécessaire de connecter une antenne externe au Raspberry Pi 3. Ses radios sont connectées à cette antenne à puce soudée directement sur la carte, afin de conserver la taille de l'appareil au minimum. Malgré sa petite taille, cette antenne devrait être plus que capable de capter les signaux LAN sans fil et Bluetooth - même à travers les murs.

✓ **Port USB (Universal Serial Bus):**



Le Raspberry Pi possède une gamme de ports, en commençant par quatre ports Universal Serial Bus (USB) au milieu et à droite du bord inférieur. Ces ports vous permettent de connecter n'importe quel périphérique compatible USB, des claviers et souris aux appareils photo numériques et lecteurs flash, au Pi. Sur le plan technique,

✓ Port Ethernet:



À gauche des ports USB se trouve un port Ethernet, également appelé port réseau. Vous pouvez utiliser ce port pour connecter le Raspberry Pi à un réseau informatique filaire à l'aide d'un câble avec ce que l'on appelle un connecteur RJ45 à son extrémité. Si vous regardez attentivement le port Ethernet, vous verrez deux diodes électroluminescentes (LED) en bas; ce sont des voyants d'état qui vous indiquent que la connexion fonctionne.

✓ HDMI (High Definition Multimedia Interface):



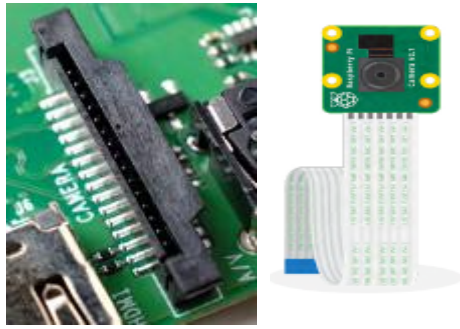
Le port HDMI (High-Definition Multimedia Interface), est le même type de connecteur que vous trouverez sur une console de jeux, top box et TV. La partie multimédia de son nom vous indique qu'il transporte des signaux audio et vidéo, tandis que la haute définition vous indique que vous pouvez vous attendre à une excellente qualité. Vous l'utiliserez pour connecter le Raspberry Pi à votre périphérique d'affichage, qu'il s'agisse d'un écran d'ordinateur, d'un téléviseur ou d'un projecteur.

✓ Prise audio (jack):



Une prise audio-visuelle (AV) de 3,5 mm. Ceci est également connu comme la prise casque, et il peut être utilisé à cette fin, bien que vous obteniez un meilleur son en le connectant à des haut-parleurs amplifiés plutôt qu'à des écouteurs. Il a cependant une fonctionnalité cachée supplémentaire: en plus de l'audio, la prise AV 3,5 mm transporte un signal vidéo qui peut être connecté à des téléviseurs, des projecteurs et d'autres écrans qui prennent en charge un signal vidéo composite à l'aide d'un câble spécial appelé -Adaptateur anneau-manchon (TRRS).

✓ Entrée caméra CSI:



Un connecteur d'aspect étrange avec un volet en plastique qui peut être tiré vers le haut; il s'agit du connecteur de la caméra, également appelé interface série de la caméra (CSI). Cela vous permet d'utiliser le module de caméra Raspberry Pi spécialement conçu.

✓ Micro USB:



Au-dessus du port HDMI se trouve un port d'alimentation micro USB, que vous utiliserez pour connecter le Raspberry Pi à une source d'alimentation. Le port micro USB est courant sur les smartphones, tablettes et autres appareils portables. Vous pouvez donc utiliser un chargeur mobile standard pour alimenter le Pi, mais pour de meilleurs résultats, vous devez utiliser l'alimentation USB officielle du Raspberry Pi.

✓ Affichage DSI:



Touch Display

Sur le bord supérieur de la carte se trouve un autre connecteur d'aspect étrange, qui à première vue semble être identique au connecteur de la caméra. Cependant, c'est exactement le contraire: un connecteur d'affichage, ou Display Serial Interface (DSI), conçu pour être utilisé avec le Raspberry Pi Touch Display.

✓ Connecteur de carte microSD:



Le Raspberry a besoin d'une mémoire externe supplémentaire pour fonctionner. Ce connecteur permet de connecter la mémoire externe. Vous pouvez utiliser votre propre carte SD dans le Raspberry Pi, mais elle devra être préparée avec une image disque d'un système d'exploitation.

✓ GPIO Pins:



Sur le bord droit de la carte, vous trouverez 40 broches métalliques, divisées en deux rangées de 20 broches.

Il s'agit de l'en-tête GPIO (entrée / sortie à usage général), une fonction du Raspberry Pi utilisée pour parler à du matériel supplémentaire, des LED et des boutons jusqu'aux capteurs de température, joysticks et moniteurs de fréquence du pouls.

- N'importe laquelle des broches GPIO peut être désignée (dans le logiciel) comme une goupille d'entrée ou de sortie et utilisée pour un large éventail de fins.
- la numérotation des broches GPIO n'est pas dans l'ordre numérique; Les broches GPIO 0 et 1 sont présentes sur le tableau (broches physiques 27 et 28) mais sont réservées à une utilisation avancée

Raspberry Pi 3 GPIO Header			
Pin#	NAME		NAME Pin#
01	3.3v DC Power		DC Power 5v 02
03	GPIO02 (SDA1 , I2C)		DC Power 5v 04
05	GPIO03 (SCL1 , I2C)		Ground 06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14 08
09	Ground		(RXD0) GPIO15 10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18 12
13	GPIO27 (GPIO_GEN2)		Ground 14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23 16
17	3.3v DC Power		(GPIO_GEN5) GPIO24 18
19	GPIO10 (SPI_MOSI)		Ground 20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25 22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08 24
25	Ground		(SPI_CE1_N) GPIO07 26
27	ID_SD (I2C ID EEPROM)		(I2C ID EEPROM) ID_SC 28
29	GPIO05		Ground 30
31	GPIO06		GPIO12 32
33	GPIO13		Ground 34
35	GPIO19		GPIO16 36
37	GPIO26		GPIO20 38
39	Ground		GPIO21 40

Figure 2.7. GPIO

II.5.1.2.6. Connexion des composants d'un Raspberry Pi

Les éléments essentiels (accessoires) dont nous aurons besoin pour notre Raspberry Pi et comment les connecter tous pour le configurer et le faire fonctionner.

- **Une alimentation (USB power supply)**

Une alimentation de courant de sortie maximum 2,5 A et une tension de 5 V avec un connecteur micro USB. L'alimentation officielle du Raspberry Pi est le choix recommandé, car elle peut faire face aux demandes d'alimentation à commutation rapide du Raspberry Pi.



Figure 2.8. Alimentation 5V--2.5A

- **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.

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 (classe 10) pouvant atteindre au maximum 8 Go pour permettre le démarrage, bien qu'une carte de 16 Go ou bien 32 Go offre plus d'espace pour se développer.

Figure 2.9. Carte Micro SD

II.5.1.3. Un Ecran : TFT LCD 5"

Un écran TFT c'est un écran hautes performances grâce au module LCD TFT 5 pouces de Displaytech. Cet écran offre une résolution RVB 800 x 480 et prend en charge le mode d'entrée RVB parallèle 24 bits. L'écran LCD utilise deux circuits d'attaque pour commander l'écran : un circuit d'attaque source avec TCON et un circuit d'attaque de grille. Les deux circuits d'attaque permettent différentes tensions d'alimentation et fonctionnalités de commande de broche, c'est pourquoi ce module d'affichage est idéal pour une variété de dispositifs numériques tels que des systèmes vidéo. L'écran LCD TFT 5 pouces est disponible avec un panneau tactile

avec technologie tactile résistive (pression à un doigt ou à stylet) ou capacitive (cinq doigts, multi gestes) [23].



Figure 2.10. Un écran TFT

II.5.1.4. Une Camera Web

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



Figure 2.11. Caméra web

II.5.1.5. Un relais

Un relais est un appareil dans lequel un phénomène électrique (courant ou tension) contrôle la commutation On / Off d'un élément mécanique ou d'un élément électronique. C'est en quelque sorte un interrupteur que l'on peut actionner à distance, et où la fonction de coupure est dissociée de la fonction de commande. La tension et le courant de commande (partie "Commande"), ainsi que le pouvoir de commutation (partie "Puissance") dépendent du relais, il faut choisir ces paramètres en fonction de l'application désirée. Dans notre projet nous avons utilisées un relais électromécanique.

Un relais électromécanique est doté d'un bobinage en guise d'organe de commande. La tension appliquée à ce bobinage va créer un courant, ce courant produisant un champ électromagnétique à l'extrémité de la bobine (il ne s'agit ni plus ni moins que d'un électro-aimant). Ce champ magnétique va être capable de faire déplacer un élément mécanique métallique monté sur un axe mobile, qui déplacera alors des contacts mécaniques. [24]

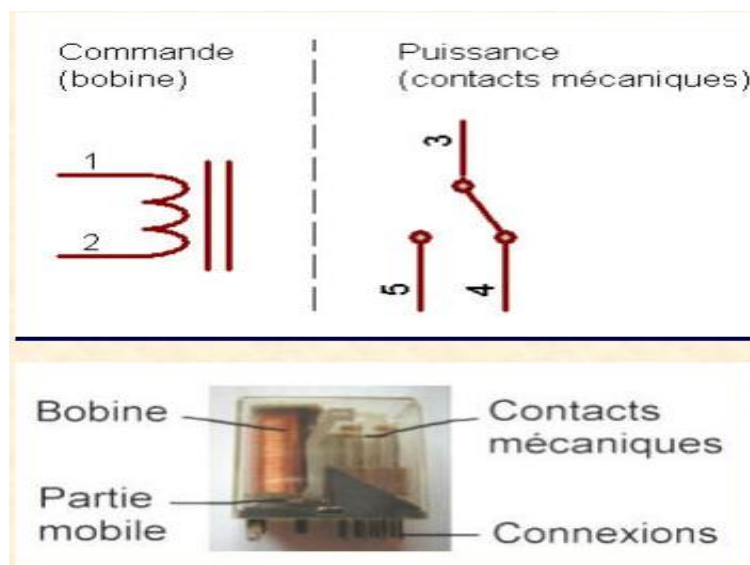


Figure 2.12. Un relais

II.5.1.6. Une serrure solénoïde

Les solénoïdes sont des électro-aimants : Ils sont composés d'une grosse bobine de cuivre avec une armature (un noyau en métal) en leur centre. Quand la bobine est alimentée, le noyau est attiré au centre de la bobine. Cela permet au solénoïde de se déplacer.

Cette solénoïde est particulièrement robuste, son noyau en métal dispose d'une extrémité avec une coupe inclinée et d'un bon support de montage. Cela en fait une serrure électrique, idéal pour un placard ou une porte. De base, la serrure est active sans alimentation, quand une tension de 9VDC à 12VDC est appliquée, la serrure s'actionne et la porte peut s'ouvrir.

Il est possible de tourner la gâche de 90°, 180° ou 270° pour l'adapter à votre application en ouvrant le boîtier avec un tournevis cruciforme.

Attention, pour piloter un solénoïde, vous devez disposer d'une bonne alimentation pouvant fournir 500mA, il ne faut pas piloter cette serrure électronique directement avec un microcontrôleur comme une carte Raspberry pi 3 [25].



Figure 2.13. Une serrure solénoïde

II.5.2. Environnement software

II.5.2.1. Raspbian

Après savoir quel Raspberry Pi choisir et les différents accessoires importants pour son fonctionnement et son utilisation, la chose de plus concret est le choix du système d'exploitation.

Les distributions les plus utiles et les plus populaires sont présentées comme suite :

- Les distributions sous 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 Raspbian qui présente une distribution polyvalente optimisée spécialement pour Raspberry Pi, et qui permettra la familiarisation très facilement avec le matériel. Il existe plusieurs versions de Raspbian : Wheezy (basée sur Debian 7) Jessie (basée sur Debian 8) Stretch (basée sur Debian 9) Dans ce travail on va utiliser la version buster full.

II.5.2.2. Langage Python

Python est un langage orienté objet qui a 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 suivent une considération dans la communauté de programmation [26]. 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 pas 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 davantage 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 [27].

Python peut être développée en un temps beaucoup plus court que d'autres programmes tels que `c ++`

Python est un logiciel source de stylo, ce qui signifie que sans identifiant il est inclus dans la plupart des distributions LINUX.

II.5.3.Bibliothèques utilisées

II.5.3.1.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.

II.5.3.2.Tkinter

Tkinter est la bibliothèque d'interface graphique standard pour Python [28]. Python, lorsqu'il est combiné à Tkinter, fournit un moyen rapide et facile pour créer des applications graphiques [29]. Tkinter fournit une puissante interface orientée objet simple et conviviale. La création d'une application graphique à l'aide de Tkinter est une tâche assez facile. Tout ce que vous avez à faire est de suivre les étapes suivantes [30]:

1. Importez le module Tkinter.
2. Créez la fenêtre principale de l'application graphique.
3. Ajoutez un ou plusieurs des widgets graphiques.
4. Entrez la boucle d'évènements principale pour agir contre chaque évènement déclenché par l'utilisateur.

II.5.3.3.OpenCv

La bibliothèque OpenCv sera utilisée dans notre projet pour suivre des objets en mouvement en utilisant une webcam.

II.6 Présentation de la bibliothèque Opencv

II.6.1.Historique

Officiellement lancé en 1999, le projet OpenCV est développé initialement par Intel pour optimiser les applications gourmandes en temps processeur [31]. Cela faisait partie d'une série de projets tels que l'affichage d'un mur en 3 dimensions. Cette bibliothèque est distribuée sous licence BSD.

Les principaux acteurs du projet sont l'équipe de développement de bibliothèque de chez Intel ainsi qu'un certain nombre d'experts dans l'optimisation de chez Intel Russie.

Les objectifs de base du projet étaient :

- Faire des recherches sur la vision par ordinateur en vue de fournir un logiciel libre et optimisé.
- Établir une infrastructure commune s'appuyant sur les développeurs pour obtenir un code plus lisible et transférable.
- Continuer à développer en rendant le code portable et permettre des performances Optimisées gratuites avec une licence qui est libre de toutes contraintes commerciales.

La première version alpha d'OpenCV fut présentée lors de la conférence IEEE sur la vision par ordinateur et la reconnaissance de formes en 2000. Après cela, cinq versions bêta ont été publiées entre 2001 et 2005 et la première version 1.0 a été publiée en 2006[31].

Au milieu de l'année 2008, OpenCV obtient l'appui de la société de robotique Willow Garage et la bibliothèque est encore développée à ce jour. Une version 1.1 est sortie en Octobre 2008 et un livre écrit par deux auteurs d'OpenCV, publié par O'Reilly Media est sorti sur le marché [31] ce même mois.

La deuxième version majeure d'OpenCV manufacturé en octobre 2009. Il s'agit d'OpenCV 2 incluant des changements majeurs au niveau du langage C++ servant à faciliter le développement de nouvelles fonctions et améliorant les performances.

II.6.2.Qu'est-ce que OpenCv

OpenCV (Open Source Computer Vision) est une bibliothèque libre de vision par Ordinateur. Cette bibliothèque est écrite en C et C++ et peut être utilisée sous Linux, Windows et Mac OS X. Des interfaces ont été développées pour Python, Ruby, Matlab et autre langage [32].

Open CV est orienté vers des applications en temps réel. Un des buts d'OpenCV est d'aider les gens à construire rapidement des applications sophistiquées de vision à l'aide d'infrastructure simple de vision par ordinateur. La bibliothèque d'OpenCV contient plus de 500 fonctions.

II.6.3.Fonctionnalités

La bibliothèque OpenCV met à disposition de nombreuses fonctionnalités très diversifiées permettant de créer des programmes partant des données brutes pour aller jusqu'à la création d'interfaces graphiques basiques [33]. Elle propose la plupart des opérations classiques en traitement bas niveau des images et des vidéos.

II.6.3.1. Traitement d'images

Elle propose la plupart des opérations classiques en traitement bas niveau des images :

- Lecture, écriture et affichage d'une image.
- Calcul de l'histogramme des niveaux de gris ou d'histogrammes couleurs [34].
- Lissage, filtrage.
- Seuillage d'image (méthode d'Otsu, seuillage adaptatif).
- Segmentation (composantes connexes, GrabCut).
- Morphologie mathématique [35].

II.6.3.2. Traitement vidéo

Cette bibliothèque s'est imposée comme un standard dans le domaine de la recherche parce qu'elle propose un nombre important d'outils issus de l'état de l'art en vision des ordinateurs tels que :

- Lecture, écriture et affichage d'une vidéo (depuis un fichier ou une caméra).
- Détection de droites, de segment et de cercles par Transformée de Hough.
- Détection de visages par la méthode de Viola et Jones.
- Cascade de classifieurs boostés.
- Détection de mouvement, historique du mouvement.
- Poursuite d'objets par mean-shift ou Camshift.
- Détection de points d'intérêts.
- Estimation de flux optique (Méthode de Lucas–Kanade).
- Triangulation de Delaunay.
- Diagramme de Voronoi.
- Enveloppe convexe.
- Ajustement d'une ellipse à un ensemble de points par la méthode des moindres carrés. [35]

II.7. Conclusion

Dans ce chapitre nous avons vu les parties principales du procédé de détection HAAR-CASCADE, nous avons présenté la bibliothèque OpenCv qui réduit le temps de programmation de la détection des visages ainsi les différents composants utilisés dans ce projet de conception.

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

CHAPITRE III

Conception et analyse

Chapitre III : Conception et analyse

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 visages connus et stockés dans une base de données csv, lequel se rapproche le plus d'un visage passé en paramètre. Il devra également indiquer à quel point il s'en rapproche, afin de pouvoir valider ou infirmer l'identification. Il utilisera la technique dite les cascades de Haar, l'une des premières et des plus 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.

III.1. Matériel de conception

Afin de mener à bien ce projet, nous avons utilisé un ensemble de matériels dont les principales caractéristiques sont les suivantes :

- ✓ Un Raspberry pi 3 Model B
- ✓ Une alimentation (USB power supply) de sortie 1A et une tension de 5V
- ✓ Une carte micro SD « classe 10 » 16 Go
- ✓ Un clavier et une souris
- ✓ Un câble HDMI si le moniteur à une entrée HDMI
- ✓ Un adaptateur HDMI-VGA si le moniteur à une entrée VGA
- ✓ Un câble Ethernet en cas ou le choix d'utilisé SSH ou bien VNC
- ✓ Un adaptateur micro SD
- ✓ Une caméra web
- ✓ Un Ecran : TFT LCD 5''
- ✓ Un relais
- ✓ Une serrure électrique

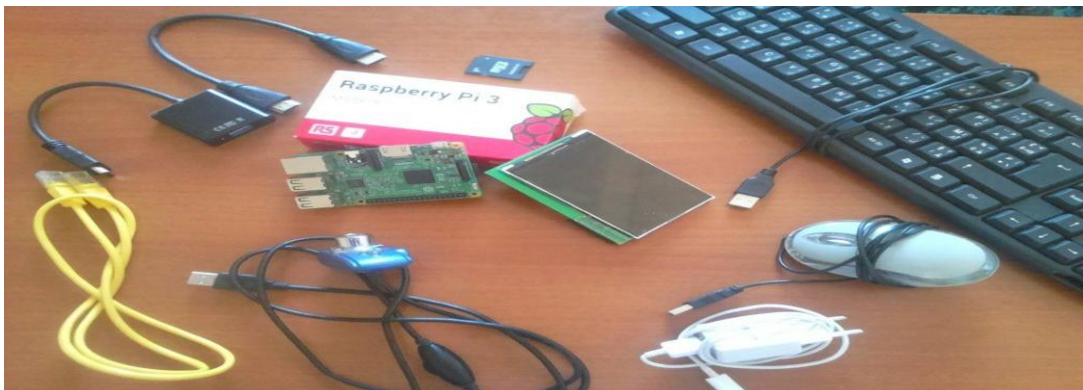


Figure 3.1. Matériels de conception

III.2. Préparation du système d'exploitation sur micro SD

III.2.1. 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, musiques, 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.

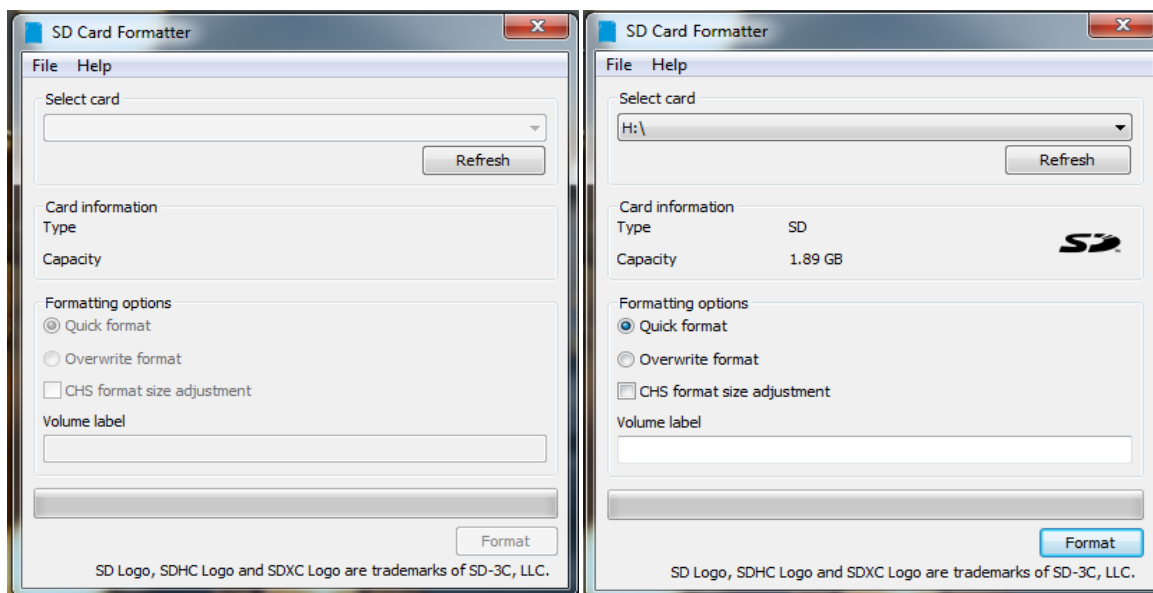


Figure 3.2. Formatage de la carte microSD

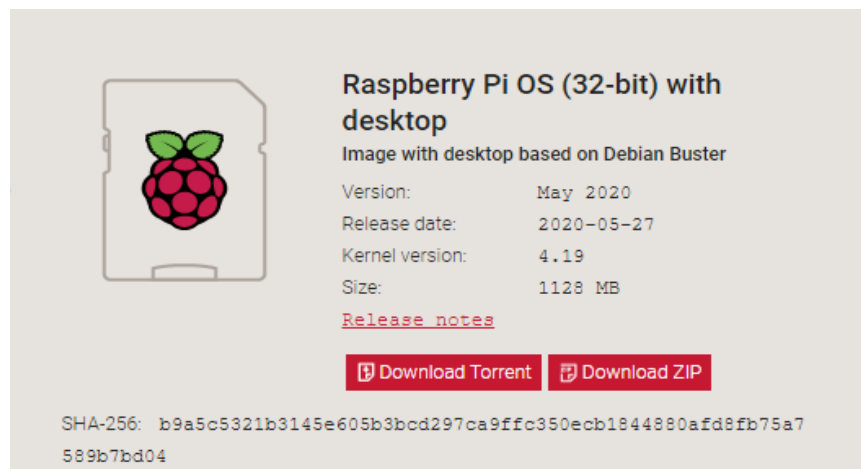
III.2.2. Installer Raspbian sur la carte SD

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/raspberry-pi-os/>
Et télécharger Raspberry Pi OS (32bits) with desktop

Chapitre III : Conception et analyse



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

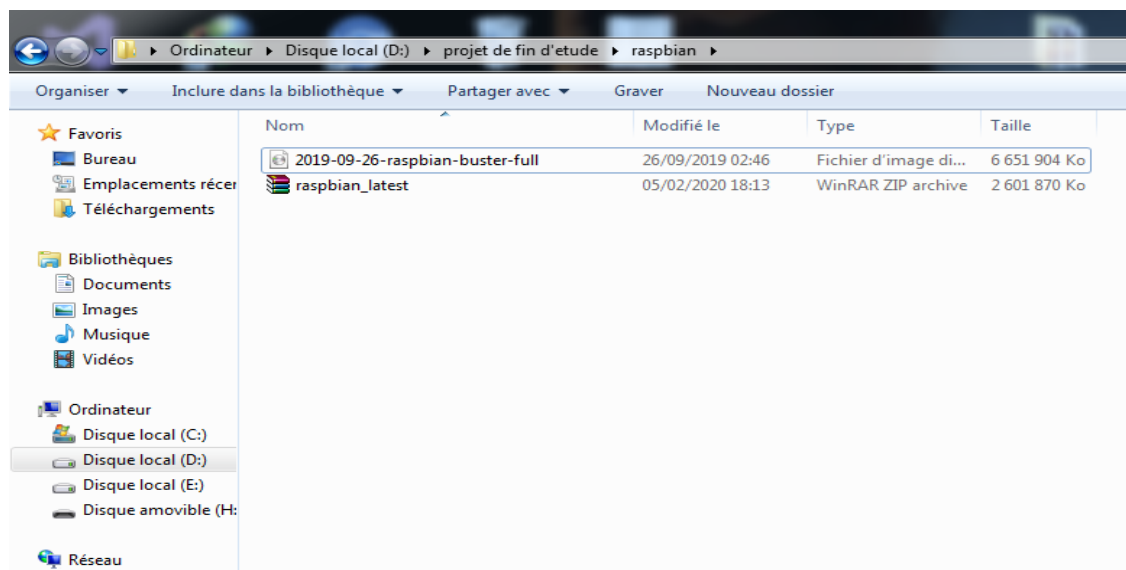


Figure 3.3. Décompression du fichier Zip

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

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

Chapitre III : Conception et analyse

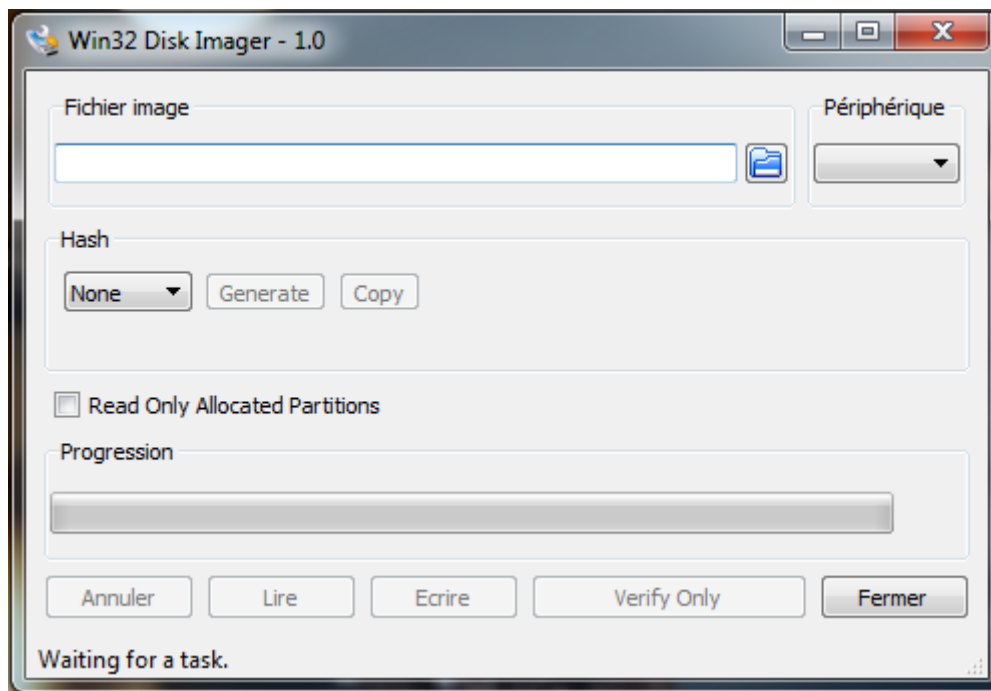


Figure 3.4. Le logiciel Win32DiskImage

Pour copier l'image sur la carte SD, c'est très simple : nous avons sélectionné l'image de Raspbian, et choisis la lettre de notre lecteur de carte SD, puis cliquons 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.

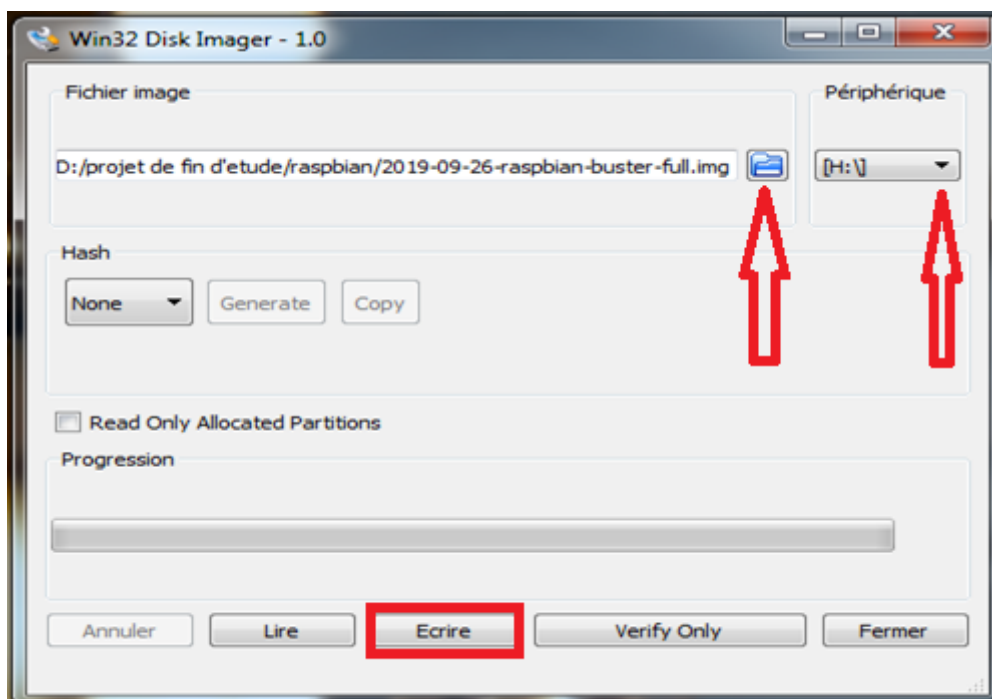


Figure 3.5. Copier le fichier image de raspbian sur la carte SD

III.2.3 Premier démarrage du système

Après le branchement de tout l'équipement, notre Raspberry Pi 3 est allumé

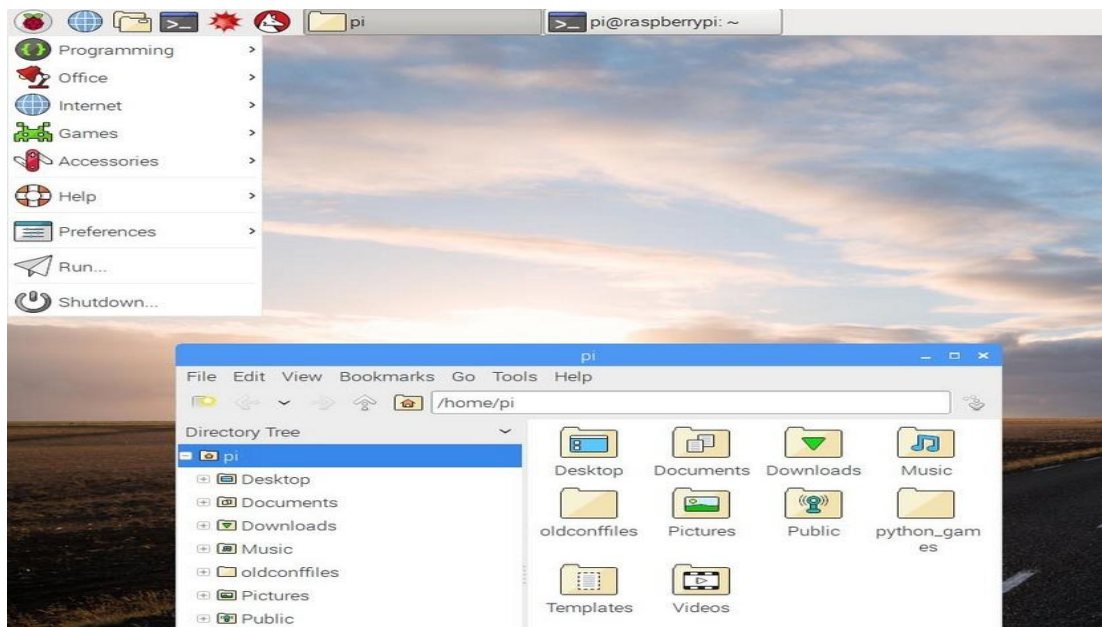


Figure 3.6. bureau du Raspberry Pi 3

III.3. Première configuration

III.3.1. Les premiers réglages : une fois que notre Raspberry est allumé et a fini de démarrer, il faut faire quelques réglages et configuration « de base » avant le commencer à l'utiliser vraiment.

III.3.2. Le clavier en français

- nous ouvrons un terminal et Saisir la commande `sudo nano /etc/default/keyboard` et valider par Entrée :

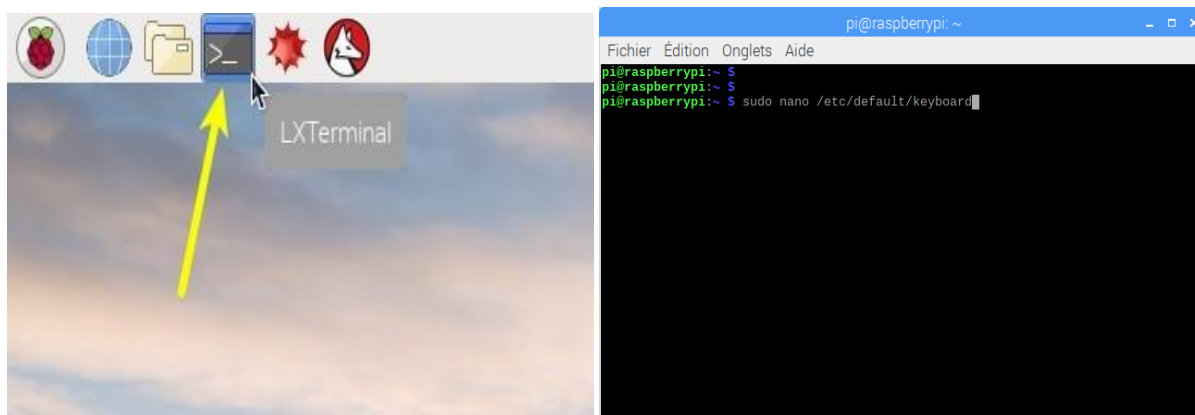
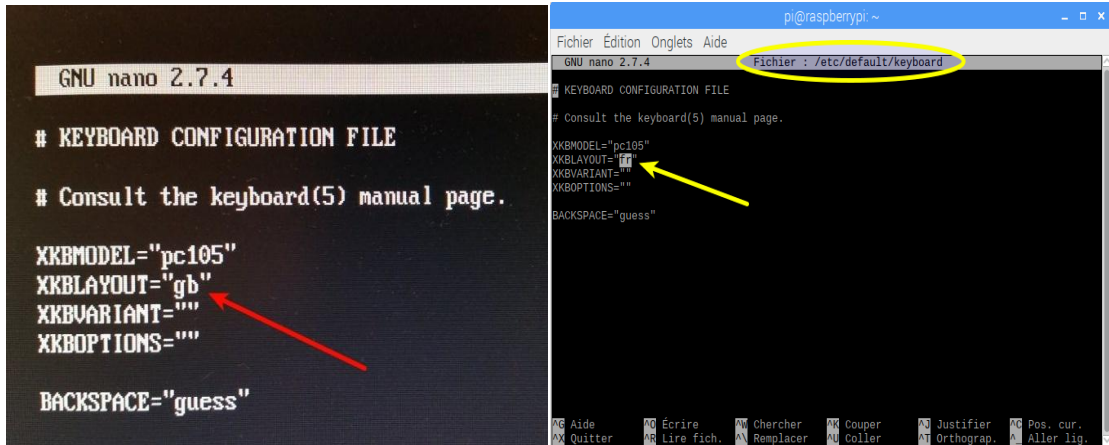


Figure 3.7. Le terminal

Chapitre III : Conception et analyse

- Le fichier s'ouvre alors *keyboard* dans l'éditeur *nano*, directement dans le terminal. Repérer la 6e ligne où on peut lire **XKBLAYOUT="gb"**



```
GNU nano 2.7.4
# KEYBOARD CONFIGURATION FILE
# Consult the keyboard(5) manual page.
XKBMODEL="pc105"
XKBLAYOUT="gb"
XKBVARIANT=""
XKBOPTIONS=""
BACKSPACE="guess"
Fichier : /etc/default/keyboard
```

Figure 3.8. Fichier *keyboard* en « gb »

Déplacement du curseur – avec les flèches du clavier, la souris est inopérante dans cette fenêtre pour aller modifier la valeur **gb** et la remplacer par **fr**, On doit lire maintenant **XKBLAYOUT="fr"**

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

On se retrouve dans la fenêtre du terminal, comme au début.



```
GNU nano 2.7.4
# KEYBOARD CONFIGURATION FILE
# Consult the keyboard(5) manual page.
XKBMODEL="pc105"
XKBLAYOUT="fr"
XKBVARIANT=""
XKBOPTIONS=""
BACKSPACE="guess"
Appuyer sur les touches Ctrl + X pour quitter.
Appuyer sur les touches Ctrl + O pour « écrire » les modifications dans le fichier.
Aide Quitter Écrire Lire fich. Chercher Remplacer Couper Coller Justifier Orthograp. Pos. cur. Aller lig.
```

Figure 3.9. Fichier *keyboard* en « fr »

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

Chapitre III : Conception et analyse

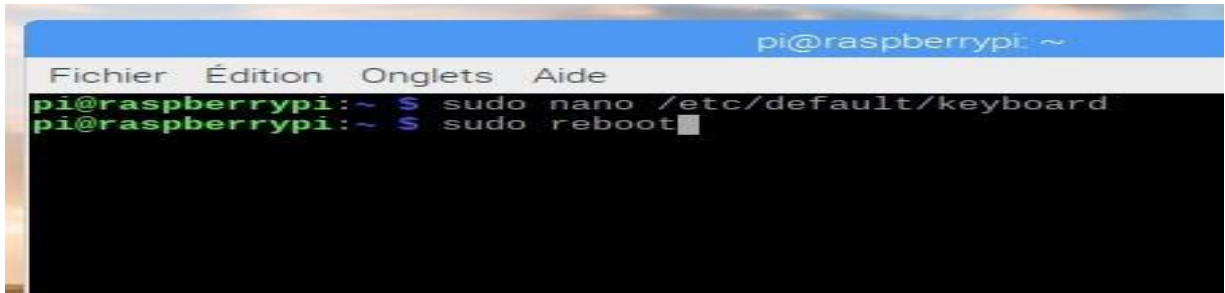


Figure 3.10. la commande du redémarrage du raspberry Pi 3

III.3.3. Configuration de la localisation

- Nous avons Accéder au menu des applications → **Préférences** → **Configuration du Raspberry Pi**.

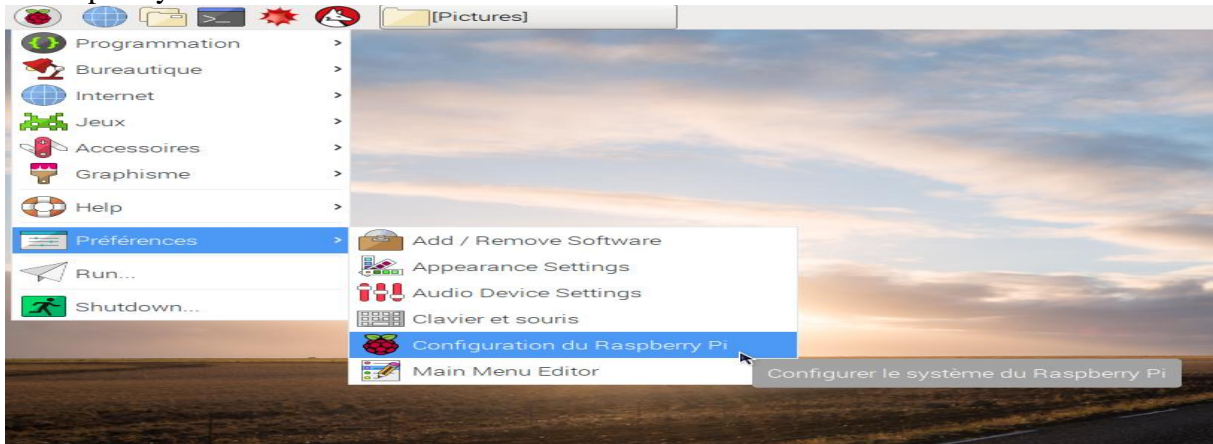


Figure 3.11. Menu des applications

- Dans la fenêtre qui s'ouvre, nous avons accéder au 4e onglet «*Localisation*» :
- Cliquer sur « Régler le clavier » :
 - Cliquer sur « Régler la région du WiFi »

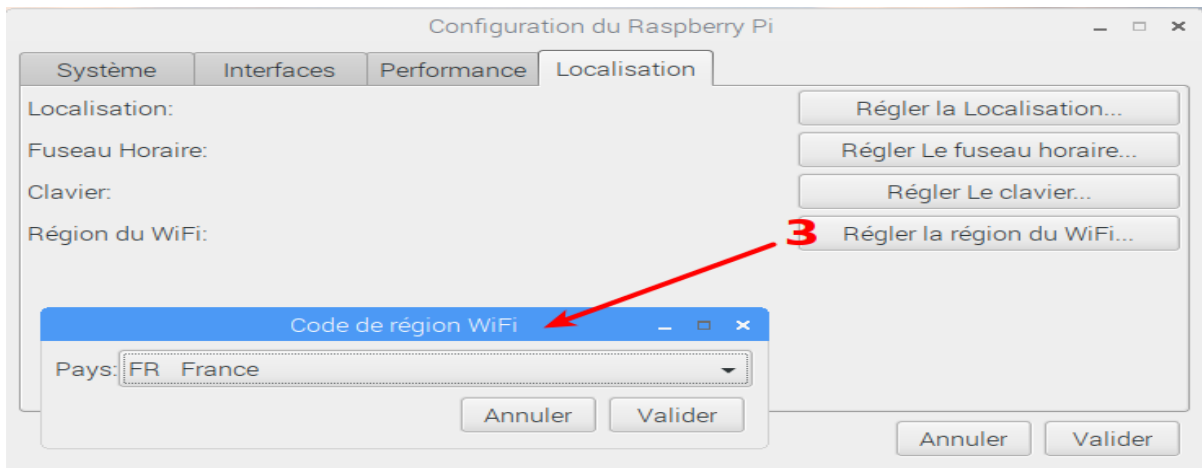


Figure 3.12. réglage du wifi

III.3.4.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 1er onglet «Système» et nous avons cliqué sur le bouton « **Changer le mot de passe** » et saisissez notre nouveau mot de passe.

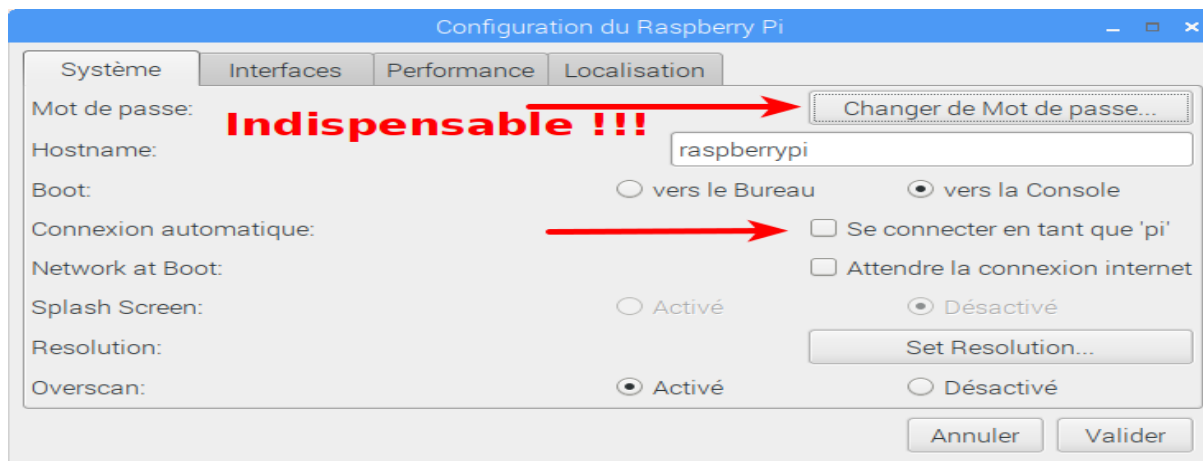


Figure 3.13.modification du mot de passe

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

Au cours du démarrage, nous serons invités à saisir :

- votre login : **pi**
- votre mot de passe : celui que nous venons de personnaliser

III.3.5.L'activation des interfaces utilisées

- Le protocole SSH (secure shell) est un protocole de communication sécurisé, dans lequel toutes les données échangées entre deux machines distantes sont chiffrées.
- Le protocole VNC
- La camera

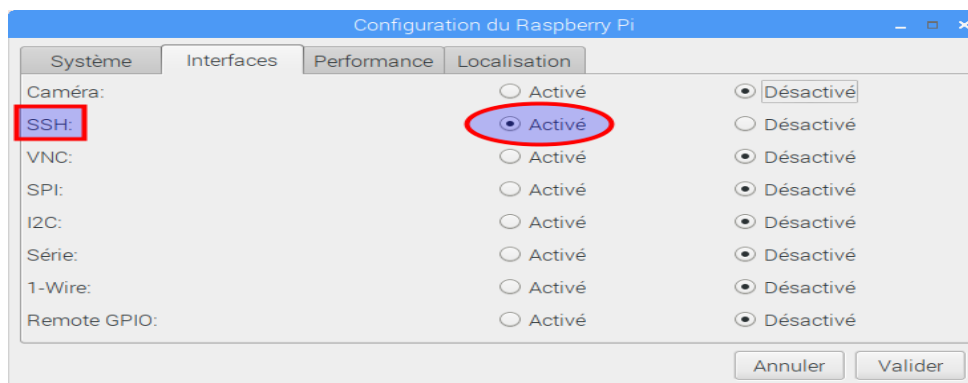


Figure 3.14.L'activation d'interfaces utilisées

Chapitre III : Conception et analyse

Nous avons toujours accéder au menu des applications → Préférences → Configuration du Raspberry Pi.

- Dans la fenêtre qui s'ouvre, accédons au premier, deuxième et troisième onglet du **Interfaces**.
 - nous avons coché le bouton « **Activé** » sur les trois premiers lignes **camera, ssh, vnc**.

Remarque :

Après chaque modification il faut faire :

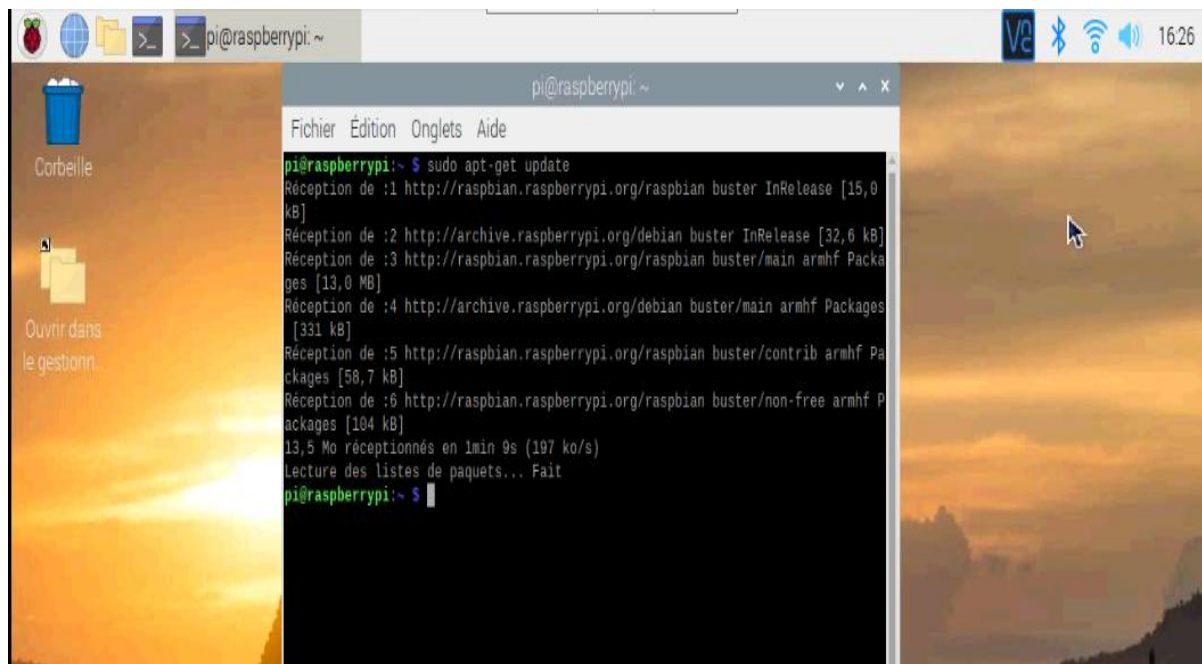
- La validation de l'ensemble des modifications effectuées avec le bouton « Valider » sur la fenêtre de configuration.
- Il va falloir redémarrer le Raspberry PI

III.4.Mettre à jour le système de la 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, etc.

III.4.1.Mettre à jour la liste des paquets

```
sudo apt-get update
```



The screenshot shows a terminal window titled 'pi@raspberrypi: ~' with a menu bar containing 'Fichier', 'Édition', 'Onglets', and 'Aide'. The terminal output is as follows:

```
pi@raspberrypi:~$ sudo apt-get update
Réception de :1 http://raspbian.raspberrypi.org/raspbian buster InRelease [15,0
kB]
Réception de :2 http://archive.raspberrypi.org/debian buster InRelease [32,6 kB]
Réception de :3 http://raspbian.raspberrypi.org/raspbian buster/main armhf Packa
ges [13,0 MB]
Réception de :4 http://archive.raspberrypi.org/debian buster/main armhf Packages
 [331 kB]
Réception de :5 http://raspbian.raspberrypi.org/raspbian buster/contrib armhf Pa
ckages [58,7 kB]
Réception de :6 http://raspbian.raspberrypi.org/raspbian buster/non-free armhf P
ackages [104 kB]
13,5 Mo réceptionnés en 1min 9s (197 ko/s)
Lecture des listes de paquets... Fait
pi@raspberrypi:~$
```

Chapitre III : Conception et analyse

Télécharger et installer les mise à jours

```
sudo apt-get upgrade
```

```
pi@raspberrypi:~$ sudo apt-get upgrade
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Calcul de la mise à jour... Fait
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  alsabase gstreamer0.10-alsa gstreamer0.10-plugins-base
  libgstreamer-plugins-base0.10-0 libgstreamer0.10-0 libxfce4util-bin
  libxfce4util-common libxfce4util7 libxfconf-0-2 pimixer point-rpi
  rpi-eeeprom-images vlc-l10n vlc-plugin-notify vlc-plugin-samba
  vlc-plugin-video-splitter vlc-plugin-visualization xfconf
Veuillez utiliser « sudo apt autoremove » pour les supprimer.
```

III.5. Activer la camera USB (webcam)



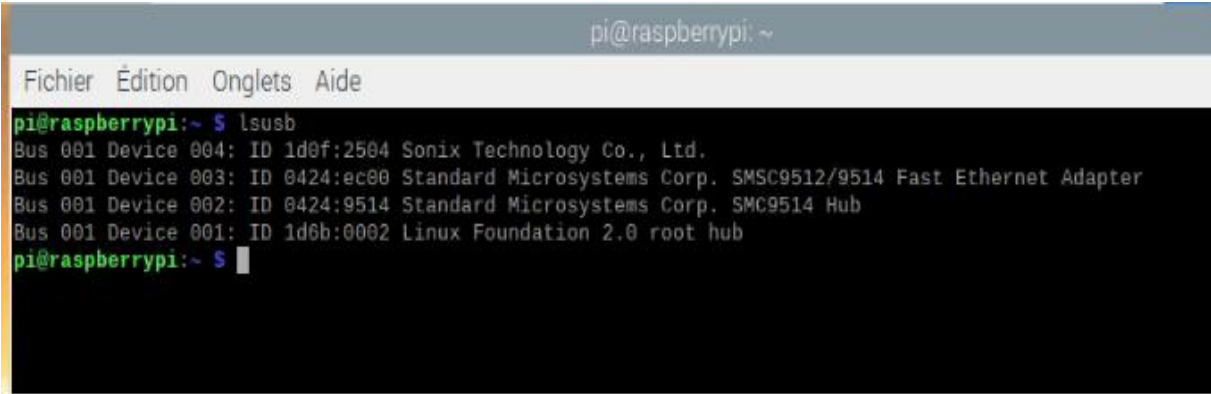
Figure 3.15. Caméra USB (Webcam)

Chapitre III : Conception et analyse

Une webcam, est une caméra conçue pour utilisée comme un périphérique d'ordinateur, pour Vérifier si la webcam est acceptable avec le Raspberry Pi, pour cela il faut connecter la webcam au port USB du Raspberry P et taper la commande :

lsusb

Cette commande nous donne la liste des périphériques USB :



```
pi@raspberrypi: ~
Fichier Édition Onglets Aide
pi@raspberrypi:~ $ lsusb
Bus 001 Device 004: ID 1d0f:2504 Sonix Technology Co., Ltd.
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp. SMC9512/9514 Fast Ethernet Adapter
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp. SMC9514 Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
pi@raspberrypi:~ $
```

Si la Webcam n'est pas définie dans cette liste il est possible que le Raspberry Pi ne fournisse pas une énergie assez pour cette Webcam, il est préférable de changer la Webcam par une autre de moins consommation d'énergie.

III.6.Installation opencv sur le Raspberry Pi

III.6.1.Mettre à jour le système d'exploitation

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Chapitre III : Conception et analyse

III.6.2.Installer les dépendances

Il est nécessaire d'ajouté les modules opencv, 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
```

III.6.3.Installation Python 3 et pip 3

Raspbian Buster est livré avec Python 3.7.3 préinstallé, il n'est donc pas nécessaire d'installer explicitement Python 3.7.3. Pour confirmer, nous avons utilisé la commande suivante pour voir quelle version de Python est installée sur notre Raspberry Pi.

```
Python --version
```

Si ce n'est pas le cas, il faut utiliser ces deux commandes pour installer python 3 et pip3 :

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

III.6.4.Installation Opencv

L'installation de opencv ce fait par cette commande :

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

Ou bien

```
Pip3 install opencv-python
```

Chapitre III : Conception et analyse

III.6.5.Dépendance supplémentaire pour Opencv et la caméra

```
sudo apt-get install libqtgui4  
sudo modprobe bcm2835-v4l2  
sudo apt-get install libqt4-test
```

III.7 Système de reconnaissance facial

Après avoir installé python 3 et opencv notre Raspberry Pi est prêt pour notre projet, Pour créer un projet complet sur la 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

Le schéma fonctionnel ci-dessous reprend ces phases:

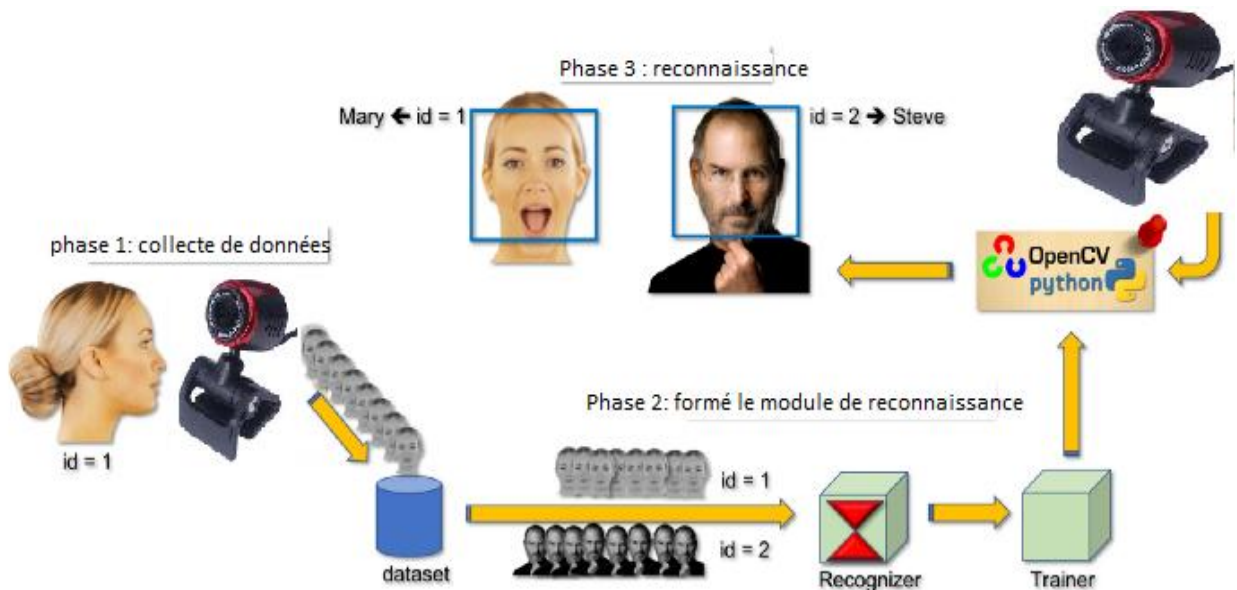


Figure 3.16.Schéma fonctionnel du système de reconnaissance facial

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:

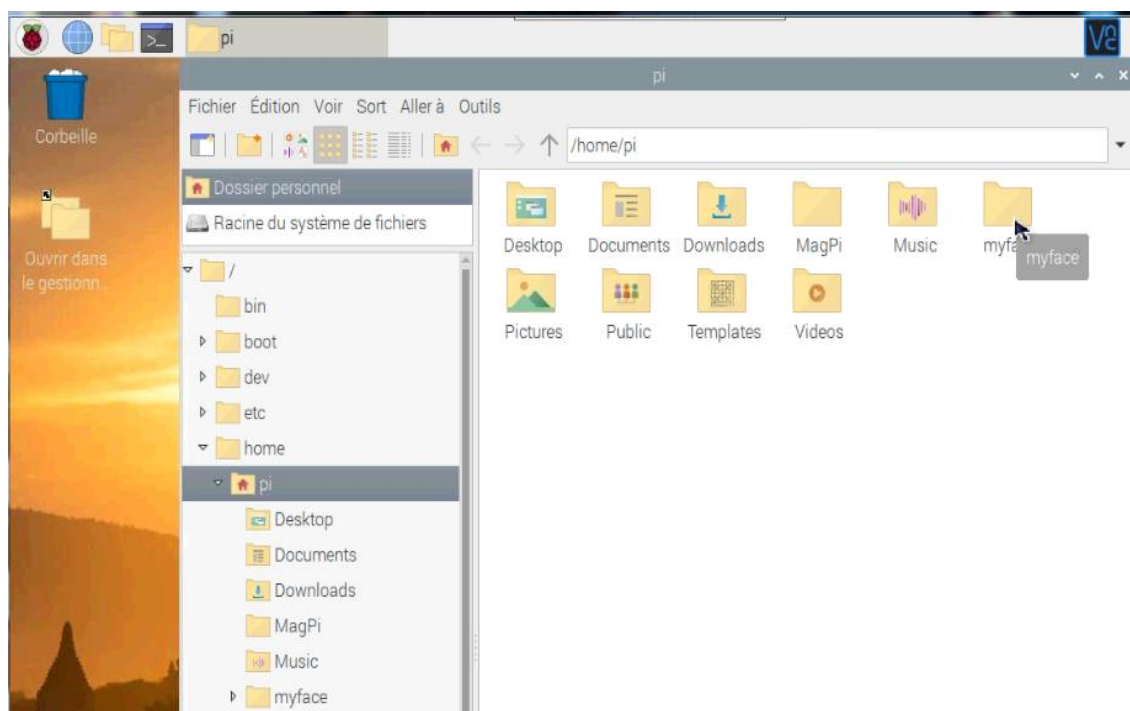
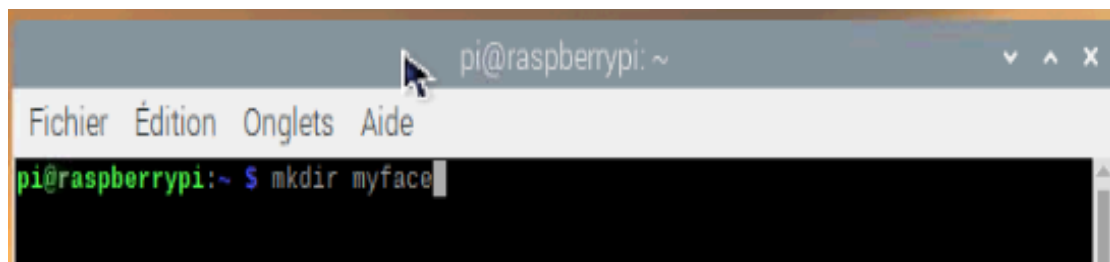


Figure 3.17. Création d'un répertoire de travail

III.7.1. Collecte de données pour la détection du visage

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 opencv classificateur préformé.

Chapitre III : Conception et analyse



Figure 3.18. Phase 1 collecte de données

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 ».

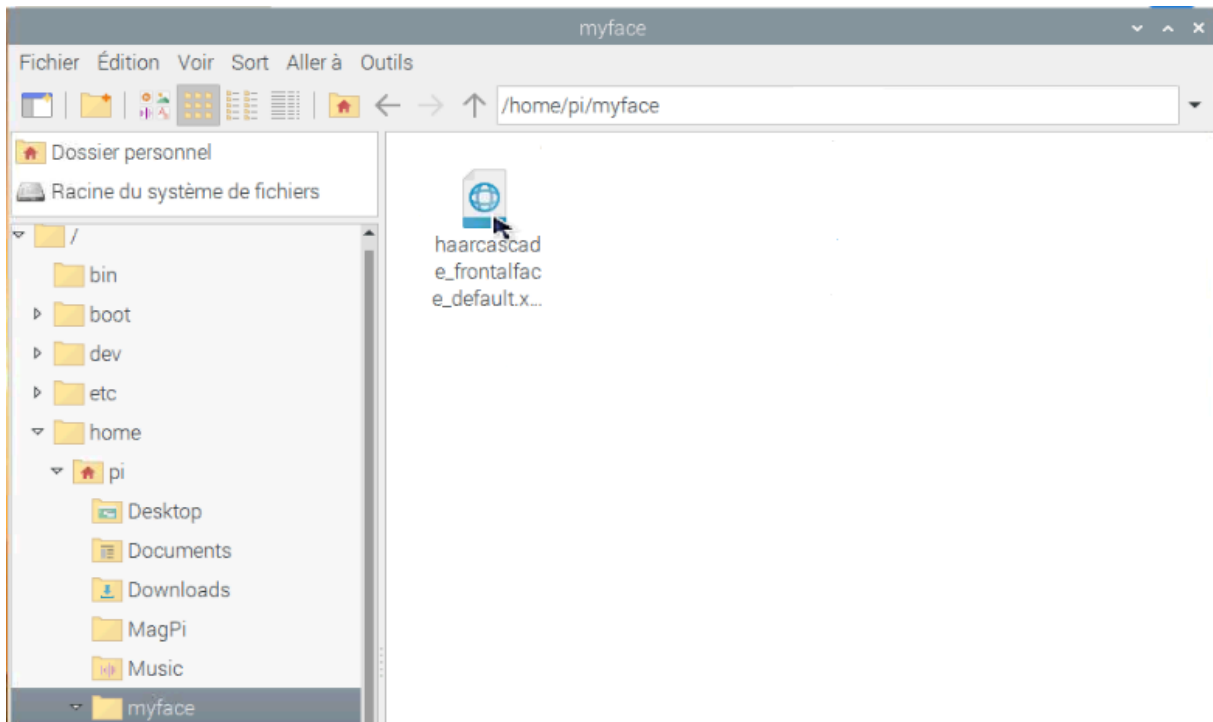


Figure 3.19. Fichier Haarcascade

Chapitre III : Conception et analyse

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".

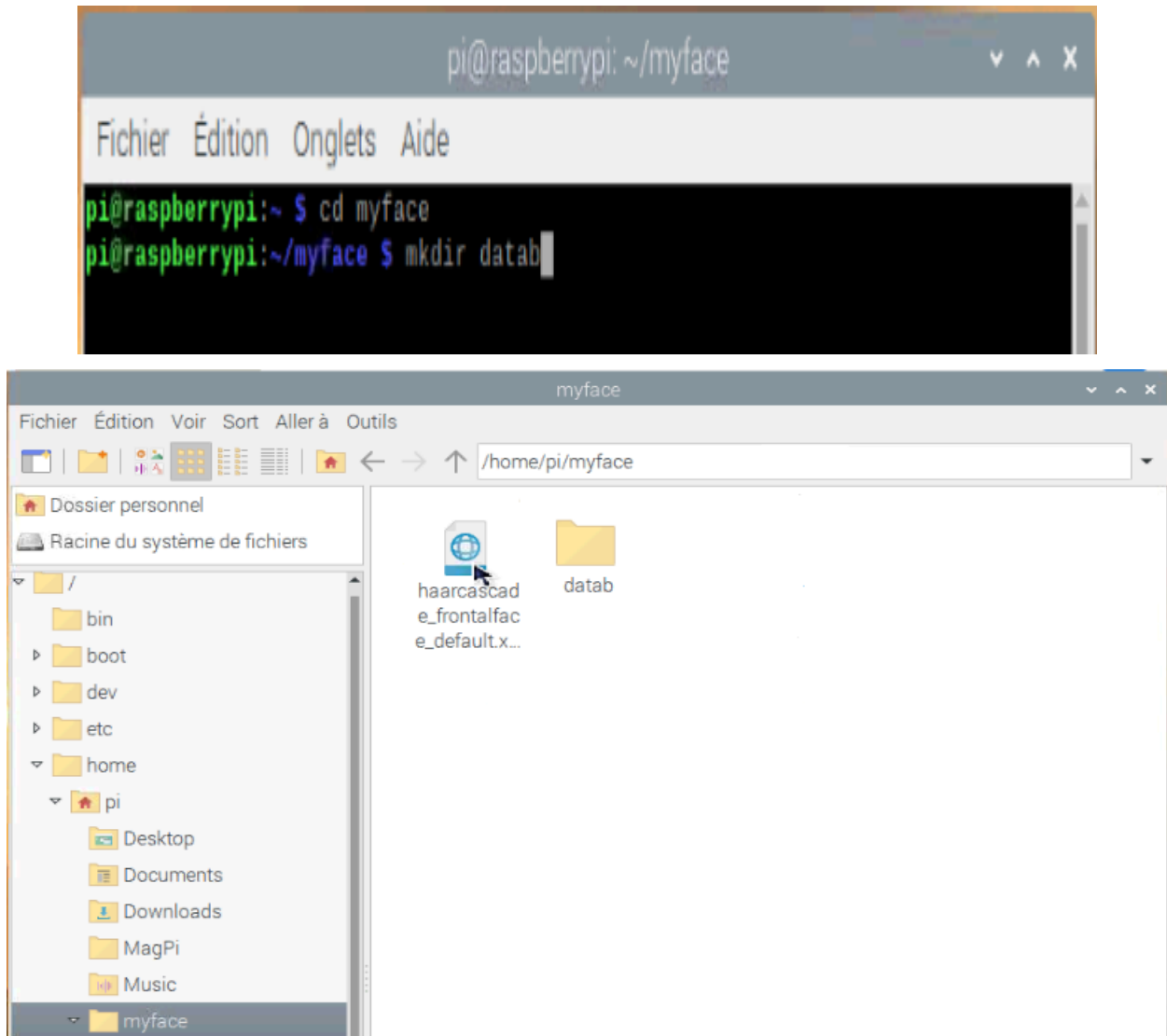


Figure 3.20. Création de dossier d'images datab

III.7.2. Procédure pas à pas de code

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
```

Chapitre III : Conception et analyse

Tout d'abord, nous importons le paquet requis.

```
import cv2  
  
import numpy as numpy
```

Chargons un fichier en cascade pour détecter les visages.

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

Ensuite, nous initialisons l'objet de la caméra qui nous permettra de jouer avec la caméra web.

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

Nous définissons une variable identifiant que nous allons changer pour chaque personne

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

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

Ce code initie une boucle infinie (qui sera interrompue plus tard par une instruction break), où nous avons ret et img définis comme cam.read (). Fondamentalement, ret est un booléen indiquant s'il y a eu ou non un retour, à l'image se trouve chaque image qui est retournée. S'il n'y a pas de cadre, nous n'obtiendrons pas d'erreur, nous n'obtiendrons aucun.

Chapitre III : Conception et analyse

```
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)
```

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.

```
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
```

La fonction ci-dessus nous donne les coordonnées rectangulaires de la zone du visage. Nous avons utilisé ces coordonnées pour extraire le visage de l'image et l'avons enregistré dans le

Chapitre III : Conception et analyse

répertoire « datab » que nous avons créé au début. Après cela, nous avons montré le visage recadré et créé un rectangle sur le cadre d'origine. Le code collectera 50 images.

Cv2.imshow montre le cadre d'origine sur la fenêtre de sortie

Cv2.waitKey(100) est une fonction de liaison de clavier. Il attend cent millisecondes pour tout événement de clavier. Il faut un argument et cet argument est le temps en millisecondes. Si la touche est enfoncée pendant ce temps, le programme continuera. Passer 0 signifie qu'il attendra.

```
cam.release()
```

```
cv2.destroyAllWindows()
```

Cela libère la webcam, puis ferme toutes les fenêtres imshow ().

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

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

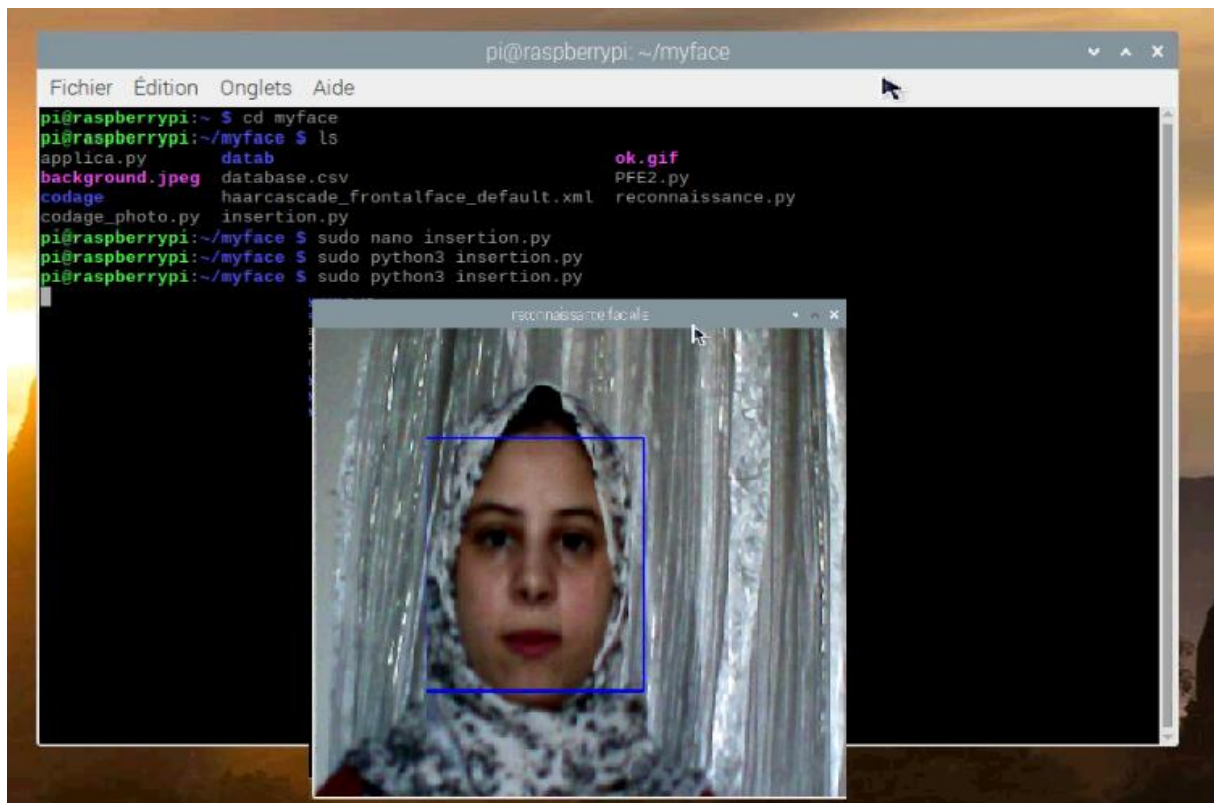


Figure 3.21. Prendre des photos avec le fichier insertion.py

Chapitre III : Conception et analyse

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

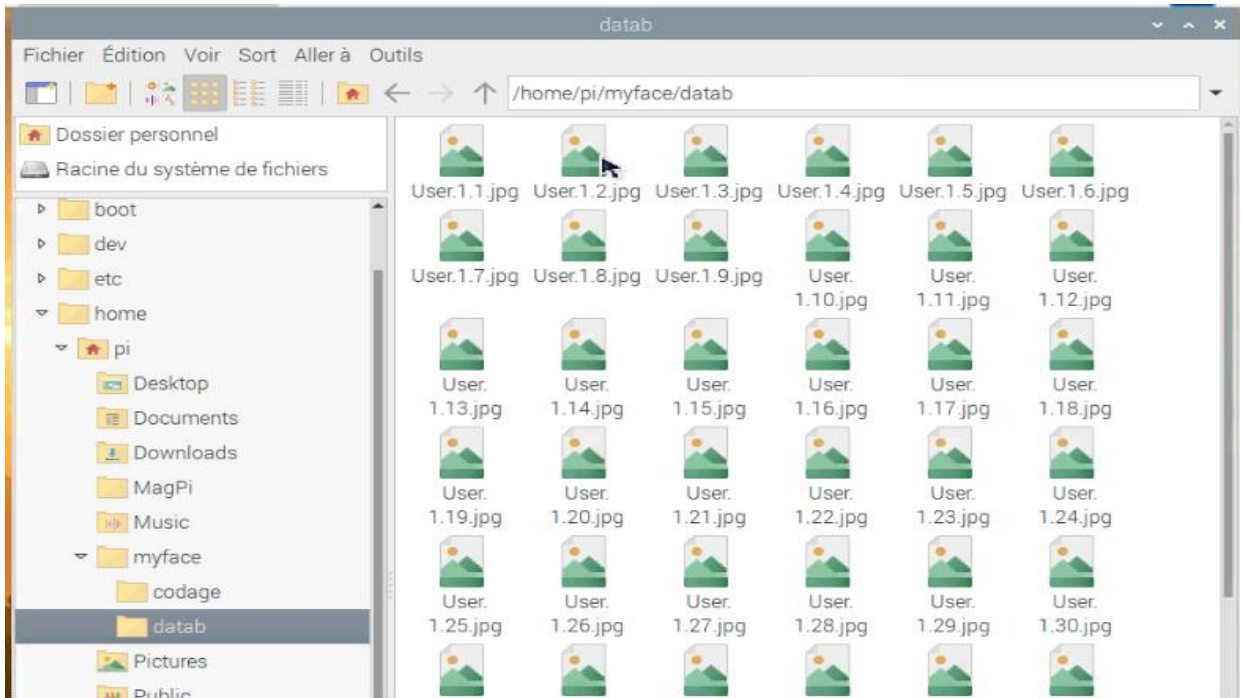
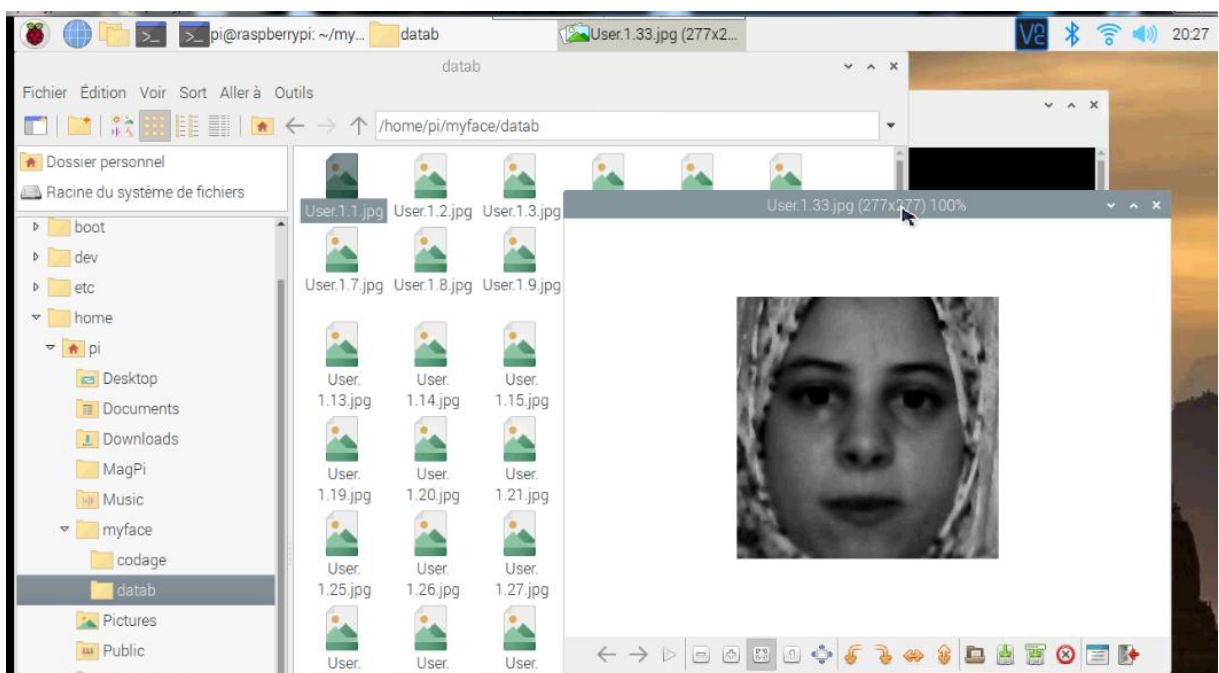


Figure 3.22. Nos 50 images enregistrées dans le répertoire datab



III.7.3. Entraînement de la 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.

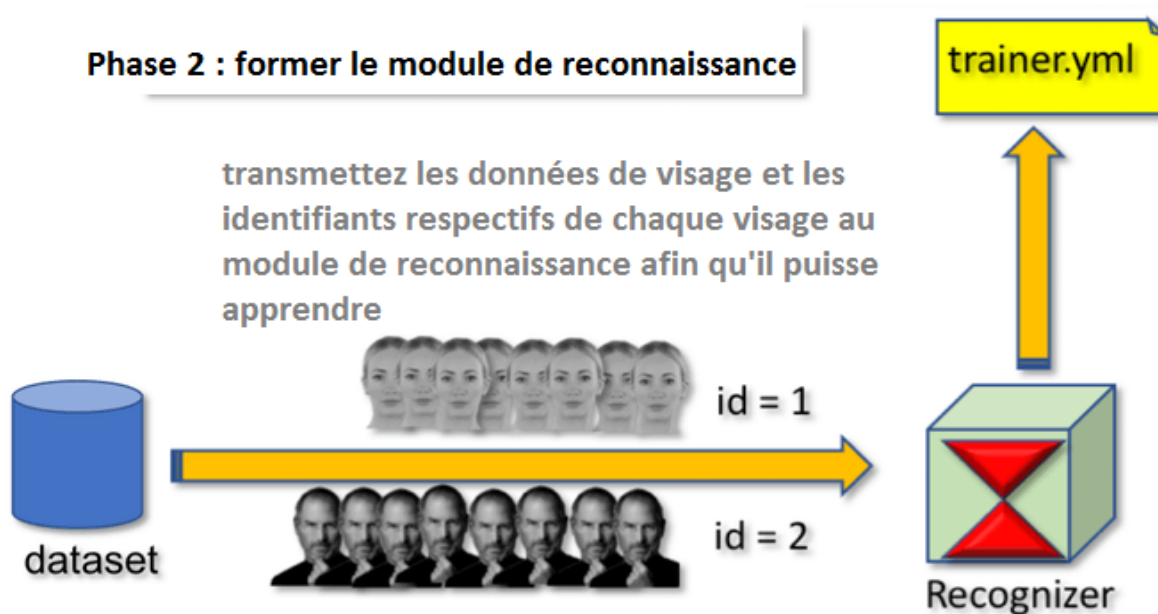


Figure 3.23. Phase 2 former le module de reconnaissance

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

```
pi@raspberrypi: ~/myface
Fichier  Édition  Onglets  Aide
pi@raspberrypi:~ $ cd myface
pi@raspberrypi:~/myface $ mkdir codage
```

Chapitre III : Conception et analyse

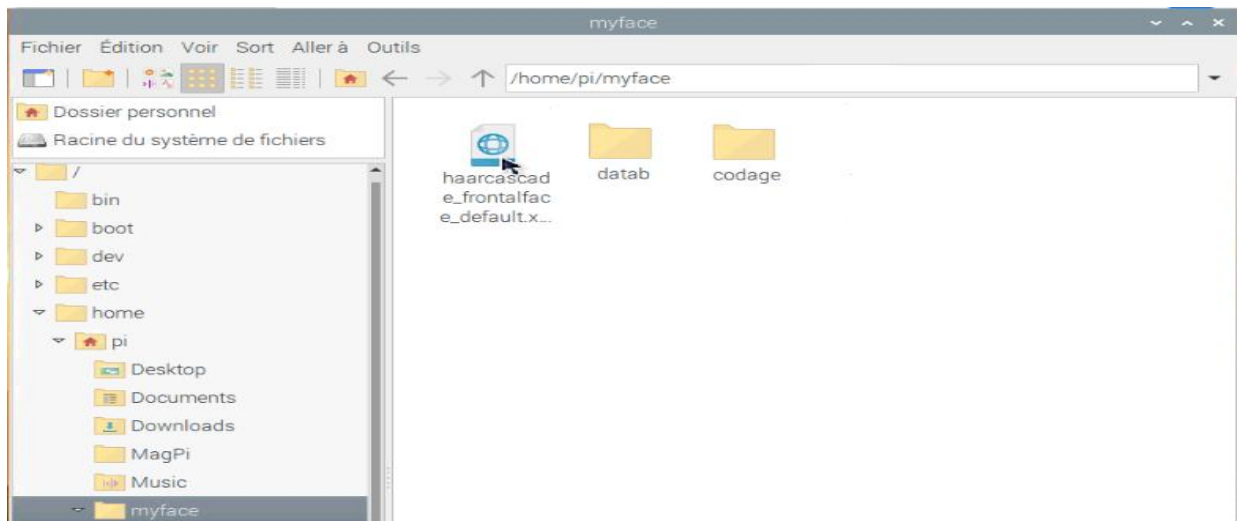


Figure 3.24. création de dossier codage

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
from PIL import Image
```

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"
```

Chapitre III : Conception et analyse

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)`.

· Alors maintenant, dans cette fonction, nous allons faire ce qui suit.

- a. Chargez les images d'entraînement à partir du dossier datab
- b. capturer les visages et l'identifiant à partir des images d'entraînement
- c. Mettez-les dans une liste d'identifiants et d'échantillons de visage et renvoyez-la.

```
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
```

```
    faceImg=Image.open(imagePath).convert("L")
```

Chapitre III : Conception et analyse

```
#Maintenant, nous convertissons l'image PIL en tableau numpy
faceNp=np.array(faceImg,"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)
#Si un visage est là, ajoutez-le dans la liste ainsi que son identifiant
for (x,y,w,h) in faces:
```

```
    face.append(faceNp[y:y+h,x:x+w])
    IDs.append(ID)
cv2.imshow("training",faceNp)
cv2.waitKey(10)
return IDs,faces
```

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()
```

Chapitre III : Conception et analyse

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
```

Il commencera la formation :

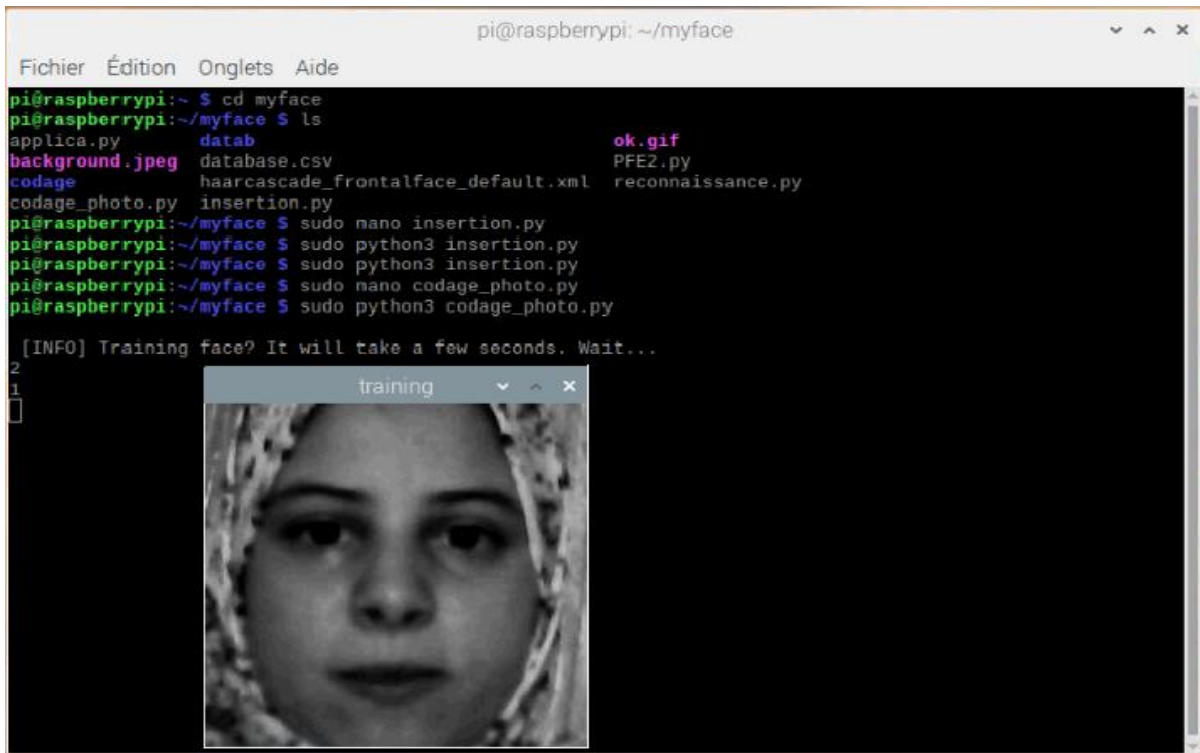


Figure 3.25. Formations des photos

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

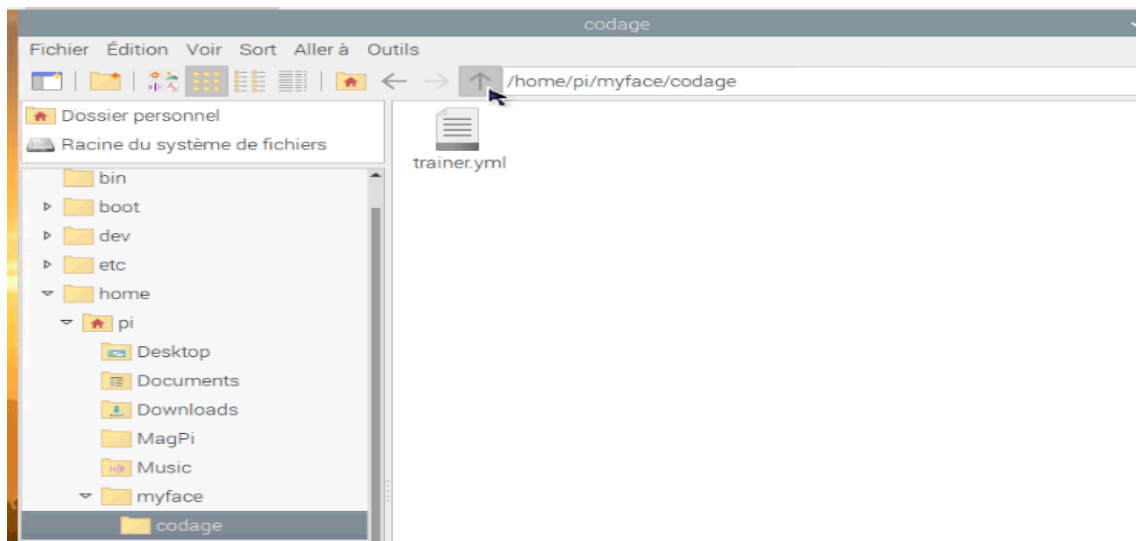


Figure 3.26. Fichier d'entraînement

Chapitre III : Conception et analyse

III .7.4. Utilisation de l'outil de reconnaissance pour la 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).



Figure 3.27. Phase 3 reconnaissance

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.

Chapitre III : Conception et analyse

```
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','sarra','imene']
```

Nous allons maintenant créer l'objet de capture vidéo

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

Ici, nous arrivons à la boucle et la fonction principale

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

Chapitre III : Conception et analyse

```
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 haasCascade. 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)
```

Chapitre III : Conception et analyse

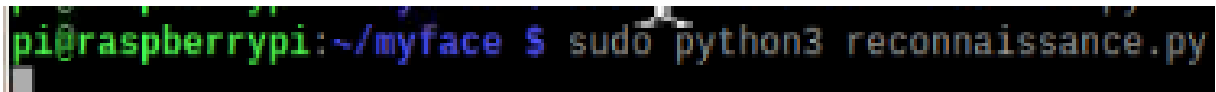
```
cv2.imshow('camera',img)
k = cv2.waitKey(10) & 0xff # Appuyez sur 'ESC' pour quitter la vidéo
if k == 27:
    break
```

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 apposée sur le visage.

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



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

Sur la photo, nous montrons quelques tests effectués avec ce projet, où nous avons également utilisé des photos pour vérifier si le module de reconnaissance fonctionne :

Chapitre III : Conception et analyse

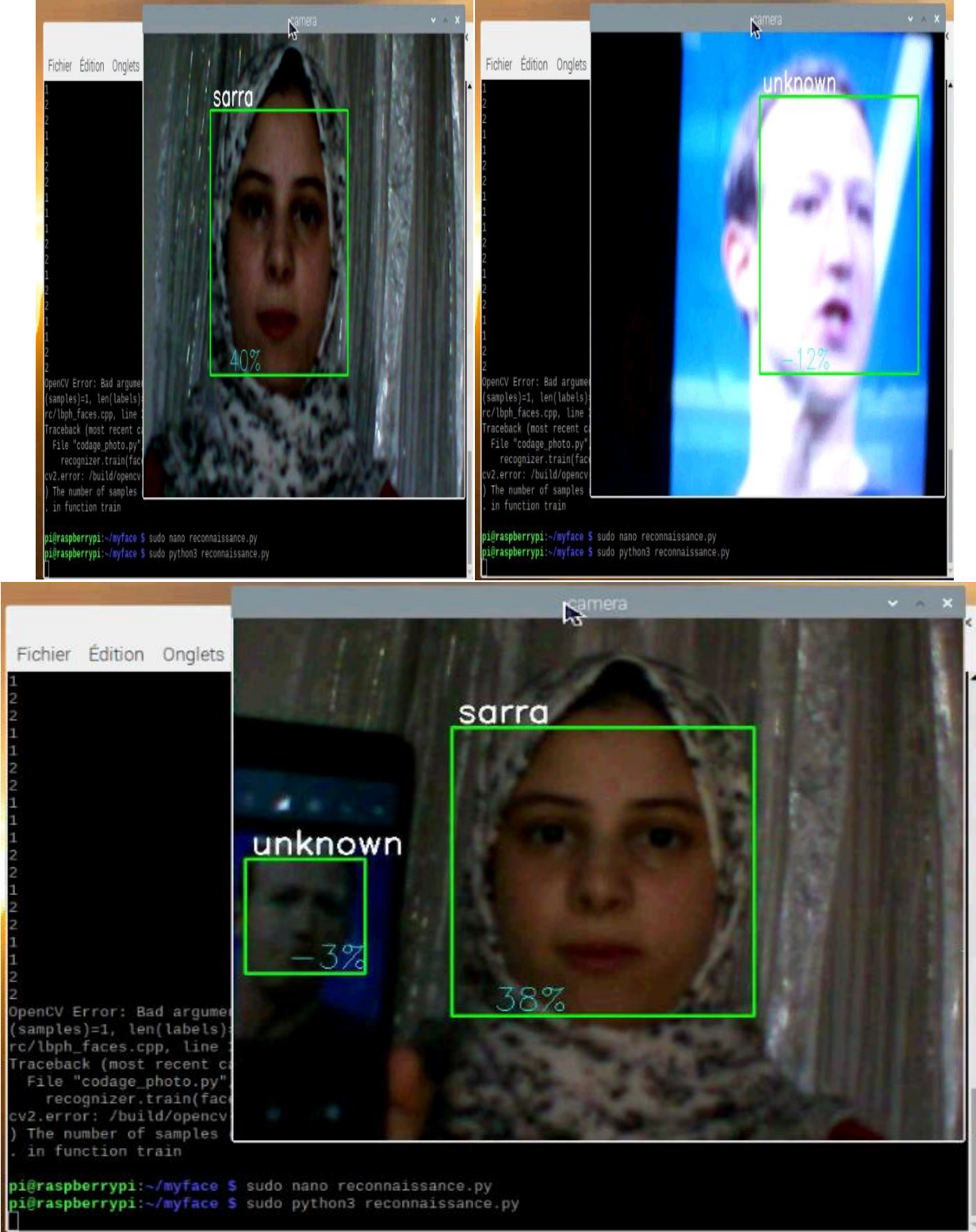


Figure 3.28. exécution de script reconnaissance.

Chapitre III : Conception et analyse

Nous avons utilisé Tkinter pour la création de l'interface graphique montrée dans les photos suivantes :

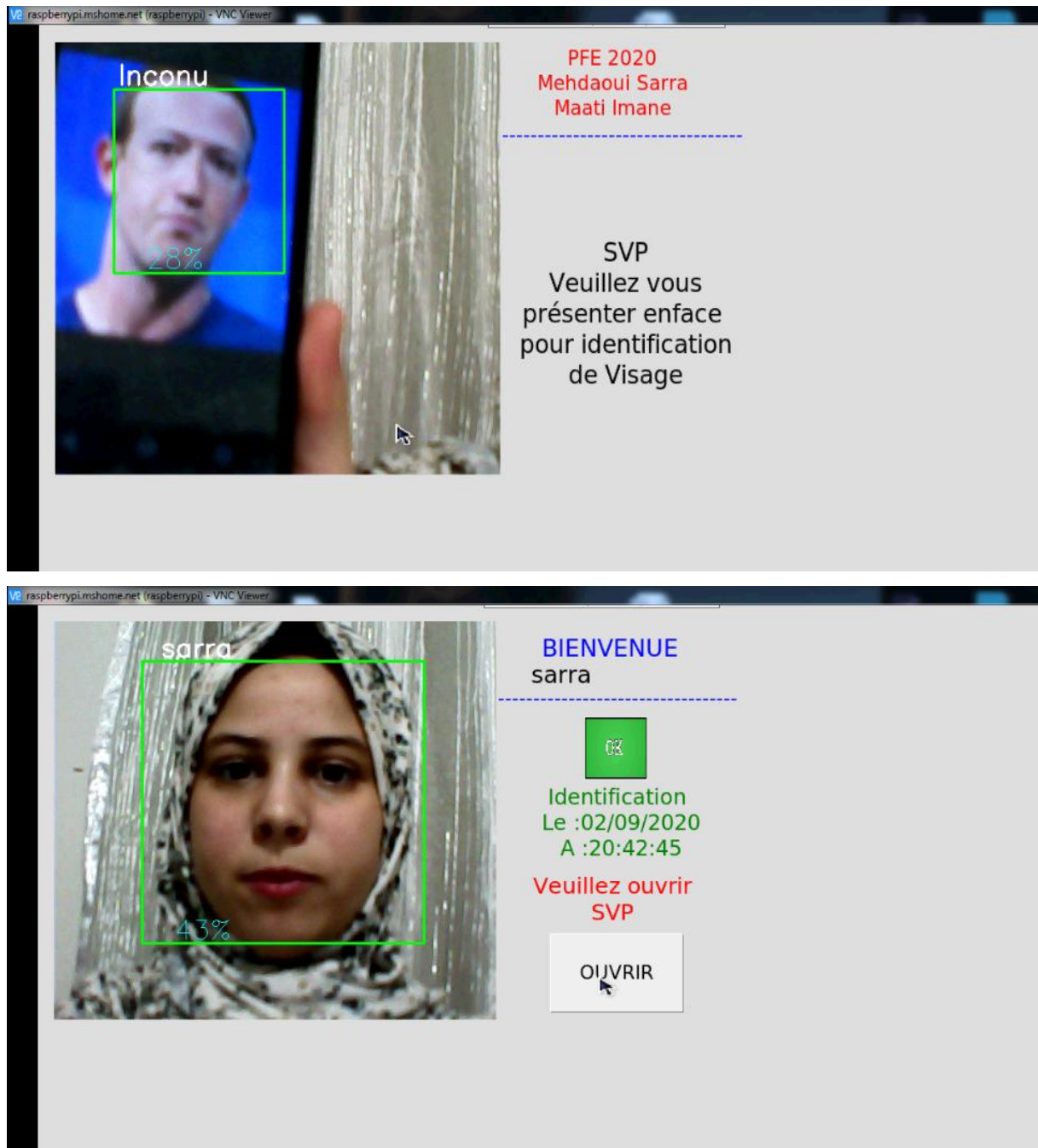


Figure 3.29. interface graphique du système de reconnaissance facial

Chapitre III : Conception et analyse

III.8 création d'une base de données avec csv file

Maintenant notre système de reconnaissance facial est prêt, nous allons créer une base de données avec csv file :

Cette base de données va collecter les données suivantes

- L'identifiant
- Le nom de l'identifiant
- La date et le temps de l'identification

Premièrement, nous allons créer le fichier database.csv dans le dossier du projet nous utilisons la commande suivante :

```
pi@raspberrypi:~/myface $ sudo nano database.csv
```

Importation des bibliothèques :

```
import csv
import time
from datetime import datetime
import locale
locale.setlocale(locale.LC_TIME,")
TIME = datetime.now().strftime('%H:%M:%S')
DATE = datetime.now().strftime('%A %d %B %Y')
DATE2 = datetime.now().strftime('%d/%m/%Y')
```

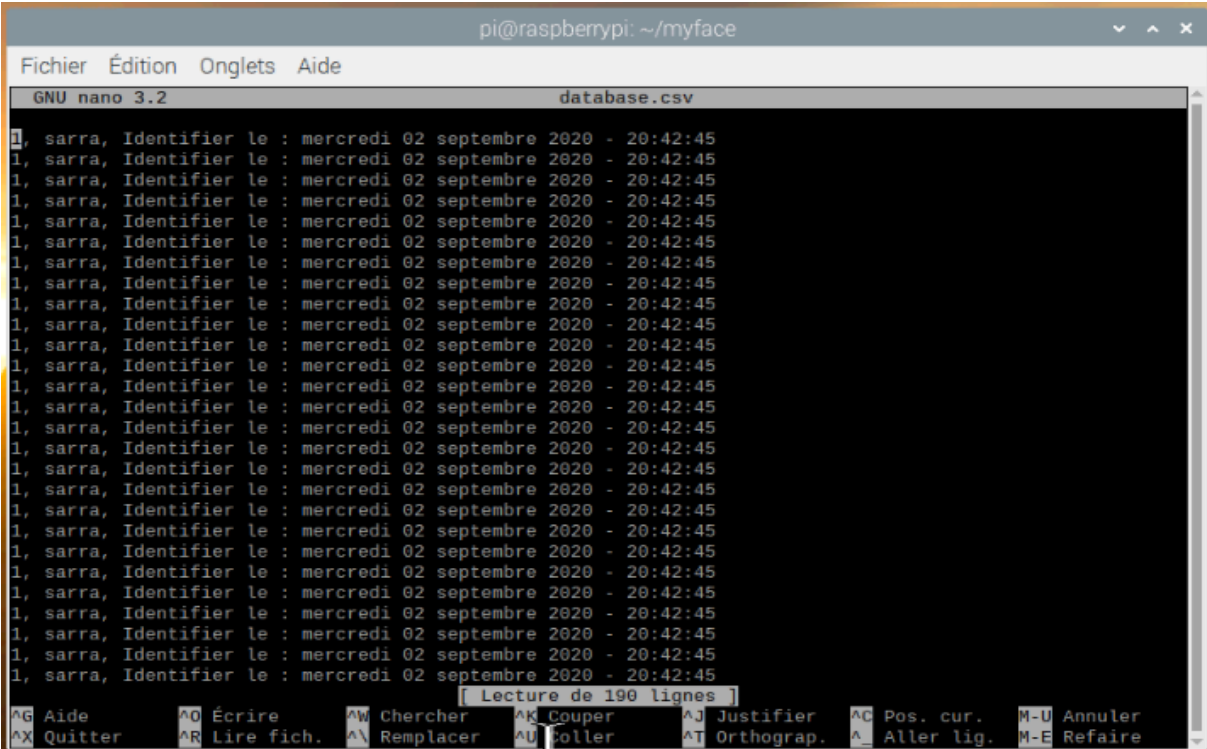
Format de l'heure et / ou la date en fonction des paramètres régionaux. Les noms de mois et de jour de la semaine et les autres chaînes dépendant de la langue respectent les paramètres régionaux actuels définis avec setlocale ().

```
if (confidence < 70):
    name = names[id]
    confidence = " {0}%".format(round(100 - confidence))
    raise_frame(frame2)
```

Chapitre III : Conception et analyse

```
Label4=Label(frame2, text=name, font='size, 20')
Label4.place(x=30, y=60)
B1.place(x=60, y=370)
file = open('database.csv','a')
file.write(str(id)+' '+names[id]+' Identifier le : '+DATE+' - '+TIME+'\n')
file.close()
else:
    name = "Inconu"
    confidence = " {0}%".format(round(100 - confidence))
    cv2.putText(img, str(name), (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)
```

Cette fonction montre que si le visage est reconnu et que la confiance est inférieure à 70%, le programme ouvre le fichier database.csv et écrit lesdites données, un bouton apparaît pour activer le relais.



The image shows a terminal window titled 'pi@raspberrypi: ~/myface' running the GNU nano 3.2 text editor. The editor is open to a file named 'database.csv'. The file contains 20 lines of text, each representing a record with the following format: '1, sarra, Identifier le : mercredi 02 septembre 2020 - 20:42:45'. The terminal window also shows a status bar at the bottom with various keyboard shortcuts for nano, such as 'Aide', 'Écrire', 'Chercher', 'Couper', 'Justifier', 'Pos. cur.', 'Annuler', 'Quitter', 'Lire fich.', 'Remplacer', 'Coller', 'Orthograp.', 'Aller lig.', and 'Refaire'.

Figure 3.30. La base de données database.csv

Chapitre III : Conception et analyse

III.9 pilotage de relais avec les GPIO de raspberry

Pour utiliser la numérotation GPIO.BCM ou GPIO.BOARD on doit utiliser la commande:

```
GPIO.setmode(GPIO.BCM)
GPIO.setmode(GPIO.BOARD)
```

Pour lire la valeur de n'importe quelle broche GPIO, il faut écrire simplement:

```
GPIO.input (pin)
```

Pour gérer un canal de broche GPIO il faut écrire :

```
GPIO.output (pin, statut)
```

Nous avons utilisé le GPIO 21 pour piloter le relais

```
GPIO.setmode(GPIO.BCM)
GPIO.setup(21, GPIO.OUT)
```

```
def Ouvrir():
    raise_frame(frame1)
    GPIO.output(21, 0)
```

La fonction ouvrir (), est destiné à gérer le relais.

III.10 Diagramme de circuit et explication

Les broches GPIO de la Raspberry Pi peut donner une sortie de 3,3 V, mais la serrure électrique nécessite 12V pour fonctionner. Pour cette raison, nous avons utilisé une source d'énergie externe et un relais pour actionner le verrou.

Connectons le VCC et le GND du module de relais à 5V et GND de Raspberry Pi. Connectons ensuite la goupille de signal du module de relais au GPIO 21 de Raspberry Pi. De l'autre côté du module de relais, connectons le négative de la source d'alimentation DC au négatif de la serrure de porte solénoïde. Connectons le positif de la source d'alimentation du courant d'alimentation à la commune du module de relais, puis connectons-nous normalement ouvert du module de relais au positif de la serrure de porte solénoïde.

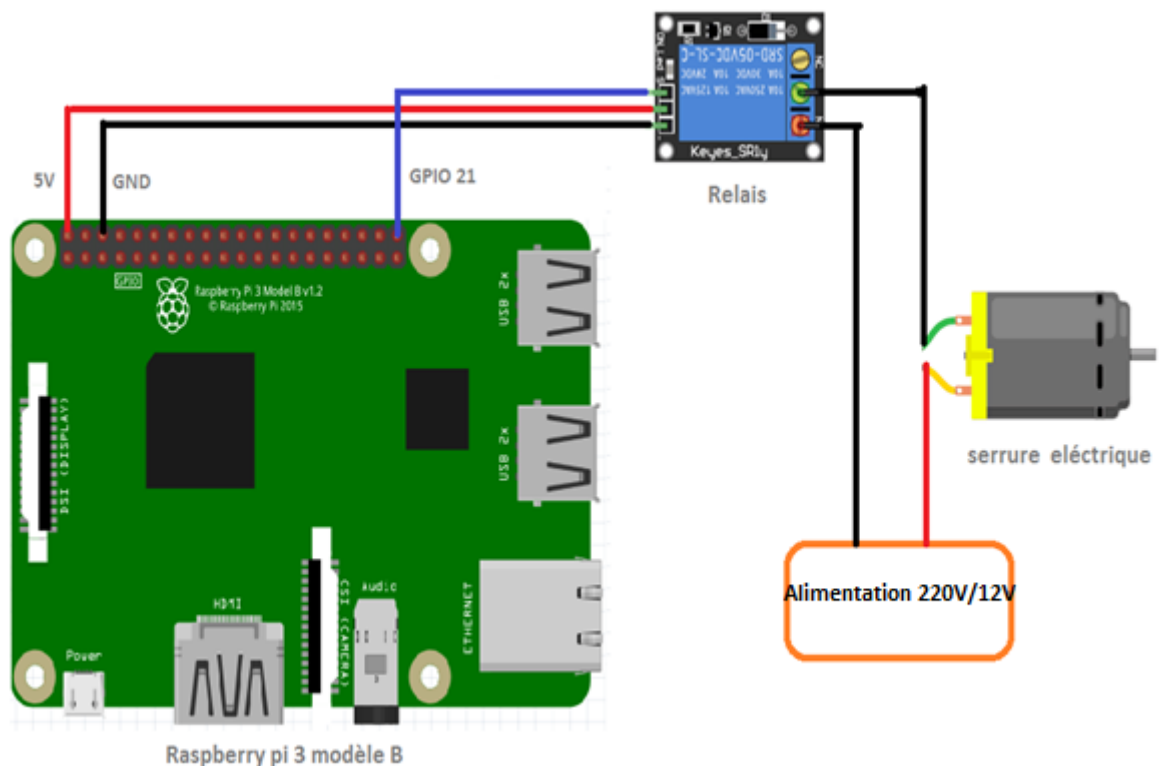


Figure 3.31 .câblage du Relais et serrure électrique avec le Raspberry Pi

Chapitre III : Conception et analyse

III.11 Création de la page HTML

Après avoir terminé le système de reconnaissance facial et la création de la base de donnée nous allons convertir le fichier databse.csv en PHP

Nous somme besoin d'installé apache2 et php :

Avant d'installer le serveur, assurons nous d'avoir une machine bien à jour.

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

Une fois la Raspberry Pi à jour, nous allons installer le serveur Apache.

```
sudo apt install apache2
```

Au passage, nous allons en profiter pour donner des droits au dossier d'apache qui nous permettra de facilement administrer les sites. Pour cela, lançons les commandes suivantes :

```
sudo chown -R pi : www-data /var/www/html/
```

```
sudo chown -R 770 /var/www/html/
```

Une fois l'installation terminée, nous pouvons tester qu'Apache fonctionne correctement en nous rendant sur l'adresse de la Raspberry.

Pour cela, il faut tenter d'accéder à la Raspberry depuis le port 80 (ce port n'étant pas encore ouvert depuis l'extérieur, il faudra le faire depuis la Raspberry elle-même). Il suffit d'ouvrir le navigateur web de la Raspberry (par défaut Midori sous Raspbian), et d'aller à l'adresse « http://127.0.0.1 ». Nous devrion alors obtenir une page avec un message du genre « *It works!* » et plein d'autre texte.

Apache utilise le répertoire `/var/www/html` comme racine pour notre site. Cela signifie que quand nous appelons notre Raspberry sur le port 80 (http), Apache cherche le fichier dans `/var/www/html`. Pour ajouter de nouveaux fichiers, sites, etc., nous devons donc les ajouter dans ce répertoire. Nous pouvons dès à présent utiliser notre Raspberry pour faire un site en HTML, CSS et JavaScript pur, en interne.

Chapitre III : Conception et analyse

Cependant, nous voudrions sans doute rapidement permettre la conversion du fichier database.csv. Pour cela, nous allons avoir besoin de PHP.

On va encore une fois faire appel à l'administrateur pour installer PHP avec la ligne de commande.

```
sudo apt install php php-mbstring
```

Pour savoir si PHP fonctionne correctement, et la méthode est relativement proche de celle employée pour Apache.

Nous allons en premier lieu supprimer le fichier « index.html » dans le répertoire « /var/www/html ».

```
sudo rm /var/www/html/index.html
```

Puis créez un fichier « index.php » dans ce répertoire, avec cette ligne de commande

```
echo '< ?php phpinfo() ; ?>' > /var/www/html/index.php
```

À partir de là, le fonctionnement est le même que pour la vérification d'Apache. Nous tentons d'accéder à notre page, et nous avons le résultat montré de cette image



System	Linux ajaniserveur 3.2.0-4-amd64 #1 SMP Debian 3.2.54-2 x86_64
Build Date	Dec 12 2013 08:42:50
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-gd.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525.NTS

Figure 3.32. Tableau généré par la commande `phpinfo` sur une Raspberry

Chapitre III : Conception et analyse

Après avoir installé apache2 et PHP nous pouvons maintenant écrire le programme qui va créer une page html affichant les informations de la base de données database.csv sous forme d'un tableau.

Pour cela nous allons utiliser le fichier index.php nouvellement créé, supprimé son contenu et écrire le code adapté à cette fonction.

```
<div class="container" id="container">
  <table class="gridtable">
    <thead>
      <tr>
        <th>ID</th>
        <th>Nom et Prénom</th>
        <th>Identifiant le</th>
      </tr>
    </thead>
    <tbody>
```

Ce code fait la création d'un tableau de trois colonnes la première est réservée à l'identifiant « ID », la deuxième pour le nom et le prénom, la troisième pour la date et l'heure de l'identification.

```
<?php
    if (($csvfile = fopen("/home/pi/myface/database.csv", "r")) !== FALSE)
    {
        while (($csvdata = fgetcsv($csvfile, 1000, ",")) !== FALSE) {
            $error="";
            $colcount = count($csvdata);
            echo '<tr>';
            if($colcount!=3) {
```

Chapitre III : Conception et analyse

```
        $error = 'Column count incorrect';
    } else {
        //check data types
        if(!is_numeric($csvdata[0])) $error.='error';
        $date = date_parse($csvdata[3]);
        if (!$date["error_count"] == 0 && checkdate($date["month"],
        $date["day"], $date["year"])) $error.='error';
        if(!is_numeric($csvdata[4])) $error.='error';
    }
    switch($error) {
        case "Column count incorrect":
            echo '<td></td>';
            echo '<td></td>';
            echo '<td class="error" >'.$error.'</td>';
            break;
        case "error":
            echo '<td class="error">'.$csvdata[0].</td>';
            echo '<td class="error">'.$csvdata[1].</td>';
            echo '<td class="error">'.$csvdata[2].</td>';
            break;
        default:
            echo '<td>'.$csvdata[0].</td>';
            echo '<td>'.$csvdata[1].</td>';
            echo '<td>'.$csvdata[2].</td>';
    }
    echo '</tr>';
}
```

Chapitre III : Conception et analyse

```
fclose($csvfile);  
}  
?>
```

La fonction suivante faire ouvrir le fichier database.csv et prendre les informations nécessaire pour les afficher dans le tableau.

Maintenant lorsque nous avons terminé le code et l'enregistrer nous pouvons tester notre page web en tapant l'adresse IP de Raspberry dans la barre de recherche de navigateur a partir de raspberry pi ou bien un périphérique connecté au même réseau local.

Nous obtiendrons l'adresse IP de raspberry Pi on tapant la commande suivante :

```
pi@raspberrypi:~/var/www/html $ ifconfig
```

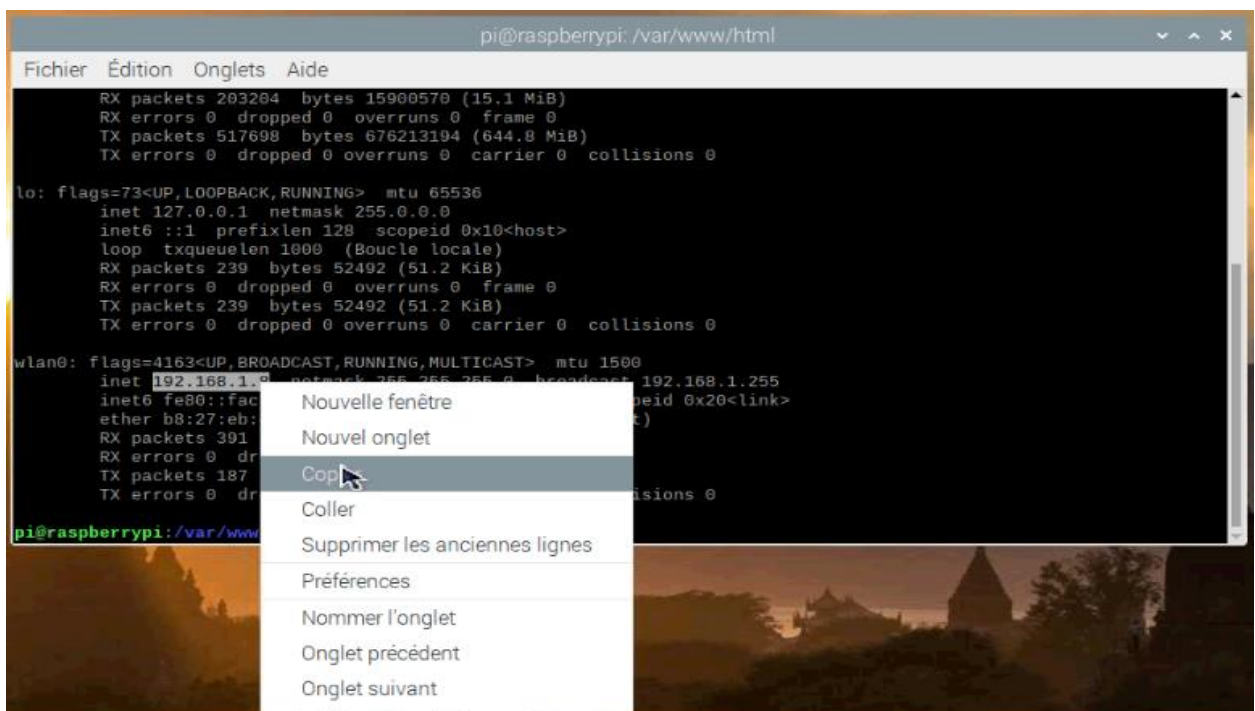


Figure 3.33.obtention de l'adresse IP avec ifconfig

Chapitre III : Conception et analyse

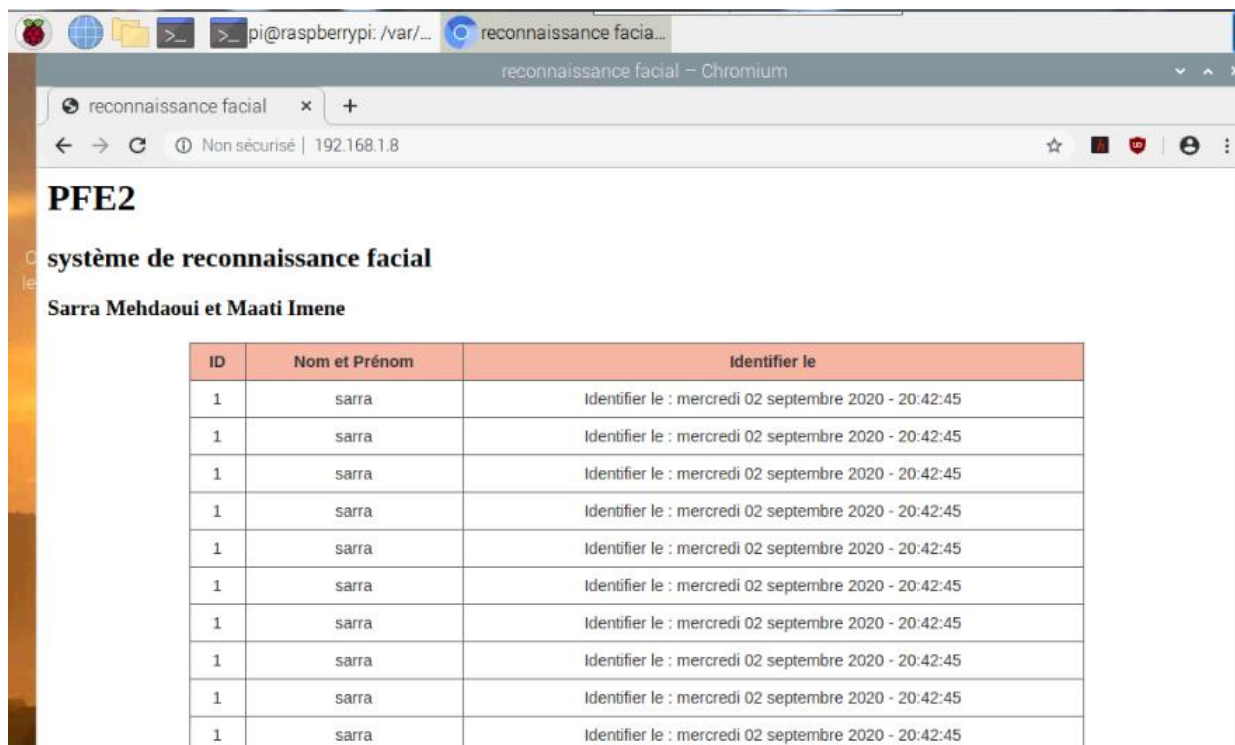


Figure 3.34. la page web de notre projet

III.12 Conception du système

Pour notre projet, le système de reconnaissance faciale doit appliquer l'algorithme suivant:

- 1- Détection de visage et collecte de données
 - 2- Formation du système de reconnaissance
 - 3- Reconnaître les visages
- Si un visage est reconnu, un identifiant est ajouté à la base de données
 - ✓ Vérification du rapport de confidentialité:
 - Si la confiance est inférieure à 70%, la serrure électrique s'ouvrira.
 - Sinon, il reste verrouillé
 - Si le visage n'est pas reconnu, l'autocollant «Inconnu» s'affiche, aucune information ne sera écrite dans la base de données et le verrou reste fermé.

III.12.1 Organigramme du système de reconnaissance facial a base de Raspberry Pi

L'organigramme ci-dessous illustre une série d'étapes allant de la configuration du système d'exploitation au déverrouillage après la reconnaissance des visages.

Chapitre III : Conception et analyse

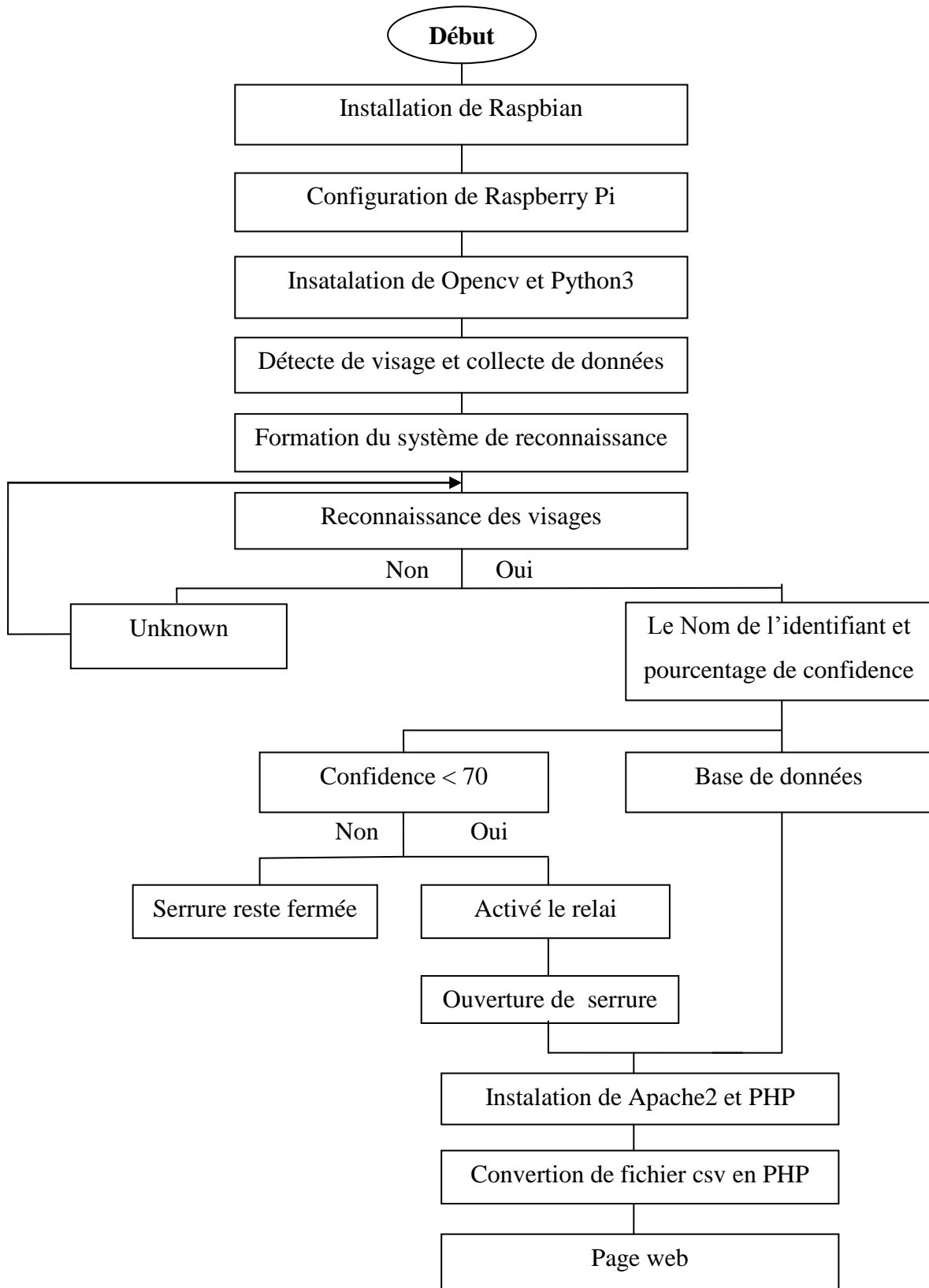


Figure 3.35. Organigramme de système

Chapitre III : Conception et analyse

La figure suivante présente le câblage des différents modules constituant le système :

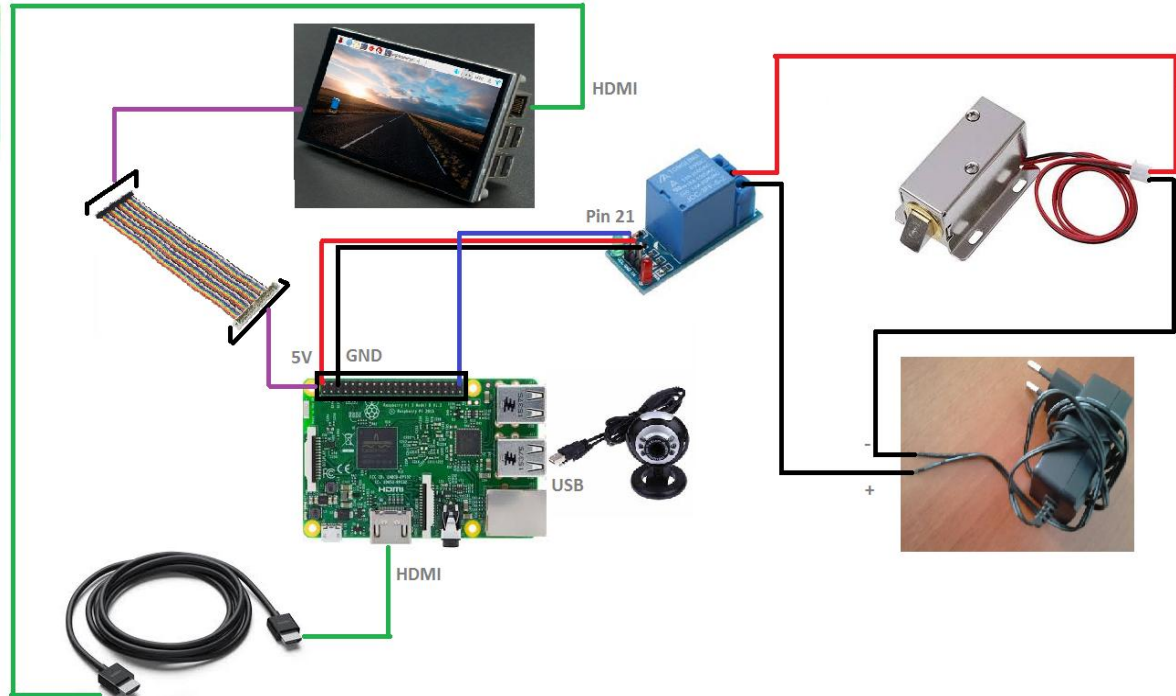


Figure 3.36. Circuit global du système

Le montage expérimental est montré dans la figure suivante :



Figure 3.37. Montage expérimental global

III.13 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 a partir d'une vidéo à l'aide d'une caméra USB et envoi des information sur le visage reconnus à une base de donnée csv. 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

Conclusion général

Conclusion générale

Ce travail illustre le fonctionnement d'un système de verrouillage électronique intelligent avec reconnaissance faciale à base d'un Raspberry Pi 3B. Les aspects de ce système sont: la détection du visage des individus par utilisation du module OpenCV puis vérification de son existence parmi plusieurs visages stockés en base de données, pour des visages inconnus aucune action n'est déclenchée, par contre pour les visages connus et stockés ; l'indentification du visage provoque l'affichage d'un nouveau cadre présentant un bouton de déverrouillage ainsi que le temps et la date de cette identification. L'utilisation de ce bouton provoque l'ouverture de la porte ainsi que l'introduction de ces informations d'accès dans une base de données csv qu'on peut consulter via une page internet sous réseau local

Nous avons rencontré quelques obstacles dans ce travail, tels que l'existence d'un défaut de certains composants, le manque des instruments dans les laboratoires, d'où la nécessité de les chercher ailleurs, le flux d'Internet faible au niveau des laboratoires, le langage de programmes que nous n'avions pas connus auparavant. Mais nous avons tiré parti de nouvelles informations et avons travaillé avec de nombreux programmes (Python, Linux, Raspbian ..). Et nous nous sommes familiarisés avec les différents types de ce système de verrouillage.

Cependant, ce projet peut être amélioré, nous pouvons ajouter plus de paramètres à l'utilisateur pour lui donner plus de contrôle, et lui permettre aussi de contrôler les accès a partir d'un réseau internet externe, ainsi de commander le verrouillage et le déverrouillage a distance.

Référence Biographique

[1] un article <https://www.cnil.fr/fr/definition/reconnaissance-faciale>

[2] Cabal (Christian), 2003. Rapport sur les méthodes scientifiques d'identification des personnes à partir de données biométriques et les techniques de mise en œuvre. Office parlementaire d'évaluation des choix scientifiques et technologiques. Enregistré à la présidence de l'Assemblée nationale Le 16 juin 2003 sous N° 938.

70 pages R. Brunelli and T. Poggio, "Face recognition: Features vs. templates," IEEE Trans. Pattern Anal. Mach. Intell., vol. 15, no. 10, pp. 1042–1053, Oct. 1993.

[3] site internet <https://www.biometrie-online.net/technologies/visage>

[4] reconnaissance faciale [en ligne]

<https://labiometrie.wordpress.com/2017/02/12/reconnaissance-faciale/>

[5] F. Perronnin and J.-L. Dugelay. "Introduction à la biométrie –Authentification des individus par traitement audio-vidéo". Traitement du signal, Vol. 19, No. 4, 2002.)

[6] journal les Echos du 15 octobre 2019

[7] Reconnaissance facial : 7 tendances à suivre pour 2020

<https://www.thalesgroup.com/fr/europe/france/dis/gouvernement/biometrie/reconnaissance-faciale>

[8] P. Jonathon Phillips and Hyeonjoon Moon and Syed A. Rizvi and Patrick J. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 22, number 10, pages 1090-1104, 2000

[9] M. Turk, A. Pentland, Eigenfaces for recognition. J. of Cognitive Neuroscience 3, 72–86,1991.

[10] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in Proc. IEEE Conf. Computer Vision Pattern Recognition, Jun. 1991, pp.586–591.

- [11] P. Jonathon Phillips and Hyeonjoon Moon and Syed A. Rizvi and Patrick J. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 22, number 10, pages 1090-1104, 2000
- [12] S. Der Phillips J., P. J. Rauss. Feret recognition algorithm development and test results. Army Research Laboratory technical report, ARL-TR-995, 1996.
- [13] P. Jonathon Phillips and Hyeonjoon Moon and Syed A. Rizvi and Patrick J. Rauss, "The FERET Evaluation Methodology for Face-Recognition Algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, volume 22, number 10, pages 1090-1104, 2000
- [14] A. Chaari, « Reconnaissance de Visages par Réseaux d'Ondelette de Gabor », Thèse de Doctorat, Discipline : Automatique, Génie Informatique, Traitement du Signal et des Images, Université Lille 1, France, Université Sfax, Tunisie, Soutenue le 08 Décembre 2009.
- [15] W. Hizem, « Capteur Intelligent pour la Reconnaissance de Visages », Thèse de Doctorat, Option : Informatique/Electronique, Institut National des Télécommunications et Université Pierre et Marie Curie, Paris, 2009.
- [16] M. Fedias, « Combinaisons de Données d'Espaces Couleurs et de Méthodes de Vérification d'Identité pour l'Authentification de Visages », Thèse pour obtenir le titre de Docteur en Sciences en Automatique, Faculté des Sciences et de la Technologie, Département de Génie Electrique, Université Mohamed Khider, Biskra, Soutenue le 21 Février 2013.
- [17] <https://translate.yandex.ru/translator/french-russian>
- [18] <https://openclassrooms.com/fr/courses/1766341-structurez-vos-donnees-avec-xml/1766421-quest-ce-que-le-xml>
- [19] <https://www.slideshare.net/marwabaich1/xml-73716331>
- [20] <https://debitoor.fr/termes-comptables/fichier-csv>
- [21] <http://doczz.fr/doc/5278225/--universit%C3%A9-de-batna-2>

[22] Raspberry Pi 3 modèle B

<https://www.ldlc.com/fiche/PB00205573.html>

[23] <https://www.digikey.fr/fr/product-highlight/d/displaytech/5-inch-color-tft-lcd-display>

[24] <http://www.zpag.net/Electroniques/relais.htm>

[25] <https://boutique.semageek.com/fr/628-serrure-electrique-12vdc-solenoid.html>

[26] https://www.lemonde.fr/pixels/article/2018/07/25/je-n-imaginai-pas-que-python-connaissait-un-tel-succes_5335917_4408996.html

[27] <https://openclassrooms.com/fr/courses/235344-apprenez-a-programmer-en-python/230722-faites-vos-premiers-pas-avec-linterpreteur-de-commandes-python>

[28] <https://www.tresfacile.net/interface-graphique-en-python-avec-tkinter/>

[29] <https://www.tresfacile.net/doc/python/programmation-graphique-python-tkinter.pdf>

[30] <https://www.it-swarm.dev/fr/14.04/comment-faire-pour-recuperer-le-mot-de-passe-mysql/961430440/>

[31] <https://www.memoireonline.com/>

[32] <http://dSPACE.univ-tlemcen.dz/bitstream/112/12164/1/Mise-au-Point-dune-Application-de-Reconnaissance-de-Formes.pdf>

[33] <https://fracademic.com/dic.nsf/frwiki/1263331>

[34] <https://fr.wikipedia.org/wiki/OpenCV>

[35] <http://fr.wikipedia.org/wiki/OpenCV>

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.

ملخص

نظام التعرف على الوجه هو تطبيق برمجي يهدف إلى التعرف على الشخص من خلال وجهه تلقائيًا. إنه مجال من مجالات رؤية الكمبيوتر يتضمن التعرف تلقائيًا على الشخص من صورة وجهه. هذا الموضوع تم دراسته بشكل خاص في مجال رؤية الكمبيوتر، مع العديد من المنشورات وبراءات الاختراع، والمؤتمرات المتخصصة

يحتوي التعرف على الوجه على العديد من التطبيقات في المراقبة بالفيديو، والقياسات البيومترية، والروبوتات، وتستخدم هذه الأنظمة عمومًا لأغراض أمنية لإلغاء تأمين الكمبيوتر / الهاتف المحمول / وحدة التحكم، وهي موضع تقدير لأنها تعتبر قليلة التوغل، مقارنة بأنظمة القياسات البيومترية الأخرى (بصمات الأصابع، التعرف على قزحية العين...). يعتمد تشغيل هذه الأنظمة على كاميرا واحدة أو أكثر للتعرف على المستخدم يمكن استخدامها أيضًا لتسهيل حياة المستخدم، لتحديد الوجه في الصور. تعتمد هذه الأنظمة بعد ذلك على الصور / مقاطع الفيديو (قاعدة بيانات) لشخص واحد أو أكثر

سيكون الهدف من مشروعنا هو تنفيذ نظام لتحديد هوية الأشخاص، ويستند هذا التحديد إلى تسلسلات الفيديو. يتم تحليل هذه التسلسلات واستخراج المعلومات المرئية المتعلقة بالوجه، ثم نبني قاعدة البيانات الخاصة بنا ونقدم التقنية للتعرف على شخص من بين المجموعة. يعتمد نظام التعرف على الوجه على مبدأ مرشح Haar ونستخدم مكتبة OpenCv وتطبيقها لخوارزمية فيولا وجونز لاكتشاف الوجه الكلمات الدالة : التعرف على الوجه، مرشح Haar، Viola و Jones، قاعدة البيانات، OpenCv، المعالجة المسبقة

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.