Royaume du Maroc - المملكة المغربية

Université Mohammed V - RABAT
جامعة محمد الخامس - الرباط

Ecole Supérieure de Technologie - Salé
المدرسة العليا للتكنولوجيا - سلا

DEPARTEMENT MAINTENANCE INDUSTRIELLE

Filière DUT :

GENIE INDUSTRIEL ET MAINTENANCE (GIM2)

Rapport de Projet de Fin d'Etude

Sous le thème :

Conception d'un Système D'accès par Reconnaissance Faciale

Soutenu et présenté le: 16 /04/2025

Par :

Kaoutar EL GHAMMARTI

Bouthaina BOUZAIDI-TIALI

Devant le jury composé de :

Encadrant: Mr I. BOUGANSSA

Membre du jury: Mme F.Z. RAMADAN

Année universitaire : 2024/2025

Remerciements

Nous souhaitons exprimer notre profonde gratitude à Monsieur Bouganssa, notre encadrant, pour sa disponibilité, sa patience et ses conseils tout au long de ce projet. Il a su nous encadrer avec sérieux et bienveillance, en nous guidant à chaque étape avec justesse. Grâce à lui, nous avons pu apprendre, progresser et donner le meilleur de nous-mêmes.

Nous tenons également à remercier Monsieur Bybi et Monsieur Oubari, qui nous ont toujours soutenus, motivés et encouragés tout au long de notre parcours. Leur confiance et leur écoute nous ont vraiment aidés à avancer, surtout dans les moments de doute. Leur présence nous a toujours été précieuse.

Nous remercions aussi l'ensemble de nos professeurs à l'École Supérieure de Technologie, pour la qualité de leur enseignement, leur accompagnement, et pour tout ce qu'ils nous ont transmis au fil des années.

Et enfin, merci à toutes les personnes qui nous ont soutenus, de près ou de loin, dans la réalisation de ce projet.

Résumé

Ce projet de fin d'études s'inscrit dans le cadre de la mise en œuvre d'une solution technologique innovante dans le domaine de la sécurité. Il porte sur la conception d'un système intelligent capable de reconnaître automatiquement un visage humain afin de contrôler l'accès à un espace. L'objectif principal est d'allier intelligence artificielle, automatisation et simplicité d'utilisation, pour offrir une alternative moderne aux moyens de sécurité traditionnels.

Ce travail nous a permis de mobiliser nos compétences en programmation, en traitement d'image, et en conception de systèmes embarqués. Il illustre concrètement la manière dont les nouvelles technologies peuvent répondre à des besoins pratiques du quotidien, tout en ouvrant la voie à des perspectives d'amélioration et d'évolution.

Abstract

This graduation project is part of the implementation of an innovative technological solution in the field of security. It focuses on the design of an intelligent system capable of automatically recognizing a human face to control access to a space. The main objective is to combine artificial intelligence, automation, and ease of use to provide a modern alternative to traditional security methods.

This work allowed us to utilize our skills in programming, image processing, and embedded system design. It concretely illustrates how new technologies can address practical everyday needs while paving the way for future improvements and developments.

ملخص

هذا المشروع التخرجي يأتي في إطار تنفيذ حل تكنولوجي مبتكر في مجال الأمن. وهو يركز على تصميم نظام ذكي قادر على التعرف تلقائياً على الوجه البشري للتحكم في الوصول إلى مكان معين. الهدف الرئيسي هو الجمع بين الذكاء الاصطناعي والأتمتة وسهولة الاستخدام، لتقديم بديل حديث لوسائل الأمن التقليدية.

لقد سمح لنا هذا العمل باستخدام مهاراتنا في البرمجة ومعالجة الصور وتصميم الأنظمة المدمجة. وهو يوضح بشكل ملموس كيف يمكن للتكنولوجيات الجديدة أن تلبي احتياجات عملية في الحياة اليومية، مع فتح الباب أمام آفاق التطوير والتحسين المستقبلية.

Table des matières

Liste des Abréviations

- **2D**: Deux Dimensions
- **3D**: Trois Dimensions
- **ADN**: Acide DésoxyriboNucléique
- **CAO** : Conception Assistée par Ordinateur
- **CNN**: Convolutional Neural Network
- **FAST:** Function analysis system technique
- **FC:** Fonctions complémentaires
- **FP:** Fonction principale
- **GND**: Ground (Masse)
- **GPIO** : General Purpose Input/Output (Entrée/Sortie Générale)
- **I2C**: Inter-Integrated Circuit
- **LCD** : Liquid Crystal Display (Affichage à Cristaux Liquides)
- **LED**: Light Emitting Diode (Diode Électroluminescente)
- **OS**: Operating System (Système d'exploitation)
- **PIR sensor** : Passive InfraRed Sensor (Capteur Infrarouge Passif)
- **PWM** : Pulse Width Modulation (Modulation de Largeur d'Impulsion)
- **SADT:** Structured Analysis and Design Technique
- **SCL**: Serial Clock Line (Ligne d'Horloge en I2C)
- **SDA** : Serial Data Line (Ligne de Données en I2C)
- **SMBus**: System Management Bus (Bus de gestion système)
- **VCC**: Voltage Common Collector (Tension d'alimentation positive)
- **VNC**: Virtual Network Computing

Liste des figures

Liste des tableaux

Introduction Générale

Les systèmes de reconnaissance faciale sont de plus en plus utilisés pour l'identification des personnes, car ils permettent d'analyser les caractéristiques du visage et d'associer une identité de manière discrète et efficace. Ce type de technologie est particulièrement adapté à des applications de surveillance, que ce soit dans les maisons, les entreprises ou les établissements scolaires et universitaires. Toutefois, les systèmes existants se basent principalement sur la détection de points clés du visage ou sur l'analyse d'images, ce qui peut présenter certaines limites en termes de précision et de fiabilité.

Dans notre projet, nous avons choisi d'intégrer la reconnaissance faciale comme moyen d'identification pour contrôler l'accès à un espace. L'objectif est simple : permettre l'ouverture d'une porte uniquement lorsque le visage reconnu appartient à un utilisateur autorisé. Cela permet de renforcer la sécurité et de limiter l'accès aux seules personnes enregistrées dans le système.

Le fonctionnement repose sur l'enregistrement préalable des visages des utilisateurs. Chaque visage est lié à un identifiant unique. Lorsqu'une personne se présente devant la caméra, l'image captée est comparée à celles enregistrées dans la base de données. Si une correspondance est trouvée, l'accès est autorisé et la porte s'ouvre. Dans le cas contraire, l'entrée est refusée. Ce système permet donc de sécuriser efficacement l'accès tout en étant simple à utiliser.

Pour la mise en œuvre, nous utilisons une carte **Raspberry Pi 4** fonctionnant sous le système **Raspbian (Linux)**. Grâce aux outils open source disponibles et au langage **Python**, nous avons pu développer un petit serveur local et intégrer un algorithme de reconnaissance faciale. Lorsque le système identifie un visage autorisé, il envoie une commande au **micro servo moteur**, qui actionne mécaniquement l'ouverture ou la fermeture de la porte.

Ainsi, notre système s'appuie sur des composants clés: une Raspberry Pi, une caméra, et un **servo moteur**, pour créer une solution simple, autonome et efficace de contrôle d'accès par reconnaissance faciale

Chapitre 1 :

Généralités d'un Système d'accès par Reconnaissance Faciale

1. Introduction au chapitre :

Aujourd'hui, les vols d'identité et les fraudes sont de plus en plus fréquents, ce qui soulève de vrais enjeux de sécurité. Les méthodes classiques comme les clés, mots de passe ou cartes peuvent facilement être perdues, oubliées ou volées. Face à ces limites, la biométrie s'impose peu à peu comme une solution plus sûre. Elle repose sur des éléments uniques à chaque personne, comme les empreintes digitales, et permet une identification plus fiable. De plus en plus utilisée, elle offre une réponse concrète aux problèmes d'accès et de sécurité.

2. Biométrie:

2.1. Définition de la biométrie :

La biométrie est une technologie émergente qui permet de vérifier l'identité d'une personne en s'appuyant sur une ou plusieurs de ses caractéristiques physiques ou comportementales. Elle est aujourd'hui largement utilisée dans de nombreuses applications pour reconnaître de manière fiable les individus [1].

2.2. Modes de fonctionnement d'un système biométrique :

Un système biométrique est un système de reconnaissance basé sur l'analyse des formes. Il commence par collecter les données biométriques de la personne à identifier, puis en extrait des caractéristiques spécifiques. Ces caractéristiques sont ensuite comparées aux modèles enregistrés dans une base de données. Selon l'usage prévu, un système biométrique peut fonctionner en mode d'enrôlement, de vérification ou d'identification. Un système biométrique fonctionne en trois modes à savoir l'enrôlement, l'identification et la vérification [3] :

- **L'enrôlement :** l'utilisateur est enregistré pour la première fois.
- **La vérification :** le système confirme que la personne est bien celle qu'elle prétend être.
- **L'identification :** le système recherche dans la base de données pour reconnaître l'individu

2.3. Modalités biométriques :

Il existe plusieurs façons d'identifier une personne grâce à la biométrie. Chaque individu possède des traits uniques que l'on peut utiliser. Ces différentes modalités biométriques offrent chacune leurs avantages selon les besoins et les contextes d'utilisation. Dans ce qui suit, nous allons découvrir les principales modalités biométriques et ce qui les rend utiles dans différents cas[1] [2].

➢ Modalités morphologiques :

Dans cette catégorie, l'authentification repose sur des caractéristiques physiques uniques à chaque personne, comme les empreintes digitales, le visage, l'iris etc.

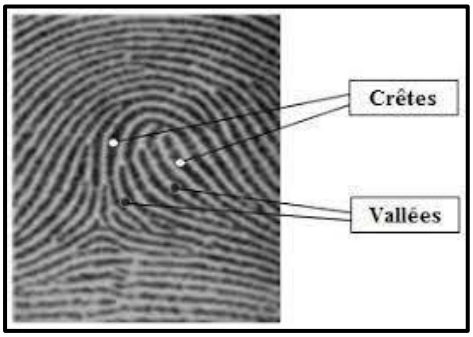

Figure 1:Les empreintes digitales

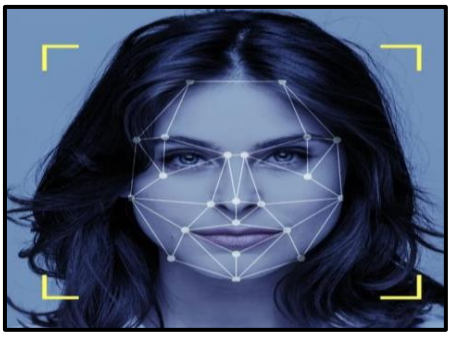

Figure 3: Les empreintes faciales

Figure 2:Reconnaissance de l'Iris

➢ Modalités biologiques :

Les modalités biologiques en biométrie reposent sur des éléments internes propres à chaque individu, comme l'**ADN** ou les caractéristiques du **système sanguin**. Ces données sont particulièrement fiables car elles sont difficiles à imiter ou à modifier. Bien que leur utilisation soit plus rare à cause de la complexité des analyses, elles offrent un haut niveau de sécurité dans des contextes très sensibles.

➢ Modalités comportementales :

Elle repose sur des habitudes propres à chaque individu. Contrairement aux traits physiques, elle s'intéresse à des actions comme la manière de marcher, de parler ou d'utiliser un clavier, des comportements qui varient naturellement d'une personne à l'autre.

La démarche : Cette méthode reconnaît une personne à partir de sa façon de se déplacer. Des éléments comme le rythme, la posture ou les mouvements du corps sont analysés à l'aide de caméras. Puisque la marche dépend de la morphologie et de la musculature, elle est difficile à imiter, ce qui en fait un bon critère d'identification.

La signature : Même si elle peut sembler facile à copier, la signature reste très personnelle. En analyse biométrique, on ne se limite plus à sa forme : on prend aussi en compte la vitesse d'écriture, la pression exercée et le mouvement de la main, ce qui la rend bien plus fiable.

La voix : L'analyse vocale s'appuie sur des éléments uniques comme le ton, l'accent, l'intonation ou le rythme de parole. Ces caractéristiques dépendent à la fois de la physiologie et du comportement, rendant chaque voix distincte. C'est aussi l'une des rares méthodes qui permet une authentification à distance, tout en restant simple à utiliser.

2.4. Architecture d'un système biométrique :

Un système biométrique peut être représenté par quatre modules principaux [3], [4] :

1. **La capture ou l'acquisition :** Cette étape consiste à recueillir les données biométriques d'une personne à l'aide d'un dispositif adapté.

2. **L'extraction de caractéristiques :** À partir des données biométriques collectées lors de la capture, cette étape vise à isoler les éléments les plus significatifs.

3. **La correspondance :** Cette phase consiste à comparer les caractéristiques extraites avec celles déjà stockées dans la base de données durant l'enrôlement.

4. **La décision :** Ce module vérifie l'identité affirmée par un utilisateur ou détermine l'identité d'une personne basée sur le degré de similitude entre les caractéristiques extraites et les modèles stockés.

3. La Reconnaissance Faciale :

3.1. Introduction sur la reconnaissance faciale :

La reconnaissance faciale consiste à identifier une personne à partir d'une image de son visage. Elle est largement utilisée pour la sécurité (aéroports, smartphones…) car elle est sans contact, rapide et généralement bien acceptée..

Bien que la reconnaissance faciale soit populaire, elle a également des limites [5]:

Tableau 1: Avantages et inconvénients des systèmes de reconnaissance faciale

Avantages	Inconvénients
Bien accepté par le public	Technologie sensible à l'expression du visage
Pas de contact physique	Sensible aux changements : barbes,lunettes,chirurgie…
Peu coûteuse	Sensible à l'environnement : éclairage, position…
Plusieurs traits caractéristiques	Risques des cyberattaques

3.2. Les techniques de la reconnaissance faciale :

La reconnaissance faciale est une technologie qui permet d'identifier ou de vérifier l'identité d'une personne à partir de son visage. Elle fonctionne en extrayant les caractéristiques faciales d'une image,

puis en les comparant à celles stockées dans une base de données. Au fil du temps, différentes approches ont été développées pour améliorer la précision et la fiabilité de cette technologie. On distingue principalement les méthodes classiques, basées sur des calculs géométriques ou statistiques, et les approches modernes, qui reposent sur l'apprentissage profond et l'intelligence artificielle. Les réseaux de neurones, en particulier les CNN (Convolutional Neural Networks), sont aujourd'hui au cœur des systèmes les plus performants. Parmi les techniques utilisées, on trouve :

L'analyse géométrique : qui mesure les distances entre différents points du visage.

Les modèles de traits : qui identifient des caractéristiques uniques comme la forme des yeux.

Les réseaux neuronaux convolutifs (CNN) : qui permettent une analyse plus fine et plus robuste, même en présence de variations d'éclairage, d'angle ou d'expression.

Dans notre projet, nous avons utilisé un CNN pour la reconnaissance faciale. Ces réseaux sont conçus pour analyser des images et en extraire automatiquement les éléments les plus pertinents, sans intervention humaine. Une fois le visage détecté par la caméra, l'image est convertie en matrice de pixels. Chaque pixel devient une donnée que le réseau traite pour reconnaître les contours, les textures et les traits distinctifs. Un CNN est généralement structuré en trois types de couches :

- **Couches convolutives** : qui appliquent des filtres pour détecter des motifs (bords, formes, textures).
- **Couches de pooling :** qui réduisent la taille des données tout en conservant l'essentiel.
- **Couches entièrement connectées :** qui exploitent les caractéristiques extraites pour identifier ou classer le visage.

Grâce à cette architecture, le système apprend à reconnaître les visages de manière autonome. Dans notre cas, le CNN compare les visages capturés en temps réel à ceux déjà enregistrés dans la base de données.

3.3. Le processus de la reconnaissance faciale :

La reconnaissance faciale ne se limite pas au déverrouillage des smartphones. Si, dans ce cas, elle sert simplement à vérifier l'identité du propriétaire, elle peut aussi être utilisée à plus grande échelle. Des caméras peuvent capturer des visages dans des lieux publics et les comparer à des listes de personnes surveillées, parfois à partir d'images issues des réseaux sociaux. Bien que les usages varient, le principe reste le même : comparer un visage en temps réel à ceux enregistrés. [6].

➢ Détection du visage :

La caméra détecte et localise l'image d'un visage, seul ou dans une foule. L'image peut montrer la personne de face ou de profil.

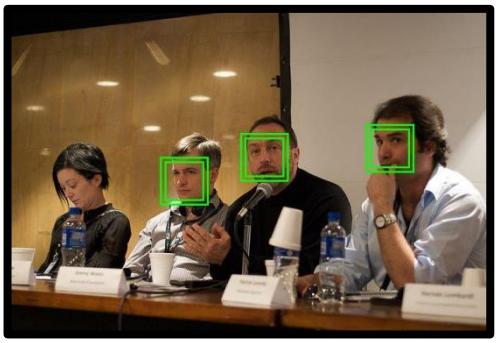

Figure 4: Exemple de détection du visage

➢ Analyse du visage :

La caméra capture une image du visage, généralement en 2D plutôt qu'en 3D. Ce choix s'explique par la facilité à comparer ces images avec les millions déjà présentes dans les bases de données. Ensuite, le logiciel entre en action. Il analyse avec précision chaque détail du visage : la distance entre les yeux, la courbure des sourcils, la hauteur du front, la forme des pommettes, des lèvres, des oreilles ou encore la ligne de la mâchoire. Toutes ces particularités sont mesurées et converties en points de repère. L'ensemble forme une empreinte faciale unique, propre à chaque individu.

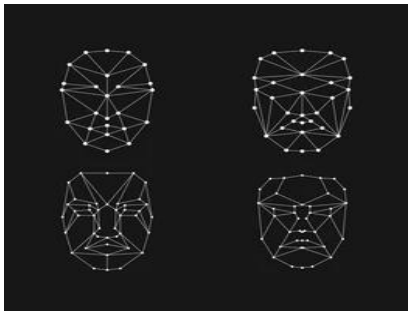

Figure 5: Analyse du visage

➢ Conversion de l'image en données :

Le visage, capturé par la caméra, se transforme en une série de données numériques. Chaque trait, chaque courbe est analysé puis converti en une formule mathématique complexe. Ce code unique, qu'on appelle "empreinte faciale", devient la signature biométrique. De la même manière que les empreintes digitales sont uniques, tout le monde a sa propre empreinte faciale.

➢ Trouver une correspondance :

L'empreinte faciale est ensuite comparée à une base de données avec d'autres visages connus.
La reconnaissance faciale est considérée comme la mesure biométrique la plus naturelle. Ce qui est logique au niveau intuitif, car nous nous reconnaissons nous mêmes et les autres en observant le visage, plutôt que les empreintes digitales ou l'iris.

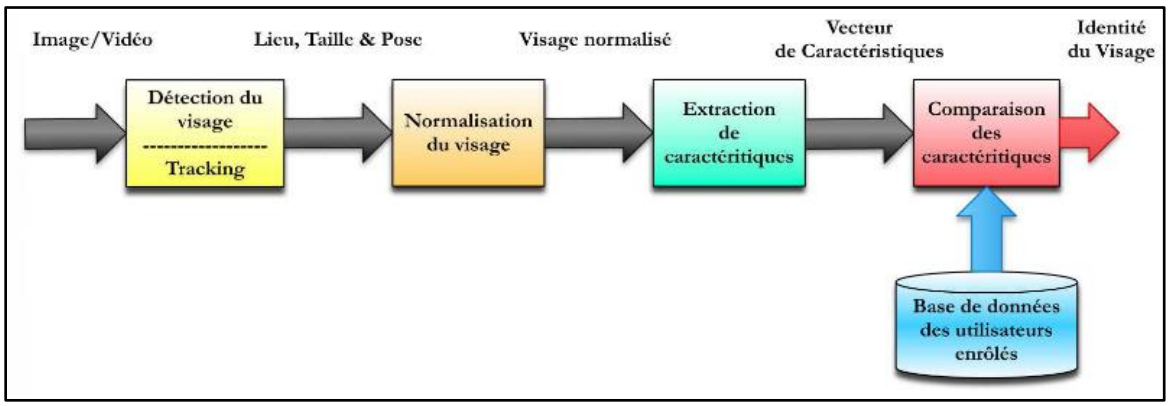

Figure 6: Schéma général d'un système de reconnaissance faciale

3.4. Reconnaissance faciale dans Raspberry Pi :

Il existe deux approches principales en reconnaissance faciale : sur image fixe ou sur vidéo, la seconde permettant parfois une modélisation 3D. Le Raspberry Pi est souvent privilégié pour ces applications grâce à ses performances, son faible coût, sa compacité et son efficacité énergétique, en comparaison avec les ordinateurs classiques. Il permet également de structurer le processus en étapes distinctes : capture d'images, entraînement du modèle et reconnaissance faciale, offrant une solution pratique et efficace pour ce type de projet.

4. Le but de ce projet :

Le but de ce projet de fin d'études est de créer un **système de reconnaissance faciale embarqué** avec un Raspberry Pi, qui permettra d'ouvrir une porte automatiquement lorsqu'un visage autorisé est détecté. On a choisi le Raspberry Pi parce qu'il est petit, pas cher et consomme peu d'énergie, ce qui le rend parfait pour un système embarqué. Ce projet montre que le Raspberry Pi est une solution pratique, efficace et économique pour des systèmes de sécurité comme le contrôle d'accès.

Ce projet démontre l'efficacité du Raspberry Pi dans la création de systèmes de sécurité abordables et fonctionnels. En combinant des composants simples comme le Raspberry Pi et une caméra, avec des algorithmes de reconnaissance faciale, on obtient un système de contrôle d'accès pratique et économique.

5. Conclusion du chapitre :

Ce premier chapitre nous a permis de mieux comprendre le contexte général de la reconnaissance faciale. En mettant en évidence les avantages de cette méthode d'identification, ainsi que les principes de fonctionnement des systèmes biométriques, nous avons posé les bases nécessaires à la conception de notre propre système. La reconnaissance faciale, combinée aux capacités du Raspberry Pi, constitue une solution accessible et adaptée aux besoins actuels en matière de sécurité.

Chapitre 2 :

Études et Analyse Fonctionnelle

1. Introduction :

Dans ce chapitre, nous allons présenter l'analyse fonctionnelle du système de contrôle d'accès par reconnaissance faciale. L'objectif est de définir les fonctions du système, ses interactions avec l'environnement, et les exigences qui guideront sa conception.

Nous commencerons par une analyse fonctionnelle qui identifie les besoins du système ainsi que les interactions entre le système et son environnement. Ensuite, nous définirons les exigences fonctionnelles et non fonctionnelles pour établir un cahier des charges précis et complet.

2. Cahier de charges :

2.1. Objectifs du projet :

Les objectifs de notre projet sont les suivants :

- Mettre en place un système de contrôle d'accès sécurisé basé sur la **reconnaissance faciale** à l'aide d'une **Raspberry Pi** 4
- Utiliser une caméra compatible avec la Raspberry Pi 4 pour capturer des images en temps réel et permettre la détection et la reconnaissance faciale des utilisateurs
- Afficher les informations relatives à l'état du système sur un écran LCD (par exemple, "Accès autorisé" ou "Accès refusé")
- Détection de présence à l'aide d'un capteur PIR sensor pour activer le système lorsque quelqu'un se trouve devant la porte
- Utiliser un servo moteur pour actionner l'ouverture et la fermeture de la porte
- Indiquer l'état du système avec des LED (vert pour accès autorisé, rouge pour accès refusé)

2.2. Exigences fonctionnelles :

2.2.1. Fonctions principales :

Reconnaissance faciale : Le système doit identifier un visage à partir d'une image capturée par la caméra. Si le visage correspond à une base de données d'images enregistrées, l'accès est autorisé

- **Accès autorisé** : Si la reconnaissance est réussie, le système doit :
  - Allumer la LED verte.
  - Afficher un message de bienvenue sur l'écran LCD.
  - Actionner le servomoteur pour ouvrir la porte.
- **Accès refusé** : Si la reconnaissance échoue, le système doit :
  - Allumer la LED rouge.
  - Afficher un message d'erreur sur l'écran LCD

2.2.2. Détection de présence :

Le ***capteur PIR*** doit détecter la présence d'une personne devant le système et déclencher le processus de reconnaissance faciale lorsque la personne se trouve dans la zone de détection.

2.2.3. Interface utilisateur :

Écran LCD : L'écran LCD servira à afficher des informations telles que :

- **"Bienvenue"** ou **"Accès refusé"** en fonction de la reconnaissance.

LEDs : Deux LEDs seront utilisées pour signaler l'état d'accès :

- **LED verte** : Signal d'accès autorisé.
- **LED rouge** : Signal d'accès refusé.

2.2.4. Actionnement de la porte :

Un servomoteur sera utilisé pour actionner une porte d'entrée, la déverrouiller ou l'ouvrir lorsque l'accès est autorisé.

2.3. Exigences non fonctionnelles :

- **Temps de réponse :**
  - Le système doit être capable de reconnaître un visage en moins de 5 secondes
  - Le capteur PIR doit détecter une personne en moins de 1 seconde et initier le processus de reconnaissance faciale.

- **Fiabilité :**

Le taux de reconnaissance faciale doit être d'au moins **95%** pour garantir une identification correcte et fiable.

- **Sécurité :**

Le système doit être conçu de manière à protéger la vie privée des utilisateurs et à sécuriser les données personnelles, notamment les images faciales.

2.4. Contraintes techniques :

2.4.1. Matériel :

- Raspberry Pi 4
- Caméra
- LEDs
- Servo-moteur
- Ecran LCD I2C
- PIR sensor

2.4.2. Logiciel :

- **Système d'exploitation :** Raspberry Pi Os

- **Language de programmation :** Le code sera écrit en Python pour une compatibilité optimale avec les bibliothèques disponibles pour le Raspberry Pi 4 telles que OpenCv, RPI.GPIO ou aussi numpy

2.5. Budget estimé :

- Le projet est d'un total de cahier de charge de :

Tableau 2 : Cahier de Charge

Nom du composant électronique	Quantité	Le Prix (en DHs)
Raspberry Pi 4	1	1150,00
Caméra Microsoft HD-3000	1	241,25
PIR Sensor	1	25,00
Servo motor SG90	1	30,00
Afficheur LCD I2C	1	50,00
LED	2	4,00
Arduino Uno	1	140,00
Carte micro-SD (32GB)	1	70,00
Total		**1710,25**

3. Recensement des fonctions de service :

3.1. Identification des fonctions de service :

Le diagramme **pieuvre** est utilisé pour identifier la fonction principale d'un système et ses fonctions complémentaires (services ou contraintes) en interaction avec les différents éléments de l'evironnement

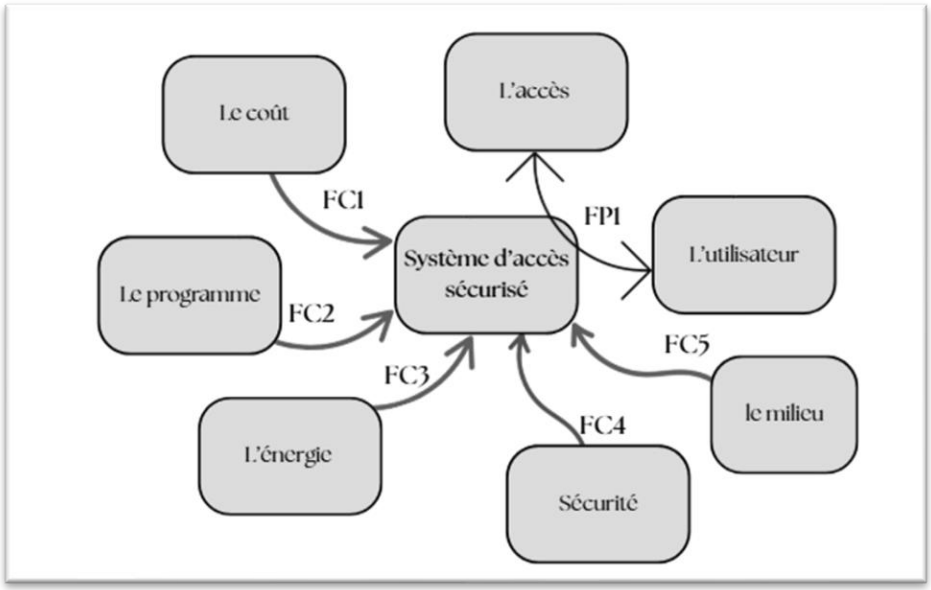

Figure 7: Diagramme de pieuvre

- **Fonction principale :**

FP1 : Limiter l'accès uniquement aux individus autorisés

- **Fonctions complémentaires :**

FC1 : Réduire les coûts

FC2 : Développer le système

FC3 : Optimiser la consommation énergétique

FC4 : Assurer la conformité aux normes de sécurité

FC5 : Protéger l'environnement immédiat

3.2. Analyse descendante :

Nous présenterons cette analyse à l'aide du diagramme SADT (Structured Analysis and Design Technique), qui permet de modéliser de manière hiérarchique et fonctionnelle les différentes étapes du système.

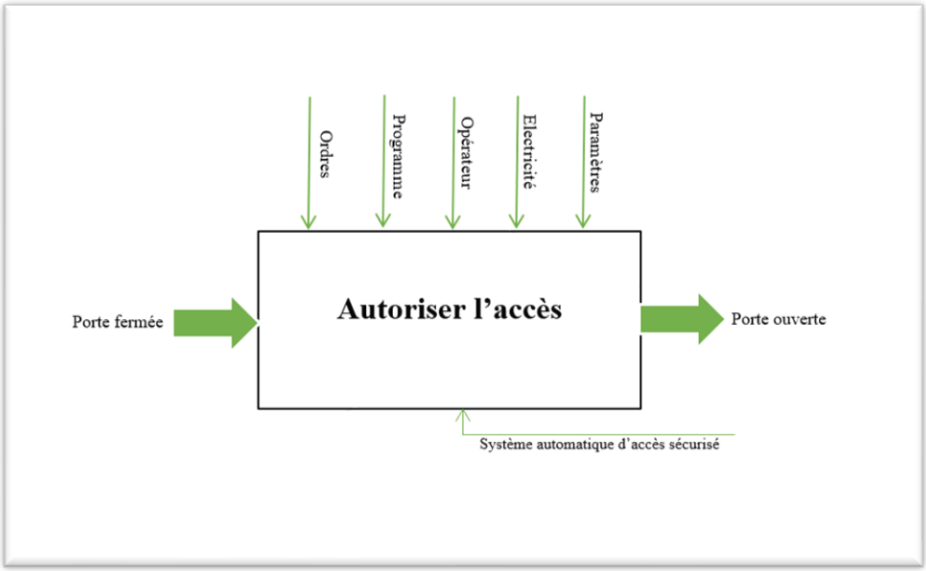

Figure 8: Diagramme SADT

3.3. Diagramme FAST :

Pour une meilleure compréhension des relations logiques entre les différentes fonctions du système, nous avons élaboré un diagramme FAST (Function Analysis System Technique). Ce diagramme met en évidence la fonction principale du projet et détaille les sous-fonctions nécessaires à sa réalisation, permettant ainsi de structurer l'analyse fonctionnelle

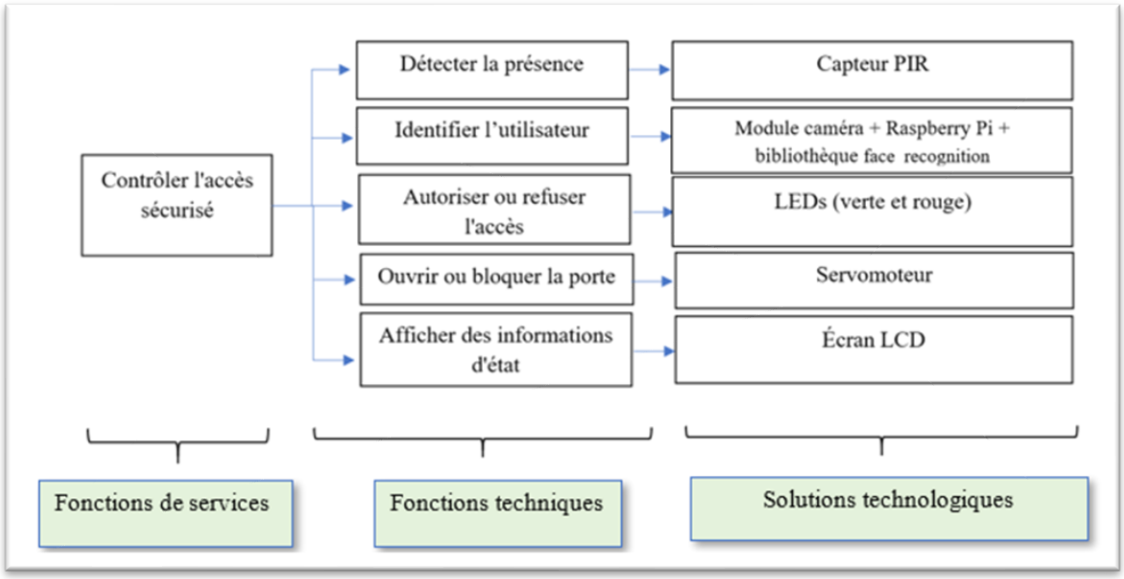

Figure 9 : Diagramme FAST

4. Conclusion du chapitre :

Dans ce chapitre, l'analyse fonctionnelle nous a permis d'identifier les fonctions essentielles garantissant la conception et le fonctionnement optimal du système d'accès sécurisé automatique. Cette démarche a également aidé à sélectionner les différents composants adaptés à la réalisation de notre projet.

Chapitre 3 :

Les Outils, Logiciels et Matériels

1. Introduction au chapitre :

Dans ce chapitre, nous présentons les différents outils, logiciels, et matériels utilisés pour concevoir et mettre en œuvre le système d'accès sécurisé basé sur la reconnaissance faciale. Ces éléments ont été choisis en fonction des exigences du projet et des fonctionnalités attendues, garantissant ainsi une performance optimale et une intégration harmonieuse des différentes composantes. Cette section mettra en lumière le rôle de chaque composant dans la réalisation du système.

2. Présentation des Matériels & Logiciels :

2.1. Matériel utilisé :

2.1.1. Raspberry Pi 4 :

Figure 10 : Raspberry Pi 4

- **Introduction :**

Le Raspberry Pi est un nano-ordinateur monocarte, low-cost, conçu pour favoriser l'apprentissage de la programmation et des projets électroniques. Malgré sa petite taille, il possède les fonctionnalités d'un PC classique et peut exécuter divers systèmes d'exploitation, notamment des distributions Linux

➢ Fonction et rôle dans le projet :

La Raspberry Pi 4 agit comme l'unité centrale du système, chargé d'exécuter le logiciel de reconnaissance faciale et de coordonner l'ensemble des composants. Dans notre projet elle traite les images capturées par la caméra pour effectuer la reconnaissance, contrôle l'affichage des messages sur l'écran LCD, gère l'activation des LEDs (indication d'état) et du servomoteur (verrouillage/déverrouillage) en fonction des résultats, tout en supervisant les signaux du capteur PIR pour déclencher le processus de détection. Son rôle principal est de gérer les interactions entre ces différents composants

➢ Plan de connexion :

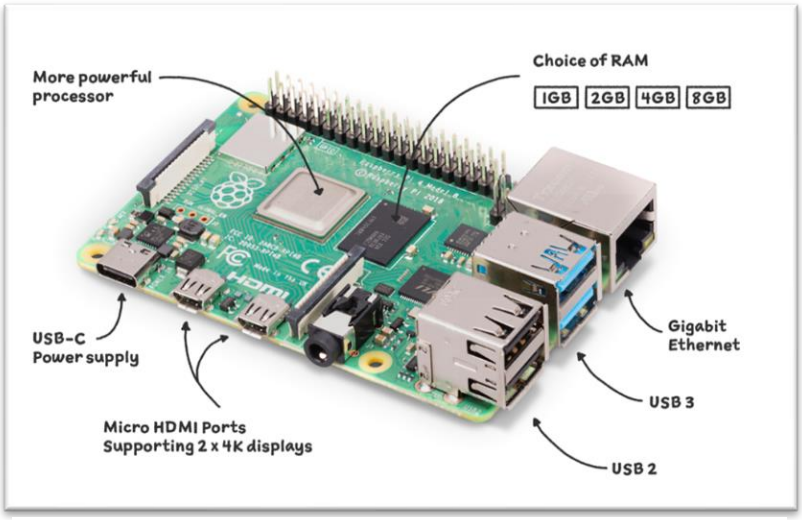

Figure 11: Les caractéristiques techniques de la Raspberry Pi 4

Tableau 3 : Les caractéristiques techniques de la Raspberry Pi 4 Modèle B [7]

Caractéristique	Connecteur/Interface	Détails Techniques
Processeur plus puissant	_	BCM2711 (quad-core ARM Cortex-A72 à 1.5 GHz)
Alimentation	Port USB-C	5V/3A recommandé (minimum 2.5A requis)
Port vidéo	2x Micro HDMI	Support 2 écrans 4K @ 60Hz simultanés
Mémoire RAM	Soudée sur carte	Options: 1GB, 2GB, 4GB ou 8GB LPDDR4
Réseau filaire	Port RJ45 Gigabit Ethernet	Débit jusqu'à 1 Gbit/s
Connectivité USB	- 2x USB 3.0 (bleu) - 2X USB 2.0 (noir)	- Débit jusqu'à 5 Gbit/s - Pour périphériques bas débit
Broches GPIO	Connecteur 40 broches	3.3V logique, UART, I2C, SPI, PWM disponibles

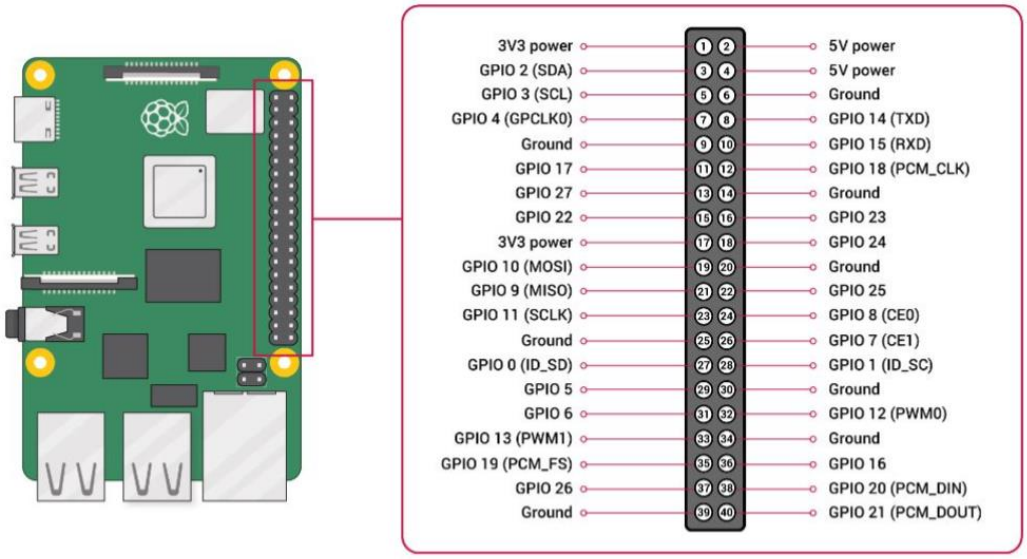

Figure 12: Le Port GPIO de la Raspberry Pi 4

2.1.2. Caméra Microsoft HD-3000 :

Figure 13: Camera Microsoft HD-3000

- **Introduction :**

La Microsoft HD-3000 est une webcam HD conçue pour offrir une qualité d'image claire et fluide, adaptée aux visioconférences, aux appels vidéo et au streaming. Dotée d'un capteur HD (720p) et d'un microphone intégré, elle est compatible avec les principaux systèmes d'exploitation (Windows, macOS) et plateformes (Zoom, Skype, etc.).

- **Fonction et rôle dans le projet :**

La caméra Microsoft HD-3000 est utilisée pour capturer des images du visage de la personne souhaitant accéder à la zone sécurisée, son rôle dans notre projet sera fournir les images nécessaires à l'algorithme de reconnaissance faciale pour déterminer si la personne est autorisée ou non à entrer.

2.1.3. Afficheur LCD avec module I2C :

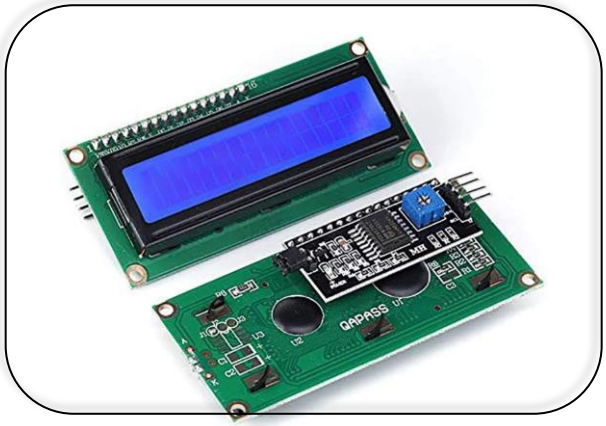

Figure 14: Afficheur LCD I2C

- **Introduction :**

L'**afficheur LCD (Liquid Crystal Display)** couplé à un **module d'interface I2C** est une solution pratique pour afficher des informations dans des projets électroniques (Arduino, Raspberry Pi, etc.). Grâce au protocole **I2C (Inter-Integrated Circuit)**, le câblage est simplifié, réduisant le nombre de connexions nécessaires tout en permettant un contrôle facile via un microcontrôleur

- **Fonction et rôle dans le projet :**

L'afficheur LCD I2C permet d'afficher des informations textuelles en utilisant l'interface I2C, son rôle dans ce projet sera d'afficher « Bienvenue {le nom de la personne} » si l'accès est autorisé ou afficher « Accès refusé » si l'accès est non permis

➢ **Plan de connexion :**

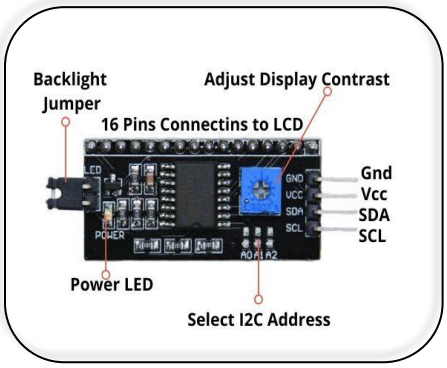

Figure 15: Module I2C pour LCD

Le module I2C se greffe directement sur les broches de l'afficheur LCD. Il comporte généralement 4 broches de connexion:

- **VCC** : Connecté à la source d'alimentation positive (+5V ou +3.3V) de la Raspberry Pi
- **GND** : Connecté à la masse de la Raspberry Pi
- **SDA (Serial Data)** : Connecté à la ligne de données série (SDA) de la Raspberry Pi (GPIO 2)
- **SCL (Serial Clock)** : Connecté à la ligne d'horloge série (SCL) de la Raspberry Pi (GPIO 3)

2.1.4. PIR sensor:

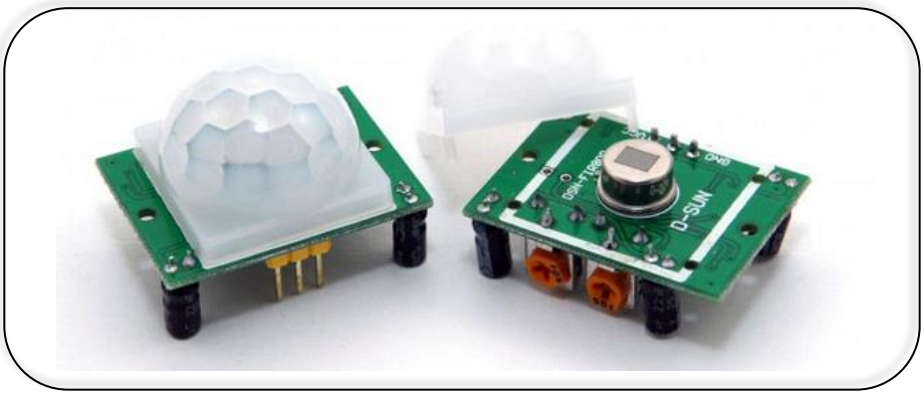

Figure 16: PIR sensor

➢ Introduction :

Le **capteur PIR** (*Passive Infrared Sensor* ou *Détecteur de Mouvement Infrarouge*) est un composant électronique largement utilisé pour détecter les mouvements en analysant les variations de rayonnement infrarouge émis par les corps chauds (humains, animaux, etc.). Simple d'utilisation et peu coûteux, il est idéal pour des applications domotiques, de sécurité ou d'automatisation.

➢ Fonction et rôle dans le projet :

Il détecte la présence d'un individu en mesurant les changements de rayonnement infrarouge dans le champ de vision. Dans notre projet lorsqu'un mouvement sera détecté, il envoie un signal permettant d'activer la caméra.

➢ Plan de connexion :

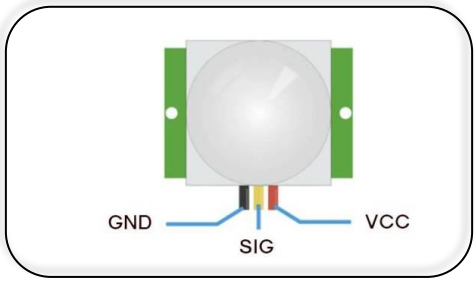

Figure 17: les pins du PIR sensor

. Voici le plan de connexion typique pour un capteur PIR

- **VCC** : Connecté à la source d'alimentation positive (+5V ou +3.3V) de la Raspberry Pi
- **GND** : Connecté à une masse de la Raspberry Pi
- **OUT** : Sortie (connecté au GPIO 4 de la Raspberry Pi)

2.1.5. Diode électroluminescente (LED) :

Figure 18: Diode électroluminescente (LED)

➢ Introduction :

Les **LED** (*Light Emitting Diodes ou Diodes Électroluminescentes*) sont des composants électroniques semi-conducteurs qui émettent de la lumière lorsqu'un courant électrique les traverse. Elles sont omniprésentes dans nos vies, des indicateurs d'état sur les appareils électroniques à l'éclairage domestique et urbain, en passant par les écrans modernes.

➢ Fonction et rôle dans le projet :

Les LEDs sont utilisées comme indicateurs visuels pour représenter l'état du système. Dans notre projet on utilisera deux LEDs : une LED verte qui s'allumera en cas d'accès autorisé et une LED rouge en cas d'accès refusé

➢ Plan de connexion :

LED verte : **Anode de la LED** Connectée à une broche GPIO de la Raspberry Pi (GPIO 22) et la **cathode de la LED** connectée à la **terre (GND)** via une résistance de 220Ω pour limiter le courant

LED rouge : **Anode de la LED** : Connectée à une broche GPIO de la Raspberry Pi (GPIO 27) et la **cathode de la LED** Connectée à la **terre (GND)** via une résistance de 220Ω pour limiter le courant

2.1.6. Servo moteur (SG90) :

➢ Introduction :

Le **servomoteur SG90** est un actionneur compact et populaire dans le domaine de la robotique et des projets électroniques. Contrairement aux moteurs classiques, il offre un contrôle précis de position angulaire, généralement sur 180 degrés, grâce à son système intégré de rétroaction.

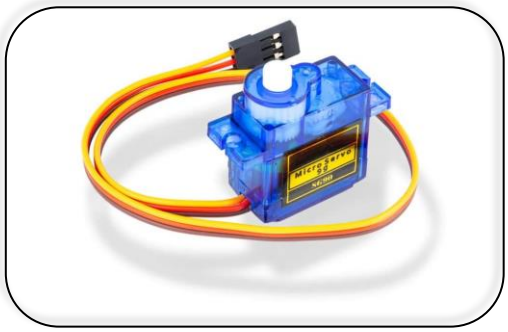

Figure 19: Servo-moteur (SG90)

➢ Fonction et rôle dans le projet :

Dans notre projet, le servomoteur SG90 est chargé d'actionner mécaniquement l'ouverture et la fermeture d'une petite porte. Positionné à l'entrée du système, il joue le rôle de barrière physique contrôlée électroniquement

Pour assurer une alimentation stable et mobile du servomoteur SG90 dans le cadre de ce projet, nous avons opté pour l'utilisation d'un ***Arduino Uno*** pour éviter la surcharge de la Raspberry Pi

Figure 20: Arduino Uno

➢ Plan de connexion :

Le servomoteur SG90 est alimenté directement par un Arduino, qui fournit une alimentation stable via ses broches de sortie : le fil rouge (+5V) et le fil noir (GND) du servo sont connectés respectivement aux broches 5V et GND de l'Arduino, tandis que le fil orange (signal PWM) est relié à une broche de commande de l'a Raspberry Pi pour le contrôle du signal PWM.

2.2. Architecture du système :

Voici l'architecture globale de notre système, illustrant les différentes composantes et leur interaction :

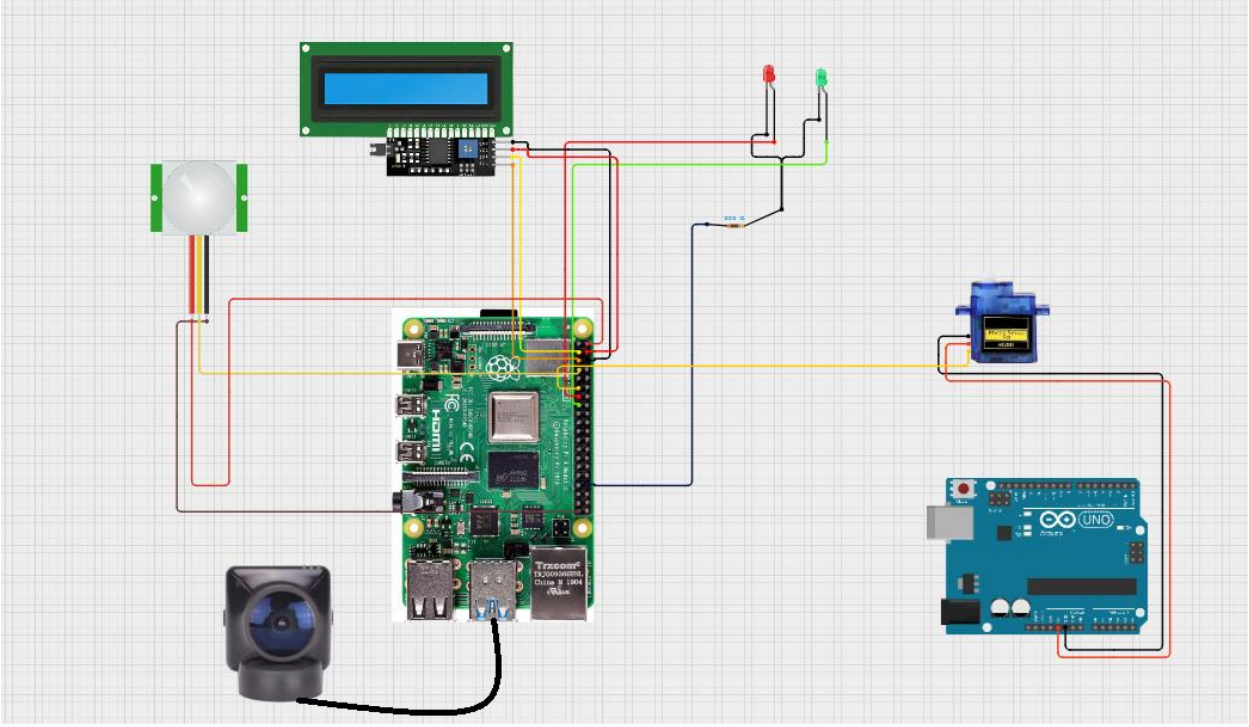

Figure 21: Architecture du système

2.3. Outils Logiciels et Dépendances :

2.3.1. CATIA V5R20 :

CATIA V5R20 est un logiciel professionnel de conception assistée par ordinateur (CAO) développé par Dassault Systèmes, particulièrement utilisé pour la modélisation paramétrique de pièces mécaniques et de structures complexes**.** Doté d'outils puissants comme le Sketcher pour les plans 2D et le Part Design pour les volumes 3D, il permet une conception précise et modifiable à tout moment grâce à son approche paramétrique. **Dans le cadre de ce projet, CATIA V5R20 a été indispensable pour modéliser la maison**, depuis la création des murs et ouvertures jusqu'à l'assemblage final des différents éléments architecturaux, offrant ainsi une vision claire et détaillée du résultat avant toute réalisation physique

Figure 22: CATIA V5R20

2.3.2. Python :

Python est un langage de programmation interprété, multiplateforme et open source, réputé pour sa syntaxe claire et intuitive qui en fait un outil idéal pour le prototypage rapide et le développement d'applications complexes. Conçu pour optimiser la productivité des développeurs, il combine une approche orientée objet avec des fonctionnalités de script, tout en offrant une vaste bibliothèque standard et un riche écosystème de modules tiers.

Figure 23: Python

2.3.3. PuTTY :

PuTTY est un logiciel gratuit et open source qui permet d'établir des connexions distantes sécurisées entre un ordinateur local et un serveur ou un appareil distant, principalement via des protocoles tels que SSH (Secure Shell), Telnet, et Serial. Il est principalement utilisé pour gérer des systèmes distants en ligne de commande et convient particulièrement aux développeurs, administrateurs système, et utilisateurs de serveurs Linux ou Raspberry Pi

Figure 24: PuTTY

2.3.4. RealVNC Viewer :

RealVNC Viewer est un logiciel de contrôle à distance qui permet d'accéder à un autre ordinateur ou appareil via le protocole **VNC (Virtual Network Computing)**. Il permet à l'utilisateur de voir et d'interagir avec l'interface graphique de l'appareil distant comme s'il était assis devant celui-ci. RealVNC Viewer est particulièrement connu pour sa sécurité renforcée grâce au cryptage des connexions et est compatible avec divers systèmes d'exploitation comme Windows, macOS, Linux, ainsi que les appareils mobiles sous Android ou iOS.

Figure 25: RealVNC Viewer

2.3.5. OpenCV :

OpenCV est une bibliothèque open-source de vision par ordinateur offrant des fonctions avancées pour le traitement d'images et la détection d'objets. Elle permet notamment la capture vidéo, la détection de visages et la reconnaissance faciale grâce à des algorithmes optimisés

Figure 26:OpenCV

2.3.6. NumPy:

NumPy est la bibliothèque fondamentale pour le calcul scientifique en Python. Elle fournit des tableaux multidimensionnels performants et des fonctions mathématiques essentielles pour le traitement du signal et des images, notamment avec OpenCV

Figure 27:NumPy

2.3.7. smbus2:

smbus2 est une implémentation Python du protocole SMBus (System Management Bus), utilisée pour communiquer avec des périphériques I2C comme les afficheurs LCD. Elle simplifie les opérations de lecture/écriture sur le bus I2C

2.3.8. RPI.GPIO :

RPI.GPIO est la bibliothèque Python officielle pour contrôler les broches GPIO de la Raspberry Pi. Elle permet de gérer facilement les entrées/sorties numériques, le PWM pour les servomoteurs, et d'interagir avec des composants électroniques comme les LEDs et capteur

2.3.9. face_ recognition :

face_recognition est une bibliothèque Python construite sur OpenCV et dlib, spécialisée dans la reconnaissance faciale. Elle permet de détecter, comparer et identifier des visages à partir d'images avec seulement quelques lignes de code

3. Conclusion du chapitre :

En conclusion, le choix des outils, logiciels et matériels détaillés dans ce chapitre a été réalisé en adéquation avec les besoins et les objectifs du projet. Chaque élément joue un rôle spécifique dans la conception, le développement et le fonctionnement du système d'accès sécurisé. Cette sélection rigoureuse garantit une intégration cohérente et une performance optimale, posant ainsi les bases pour la mise en œuvre efficace du système.

Chapitre 4 :

Conception et Réalisation

1. Introduction au chapitre :

Ce chapitre présente la phase concrète de réalisation de notre système de contrôle d'accès par reconnaissance faciale, structurée autour de trois piliers fondamentaux. La conception logicielle détaille l'architecture algorithmique, avec l'implémentation des traitements d'images et la gestion intelligente des périphériques. La conception mécanique expose les solutions retenues pour l'intégration physique des composants, en mettant l'accent sur l'optimisation des mouvements du moteur et la robustesse de l'assemblage. Enfin, la partie développement du code révèle la concrétisation programmatique de ces choix, depuis l'acquisition vidéo jusqu'à la prise de décision en temps réel.

2. Conception Mécanique :

2.1. Objectifs principaux :

La conception mécanique de notre système de contrôle d'accès repose sur **la modélisation 3D de l'environnement d'installation** (maison, porte, emplacements des composants) pour :

- Visualiser l'intégration des éléments (caméra, moteur, LEDs)
- Valider les dimensions et l'accessibilité avant le montage réel
- Servir de support pour les ajustements pratiques

2.2. Outil utilisé :

Nous avons utilisé le logiciel CATIA V5 R20 ; un logiciel de conception asistée par ordinateur pour :

- Modéliser la structure de la maison (murs,porte,zone d'entrée)
- Positionner virtuellement les composants (caméra,LCD,moteur)

2.3. Apports de la modélisation :

La modélisation 3D sous CATIA nous a permis de visualiser et valider l'intégration des composants avant leur installation réelle. Grâce à cette approche, nous avons pu optimiser le positionnement de la caméra, du moteur et des autres éléments, tout en évitant les problèmes d'encombrement. Ce travail préparatoire a facilité le montage final et assuré une bonne cohérence entre les parties électroniques et mécaniques du système

2.4. Résultat obtenu :

La figure ci-dessous présente notre modélisation 3D de la chambre réalisée sous CATIA V5 R20, offrant une vision claire de l'intégration des différents composants du système. Cette représentation visuelle permet d'apprécier l'implantation cohérente de la caméra, du moteur et des autres éléments dans leur environnement physique.

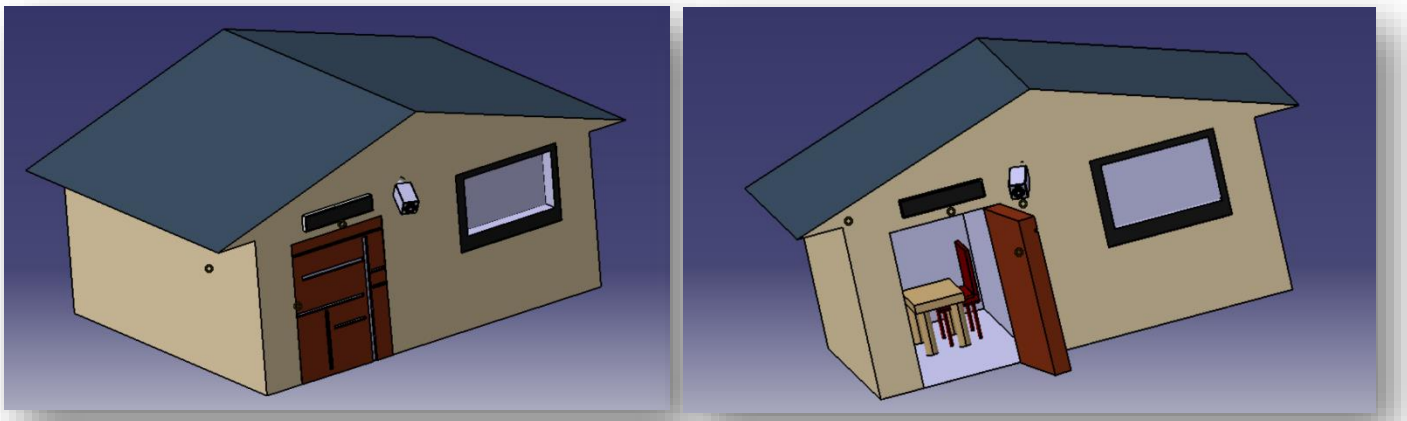

Figure 28 : Chambre modélisée sous CATIAV5R20

3. Organigramme de fonctionnement :

Workflow de décision du système

Ce diagramme illustre la logique opérationnelle complète de notre système de contrôle d'accès, depuis l'initialisation jusqu'aux actions différenciées selon la reconnaissance faciale:

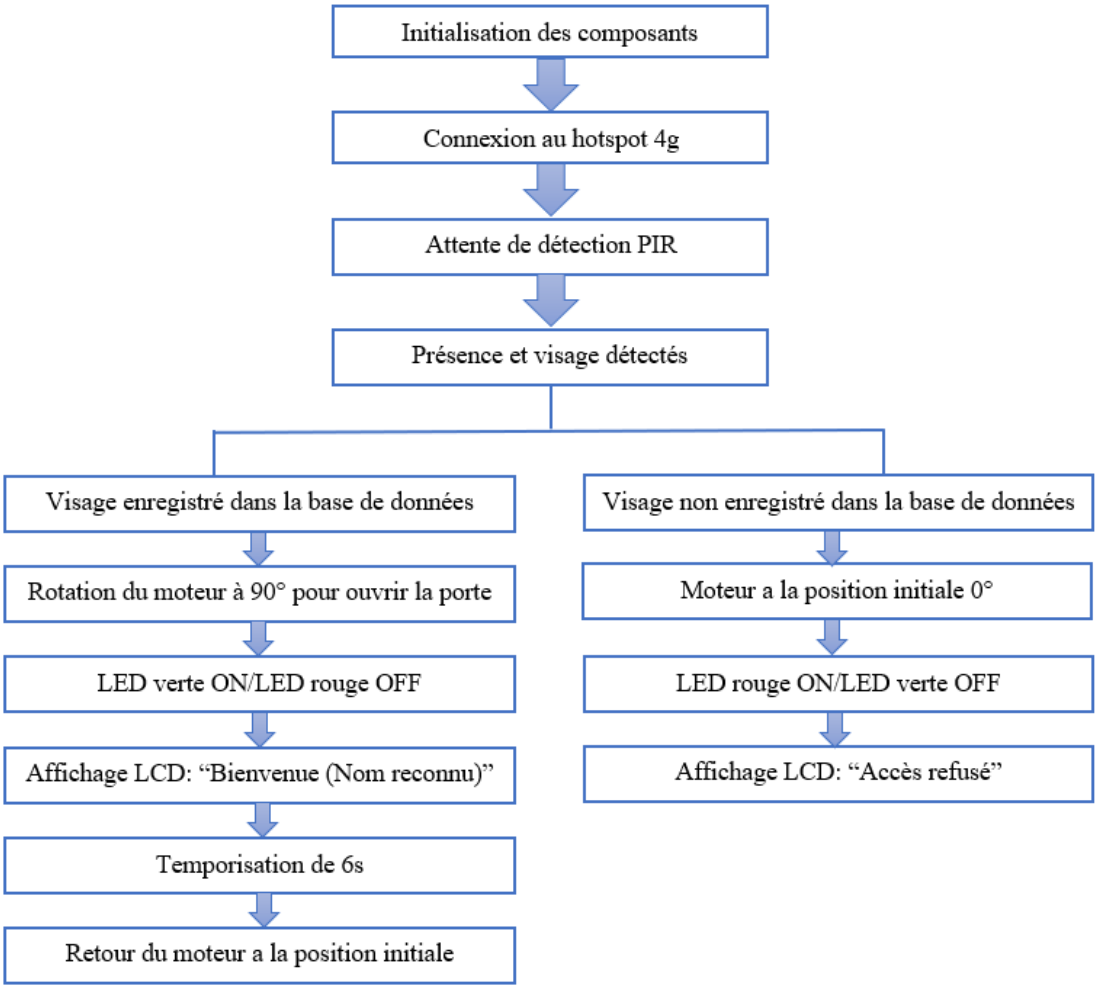

Figure 29: Logique de fonctionnement du système

4. Conception Logicielle :

➢ Importation des bibliothèques :

```
import face_recognition
import cv2
import numpy as np
import os
import RPi.GPIO as GPIO
import time
import smbus2
from smbus2 import SMBus
```

- **face_recognition** : Utilisée pour la détection et la reconnaissance des visages.
- **cv2** : Fournit des outils pour le traitement d'image et la manipulation vidéo.
- **numpy** : Permet la manipulation efficace de données sous forme de matrices.
- **RPi.GPIO** : Gère les interfaces GPIO pour interagir avec les composants matériels.
- **smbus2** : Offre des outils pour communiquer avec des périphériques I2C comme un écran LCD

➢ Configuration des composants :

Chaque composant matériel utilisé dans le projet a été configuré :

- **Configuration des broches GPIO :**

```
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
```

- **Capteur PIR :**

```
PIR_PIN = 4
GPIO.setup(PIR_PIN, GPIO.IN)
```

- **Servo-moteur :**

```
SERVO_PIN = 17
GPIO.setup(SERVO_PIN, GPIO.OUT)
pwm = GPIO.PWM(SERVO_PIN, 50)  # 50Hz = 20ms
pwm.start(0)  # Position initiale à 0°
```

- **LEDs :**

```
LED_VERTE = 22  # GPIO22 pour LED verte (accès autorisé)
LED_ROUGE = 27  # GPIO27 pour LED rouge (accès refusé)
GPIO.setup(LED_VERTE, GPIO.OUT)
GPIO.setup(LED_ROUGE, GPIO.OUT)
```

- **LCD I2C :**

```
# Configuration LCD 16x2 I2C
I2C_ADDR = 0x27  # Adresse I2C typique pour LCD
LCD_WIDTH = 16  # Largeur du LCD en caractères
LCD_CHR = 1
LCD_CMD = 0
LCD_LINE_1 = 0x80  # Adresse 1ère ligne
LCD_LINE_2 = 0xC0  # Adresse 2ème ligne
LCD_BACKLIGHT = 0x08  # Rétroéclairage activé

# Timing
E_PULSE = 0.0005
E_DELAY = 0.0005

# Initialisation du bus I2C
try:
    bus = smbus2.SMBus(1)  # Raspberry Pi version 2 utilise
except:
    bus = smbus2.SMBus(0)  # Raspberry Pi version 1 utilise
```

➢ Initialisation du LCD :

Les fonctions suivantes ont été implémentées pour initialiser et contrôler l'écran LCD :

- **lcd_init() :** Initialise l'écran
- **lcd_byte(bits, mode)** : Envoie des commandes ou des données à l'écran
- **lcd_string(message, line)** : Affiche une chaîne de caractères sur une ligne spécifique.

```python
def lcd_init():  1 usage
    lcd_byte( bits: 0x33, LCD_CMD)
    lcd_byte( bits: 0x32, LCD_CMD)
    lcd_byte( bits: 0x06, LCD_CMD)
    lcd_byte( bits: 0x0C, LCD_CMD)
    lcd_byte( bits: 0x28, LCD_CMD)
    lcd_byte( bits: 0x01, LCD_CMD)
    time.sleep(E_DELAY)
```

```python
def lcd_byte(bits, mode):  9 usages
    bits_high = mode | (bits & 0xF0) | LCD_BACKLIGHT
    bits_low = mode | ((bits << 4) & 0xF0) | LCD_BACKLIGHT

    bus.write_byte(I2C_ADDR, bits_high)
    lcd_toggle_enable(bits_high)

    bus.write_byte(I2C_ADDR, bits_low)
    lcd_toggle_enable(bits_low)
```

```python
def lcd_string(message, line):  11 usages
    """Affiche une chaîne sur le LCD"""
    message = message.ljust(LCD_WIDTH, " ")
    lcd_byte(line, LCD_CMD)
    for i in range(LCD_WIDTH):
        lcd_byte(ord(message[i]), LCD_CHR)
```

- Commandes du Servo-moteur :

Contrôle de l'angle du servo via un cycle de travail PWM

```python
def set_servo_angle(angle):  3 usages
    duty = angle / 18 + 2  # Conversion angle -> duty cycle
    GPIO.output(SERVO_PIN, True)
    pwm.ChangeDutyCycle(duty)
    time.sleep(0.5)  # Temps pour atteindre la position
    GPIO.output(SERVO_PIN, False)
    pwm.ChangeDutyCycle(0)  # Évite les vibrations
```

- **Contrôle des LEDs :**

Allume ou éteint les LEDs en fonction de l'accès autorisé ou refusé

```
def control_leds(acces_autorise):  3 usages
    if acces_autorise:
        GPIO.output(LED_VERTE, GPIO.HIGH)
        GPIO.output(LED_ROUGE, GPIO.LOW)
    else:
        GPIO.output(LED_VERTE, GPIO.LOW)
        GPIO.output(LED_ROUGE, GPIO.HIGH)
```

- **Chargement des visages connus :**

```
KNOWN_FACES_DIR = "known_faces"
print("Chargement des visages connus...")
known_face_encodings = []
known_face_names = []
```

- **Boucle principale**

```
try:
    while True:
        # Vérifier l'état du PIR
        pir_state = GPIO.input(PIR_PIN)
```

▪▪▪▪

```
if cv2.waitKey(1) & 0xFF == ord('q'):
    break
```

5. Réalisation du prototype physique :

Les figures ci-dessous présentent le prototype physique du système d'accès par reconnaissance faciale, illustrant les différentes étapes de fonctionnement :

Détection de présence à l'aide d'un capteur PIR. Lorsqu'une personne s'approche du système, ce capteur déclenche automatiquement la caméra pour lancer le processus de reconnaissance faciale

Figure 30: Détection de présence

Le système identifie la personne comme autorisée. Il affiche le message « Bienvenue », ouvre la porte et allume LED verte en guise de validation visuelle

Figure 31: Premier cas – Visage reconnu

Le système ne reconnaît pas la personne. Il affiche un message de refus, n'ouvre pas la porte, et active la LED rouge pour indiquer un accès non autorisé

Figure 32: Deuxième cas – Visage non reconnu

6. Conclusion du chapitre :

En conclusion, cette phase de réalisation a permis de concrétiser le système de contrôle d'accès par reconnaissance faciale en alliant harmonieusement les aspects logiciels, mécaniques et programmatiques. Chaque étape a contribué à transformer les concepts initiaux en un système fonctionnel, alliant précision dans le traitement des images, fiabilité des mouvements mécaniques et efficacité des algorithmes développés. Cette synthèse technique marque une étape décisive dans l'avancement du projet, posant les bases pour son évaluation et son optimisation

Annexe

```
import face_recognition
import cv2
import numpy as np
import os
import RPi.GPIO as GPIO
import time
import smbus2
from smbus2 import SMBus

# Configuration des GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

# Configuration du PIR
PIR_PIN = 4
GPIO.setup(PIR_PIN, GPIO.IN)

# Configuration du servo-moteur
SERVO_PIN = 17
GPIO.setup(SERVO_PIN, GPIO.OUT)
pwm = GPIO.PWM(SERVO_PIN, 50)  # 50Hz = 20ms
pwm.start(0)  # Position initiale à 0°

# Configuration des LEDs
LED_VERTE = 22  # GPIO22 pour LED verte (accès autorisé)
LED_ROUGE = 27  # GPIO27 pour LED rouge (accès refusé)
GPIO.setup(LED_VERTE, GPIO.OUT)
GPIO.setup(LED_ROUGE, GPIO.OUT)

# Configuration LCD 16x2 I2C
I2C_ADDR = 0x27  # Adresse I2C typique pour LCD
LCD_WIDTH = 16  # Largeur du LCD en caractères
LCD_CHR = 1
LCD_CMD = 0
LCD_LINE_1 = 0x80  # Adresse 1ère ligne
LCD_LINE_2 = 0xC0  # Adresse 2ème ligne
LCD_BACKLIGHT = 0x08  # Rétroéclairage activé

# Timing
E_PULSE = 0.0005
E_DELAY = 0.0005

# Initialisation du bus I2C
try:
    bus = smbus2.SMBus(1)  # Raspberry Pi version 2 utilise 1
except:
    bus = smbus2.SMBus(0)  # Raspberry Pi version 1 utilise 0
```

```python
def lcd_init():  1 usage
    lcd_byte( bits: 0x33, LCD_CMD)
    lcd_byte( bits: 0x32, LCD_CMD)
    lcd_byte( bits: 0x06, LCD_CMD)
    lcd_byte( bits: 0x0C, LCD_CMD)
    lcd_byte( bits: 0x28, LCD_CMD)
    lcd_byte( bits: 0x01, LCD_CMD)
    time.sleep(E_DELAY)


def lcd_byte(bits, mode):  9 usages
    bits_high = mode | (bits & 0xF0) | LCD_BACKLIGHT
    bits_low = mode | ((bits << 4) & 0xF0) | LCD_BACKLIGHT

    bus.write_byte(I2C_ADDR, bits_high)
    lcd_toggle_enable(bits_high)

    bus.write_byte(I2C_ADDR, bits_low)
    lcd_toggle_enable(bits_low)


def lcd_toggle_enable(bits):  2 usages
    """Toggle enable"""
    time.sleep(E_DELAY)
    bus.write_byte(I2C_ADDR, (bits | 0x04))
    time.sleep(E_PULSE)
    bus.write_byte(I2C_ADDR, (bits & ~0x04))
    time.sleep(E_DELAY)


def lcd_string(message, line):  11 usages
    """Affiche une chaîne sur le LCD"""
    message = message.ljust(LCD_WIDTH, " ")
    lcd_byte(line, LCD_CMD)
    for i in range(LCD_WIDTH):
        lcd_byte(ord(message[i]), LCD_CHR)

def set_servo_angle(angle):  3 usages
    duty = angle / 18 + 2  # Conversion angle -> duty cycle
    GPIO.output(SERVO_PIN, True)
    pwm.ChangeDutyCycle(duty)
    time.sleep(0.5)  # Temps pour atteindre la position
    GPIO.output(SERVO_PIN, False)
    pwm.ChangeDutyCycle(0)  # Évite les vibrations
```

```
def control_leds(acces_autorise):  3 usages
    if acces_autorise:
        GPIO.output(LED_VERTE, GPIO.HIGH)
        GPIO.output(LED_ROUGE, GPIO.LOW)
    else:
        GPIO.output(LED_VERTE, GPIO.LOW)
        GPIO.output(LED_ROUGE, GPIO.HIGH)


# Initialisation LCD
lcd_init()
lcd_string( message: "Systeme pret", LCD_LINE_1)
lcd_string( message: "En attente...", LCD_LINE_2)


KNOWN_FACES_DIR = "known_faces"
print("Chargement des visages connus...")
known_face_encodings = []
known_face_names = []

for name in os.listdir(KNOWN_FACES_DIR):
    image_path = os.path.join(KNOWN_FACES_DIR, name)
    try:
        image = face_recognition.load_image_file(image_path)
        encodings = face_recognition.face_encodings(image)

        if len(encodings) > 0:
            known_face_encodings.append(encodings[0])
            known_face_names.append(os.path.splitext(name)[0])
            print(f"Visage chargé: {name}")
        else:
            print(f"Aucun visage détecté dans {name} - Vérifiez la qualité de l'image")
    except Exception as e:
        print(f"Erreur avec {name}: {str(e)}")

print(f"{len(known_face_names)} visages valides chargés")

if not known_face_encodings:
    print("Aucun visage valide trouvé. Vérifiez votre dossier known_faces.")
    exit()

# Initialiser la caméra
video_capture = cv2.VideoCapture(0)
video_capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
video_capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

# Variables pour la détection
process_this_frame = True
servo_active = False
face_detection_active = False
pir_timeout = 5
last_pir_detection = 0
```

```
print("En attente de détection PIR...")

try:
    while True:
        # Vérifier l'état du PIR
        pir_state = GPIO.input(PIR_PIN)

        if pir_state:
            last_pir_detection = time.time()
            if not face_detection_active:
                print("Personne détectée par PIR - Activation reconnaissance faciale")
                face_detection_active = True
                lcd_string( message: "Detection en", LCD_LINE_1)
                lcd_string( message: "cours...", LCD_LINE_2)

        # Désactiver la détection faciale si timeout PIR
        if face_detection_active and (time.time() - last_pir_detection > pir_timeout):
            print("Timeout PIR - Retour à l'état initial")
            face_detection_active = False
            if servo_active:
                set_servo_angle(0)  # Fermer la porte
                servo_active = False
            control_leds(False)
            lcd_string( message: "Systeme pret", LCD_LINE_1)
            lcd_string( message: "En attente...", LCD_LINE_2)
            continue

        # Si personne détectée par PIR, faire la reconnaissance faciale
        if face_detection_active:
            ret, frame = video_capture.read()
            if not ret:
                print("Erreur de capture vidéo")
                break

            # Redimensionner et convertir l'image
            small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
            rgb_small_frame = cv2.cvtColor(small_frame, cv2.COLOR_BGR2RGB)

            if process_this_frame:
                # Détection des visages
                face_locations = face_recognition.face_locations(rgb_small_frame)
                name = "Inconnu"  # Par défaut, accès refusé
```

```
            if face_locations:
                # Encodage des visages détectés
                face_encodings = face_recognition.face_encodings(
                    rgb_small_frame,
                    face_locations,
                    num_jitters=1
                )

                for face_encoding in face_encodings:
                    # Comparaison avec les visages connus
                    matches = face_recognition.compare_faces(known_face_encodings, face_encoding, tolerance=0.6)

                    # Trouver la meilleure correspondance
                    face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
                    best_match_index = np.argmin(face_distances)

                    if matches[best_match_index]:
                        name = known_face_names[best_match_index]
                        if not servo_active:
                            print(f"Personne autorisée détectée: {name}")
                            set_servo_angle(90)  # Rotation à 90°
                            control_leds(True)  # Allumer LED verte
                            lcd_string( message: "Bienvenue", LCD_LINE_1)
                            lcd_string(name[:16], LCD_LINE_2)  # Tronquer si trop long
                            servo_active = True
                    else:
                        print("Personne non autorisée détectée")
                        control_leds(False)  # Allumer LED rouge
                        lcd_string( message: "Acces refuse", LCD_LINE_1)
                        lcd_string( message: "Non autorise", LCD_LINE_2)

        process_this_frame = not process_this_frame

        # Affichage des résultats
        for (top, right, bottom, left), display_name in zip(face_locations, [name] * len(face_locations)):
            top *= 4;
            right *= 4;
            bottom *= 4;
            left *= 4
            color = (0, 255, 0) if display_name != "Inconnu" else (0, 0, 255)
            cv2.rectangle(frame, (left, top), (right, bottom), color, 2)
            cv2.rectangle(frame, (left, bottom - 35), (right, bottom), color, cv2.FILLED)
            cv2.putText(frame, display_name, (left + 6, bottom - 6), cv2.FONT_HERSHEY_DUPLEX, 0.8, (255, 255, 255),
                        1)

        cv2.imshow('Reconnaissance Faciale', frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

finally:
    # Nettoyage à la fin du programme
    set_servo_angle(0)  # Remise à zéro
    pwm.stop()
    GPIO.output(LED_VERTE, GPIO.LOW)
    GPIO.output(LED_ROUGE, GPIO.LOW)
    lcd_byte( bits: 0x01, LCD_CMD)  # Effacer l'écran
    lcd_string( message: "Systeme arrete", LCD_LINE_1)
    GPIO.cleanup()
    video_capture.release()
    cv2.destroyAllWindows()
    print("Programme arrêté")
```

Conclusion :

À travers ce projet de fin d'études, nous avons pu concevoir et développer un système de reconnaissance faciale embarqué permettant l'ouverture automatique d'une porte. Ce projet nous a permis de mettre en pratique les connaissances acquises durant notre formation, notamment en électronique, en programmation embarquée, et en intelligence artificielle.

En intégrant des technologies comme le Raspberry Pi, l'apprentissage automatique et les composants électroniques (caméra, servo-moteur, écran LCD, LEDs), nous avons pu réaliser un système fonctionnel, autonome et à faible coût, répondant à un besoin réel en matière de sécurité d'accès.

Au-delà de l'aspect technique, ce projet nous a appris à travailler en équipe, à gérer un projet de manière structurée et à surmonter les imprévus, aussi bien matériels que logiciels. Il a également renforcé notre intérêt pour les technologies embarquées et l'intelligence artificielle, et ouvre la voie à de nombreuses perspectives d'amélioration.

En somme, ce projet a été une expérience formatrice à tous les niveaux. Il nous a permis de mesurer l'importance de la collaboration dans la réussite d'un travail technique. Nous clôturons cette étape fiers du chemin parcouru, et motivés à poursuivre notre évolution dans le monde de l'innovation technologique

Bibliographie / Webographie

[1] : Mémoire de fin d'étude pour l'obtention du diplôme Master de Recherche en Informatique, "Cryptosystème biométrique pour la protection du template d'empreinte digitale".

[2] : RAMANAMBE Zo Ismaël, "APPLICATION DE GESTION DE PRESENCE PAR RECONNAISSANCE FACIALE", MEMOIRE en vue de l'obtention du DIPLÔME de Licence

[3] :Benoît Vibert, "Contributions à l'évaluation de systèmes biométriques embarqués," PhD Thesis, Normandie, 2017.

[4] : Mohamad El-Abed, "Évaluation de système biométrique," PHD thesis, Université de Caen, 2011. Accessed: Mar. 03, 2023. [Online]. Available: https://theses.hal.science/tel01007679.

[5] : Médégnonmi Houssou, "Amélioration des systèmes de reconnaissance faciale par l'utilisation de caméra MSFA et des techniques d'intelligence artificielle" Thèse de Doctorat.

---

[6] : https://www.kaspersky.fr/resource-center/definitions/what-is-facial-recognition

[7] : https://fr.vittascience.com/shop/150/raspberry-pi-4-modele-b---4gb