



RAPPORT
De Projet de Fin d'Études
Présenté en vue de l'obtention de
LICENCE SCIENCES ET TECHNIQUES
Spécialité : **MATHEMATIQUES-INFORMATIQUE.**

**Conception d'une solution de gestion de la présence des étudiants
intégrant la reconnaissance faciale**

Réalisé par :

ALI ELYOUSFI
 RABAH FARES

Encadré par :

ABDELHAMID TADMORI

Co-encadrante :

SARA OUALD CHAIB

Soutenu le 25-juin-2024 devant le jury composé de :

AHMED. TOUKMATI
 ABDERRAHIM ZANNOU
 ABDELHAMID TADMORI
 SARA OUALD CHAIB

Président
Examinateur
Encadrant
Co-encadrante

Année universitaire 2023-2024

DÉDICACE



À nos chers parents

C'est grâce à vous que nous sommes devenus ce que nous sommes aujourd'hui. Votre amour inconditionnel et votre soutien indéfectible ont été notre moteur.

À nos chères sœurs et nos frères,

Vous avez toujours été à nos côtés pour apporter soutien et encouragement. Ce travail exprime notre gratitude et notre amour.

À nos chers amis,

Merci pour les moments agréables que nous passons avec vous, Tout l'amour pour vous aussi.

Et à vous chers lecteurs.

REMERCIEMENT

Avant tout, nous remercions notre Dieu qui illuminé notre chemin et qui nous a armé du courage et de bonne volonté pour réaliser ce modeste travail.

Au terme de ce travail, Nous voulons dire un grand merci à toute l'équipe enseignante de la Faculté des sciences et techniques d'Al-Hoceima, ainsi qu'aux professionnels qui nous ont aidés dans nos cours de maths et d'informatique. Ils ont été vraiment géniaux et ont beaucoup contribué à notre apprentissage. Grâce à eux, on a vraiment bien progressé dans nos études.

Nos remerciements s'adressent particulièrement au Professeur Docteur TADMORI ABDELHAMID notre encadrant de projet de fin d'étude, qui nous a toujours accordé sa confiance, pour sa patience et surtout ses judicieux conseils. On le remercie profondément pour ses encouragements et son soutien constant.

Nous souhaitons également exprimer notre profonde gratitude envers notre encadrante, Madame SARA OUALD CHAIB, pour sa confiance continue, sa patience inébranlable et ses conseils avisés. Ses encouragements et son soutien constants ont été précieux tout au long de notre parcours. Merci infiniment pour son engagement et sa bienveillance

Nos très chers parents pour leur soutien et encouragement durant toutes nos années d'études et sans lesquels on n'aurait jamais réussi. Nos frères et sœurs, pour leur présence, leur soutien moral et leurs encouragements.

Merci à tous.



RÉSUMÉ

Dans le cadre des projets de fin d'étude, chaque étudiant reçoit un thème de projet. Ceci dans le but de permettre à chaque étudiant sortant de pouvoir s'appuyer sur ces projets pour s'insérer aisément dans le monde professionnel ou même de développer de nouvelles aptitudes. C'est dans ce sens que nous avons eu pour thème de projet « **Conception d'une solution de gestion de la présence d'étudiante intégrant La reconnaissance Faciale** ». Ce projet part du constat que l'édition des fiches de présences en salle de classe est très gourmand en temps et devient une tâche difficile pour les enseignants ou pour les délégués de classe lorsque le nombre d'étudiant est très élevé. De plus, avec la méthode actuelle, les fraudes sont de plus en plus récurrentes, dans la mesure où c'est possible pour un étudiant de répondre à l'appel d'un autre. En outre, les fiches de présences sont faites manuellement et sur papier, donc la probabilité de faire des erreurs ou de perdre des listes de présences est assez élevée.

Notre mission était donc de mettre en place une application de reconnaissance faciale qui sera pour objectif de détection des visages des étudiants en temps réel, et les comparés avec une série des visages enregistrées dans une base de données, et dans le cas où il y'a une correspondance, les informations de l'étudiant seront inscrites dans une liste de présence.

Le projet se divise en deux grandes parties :

- Détection des visages : Le système utilise des algorithmes avancés tels que l'histogramme des gradients orientés HOG pour détecter les visages dans les images capturées par la caméra.
- Reconnaissance des visages : Une fois les visages détectés, le système utilise des modèles de la bibliothèque Dlib pour reconnaître les visages des étudiants enregistrés dans la base de données du système.

ABSTRACT

In the context of end-of-studies projects, each student is assigned a project theme. This is aimed at enabling each graduating student to easily integrate into the professional world or even develop new skills through these projects. It is in this sense that our project theme was "Designing a Student Attendance Management Solution Integrating Facial Recognition." This project stems from the observation that manually taking attendance in the classroom is very time-consuming and becomes a challenging task for teachers or class representatives when the number of students is very high. Moreover, with the current method, frauds are becoming more frequent, as it is possible for one student to answer for another. Furthermore, attendance sheets are done manually and on paper, so the probability of making errors or losing attendance sheets is quite high.

Our mission was therefore to implement a facial recognition application aimed at detecting students' faces in real-time, comparing them with a series of faces recorded in a database, and in the event of a match, the student's information will be recorded in an attendance list

The project is divided into two main parts:

- Face Detection: The system utilizes advanced algorithms such as Histogram of Oriented Gradients (HOG) to detect faces in images captured by the camera.
- Face Recognition : Once faces are detected, the system uses models from the Dlib library to recognize the faces of students enrolled in the system's database.

Table des figures

Figure 1: Les besoins non fonctionnels	3
Figure 2: Tableau des taches.....	5
Figure 3 : Diagramme de gant	6
Figure 4 : Le cycle de vie de l'application.	7
Figure 5:Reconnaissance faciale.....	10
Figure 6 : Fonctionnement de la reconnaissance faciale	10
Figure 7: Application de la reconnaissance faciale	11
Figure 8:La procédure de DeepFace pour la reconnaissance	12
Figure 9 : Image originale.....	15
Figure 10 : Image en niveaux gris.....	15
Figure 11 : Image originale.....	16
Figure 12 : Image en cellules	16
Figure 13 : Sobel vertical (Y)	16
Figure 14 : Sobel horizontal (X).....	16
Figure 15 : Image en cellules	17
Figure 16 : Dégradation en gris.....	17
Figure 17:Cellule 4*4 pixels	17
Figure 18 : Image en cellules	19
Figure 19 : Histogramme des cellules	19
Figure 20 : Histogramme de la cellule (cellule de figure 17).....	20
Figure 21 : Image en cellules (rouge) et en blocs (jaune)	21
Figure 22 : Histogramme de quatre cellules	21
Figure 23 : Histogramme de bloc	21
Figure 24 : Normalisation de bloc.....	22
Figure 25 : Une fenêtre glissante sur l'image à différentes échelles pour détecter des visages.....	23
Figure 26 : Visage détecté.....	23
Figure 27 : Visualiser les coordonnées des 68 points de repère faciaux	24
Figure 28:Les caractéristiques de chaque visage.....	25
Figure 29: Les caractéristiques des visages dans le fichier CSV	25
Figure 30 : La distance euclidienne	26
Figure 31 : Visage connu	26
Figure 32 : Visage inconnu	26
Figure 33 : Schéma générale des technique de reconnaissance facial utilisée	27
Figure 34:Logo de langage de modulation UML	28
Figure 35: Types des diagrammes UML	28
Figure 36: Diagramme de cas d'utilisation d'administrateur.....	29
Figure 37: Diagramme de cas d'utilisation d'étudiant	29
Figure 38: Diagramme de séquence d'étudiant dans la phase d'inscription	30
Figure 39: Diagramme de séquence d'étudiant dans la phase de marquage de présence	31
Figure 40 : Diagramme de séquence d'administrateur dans la phase d'accéder au liste des étudiants	32
Figure 41: Diagramme de séquence dans la phase d'accéder au la liste de présence.....	34
Figure 42 : Architecture global de l'application.....	36
Figure 43 : Logo de langage python.....	38
Figure 44 : logo de visual studio code.....	39
Figure 45 : Acceptation du contrat.....	40
Figure 46 : Sélectionner les taches supplémentaires.....	40
Figure 47 : Installation du logiciel.....	41
Figure 48: Logo de GanttProject	41
Figure 49 : Logo de EDRAW MAX.....	42

<i>Figure 50 : L'accès à la page d'installation.....</i>	43
<i>Figure 51 : Choix de type de téléchargement</i>	43
<i>Figure 52 : Installation.....</i>	44
<i>Figure 53 : Logo de MySQL.....</i>	44
<i>Figure 54 : Logo MySQL Workbench.....</i>	45
<i>Figure 55 : Les 10 meilleures bibliothèques de python</i>	46
<i>Figure 56 : Logo de OpenCV</i>	46
<i>Figure 57: Installation OpenCV</i>	47
<i>Figure 58 : Importation de OpenCV.....</i>	47
<i>Figure 59 : Logo de Dlib</i>	47
<i>Figure 60 : Installation de Dlib</i>	48
<i>Figure 61 : Logo de Tkinter.....</i>	48
<i>Figure 62 : Logo de NumPy</i>	49
<i>Figure 63 : Installation de NumPy</i>	49
<i>Figure 64 : Importation de NumPy.....</i>	50
<i>Figure 65: Logo de pandas.....</i>	50
<i>Figure 66 : Installation de pandas.....</i>	50
<i>Figure 67 : Importation de pandas</i>	51
<i>Figure 68 : Interface principal</i>	53
<i>Figure 69 : Interface de connexion.....</i>	54
<i>Figure 70 : Interface principale d'administrateur</i>	55
<i>Figure 71 : Interface d'étudiant.....</i>	55
<i>Figure 72 : Fonctionnement du bouton "about".....</i>	56
<i>Figure 73 : Fonctionnement du bouton "contacter nous"</i>	57
<i>Figure 74 : L'interface principale.....</i>	58
<i>Figure 75 : Espace étudiant.....</i>	58
<i>Figure 76 : Espace d'inscription.....</i>	59
<i>Figure 77 : Zone de détection visage et capture des images</i>	59
<i>Figure 78 : Zone de saisir des informations personnelles</i>	60
<i>Figure 79 : Dossier des images captures</i>	60
<i>Figure 80 : Espace d'étudiant.....</i>	61
<i>Figure 81 : Détection et reconnaissance de visage</i>	62
<i>Figure 82 : Le cas où le visage n'est pas reconnu</i>	62
<i>Figure 83 : Interface principal</i>	63
<i>Figure 84 : Interface de connexion.....</i>	64
<i>Figure 85 : Espace d'administrateur</i>	65
<i>Figure 86 : Liste des étudiants</i>	65
<i>Figure 87 : Créer compte</i>	66
<i>Figure 88 : Espace d'administrateur</i>	66
<i>Figure 89 : Interface de liste de présence et d'absence</i>	67
<i>Figure 90 : Liste de présence</i>	67
<i>Figure 91 : Liste d'absence.....</i>	68

Table des matières

Dédicace	i
Remerciement.....	ii
Résumé	iii
Abstract	iv
Introduction général.....	1
Présentation et cadrage du projet	2
I. Introduction :	2
II. Problématique :.....	2
III. Objectifs :	2
1. L'objectif principale :	2
2. L'objectif spécifique :	2
IV. Solution proposée :	3
V. Les besoins non fonctionnels :.....	3
CHAPITRE I : DÉROLEMENT DE PROJET	4
I. Planification opérationnelle :.....	5
1. Tableau des taches :	5
2. Diagramme de Gantt :	5
II. Processus de développement :	6
1. Présentation du cycle de vie de développement :	6
2. Phase du cycle de vie en V :	7
III. Conclusion :.....	8
CHAPITRE II : RECONNAISSANCE FACIALE	9
I. Introduction :	10
II. Fonctionnement :	10
III. Les domaines d'utilisations :	11
IV. Historique :	12
V. Conclusion :.....	12
CHAPITRE III : CONTEXTE THÉORIQUE ET CONCEPTION DE L'APPLICATION... 	13
I. Techniques de détection des visages	14
1. Introduction	14
2. Détection des visages via la méthode HOG	14
a. Définition et description générale.....	14
b. Fonctionnement :	15
3. Extraction des caractéristiques de visage :	24
II. Technique de reconnaissance des visages :	25
1. Introduction :	25

2.	La reconnaissance des visages par calcule de la distance euclidienne :	26
III.	Schéma générale des techniques de reconnaissance faciale utilisés :	27
IV.	Conception de l'application :	28
1.	Langage de modulation UML :	28
2.	Diagramme de cas d'utilisation :	29
a.	Définition de diagramme de cas d'utilisation :	29
b.	Diagramme de cas d'utilisation d'administrateur :	29
c.	Diagramme de cas d'utilisation d'étudiant :	29
3.	Diagramme de séquence :	30
a.	Définition de diagramme de séquence :	30
b.	Diagramme de séquence d'étudiant :	30
c.	Diagramme de séquence d'administrateur :	32
4.	Architecture global de l'application :	35
V.	Conclusion :	36
CHAPITRE IV : PROGRAMMES ET OUTILS UTILISÉS		37
I.	Introduction :	38
II.	Langage Python :	38
1.	Définition de Python :	38
2.	Les avantages de Python :	38
III.	Visual Studio Code :	39
1.	Définition de Visual Studio Code :	39
2.	Les étapes d'installation de Visual Studio Code :	39
IV.	GanttProject :	41
1.	Définition de GanttProject :	41
2.	Les avantages de GanttProject :	41
V.	EDRAW MAX :	42
1.	Définition de Edraw Max :	42
2.	Les étapes d'installation de Edraw Max :	42
VI.	MySQL :	44
1.	Les avantages de MySQL :	45
2.	MySQL Workbench :	45
VII.	Bibliothèques de python utilisés :	45
1.	La bibliothèque OpenCV :	46
a.	Définition de OpenCV :	46
b.	Installation :	46
2.	La bibliothèque Dlib :	47
a.	Définition de Dlib :	47
b.	Installer :	48

3.	La bibliothèque Tkinter :	48
4.	La bibliothèque NumPy :	49
a.	Définition de NumPy.....	49
b.	Installation :.....	49
5.	Pandas :	50
a.	Définition de Pandas :.....	50
b.	Installation :.....	50
VIII.	Conclusion :.....	51
CHAPITRE V : IMPLÉMENTATION.....		52
I.	Introduction :	53
II.	Les interfaces de l'application :.....	53
1.	Interface principale :	53
2.	Interface d'administrateur	54
a.	Interface de connexion :	54
b.	Interface principale d'administrateur :	55
3.	Interface d'étudiant :	55
III.	Démonstration :	56
1.	Interface principal :	56
2.	Espace d'étudiant :	57
a.	Inscription :.....	57
b.	Marquage de présence :	61
3.	Espace d'administrateur :	63
a.	Connexion :	63
b.	La liste des étudiants :	64
c.	Liste de présence	66
d.	Marquage de présence :	68
IV.	Conclusion :.....	68
Conclusion générales		69
Bibliographie.....		70

Introduction général

Avec l'avancée technologique rapide dans le domaine de la reconnaissance faciale, de nouvelles opportunités émergent pour améliorer le processus de gestion de la présence des étudiants. La reconnaissance faciale, qui était autrefois limitée aux domaines de la sécurité et de l'identification personnelle, trouve désormais des applications innovantes dans le secteur de l'éducation, remplaçant ainsi les méthodes de gestion traditionnelles basées sur des méthodes manuelles ou semi-automatisées telles que les feuilles de présence.

Le principal objectif de ce projet est de développer une application solide et fiable, capable d'assurer une gestion efficace de la présence des étudiants tout en garantissant la confidentialité et la sécurité des données biométriques. En intégrant la reconnaissance faciale, ces solutions visent à simplifier le processus de signature, à réduire les risques d'erreurs.

La technologie de reconnaissance faciale est un domaine très actif dans le domaine de la vision par ordinateur et de la biométrie. Elle a fait l'objet d'études intensives avec pour objectif de produire des applications dans la sécurité, la robotique, les interfaces homme-machine, les caméras numériques, les jeux et le divertissement.

La reconnaissance faciale comprend généralement deux étapes :

- La détection du visage, qui consiste à rechercher un ou plusieurs visages et à les détecter dans une image numérique ou une vidéo
- La reconnaissance du visage, qui implique de comparer le visage détecté avec ceux présents dans la base de données des visages reconnus, afin de déterminer à qui appartient le visage identifié.

Le reste du rapport se présente de la manière suivante :

- ✓ **Présentation et cadrage du projet :** dans cette première partie, on expose les exigences du projet. Il expose la situation globale et la structure de l'application, en exposant la problématique et la solution.
- ✓ **Chapitre 1–Déroulement de projet :** dans ce chapitre, on va voir comment on a organisé notre projet étape par étape. On commencera par faire tableau de tâches contenant les choses à faire. Ensuite, en utilisant un diagramme de Gantt, qui est comme un planning visuel. Enfin, on parlera du modèle de cycle de vie en V, qui nous aide à bien organiser toutes les étapes du projet.
- ✓ **Chapitre 2–Reconnaissance faciale :** dans ce deuxième chapitre, nous aborderons la reconnaissance facial en fournissant sa définition, en expliquant son fonctionnement et ses applications, ainsi son historique.
- ✓ **Chapitre 3–Contexte théorique et conception de l'application :** dans ce chapitre, nous explorerons comment détecter les visages dans les images. Nous verrons les différentes méthodes utilisées, des plus anciennes aux plus récentes, pour reconnaître les caractéristiques faciales. Ensuite, nous aborderons la conception d'une solution de suivi de présence pour les étudiants, en utilisant les diagrammes pour représenter les interactions entre les utilisateurs et le système.
- ✓ **Chapitre 4 – Programmes et outils utilisés :**
Dans ce chapitre, nous discuterons des outils qui ont été utilisés dans notre projet.
- ✓ **Chapitre 5 – Implémentation :**
Dans cette partie, nous allons présenter une démonstration de notre application, au cours de laquelle seront découvertes les différentes fonctionnalités qu'elle propose.

Présentation et cadrage du projet

I. Introduction :

Aujourd'hui, la gestion de la présence des étudiants est un défi majeur pour les établissements d'enseignement. Les techniques classiques d'enregistrement des présences sont chronophages, et sujettes à de nombreuses erreurs. Il est donc crucial de concevoir des nouvelles solutions performantes afin de gérer la présence des étudiants.

Ce projet a pour objectif de concevoir et de développer une solution technologique innovante pour la gestion de la présence des étudiants, basée sur la reconnaissance faciale. Cette solution vise à simplifier et à accélérer la processus d'enregistrement des présences, à en améliorer la précision et à réduire les cas de fraude, contribuant ainsi à une meilleure gestion de la présence dans les établissements éducatifs.

II. Problématique :

Au début de chaque cours dans les établissements éducatifs, l'enseignant ou le responsable procède à l'enregistrement de la présence des étudiants, l'édition de la fiche de présence se fait principalement de façon orale, c'est-à-dire que le délégué ou l'enseignant possède une liste sur laquelle le nom de chaque étudiant est inscrit, chaque nom devra donc être lu ; lorsqu'un étudiant suit son nom, il doit répondre présent pour que l'enseignant ou le délégué de classe puisse le marquer présent sur la fiche. Le processus actuel étant décrit, on peut constater qu'il est effectivement possible d'altérer le niveau d'intégrité de la fiche de présence. La principale fraude qui pourrait être appliquée dans ce cas est qu'un étudiant peut répondre à la place d'un autre ou encore, un individu inconnu de tous peut se faire passer pour un étudiant de la salle et répondre à sa place.

En plus des fraudes possibles, un autre problème qui réside est le fait que le temps à mettre pour réaliser l'appel dépend excessivement du nombre d'étudiants (plus les étudiants sont nombreux, plus le temps mis est élevé ; donc réaliser l'appel peut devenir un véritable supplice pour les enseignants ou les délégués.

III. Objectifs :

1. L'objectif principale :

L'objectif principale de notre projet est La création d'une solution de gestion de la présence étudiante qui intègre la reconnaissance faciale vise principalement à offrir un système efficace et fiable pour surveiller la présence des étudiants dans un cadre éducatif. L'objectif de cette solution est d'automatiser la détection de présence en utilisant la technologie de reconnaissance faciale afin d'identifier les étudiants à leur arrivée en classe ou dans d'autres zones spécifiques.

2. Les objectifs spécifiques :

Pour atteindre cet objectif principal, nous devons réaliser un ensemble d'objectifs spécifiques, parmi lesquels figurent :

- Créer une interface utilisateur efficace et facile à utiliser.
- Donner la capacité à notre solution de détecter un visage en temps réel.

- Permettre au système de faire une identification de visage grâce à des images déjà enregistrées.
- Incrire dans une base de données les informations (CNE, Prénom, Nom, Email ...) nécessaires de chaque étudiant.
- Marquage de présence des étudiants.

IV. Solution proposée :

À travers ce projet, nous cherchons à créer un programme développé par le langage python, grâce à des bibliothèques telles que OpenCV, Dlib et Pandas, pour la détection et la reconnaissance faciale. En ce qui concerne la base de données, MYSQL est une solution efficace, ce dernier est un système utilisé pour gérer les bases de données relationnelles, où on stock les informations des étudiants.

V. Les besoins non fonctionnels :

Les besoins non fonctionnels les plus importants pour notre application dépendent largement du contexte spécifique de déploiement et des priorités de l'organisation.

Notre application doit nécessairement assurer ces exigences non-fonctionnels :

- **Sécurité et confidentialité** : Il est essentiel que les données biométriques des étudiants soient sécurisées et confidentielles, afin de renforcer la confiance des utilisateurs dans le système, tout en respectant les réglementations en matière de protection des informations personnelles.
- **Précision et fiabilité** : Il est crucial d'avoir une identification faciale précise et fiable afin d'éviter les erreurs de présence qui pourraient avoir un impact négatif sur les relevés de présence des étudiants.
- **Facilité d'utilisation** : Une facilité d'utilisation améliorée assure que le processus de reconnaissance faciale est simple à comprendre et à utiliser, ce qui diminue la réticence des étudiants et du personnel enseignant à adopter le système.

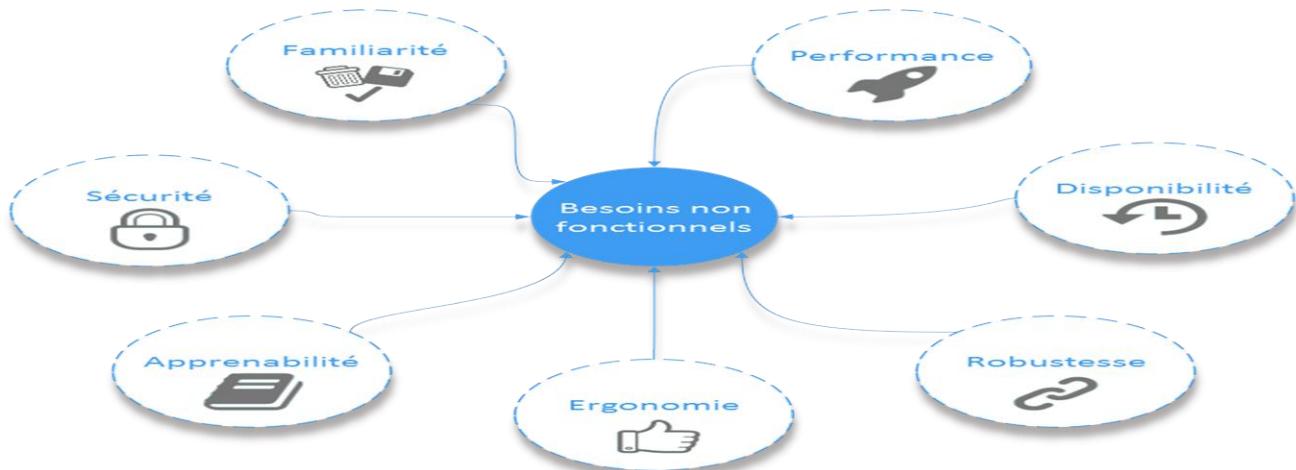


Figure 1: Les besoins non fonctionnels

CHAPITRE I : DÉROLEMENT DE PROJET

I. Planification opérationnelle :

1. Tableau des taches :

Un tableau des taches c'est comme une liste où on écrit tout ce qu'on doit faire. Ça aide à ne pas oublier les choses importantes et à s'organiser pour être sûr de tout faire à temps.



Nom	Date de début	Date de fin
Etude préalable	22/04/2024	26/04/2024
Elaboration de cahier de charge	26/04/2024	03/05/2024
Rédaction du rapport	26/04/2024	29/05/2024
Phase de conception	03/05/2024	09/05/2024
Développement de l'application	09/05/2024	29/05/2024
Présentation PowerPoint	30/05/2024	10/06/2024

Figure 2: Tableau des taches

- **Etude préalable :** nous faisons des recherches approfondies pour bien comprendre ce qu'il implique. On rassemble toutes les informations nécessaires et établissons une planification initiale.
- **Elaboration de cahier de charge :** cette tâche consiste à rédiger les différents aspects de notre projet tels que les définitions, les objectifs, la problématique, ... etc.
- **Phase de conception :** cette étape est consacrée à la modélisation et à la présentation de notre application ainsi que de son architecture globale.
- **Développement de l'application :** création et codage de notre application.
- **Rédaction de rapport :** on organise et on décrit chaque étape de notre projet de façon claire et ordonnée, pour une rédaction parfaite.
- **Présentation PowerPoint :** présentation de notre projet en PowerPoint.

2. Diagramme de Gantt :

Le diagramme de GANTT, utilisés pour planifier, exécuter et contrôler les projets, est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différents activités (tâches) qui Constituent un projet. Cela nous permet de voir et de suivre un ensemble de choses spécifiques à notre projet, telles que :

- Les différents tâches qui doit être réalisée.
- La date de début et de fin de chaque tâche.
- Durée de réalisation de chaque tâche.
- Les possibles interférence entre les tâches et leur durée.
- La date de début et de fin du projet.

Pour arriver à réaliser ce projet, nous avons suivi le planning suivant :

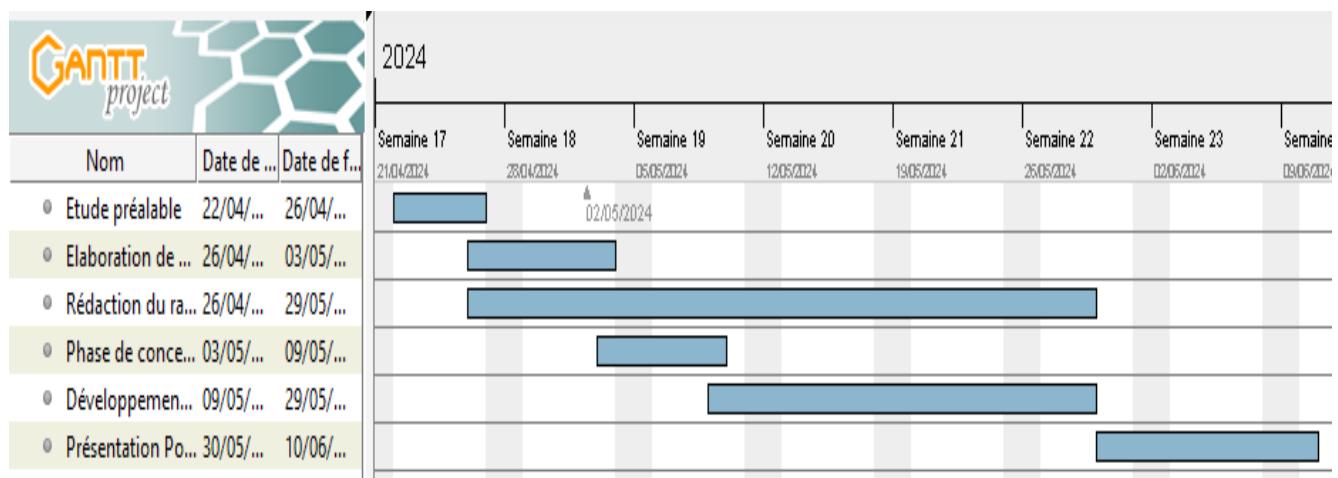


Figure 3 : Diagramme de Gantt

II. Processus de développement :

1. Présentation du cycle de vie de développement :

Le cycle de vie d'une application est l'ensemble de la durée de vie d'une application, représente tout son parcours, depuis sa création jusqu'à son retrait.

Il y a plusieurs façons de gérer le développement des applications. Voici quelques modèles les plus utilisés :

- **Modèle en cascade** : il s'agit d'un modèle linéaire avec phase séquentielles chaque phase doit être vérifiée avant de passer à la suivante et se termine à une date précisée.
- **Modèle en V** : ce modèle est similaire au modèle en cascade, mais met davantage l'accent sur les tests et la validation. Les activités de tests sont alignées avec les phases de développement correspondant, créant ainsi une forme en V.
- **Modèle en spiral** : ce modèle combine les aspects de modèle en cascade avec des itérations. Il suit un processus en spirale, où chaque itération inclut des phases de planification, de gestion des risques, de développement et de tests. Il est adapté aux projets complexes nécessitant une évaluation continue des risques.

Dans notre projet, nous avons adopté le modèle en V pour le cycle de vie de notre application, car l'avantage principal de cette méthode est d'offrir une approche structurée et séquentielle du développement d'applications, ce qui permet de mieux comprendre les exigences, de détecter les erreurs précocement, d'assurer une traçabilité et de maintenir une documentation claire.

2. Phase du cycle de vie en V :

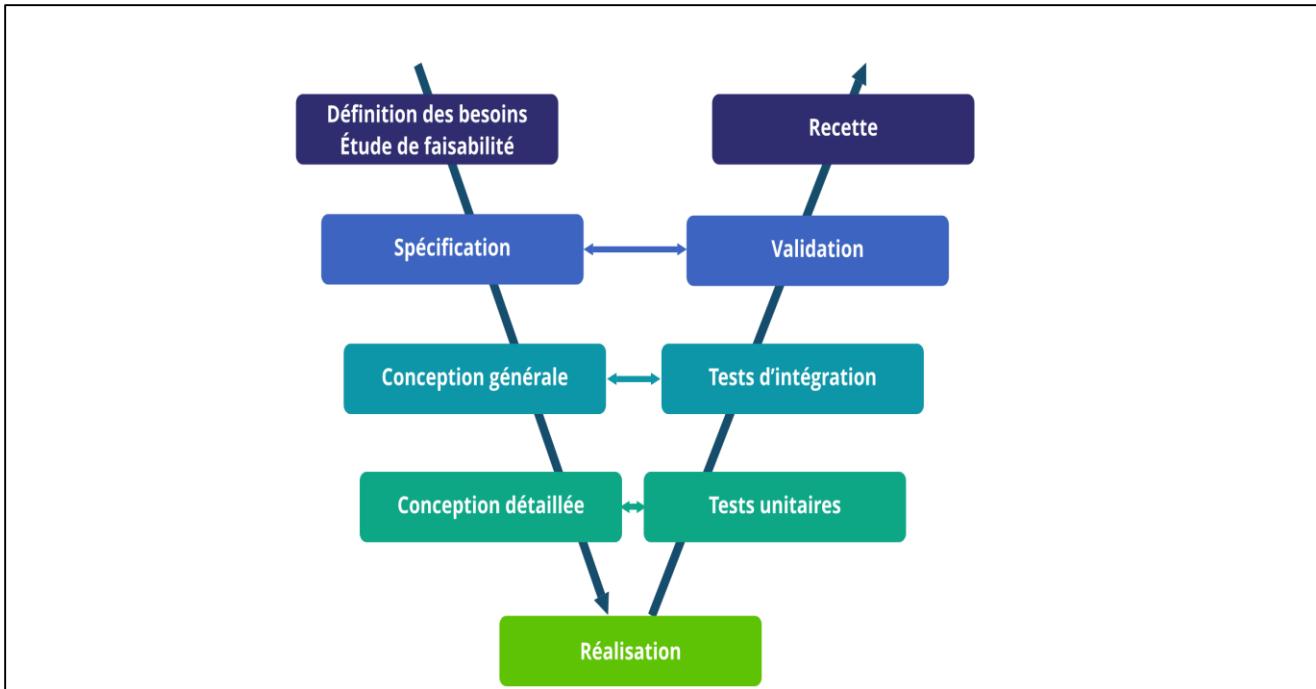


Figure 4 : Le cycle de vie de l'application.

- **Définition des besoins, étude de faisabilité** : Il s'agit de définir le cahier des charges et d'étudier les besoins des utilisateurs et les exigences fonctionnelles et non fonctionnelles de l'application.
- **Spécification** : après avoir étudié les besoins, on établit des spécifications pour l'application ; on détermine exactement ce que l'application doit pouvoir faire, comment elle doit être utilisée.
- **Conception générale** : dans cette étape on décide de comment le système sera organisé, des technologies et des pièces qu'on utilisera, et comment chaque partie sera connectée aux autres pour former l'application.
- **Conception détaillée** : la conception détaillée consiste à planifier en détail chaque partie de projet, on dirait que vous dessinez très précisément chaque pièce avant de les assembler.
- **Réalisation (codage)** : au cours de cette étape, on développe le code source de l'application en respectant la conception détaillée. En utilisant les langages de programmation appropriés, les développeurs transforment les conceptions en code utilisable.
- **Tests unitaires** : après le codage, des tests unitaires sont effectués sur chaque composant de l'application pour vérifier leur bon fonctionnement individuel, et en plus pour détecter les erreurs.
- **Tests d'intégration** : sont un ensemble des tests effectués pour vérifier le bon fonctionnement de l'ensemble du système.
- **Validation** : permettent de vérifier si toutes les exigences du client qui sont décrites pendant les spécifications sont respectées.
- **Recette** : c'est-à-dire la vérification de la conformité de l'application aux spécifications initiales

III. Conclusion :

Ce chapitre a examiné la progression du projet, mettant en évidence les étapes essentielles, l'importance de la planification, de la réalisation et du suivi. Il est crucial de saisir ces éléments afin d'assurer le succès et l'efficacité dans la concrétisation de nos objectifs.

CHAPITRE II : RECONNAISSANCE FACIALE

I. Introduction :

La technologie de pointe de la reconnaissance est une Système visant à reconnaître automatiquement une personne grâce à son visage, [1] permet d'identifier ou de confirmer l'identité d'une personne en se basant sur ses caractéristiques faciales uniques. Les algorithmes de traitement d'images et d'apprentissage automatique utilisés dans cette technologie permettent d'analyser les caractéristiques du visage d'une personne et de les comparer à une base de données déjà existante.



Figure 5:Reconnaissance faciale

II. Fonctionnement :

La reconnaissance faciale débute en prenant une photo du visage de la personne à identifier à l'aide d'une caméra.

Extraction des propriétés : On extrait de l'image les principales caractéristiques du visage, comme la distance entre les yeux, la forme du nez et des lèvres, et on les transforme en données numériques.

Analyse de la Base de Données : Les informations obtenues sont comparées à celles conservées dans une base de données regroupant les visages préalablement enregistrés.

Identification ou Vérification : En fonction du contexte d'utilisation, la reconnaissance faciale peut soit identifier une personne inconnue en la comparant à l'ensemble de la base de données, soit vérifier l'identité d'une personne en la comparant à une seule entrée dans la base de données.

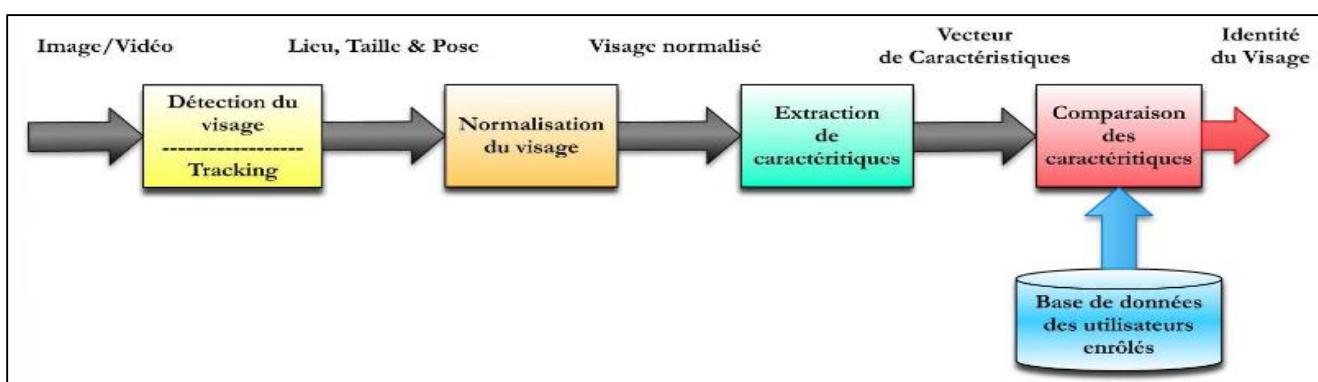


Figure 6 : Fonctionnement de la reconnaissance faciale

III. Les domaines d'utilisations :

La reconnaissance faciale est devenue une technique utilisée dans la plupart des métiers et des domaines, et parmi les plus remarquables, ces domaines :

Sécurité : La reconnaissance faciale est largement utilisée dans les systèmes de sécurité pour contrôler l'accès à des zones restreintes, comme les bâtiments gouvernementaux, les aéroports et les institutions financières.

Authentification : Elle est également utilisée pour l'authentification des utilisateurs dans les appareils mobiles, les applications bancaires en ligne et les systèmes de gestion de l'identité.

Éducation : Dans le cadre de notre projet, la reconnaissance faciale peut être utilisée pour la gestion de la présence étudiante, automatisant ainsi le processus de suivi de la présence en classe.

En revanche, on distingue un autre domaine d'application de ces systèmes qui est l'assistance à l'utilisateur. Les systèmes de reconnaissance faciale sont de plus en plus présents au quotidien. Ils sont par exemple utilisés sur les réseaux sociaux sur internet pour identifier quelqu'un sur une photo, sur les Smartphones pour les déverrouiller...



Figure 7: Application de la reconnaissance faciale

Facebook a la plus grande collection et exposition de photos au monde et a maintenant une chance d'unifier tous les visages. Oui, même les personnes qui ne sont pas identifiées. Facebook a annoncé qu'il a développé un programme appelé **DeepFace**, qui selon les chercheurs, peut déterminer si deux visages appartiennent à la même personne avec une précision de 98%. Selon Facebook, dans 97,53% des cas, les photos à différents âges de même personnes ont la même réponse au test. [2]

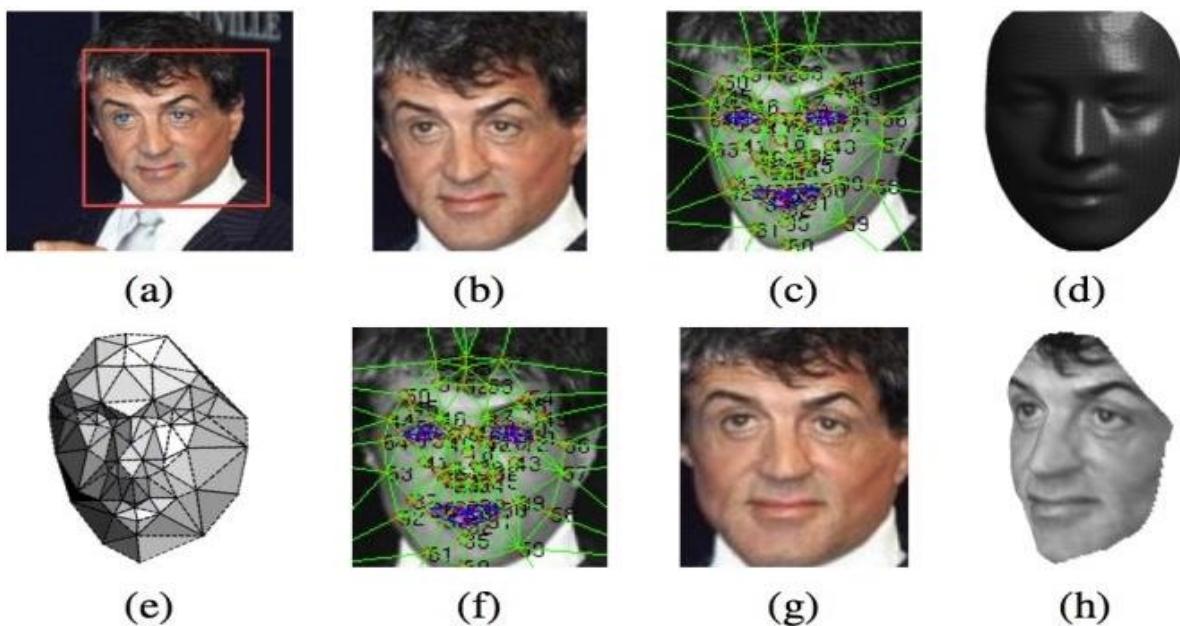


Figure 8: La procédure de DeepFace pour la reconnaissance

IV. 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 développé 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 apparaître des photographies de visages non familiers présentés sous différents éclairages et selon des angles différents et nécessitant une bonne capacité d'intégration Visio-spatiale. L'utilisation des techniques de reconnaissance faciale a connu un développement à grande échelle depuis 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, basées sur des attributs faciaux mesurables comme l'écartement des yeux, des sourcils, des lèvres, la position du menton, la forme, etc. 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. [2]

V. Conclusion :

Dans ce chapitre, nous avons parlé de la reconnaissance facial en donnant sa définition et en expliquant comment elle fonctionne, ainsi que les nombreux domaines dans lesquels elle est utilisée, en plus de son historique, ou nous avons mentionné les débuts de cette technologie

CHAPITRE III : CONTEXTE THÉORIQUE ET CONCEPTION DE L'APPLICATION

I. Techniques de détection des visages

1. Introduction

Ces dernières années, la technologie de détection faciale a connu une progression remarquable, transformant fondamentalement notre interaction avec les appareils numériques. Aujourd'hui, un éventail de méthodes de détection faciale existe, chacune présentant ses avantages et ses limites propres. Toutefois, grâce à l'émergence des algorithmes de machine learning, de nouvelles approches telles que la détection de visages, la reconnaissance faciale et la reconnaissance des expressions faciales ont gagné en précision et en fiabilité. Dans cette ère de développement technologique rapide, il est essentiel de comprendre les différentes techniques de détection faciale disponibles et leur impact potentiel sur notre vie quotidienne.

Parmi ces méthodes, nous allons nous pencher sur l'approche de l'histogramme des gradients orientés (HOG).

2. Détection des visages via la méthode HOG

a. Définition et description générale

Un histogramme de gradient orienté (HOG) est une caractéristique utilisée en vision par ordinateur pour la détection d'objet. La technique calcule des histogrammes locaux de l'orientation du gradient sur une grille dense, c'est-à-dire sur des zones régulièrement réparties sur l'image. La méthode est particulièrement efficace pour la détection de personnes.

Les HOG ont été proposés par Navneet Dalal et Bill Triggs, chercheurs au CNRS et à l'INRIA de Grenoble, à la conférence CVPR de juin 2005.

L'idée importante derrière le(descripteur) HOG est que l'apparence et la forme locale d'un objet dans une image peuvent être décrites par la distribution de l'intensité du gradient ou la direction des contours. Ceci peut être fait en divisant l'image en régions adjacentes de petite taille, appelées cellules, et en calculant pour chaque cellule l'histogramme des directions du gradient ou des orientations des contours pour les pixels à l'intérieur de cette cellule. La combinaison des histogrammes forme alors le(descripteur) HOG. Afin d'obtenir de meilleurs résultats, on normalise en contraste les histogrammes locaux en calculant une mesure de l'intensité sur des zones plus grandes que les cellules, connues sous le nom de blocs, et en utilisant cette valeur pour normaliser toutes les cellules du bloc. Cette standardisation offre une plus grande résistance aux variations d'éclairage et aux ombres.

L'algorithme de HOG comporte sept étapes principales, dont nous parlerons dans les sections suivantes :

- Prétraitement de l'image.
- Calcul des gradients.
- Calcul des magnitudes et orientations des gradients.
- Formation de l'histogramme des cellules.
- Normalisation des blocs.
- Formation du(des) descripteur(s) HOG.
- Classification ou détection.

b. Fonctionnement :**➤ Prétraitement de l'image :**

Le traitement préliminaire des images constitue une phase essentielle au sein de multiples flux de travail en vision par ordinateur, y compris dans le calcul de l'Histogramme des gradients orientés (HOG). Cette étape vise à préparer l'image brute pour une analyse ultérieure en réduisant sa complexité et en mettant en évidence les caractéristiques pertinentes.

Prétraitement de l'image se fait principalement comme suit :

L'image se converse en niveaux de gris pour la simplifie en éliminant les couleurs, mais en préservant les informations sur la luminosité. Cela facilite le traitement tout en conservant la structure de l'image. Ensuite, on peut la redimensionner sa taille, ce qui peut être utile pour s'adapter à des normes spécifiques ou accélérer le traitement. Cela peut être fait en gardant le rapport hauteur-largeur d'origine pour éviter toute distorsion.

Enfin, La réduction du bruit élimine les perturbations indésirables de l'image, comme le grain ou les taches, tout en préservant les détails importants. Cela rend l'image plus claire et plus facile à analyser.



Figure 9 : Image originale

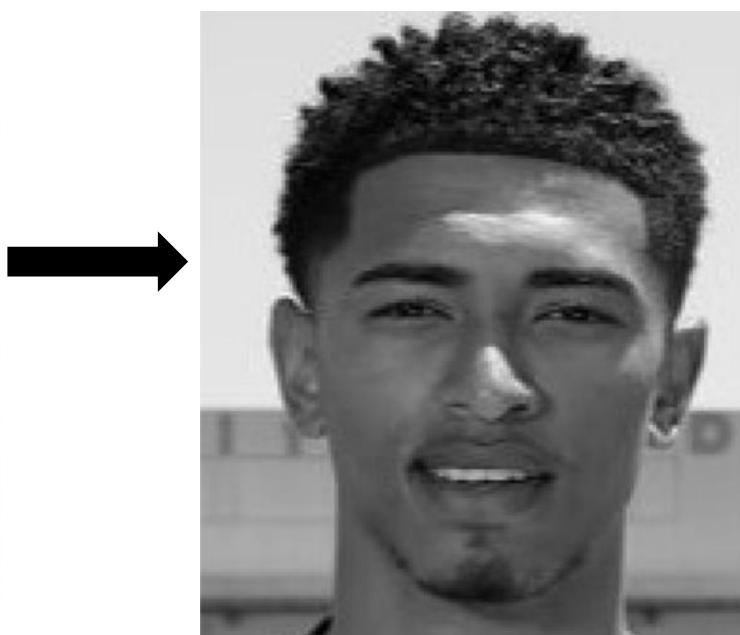


Figure 10 : Image en niveaux gris

➤ **Calcul des gradients :**

Pour construire un descripteur HOG, nous devons d'abord comprendre les changements dans l'intensité de l'image, ce que nous faisons en calculant les gradients. Les gradients nous indiquent la direction et la magnitude des changements d'intensité lumineuse dans l'image. Cela se fait en utilisant des opérateurs de dérivation comme des filtres de Sobel 3x3, ou des masques diagonaux, ou non centrés. Qui calculent les gradients dans les directions horizontale (x) et verticale (y).

Ensuite, pour mieux comprendre les variations locales d'intensité lumineuse, nous divisons l'image en petites régions appelées cellules, généralement des cellules carrées de petite taille (de 4x4 à 12x12 pixels). Cela nous permet d'analyser les gradients localement, c'est-à-dire dans des régions spécifiques de l'image, nous pouvons capturer des détails fins et des motifs locaux qui peuvent être importants pour la tâche de traitement de l'image, comme la détection des visages ou la reconnaissance faciale.



Figure 11 : Image originale

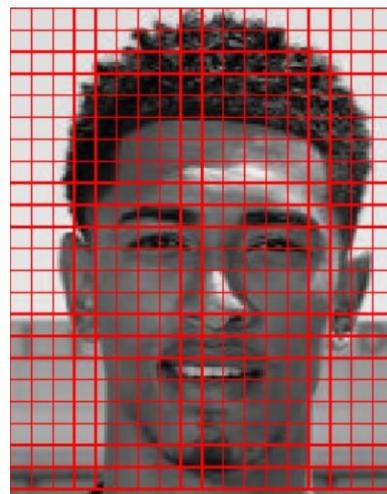


Figure 12 : Image en cellules

Le filtre de Sobel est un opérateur utilisé en traitement d'image pour la détection de contours. Il s'agit d'un des opérateurs les plus simples qui donne toutefois des résultats corrects.

L'opérateur utilise des matrices de convolution. La matrice de taille 3×3 subit une convolution avec l'image pour calculer des approximations des dérivées horizontale et verticale.

-1	0	1
-2	0	2
-1	0	1

Figure 14 : Sobel horizontal (X)

-1	-2	-1
0	0	0
1	2	1

Figure 13 : Sobel vertical (Y)

Soit une partie de l'image source, et G_x et G_y sont respectivement les gradients horizontales et verticales.

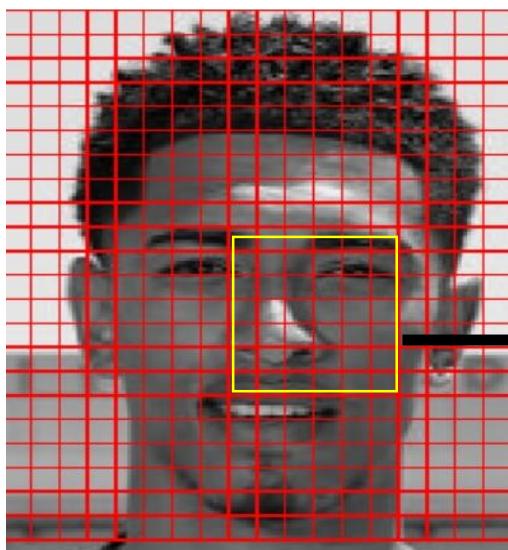


Figure 15 : Image en cellules

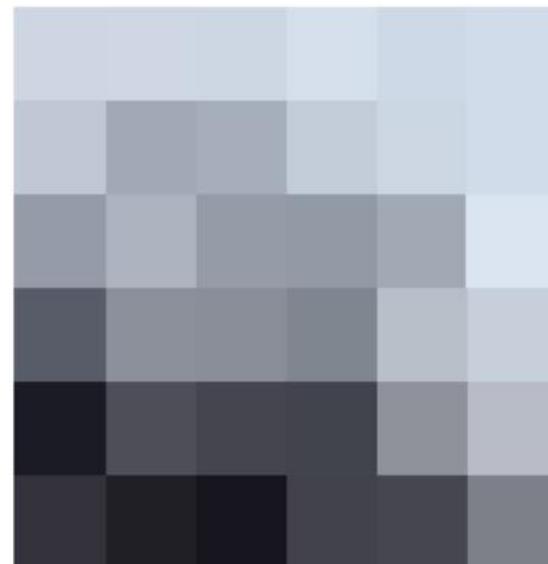


Figure 16 : Dégradation en gris

Pour calculer les gradients horizontales et verticales, nous allons prendre une cellule de cette dernière image en niveau gris. Nous allons calculer ces gradients pour chaque pixel :

Cette matrice contient les valeurs de chaque pixel de la cellule 4*4 pixels sur lequel on va travailler.

121	10	78	96
48	152	64	125
145	78	85	89
154	214	56	200

→ 85

Figure 17:Cellule 4*4 pixels

Pour faciliter les calculs des gradients horizontal G_x de la pixel 85, on fait la différence entre la valeur du pixel à droite de ce pixel en surbrillance et la valeur du pixel à gauche, et pour les gradients verticaux G_y , on fait la différence entre la valeur du pixel au-dessus de ce pixel en surbrillance et la valeur du pixel en dessous, comme suit :

Calculons le gradient horizontal (G_x) :

$$G_x = 89 - 78 = 11$$

Calculons maintenant le gradient vertical (G_y) : $G_y = 64 - 56 = 8$

Nous répétons ce processus pour chaque pixel de l'image, nous obtenons deux matrices :

$$\begin{pmatrix} 10 & 43 & 86 & 78 \\ 152 & 16 & 27 & 64 \\ 78 & 60 & 11 & 85 \\ 214 & 98 & 14 & 56 \end{pmatrix}$$

Ensembles des gradients horizontaux G_x de tous les pixels de la cellule

$$\begin{pmatrix} 48 & 152 & 64 & 125 \\ 24 & 68 & 7 & 7 \\ 106 & 62 & 8 & 75 \\ 145 & 78 & 85 & 200 \end{pmatrix}$$

Ensembles des gradients verticaux G_y de tous les pixels de la cellule

➤ Calcul des magnitudes et des orientations des gradients :

À partir des gradients G_x et G_y , nous pouvons calculer l'orientation(direction) et magnitude du gradient en utilisant la formule :

$$\mathbf{G} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

Magnitude

$$\Theta = \text{atan}\left(\frac{\mathbf{G}_y}{\mathbf{G}_x}\right)$$

Direction

L'orientation nous donne l'angle de la direction dans laquelle l'intensité lumineuse augmente le plus rapidement. et magnitude nous donne la quantité de changement d'intensité lumineuse dans cette direction.

Calcul de la magnitude du gradient :

La magnitude du gradient est donnée par la formule :

$$\text{magnitude} = \sqrt{\mathbf{G}_x^2 + \mathbf{G}_y^2}$$

$$\text{magnitude} = \sqrt{11^2 + 8^2} = 13.601$$

Calcul de l'orientation du gradient :

L'orientation du gradient est donnée par la formule :

$$\text{Orientation} = \tan^{-1}\left(\frac{8}{11}\right) = 36 \text{ degrés}$$

Donc, pour l'élément 85 de la matrice :

La magnitude du gradient est de 13,601

L'orientation du gradient est de 36 degré

Après avoir terminé ce processus pour chaque pixel, nous obtenons deux matrices :

Magnitudes des gradients :

49.03	157.97	107.2	147.34
153.88	69.86	27.89	64.38
131.61	86.28	13.6	113.36
260.19	125.25	86.15	207.69

Orientations des gradients :

78.23	74.2	36.66	58.04
8.97	76.76	14.53	6.24
53.65	45.94	36.03	41.42
34.67	38.52	80.65	74.36

➤ Formation de l'histogramme des cellules :

Après avoir calculé les gradients d'intensité de l'image, et divisant celle-ci en cellules, Chaque pixel de la cellule vote alors pour une classe de l'histogramme, en fonction de l'orientation du gradient à ce point. Pour chaque cellule, nous calculons un histogramme qui compte le nombre de gradients orientés dans différentes directions. Les orientations des gradients sont pondérées par leurs magnitudes, ce qui signifie que les gradients plus forts ont un impact plus important sur l'histogramme. Les histogrammes sont uniformes de 0 à 180° ou de 0 à 360°, mais le meilleur résultat est obtenu avec un histogramme à 9 classes

Cette étape nous permet de capturer les informations sur la direction des changements d'intensité lumineuse dans chaque région de l'image de manière simple et efficace.

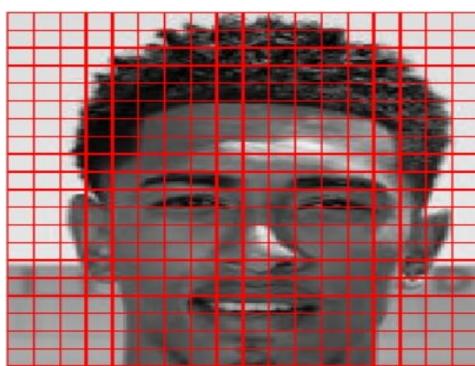


Figure 18 : Image en cellules



Figure 19 : Histogramme des cellules

D'après les résultats des orientations et des magnitudes de cellule **A**, on obtient l'histogramme ci-dessous :

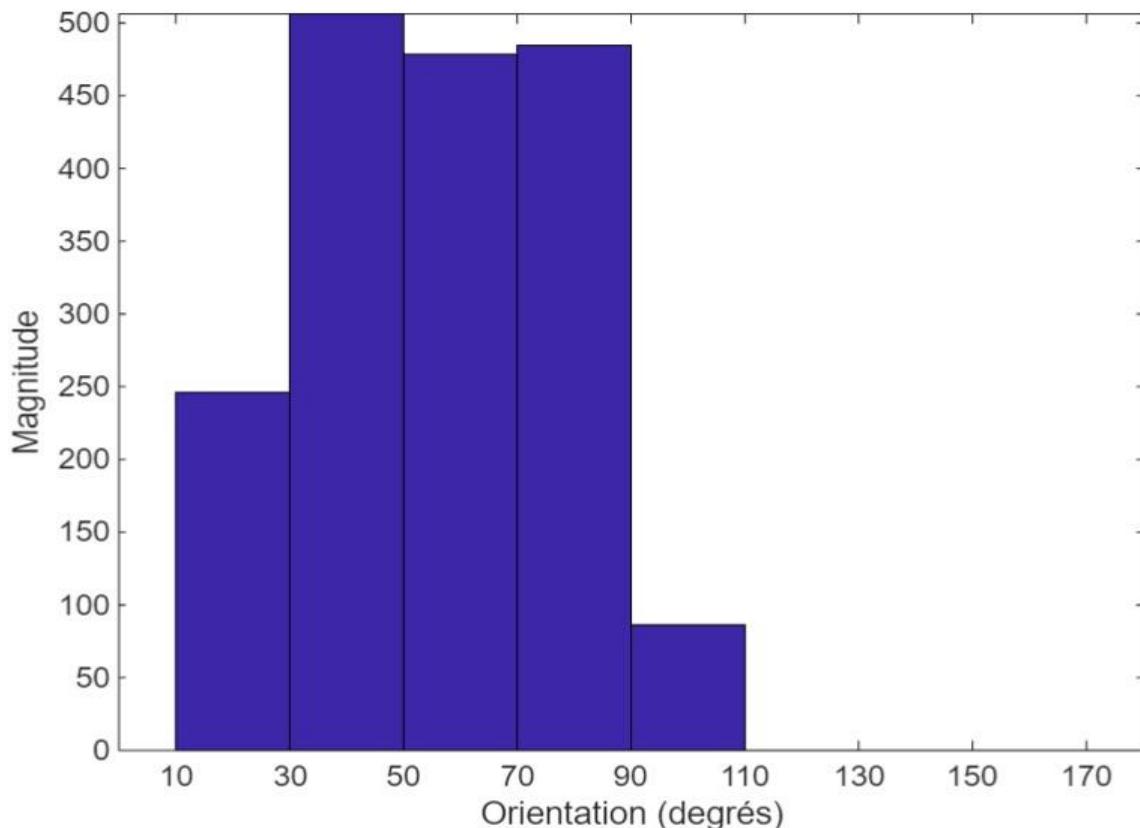


Figure 20 : Histogramme de la cellule (cellule de figure 17)

Remarque :

On fait les mêmes étapes pour chaque cellule de l'image, pour construire son histogramme.

➤ Normalisation des blocs :

Après avoir calculé l'histogrammes de chaque cellule, il est crucial de normaliser les blocs dans l'algorithme HOG afin d'assurer la fiabilité de la détection des objets, comme les visages, dans une image. Les cellules voisines doivent être regroupées en blocs (les blocs peuvent avoir une taille de 2x2 ou 3x3 cellule) et les histogrammes de gradients de chaque bloc doivent être normalisés. La normalisation, généralement effectuée à l'aide de la méthode L2, permet aux descripteurs d'être autonomes face aux variations d'éclairage et de contraste, ce qui améliore la précision de la détection. En associant cette normalisation à une gestion efficace des fenêtres de détection et de recouvrement, l'algorithme HOG devient plus performant pour extraire des caractéristiques distinctives des objets présents dans une image.

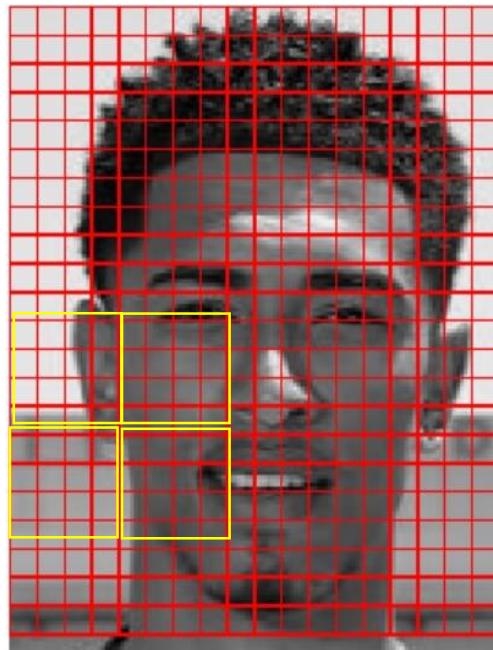


Figure 21 : Image en cellules (rouge) et en blocs (jaune)

Dans l'Histogramme des Gradients Orientés (HOG), la formation de blocs est une étape où les données de gradient sont regroupées à une échelle plus étendue afin de donner une représentation plus solide des caractéristiques dans une image. À l'origine, on subdivise l'image en petites cellules, chacune calculant un histogramme des orientations des gradients. Par la suite, on regroupe ces cellules en blocs, habituellement de taille 2×2 cellules. Ensuite, on regroupe les histogrammes des orientations des gradients de chaque cellule afin de créer le descripteur HOG du bloc.

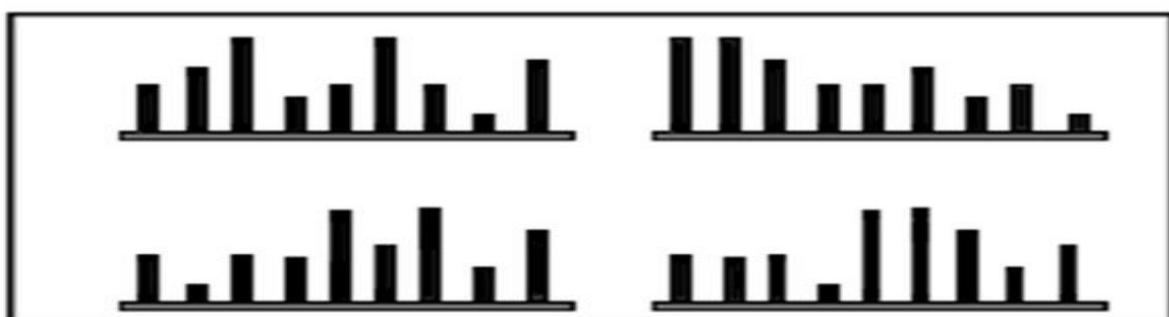


Figure 22 : Histogramme de quatre cellules

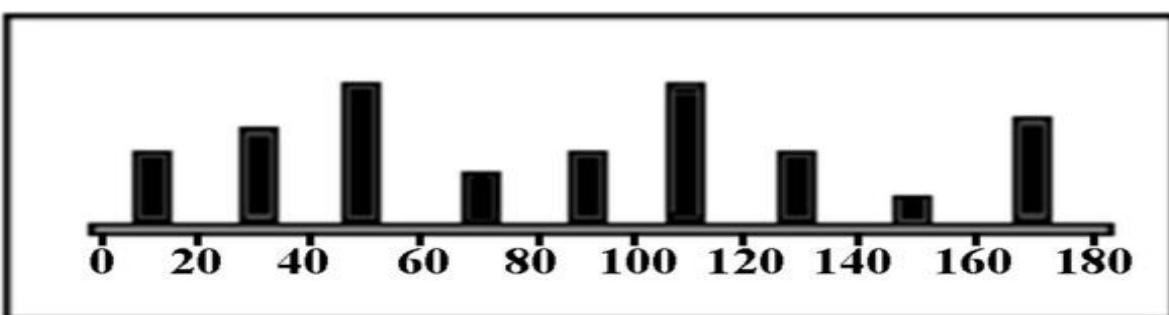


Figure 23 : Histogramme de bloc

Une fois le vecteur d'histogramme de bloc formé, il faut le normaliser.

Soit $V = [A_1, A_2, A_3, \dots, A_n]$ est un descripteur de bloc, ces valeurs sont l'ensemble des magnitudes des gradients de chaque cellule dans ce bloc.

$$\text{Et } L2 = \sqrt{A_1^2 + A_2^2 + \dots + A_n^2}$$

Le descripteur de block après la normalisation :

$$V = \left[\frac{A_1}{L2}, \frac{A_2}{L2}, \frac{A_3}{L2}, \dots, \frac{A_n}{L2} \right]$$

Exemple :

Supposons que nous ayons un bloc contenant trois valeurs de descripteur HOG : [2, 3, 4]

Calcul de la Norme L2 :

- Nous calculons d'abord la norme L2 en prenant la racine carrée de la somme des carrés des valeurs du descripteur : $\sqrt{2^2 + 3^2 + 4^2} = \sqrt{29} = 5,4$

Normalisation des Valeurs :

- Ensuite, chaque valeur du descripteur est divisée par cette norme L2 : $\left[\frac{2}{5,4}, \frac{3}{5,4}, \frac{4}{5,4} \right]$
- Cela donne environ [0.387, 0.581, 0.774]

La normalisation des blocs avec la norme L2 dans l'Histogramme des Gradients Orientés (HOG) garantit que nos descripteurs sont comparables et cohérents [3], peu importe les variations d'échelle et d'éclairage dans l'image. C'est essentiel pour une détection d'objets précise et robuste.

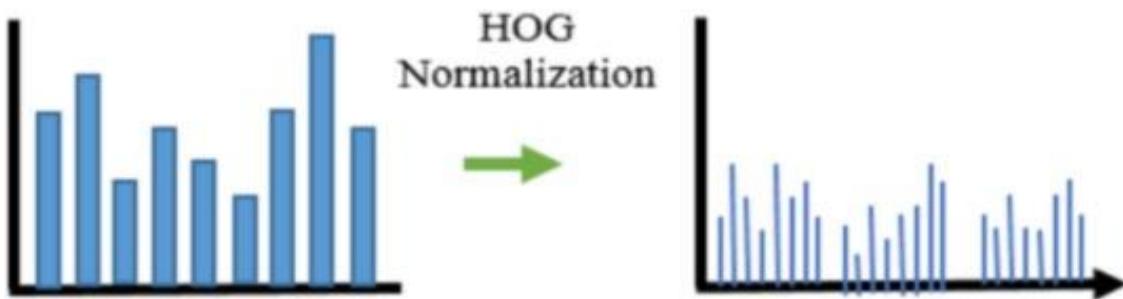


Figure 24 : Normalisation de bloc

Remarque :

On fait les mêmes étapes pour chaque bloc de l'image, pour construire son histogramme.

➤ Formation du descripteur HOG :

Une fois que nous avons calculé et normalisé les descripteurs de blocs pour chaque bloc de l'image, nous pouvons les regrouper pour former un unique vecteur de descripteur HOG pour l'ensemble de l'image. Cela implique d'aligner les descripteurs de chaque bloc dans une séquence continue.

Le vecteur de descripteur HOG résume les informations clés de l'image en utilisant les orientations des gradients. Son rôle est de capturer les caractéristiques importantes de l'image, comme les contours et les textures, pour des tâches telles que la détection des visages et la reconnaissance faciale.

Supposons chaque bloc a un vecteur de taille M et que nous avons N blocs, le vecteur de descripteur HOG final aura une taille de $N \times M$.

➤ Classification et détection :

Pour entraîner un modèle de détection de visages, on commence par rassembler un ensemble d'images étiquetées, comprenant certaines avec des visages (images positives) et d'autres sans visages (images négatives). Ensuite, on extrait les descripteurs HOG de chaque image, ce qui les transforme en vecteurs de caractéristiques. Ces vecteurs sont utilisés pour entraîner un modèle d'apprentissage automatique, tel qu'une machine à vecteurs de support (SVM) [4], qui apprend à distinguer les visages et non-visages. Une fois le modèle entraîné, on peut détecter des visages dans une nouvelle image en extrayant ses descripteurs HOG et en appliquant le modèle pour prédire la présence de visages. On utilise une fenêtre glissante sur l'image à différentes échelles pour détecter des visages de différentes tailles, si les descripteurs HOG extraits d'une région de l'image sont similaires à ceux des visages dans l'ensemble d'entraînement, le SVM va prédire que cette région contient un visage. Sinon, il prédira qu'il n'y a pas de visage dans cette région. Les régions de l'image où des visages sont détectés sont ensuite marquées par des rectangles. Enfin, on affiche l'image avec ces marques pour visualiser les résultats de la détection.



Figure 25 : Une fenêtre glissante sur l'image à différentes échelles pour détecter des visages



Figure 26 : Visage détecté

3. Extraction des caractéristiques de visage :

Après avoir identifié les visages dans l'image, nous avons utilisé un prédicteur de repères faciaux pour déterminer les emplacements précis des points caractéristiques du visage (68 points), tels que les coins des yeux, du nez et de la bouche. Ce prédicteur, qui utilise une technique appelée cascade de régressions linéaires, fonctionne comme un guide expert qui marque les contours du visage avec une précision remarquable. En comprenant la disposition spatiale de ces points, nous sommes en mesure d'extraire des informations détaillées sur la structure du visage, ce qui est essentiel pour des applications telles que la reconnaissance faciale et l'analyse des expressions faciales.

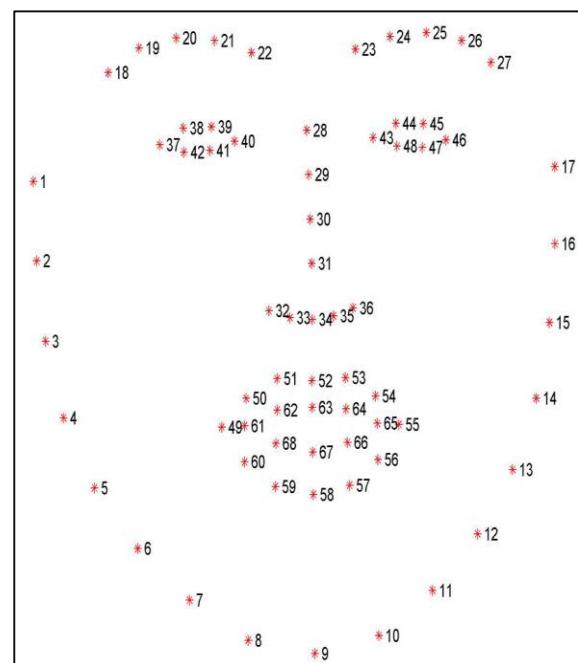


Figure 27 : Visualiser les coordonnées des 68 points de repère faciaux

Après avoir déterminer les emplacements précis des points caractéristiques du visage, on extrait les caractéristiques 128D(dimensions) à l'aide du modèle Dlib ResNet-50, ce modèle prend en entrée l'image du visage et les repères faciaux. Le vecteur des caractéristiques encapsule des informations sur la forme du visage, les proportions des différentes parties du visage, les textures de la peau, ainsi que des traits faciaux spécifiques tels que la forme des yeux, la courbure des lèvres, etc. Ces caractéristiques servent à représenter de manière unique les traits faciaux de chaque personne.

Une fois les caractéristiques extraites pour chaque visage, on calcule la moyenne de ces caractéristiques pour chaque personne. Cette moyenne permet de créer un vecteur de caractéristiques plus stable et représentatif des traits faciaux d'une personne. Ces vecteurs moyens sont ensuite stockés dans un fichier CSV, associés au CNE de la personne correspondante

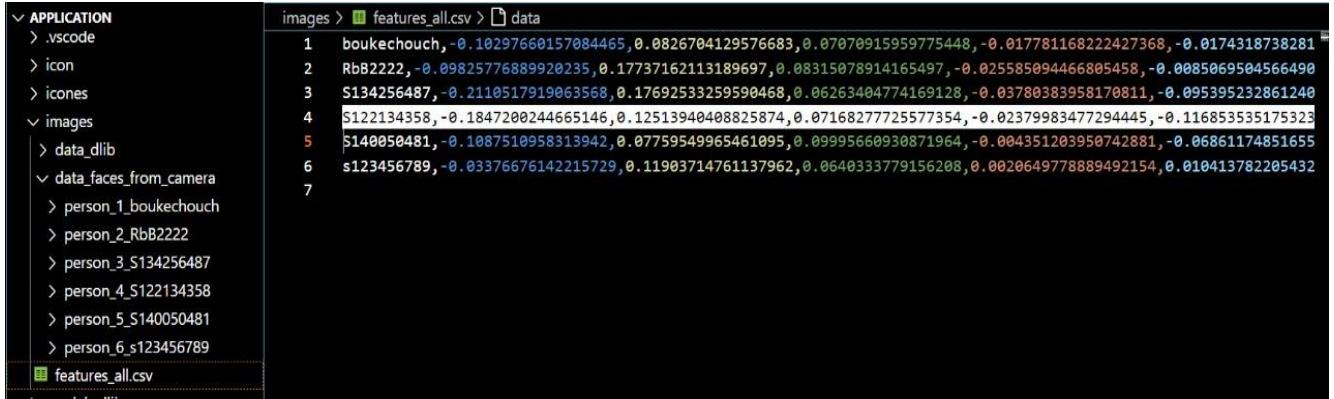


Figure 28:Les caractéristiques de chaque visage

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	boukechouch,-0.10297660157084465,0.0826704129576683,0.07070915959775448,-0.017781168222427368,-0.0174318738281														
2	RbB2222,-0.09825776889920235,0.17737162113189697,0.08315078914165497,-0.02558509446805458,-0.0085069504566490														
3	S134256487,-0.2110517919063568,0.17692533259590468,0.06263404774169128,-0.03780383958170811,-0.095395232861248														
4	S122134358,-0.1847200244665146,0.12513940408825874,0.07168277725577354,-0.02379983477294445,-0.116853535175323														
5	S140050481,-0.1087510958313942,0.07759549965461095,0.09995660930871964,-0.004351203950742881,-0.06861174851655														
6	s123456789,-0.03376676142215729,0.11903714761137962,0.0640333779156208,0.0020649778889492154,0.010413782205432653,-0.024660899536684155,-0.06549434550106525,-0.0633639432489872,0														
7															

Figure 29: Les caractéristiques des visages dans le fichier CSV

Chaque ligne du fichier CSV (figure 29) correspond à une personne et contient son CNE (ou un identifiant) suivi du vecteur de caractéristiques 128D.

II. Technique de reconnaissance des visages :

1. Introduction :

La reconnaissance faciale est une technologie qui vise à identifier et à vérifier l'identité d'une personne en se basant sur les caractéristiques de son visage. Elle a connu des avancées significatives ces dernières années grâce aux progrès de la vision par ordinateur, de l'apprentissage automatique et des techniques de traitement d'images.

La reconnaissance faciale peut utiliser différentes méthodes et techniques, telles que la reconnaissance basée sur les caractéristiques, les textures, les modèles 3d ou l'apprentissage automatique. Chaque méthode a ses propres forces et faiblesses en termes de précision, de robustesse aux variations et de performances.

Parmi ces techniques, nous allons étudier l'approche qui consiste à reconnaître les visages en calculant la distance euclidienne entre les caractéristiques du visage détecté et celles des visages connus

2. La reconnaissance des visages par calcul de la distance euclidienne :

La reconnaissance faciale avec des modèles pré-entraînés est un processus en plusieurs étapes. Tout d'abord, la détection des visages est effectuée à l'aide de techniques telles que la méthode des gradients orientés (HOG). Une fois les visages détectés, des repères faciaux sont extraits pour chaque visage. Ces repères, tels que les points des yeux, du nez et de la bouche, sont utilisés pour aligner les visages de manière cohérente et pour les représenter de manière uniforme. Ensuite, un modèle de réseau de neurones, tel que ResNet-50, est utilisé pour calculer un vecteur de caractéristiques de 128 dimensions pour chaque visage. Ces vecteurs représentent de manière unique les traits faciaux de chaque personne, capturant des informations telles que la forme du visage, les yeux, le nez et la bouche.

Une fois les caractéristiques extraites pour chaque visage, elles sont comparées à celles qui ont été enregistrées dans la base de données. Cela peut se faire en calculant **la distance euclidienne** entre les vecteurs de caractéristiques de chaque visage détecté et ceux de la base de données. Si la distance est inférieure à un certain seuil, le visage est reconnu comme appartenant à une personne spécifique. Le CNE (ou un identifiant) de la personne est alors affiché à côté du visage détecté. Sinon, le visage est marqué comme inconnu.

➤ La distance euclidienne :

C'est la racine carrée de la somme des carrés des différences entre les coordonnées de chaque dimension.

Dans le cas de la reconnaissance faciale, la distance euclidienne entre deux vecteurs de caractéristiques est utilisée pour mesurer la similarité entre deux visages. En d'autres termes, c'est la racine carrée de la somme des carrés des différences entre les coordonnées de chaque dimension des vecteurs de caractéristiques.

$$d(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

Figure 30 : La distance euclidienne

Avec P et Q sont les vecteurs de caractéristiques de 128 dimensions des visages à comparer.

P : Vecteur de caractéristique du visage connu enregistrer dans la base de données.

Q : Vecteur de caractéristiques du visage détecté, extrait de l'image ou de la vidéo en cours de traitement.

n : Représente le nombre de dimensions du vecteur de caractéristiques. Dans ce cas n=128.



Figure 31 : Visage connu



Figure 32 : Visage inconnu

III. Schéma générale des techniques de reconnaissance faciale utilisés :

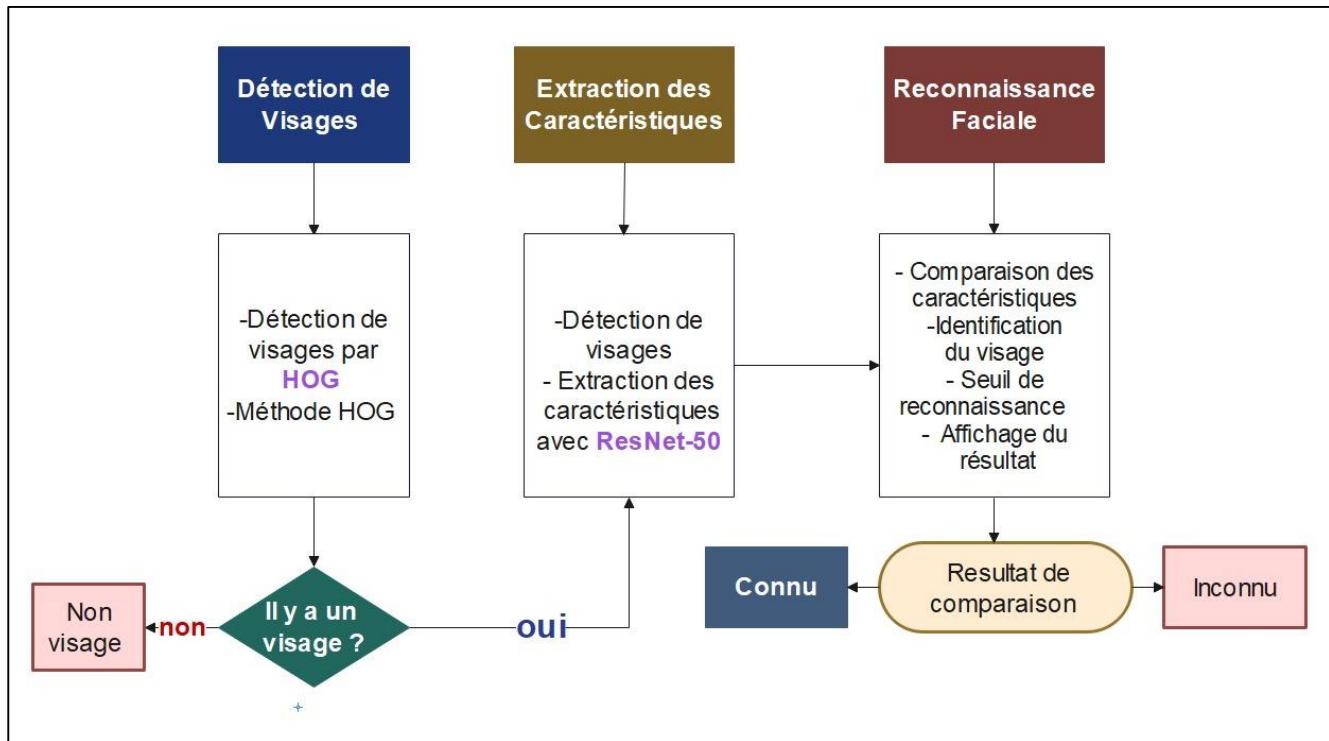


Figure 33 : Schéma générale des technique de reconnaissance facial utilisée

IV. Conception de l'application :

1. Langage de modulation UML :



Figure 34: Logo de langage de modulation UML

L'UML (Unified Modeling Language) langage de modélisation unifie, est un langage permettant de modéliser non classes et leurs interactions. Autrement c'est un ensemble de notation graphique s'appuyant sur les diagrammes et permettant de spécifier, visualiser et documenter les systèmes logiciels.

UML propose plusieurs types de diagrammes, chaque étant utilisé pour présenter un aspect spécifique d'un système logiciel, comme les diagrammes de classe, les diagrammes de séquence, les diagrammes d'activité, ...

En utilisant UML, les développeurs peuvent communiquer efficacement entre eux, ainsi qu'avec les clients, en fournissant une représentation visuelle claire et précise des différents aspects d'un système logiciel.

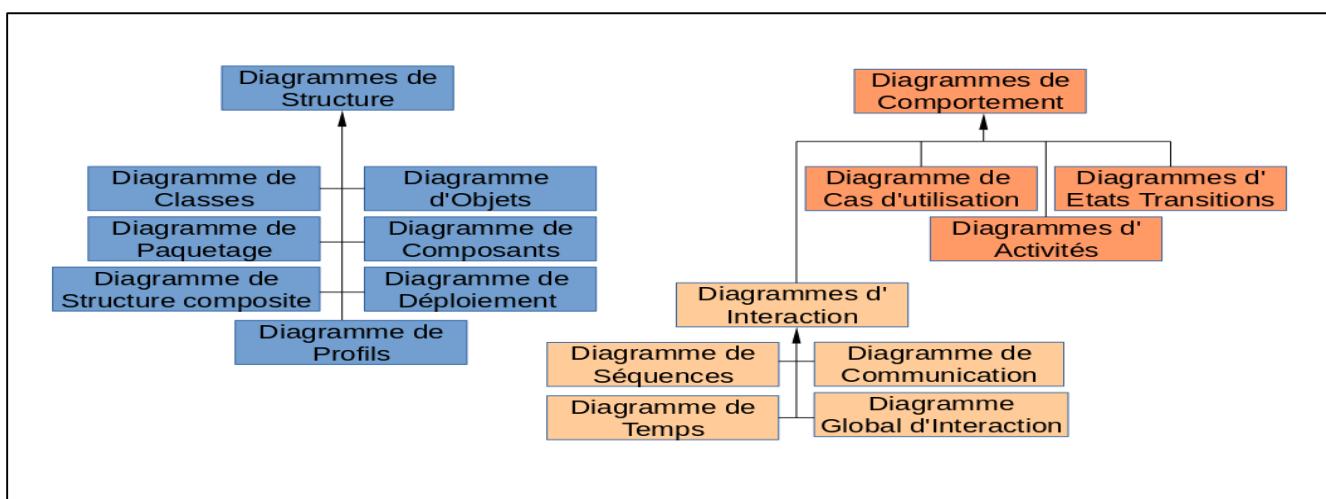


Figure 35: Types des diagrammes UML

2. Diagramme de cas d'utilisation :

a. Définition de diagramme de cas d'utilisation :

Diagramme de cas d'utilisation est un schéma qui représente visuellement les différentes actions qu'un utilisateur peut effectuer dans un système, ainsi que les réponses du système à ces actions.

b. Diagramme de cas d'utilisation d'administrateur :

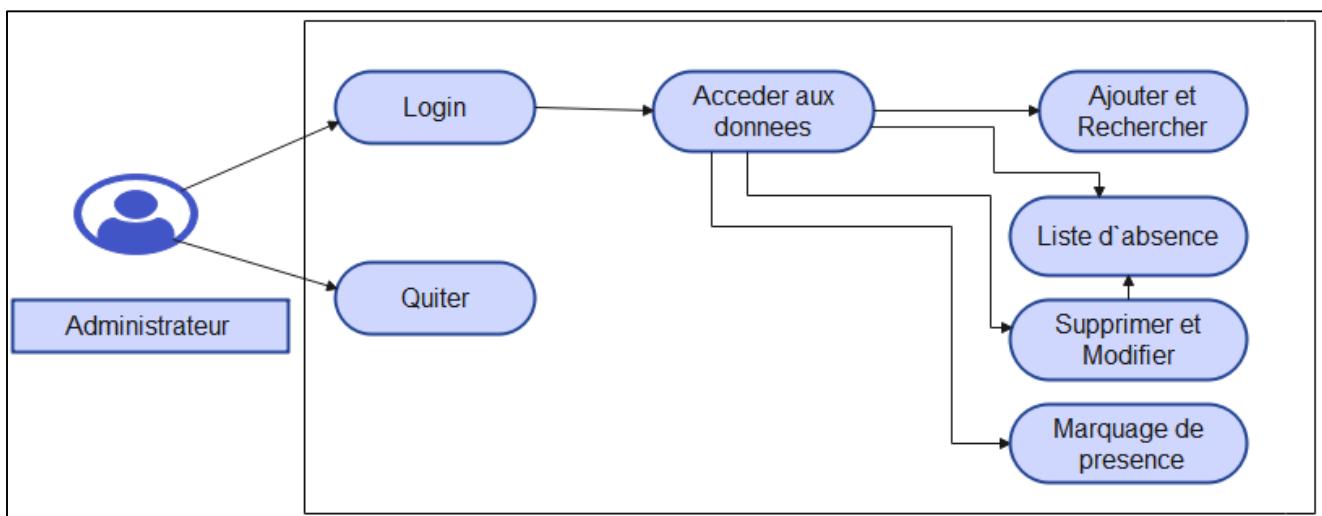


Figure 36: Diagramme de cas d'utilisation d'administrateur

c. Diagramme de cas d'utilisation d'étudiant :

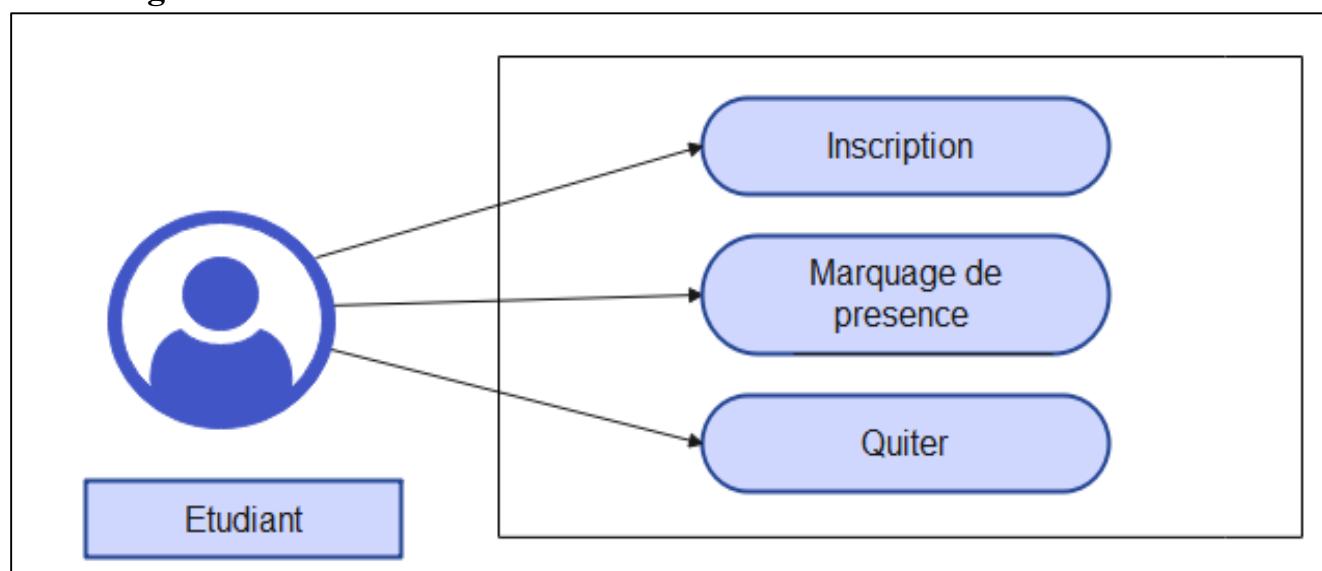


Figure 37: Diagramme de cas d'utilisation d'étudiant

3. Diagramme de séquence :

a. Définition de diagramme de séquence :

Un diagramme de séquence est un type de diagramme d'interaction, il montre simplement comment les différentes parties d'un système travaillent ensemble de manière organisée, cela aide à voir comment les actions se passent au fil du temps. Par exemple le diagramme de séquence pour notre application pourrait montrer comment l'application capture et analyse le visage de l'étudiant, vérifie son identité, puis enregistre sa présence.

b. Diagramme de séquence d'étudiant :

➤ Phase d'inscription :

Dans la figure ci-dessus, l'étudiant choisit à s'inscrire au programme. Le programme va afficher une interface contenant une zone pour saisir les informations d'étudiant (Nom, Prénom, CNE ...), une zone va lancer la caméra pour capturer des images de visage, et créer un nouveau profil dans la base de données.

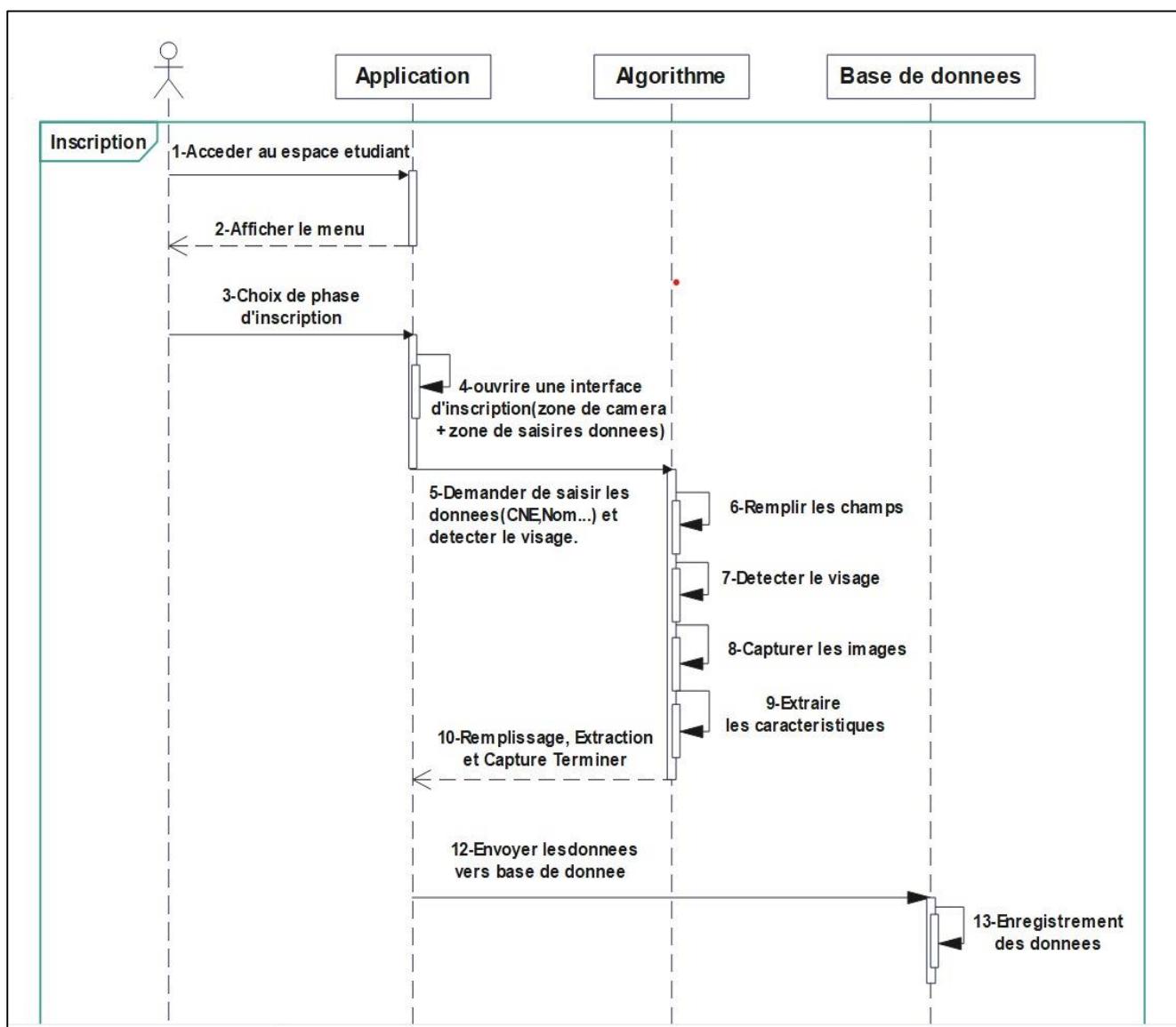


Figure 38: Diagramme de séquence d'étudiant dans la phase d'inscription

➤ Phase de marquage de présence :

Dans la figure ci-dessous, l'étudiant choisit à marquer son présence, le programme va lancer la caméra pour détecter son visage et vérifier s'il existe dans la base de données ou non. S'il existe sa présence sera marquer automatiquement dans la liste de présence, sinon il sera considéré inconnue.

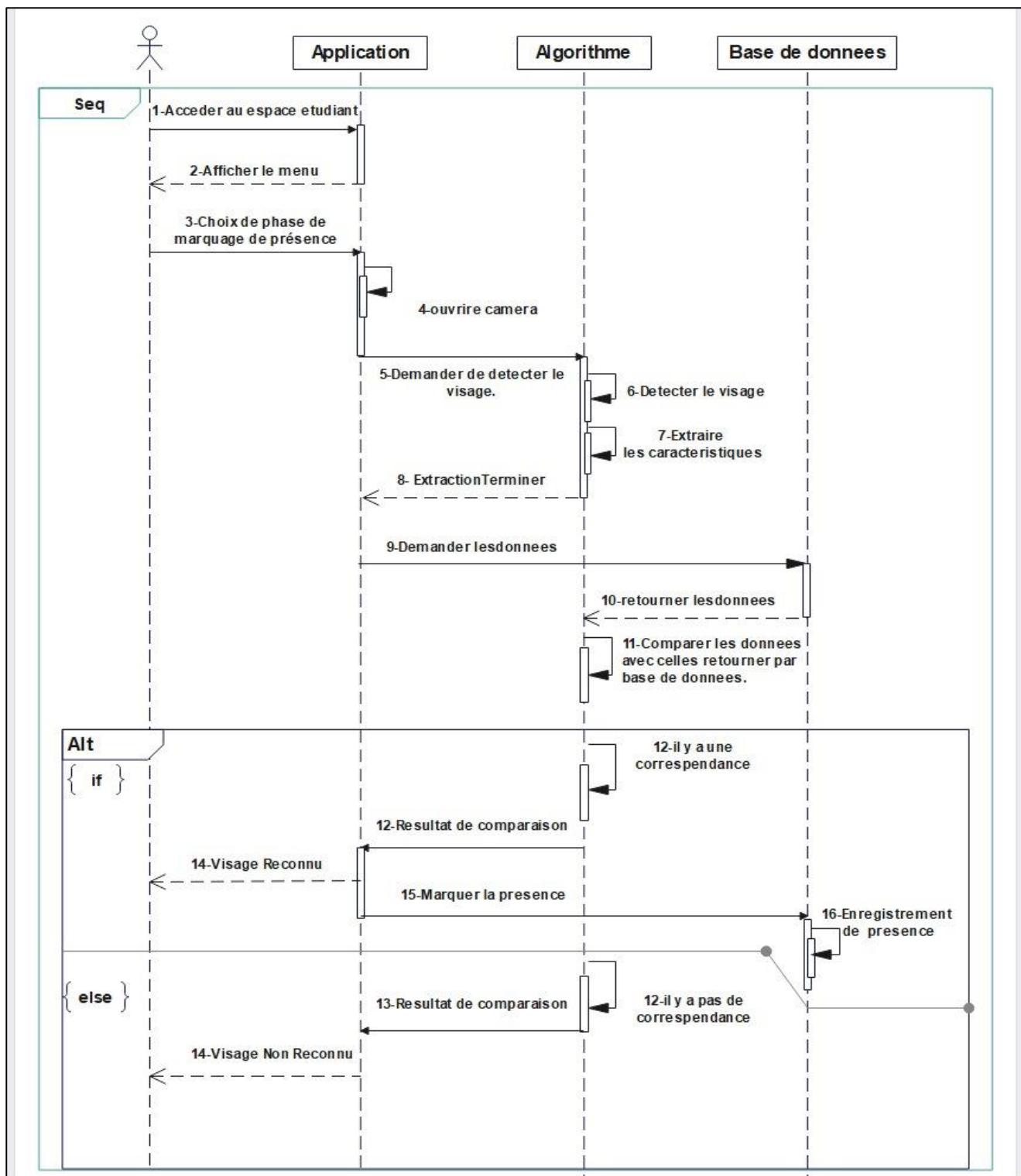


Figure 39: Diagramme de séquence d'étudiant dans la phase de marquage de présence

c. Diagramme de séquence d'administrateur :

➤ Phase d'accéder au liste des étudiants :

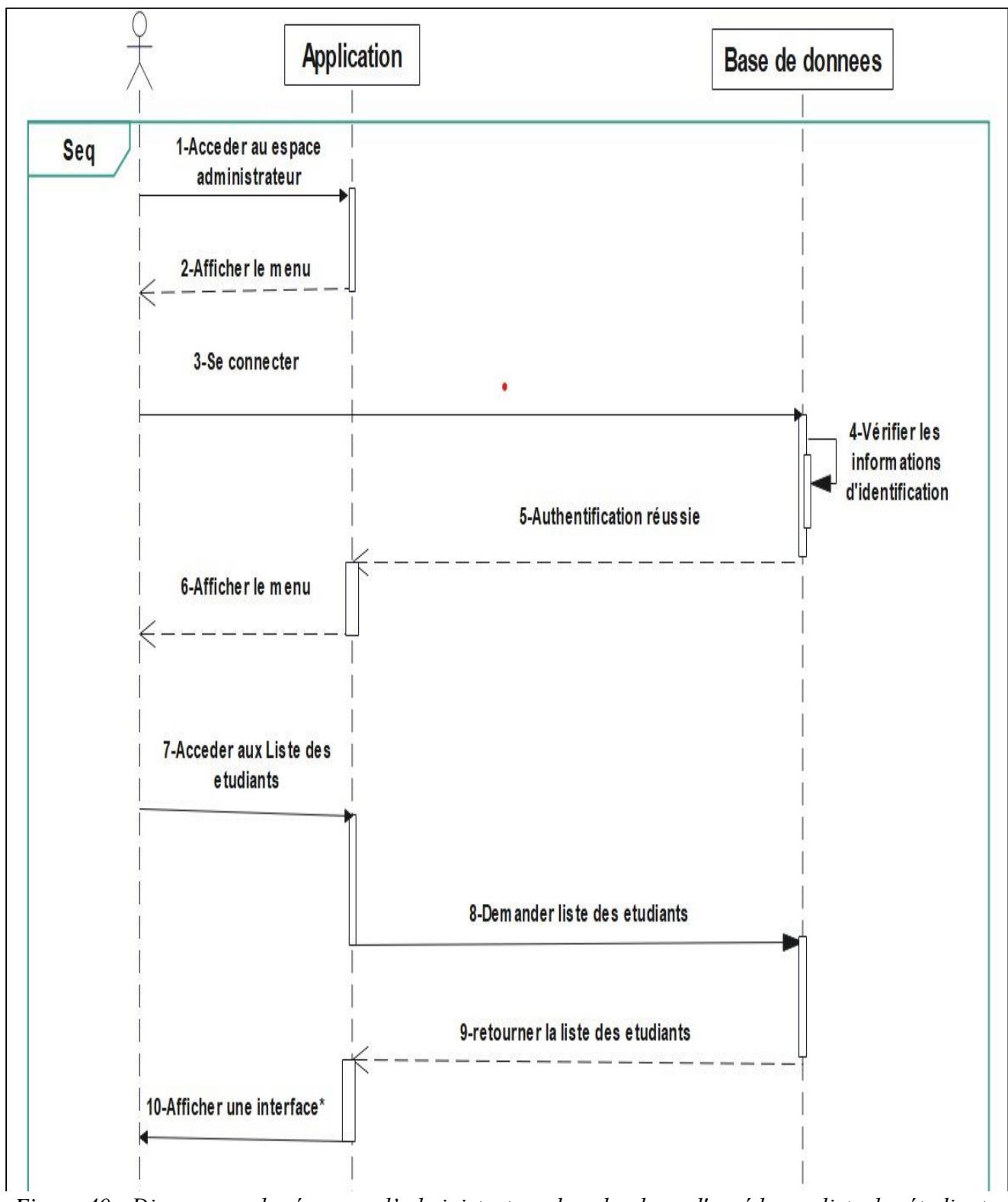


Figure 40 : Diagramme de séquence d'administrateur dans la phase d'accéder au liste des étudiants

Dans la figure ci-dessus, l'administrateur se connecte à l'application. Ensuite, l'application affiche le menu principal pour l'administrateur, qui contient trois options (liste des étudiants, liste de présence, et marquage de présence). Si l'administrateur sélectionne l'option qui permettant d'afficher la liste des étudiants, l'application envoie une requête à la base de données pour récupérer la liste des étudiants. Ensuite l'application affiche cette liste dans une interface permet à l'administrateur de contrôler cette liste en modifiant, ajoutant, supprimant et cherchant les informations des étudiants dans cette liste.

➤ **Phase de marquage de présence :**

Dans cette partie, l'administrateur peut aussi marquer la présence des étudiants, de même manière que on a expliquée dans diagramme de séquence d'étudiant dans la phase de marquage de présence.

➤ **Phase d'accéder au liste de présence :**

Dans la figure ci-dessous, l'administrateur se connecte à l'application. Ensuite, l'application affiche le menu principal pour l'administrateur, qui contient trois options de même manière que dans la phase d'accès à la liste des étudiants. Si l'administrateur sélectionne l'option qui permettant d'afficher la liste de présence, l'application envoie une requête à la base de données pour récupérer la liste des étudiants qui sont présences. Ensuite l'application affiche une interface contient deux options (liste de présence et liste d'absence), permet à l'administrateur d'affiche les listes des étudiants qui sont présences et qui sont absences.

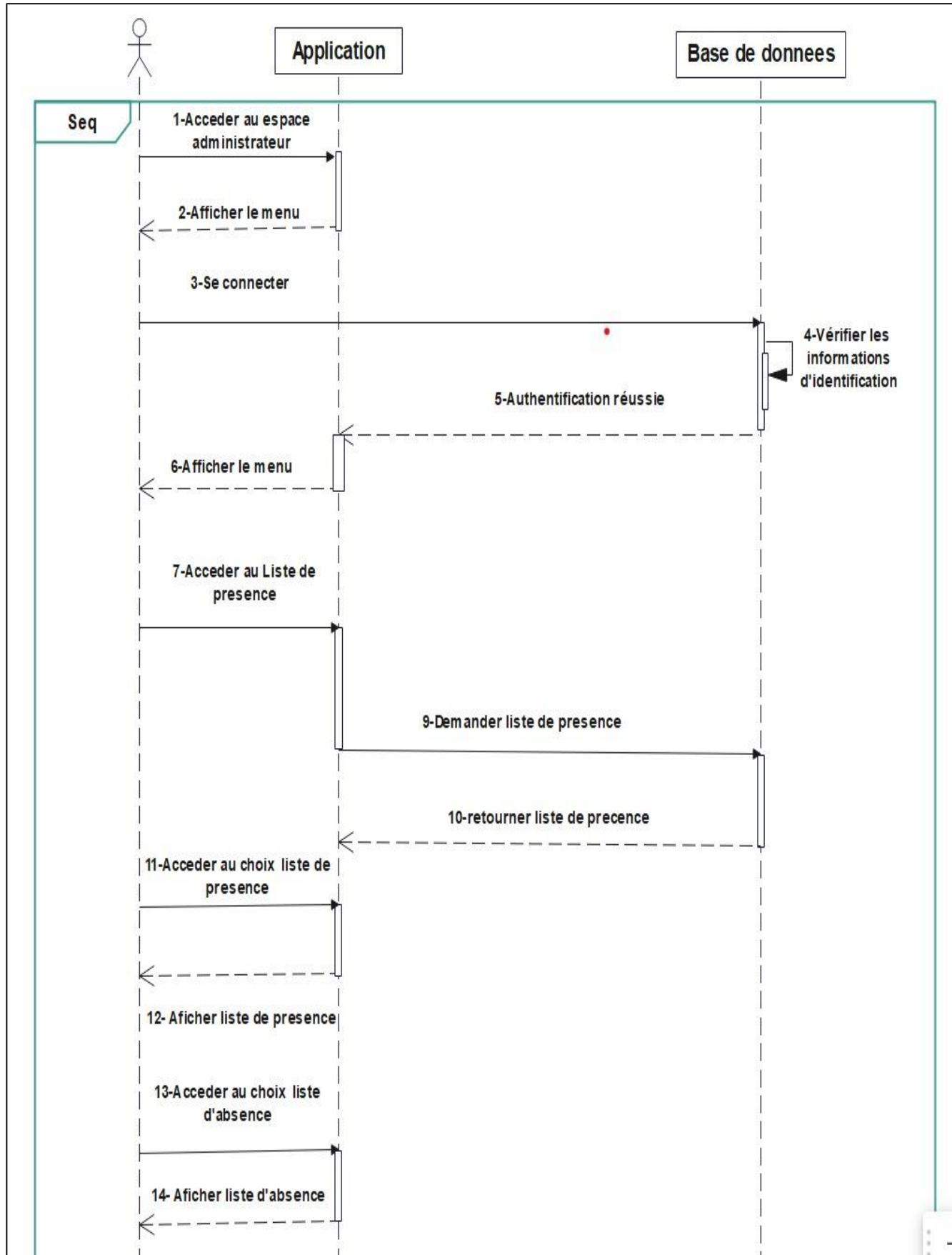


Figure 41: Diagramme de séquence dans la phase d'accéder au la liste de présence

4. Architecture global de l'application :

La structure ou l'architecture d'une application, c'est comme le plan de construction d'une maison, il définit les schémas et les méthodes employés pour concevoir et élaborer une application. Cette architecture globale donne des instructions et des étapes recommandées pour créer une application bien structurée.

Dans l'architecture de notre application, il existe deux passages possibles, soit le cas d'un étudiant ou bien d'un administrateur. Par exemple un étudiant il peut quitter le programme, s'inscrire ou marquer sa présence.

Dans l'espace d'étudiant, le cas de l'inscription est réservé aux nouveaux étudiants. A cette étape, le programme demande à l'étudiant de remplir ses informations telles que CNE, Nom, Prénom, Email, Tel. Après cette étape, le programme demande à l'étudiant de présenter son visage devant la caméra afin de prendre des images. En fin le programme extrait les caractéristiques de son visage et les informations d'étudiant et les enregistrées dans la base de donnée.

À la phase de marquage de présence, qui est réservée pour les étudiants qui sont déjà inscrits. Pour marquer la présence, l'étudiant doit présenter son visage face à la caméra. Le système extrait et calcule les caractéristiques de son visage, et les comparent avec celles déjà existantes. S'il y a une ressemblance, le visage est reconnu et par conséquence la présence d'étudiant sera marquée dans la base de données.

Pour l'administrateur, il peut également quitter le programme ou se connecter pour accéder et visualiser des informations telles que la liste des étudiants, liste de présence et d'absence. De plus, il peut marquer la présence des étudiants de la même manière que la prise de présence effectuée par l'étudiant lui-même.

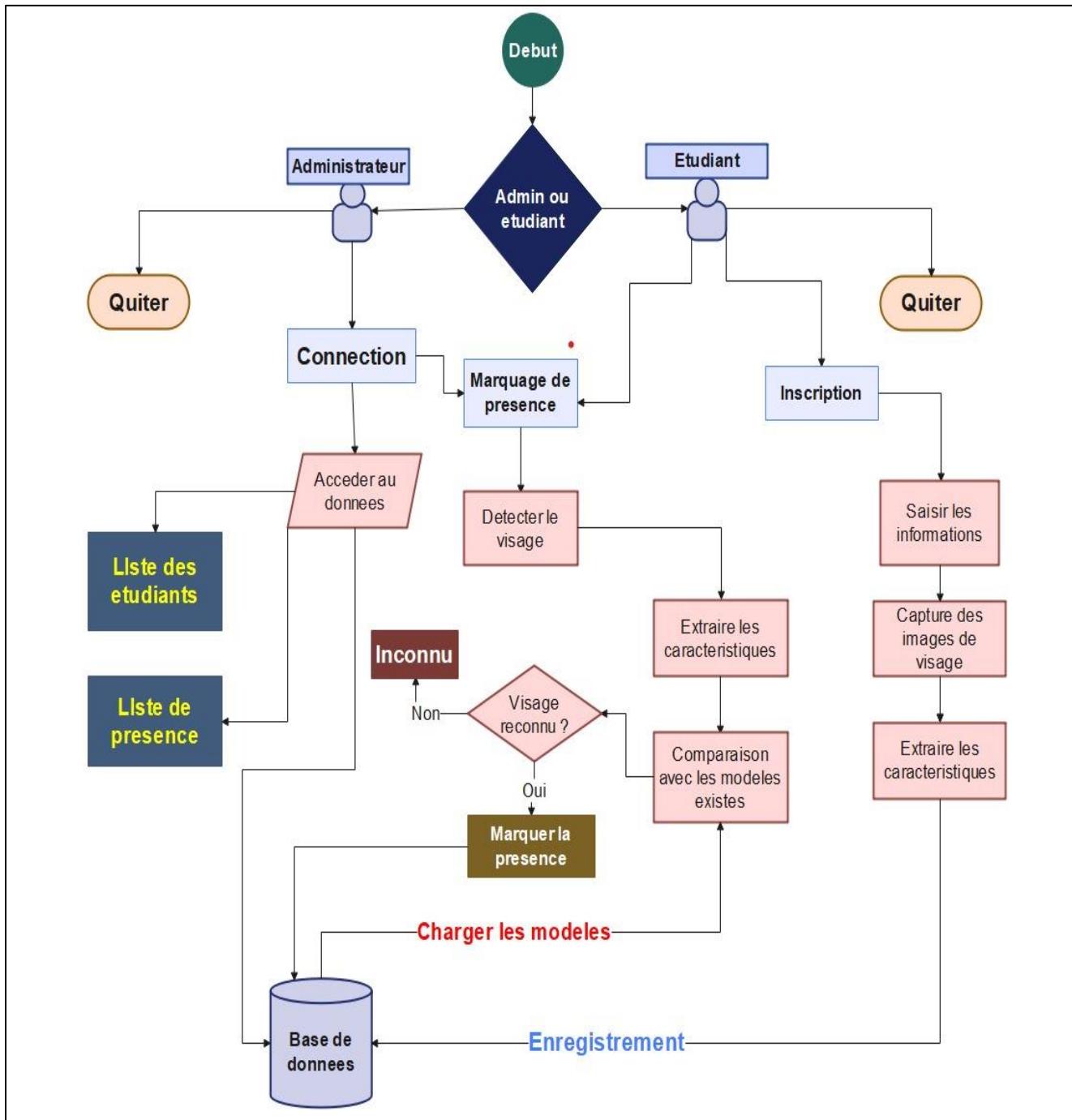


Figure 42 : Architecture globale de l'application

V. Conclusion :

Dans cette chapitre, nous avons expliqué les techniques et les algorithmes pour détecter les visages et reconnaître les étudiants dans notre application, appuyées par les diagrammes d'utilisation qui modélisent les interactions entre les utilisateurs et le système, et les diagrammes de séquence qui montrent simplement comment les différentes parties d'un système travaillent ensemble de manière organisée. Enfin, nous avons présenté l'architecture globale de l'application, offrant ainsi une synthèse claire de sa conception et de ses capacités.

CHAPITRE IV : PROGRAMMES ET OUTILS UTILISÉS

I. Introduction :

Dans cette chapitre, nous détaillerons les programmes et les outils et les bibliothèques que nous avons utilisées dans notre travaille. On a utilisé essentiellement le langage python qui contient plusieurs bibliothèques nous a aidés dans le processus d'identification et de reconnaissance des visages comme Opencv, dlib, NumPY,

II. Langage Python :

1. Définition de Python :

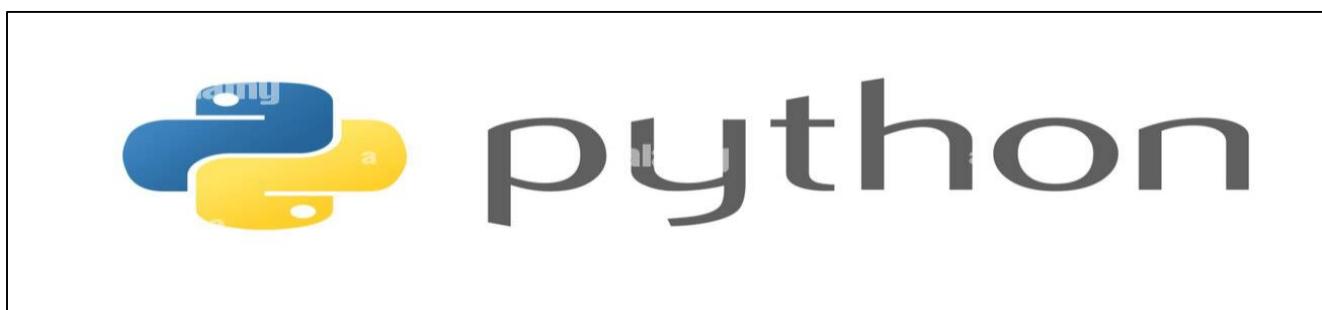


Figure 43 : Logo de langage python

Python est un langage de programmation de haute niveau populaire, crée par le programmeur Guido van Rossum en 1991 [5]. Il s'agit d'un langage de programmation interprété, qui ne nécessite pas d'être compilé pour fonctionner. Ceci permet de voir rapidement les résultats dans le code. Python possède plusieurs bibliothèques qui facilite le développement de programme. En raison de sa simplicité, Python est devenu l'un des langages de programmation les plus utilisés dans le monde.

2. Les avantages de Python :

Python est souvent un langage de premier choix pour les développeurs qui veulent crée des applications, car Python de nombreux avantages [5] :

- **Facilité d'apprentissage :** La syntaxe de Python est simple et facile à comprendre.
- **Polyvalent :** Python est polyvalent et peut être employé pour une multitude de tâches, comme le développement web, l'analyse de données, l'intelligence artificielle, ... etc.
- **Bibliothèques :** Python offre une grande collection de modules et de fonctions, ce qui permet aux développeurs d'économiser du temps en évitant de recréer des fonctions déjà disponibles.
- **Facilité d'intégration :** Python combine facilement avec d'autres langages, ce qui le rend parfait pour travaille sur des projets complexe.

III. Visual Studio Code :

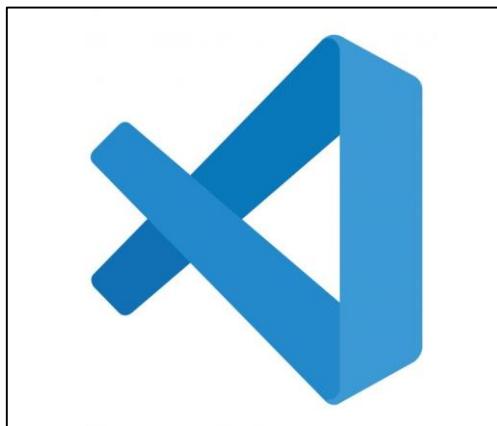


Figure 44 : logo de visual studio code

1. Définition de Visual Studio Code :

Visual studio code est un éditeur de code extensible développé par Microsoft. C'est un logiciel gratuit utilisé par les développeurs pour créer toutes sortes de programmes, des sites web aux applications mobiles...

Grâce à son système d'extensions, Visual Studio Code peut être étendu pour prendre en charge pratiquement n'importe quel langage de programmation, c'est pourquoi tant de développeurs l'utilisent.

2. Les étapes d'installation de Visual Studio Code :

➤ Étape 1 :

Allez à la page de Microsoft Visual Studio Code sur Academic Software et cliquez sur le bouton 'Télécharger Visual Studio Code' pour télécharger le logiciel. Ouvrez le fichier d'installation .exe dans votre dossier Téléchargements.

[6]

➤ Étape 2 :

Lisez et acceptez le contrat de licence et cliquez sur Next. Vous pouvez modifier l'emplacement du dossier d'installation ou conserver les paramètres par défaut. Cliquez sur Next pour continuer.

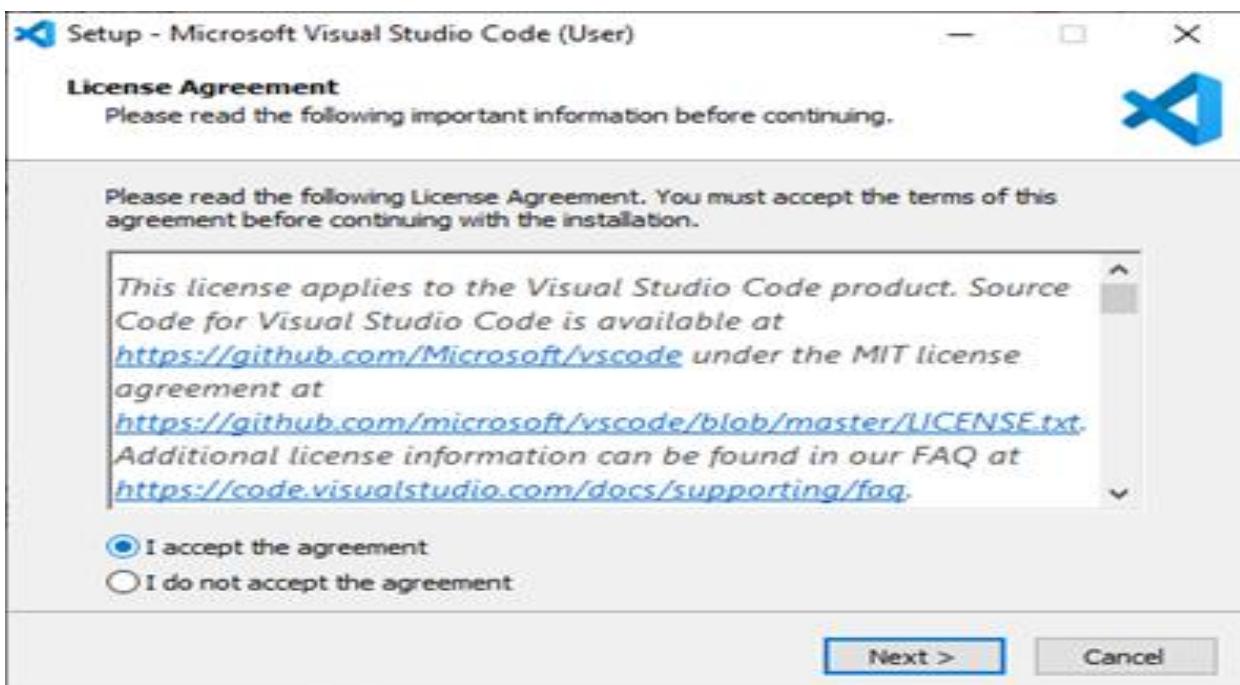


Figure 45 : Acceptation du contrat

➤ Étape 3 :

Sélectionnez les tâches supplémentaires, par exemple 'Create a desktop icon' (créer une icône sur le bureau) ou ajouter des options au menu du clic droit de l'explorateur Windows. Cliquez sur Next.

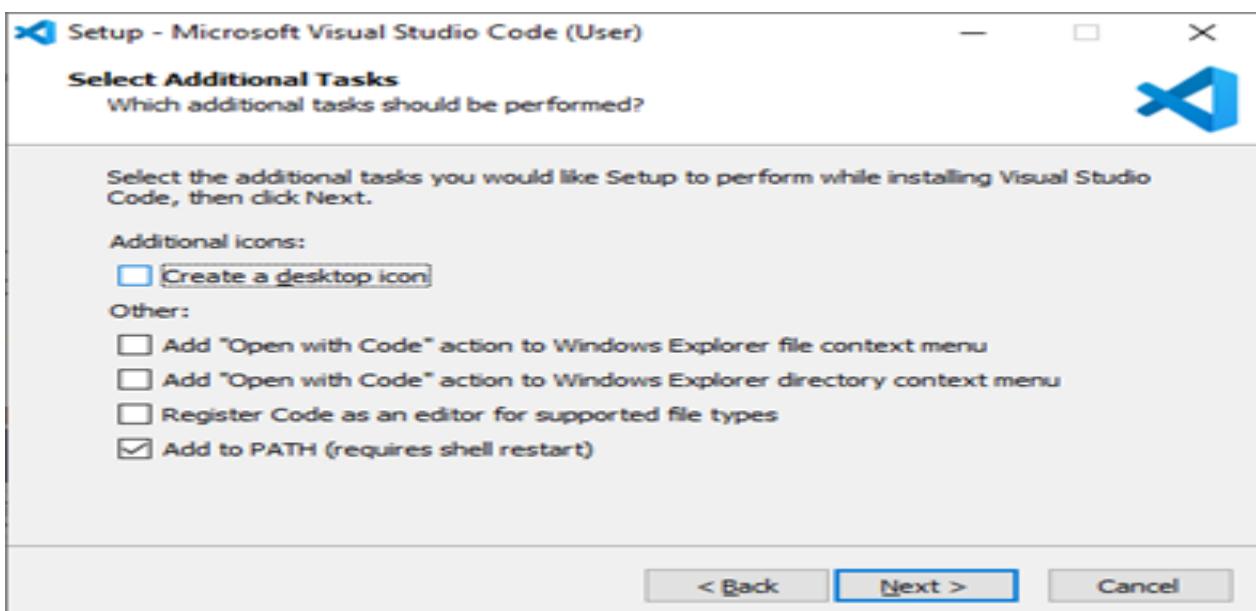


Figure 46 : Sélectionner les tâches supplémentaires

➤ **Étape 4 :**

Cliquez sur Install pour commencer l'installation du logiciel.

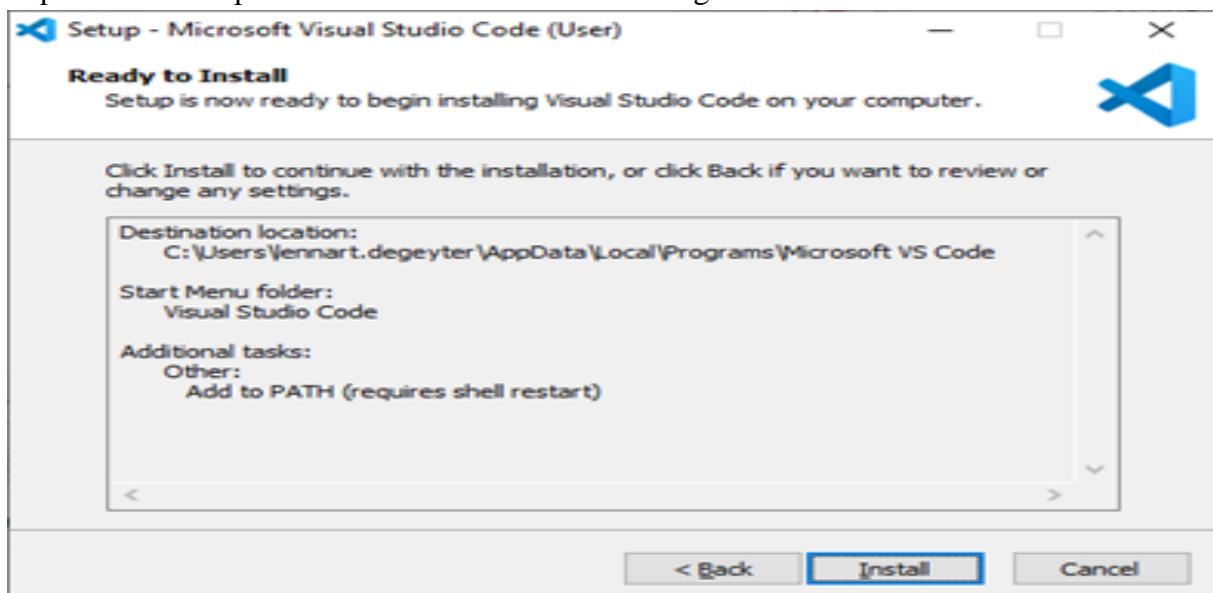


Figure 47 : Installation du logiciel

IV. GanttProject :



Figure 48: Logo de GanttProject

1. Définition de GanttProject :

GanttProject est un logiciel de gestion de projet gratuit utilisé pour planifier et gérer des projets. Il permet aux utilisateurs de créer des diagrammes de Gantt, qui sont des représentations visuelles des tâches d'un projet sur un linge de temps.

2. Les avantages de GanttProject :

GanttProject présente plusieurs avantages pour la gestion de projet :

- **Planification visuelle** : Grâce à ses diagrammes de Gantt, GanttProject offre une représentation visuelle claire des tâches de projet.

- **Gratuit :** GanttProject est disponible gratuitement, ce qui le rend accessible à tous.
- **Suivre de l'avancement :** permet de suivre l'avancement réel du projet par rapport au planning initiale.
- **Personnalisable :** avec GanttProject, vous pouvez ajuster les réglages de votre besoin.
- **Exportation et partage :** offre la possibilité d'exporter les diagrammes de Gantt au format image, PDF ce qui facilite le partage des informations avec les membres de l'équipe.

V. EDRAW MAX :



Figure 49 : Logo de EDRAW MAX

1. Définition de Edraw Max :

EdrawMax est un logiciel de création de diagrammes technique commerciaux 2D qui aide à créer des organigrammes, des cartes mentales, des diagrammes de réseau, des plans d'étages, des diagrammes de flux de travail, des graphiques commerciaux des diagrammes d'ingénierie.

2. Les étapes d'installation de Edraw Max :

➤ Étape 1 :

Allez à la page de Edraw Max et cliquez sur le bouton Démarrer l'essai gratuit pour télécharger le logiciel.

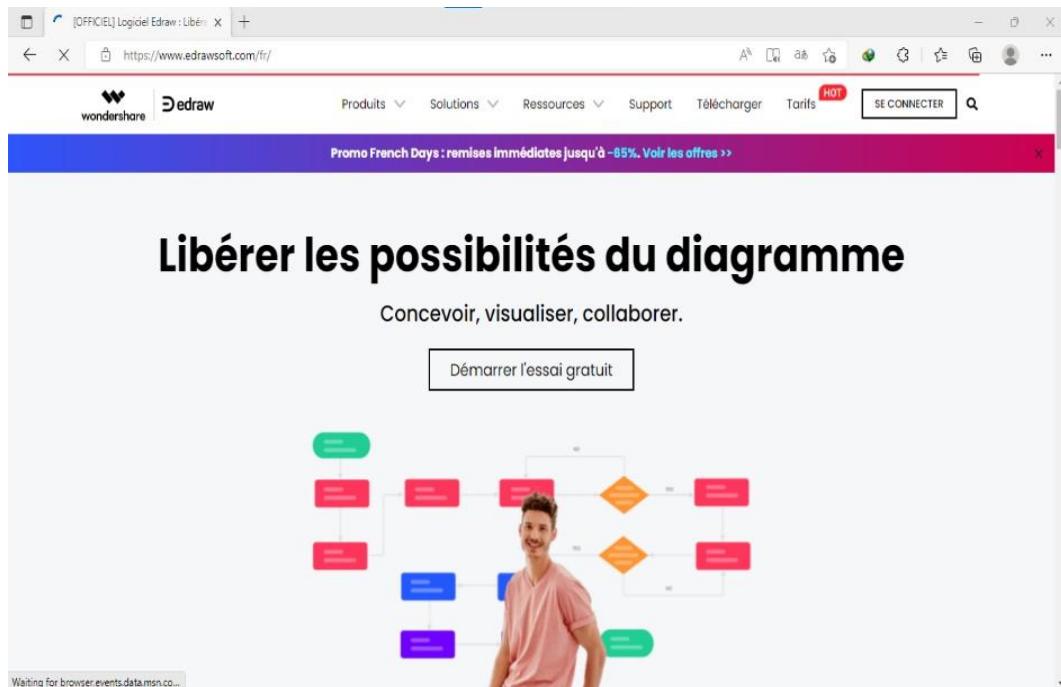


Figure 50 : L'accès à la page d'installation

➤ Étape 2 :

Cliquez sur le bouton Essai gratuit. Ouvrez le fichier d'installation .exe dans votre dossier Téléchargements.

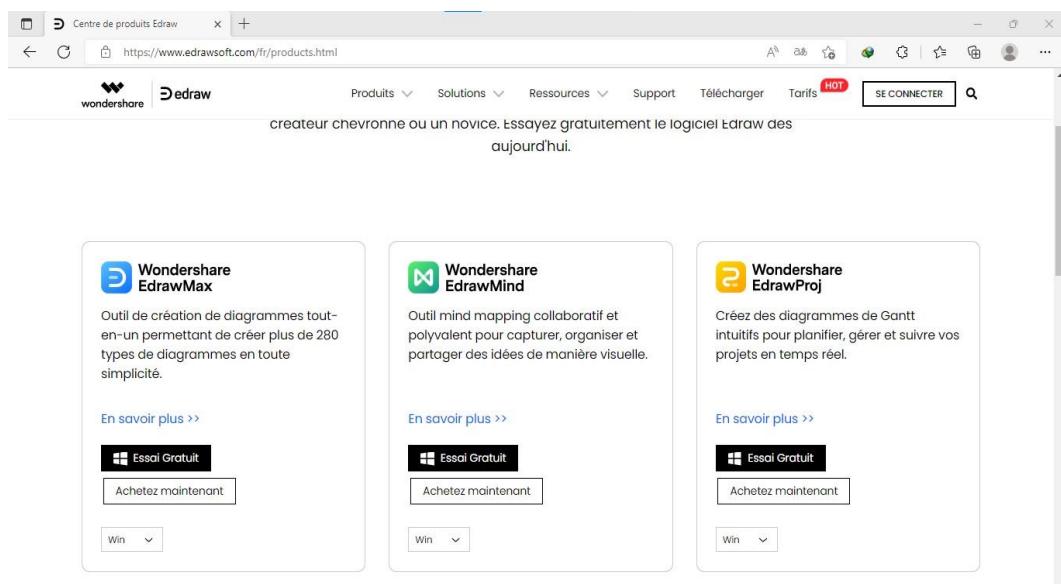


Figure 51 : Choix de type de téléchargement

Étape 3 :

Installation de programme.

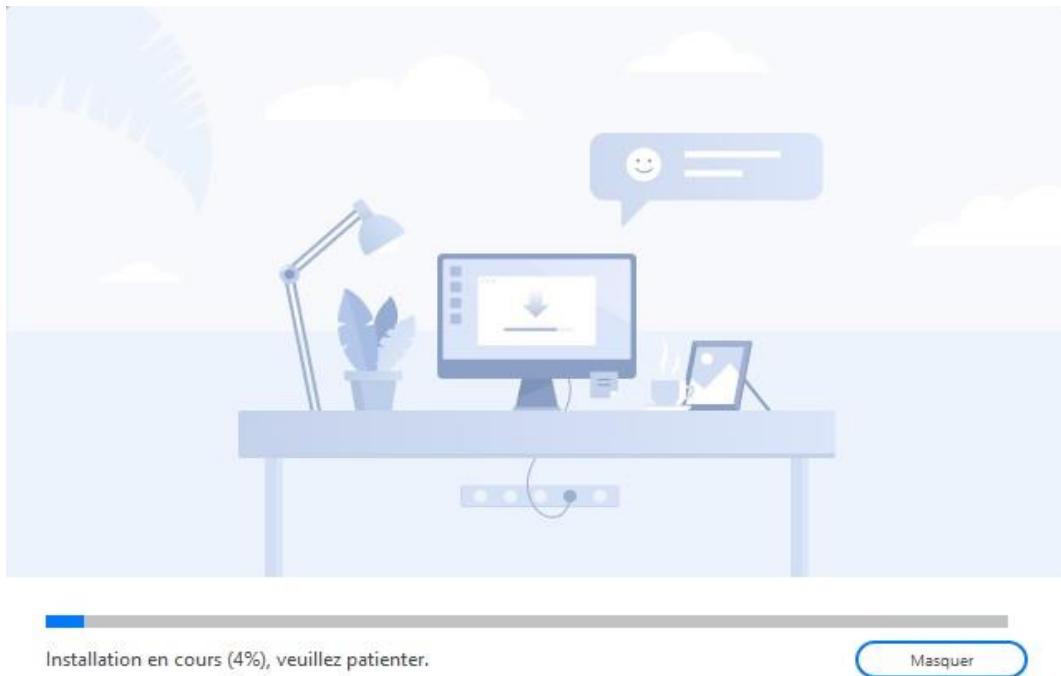


Figure 52 : Installation

VI. MySQL :

MySQL est un système de gestion de base de données open-source très populaire. Il permet de stocker et de gérer des données de manière structurée. MySQL utilise le langage SQL, qui est un langage de programmation utilisé pour communiquer avec la base de données, permettant de créer, supprimer et modifier les données par des outils de conception visuelle de base de données comme MySQL Workbench qui on a utilisé dans notre travaille.



Figure 53 : Logo de MySQL

1. Les avantages de MySQL :

MySQL a plusieurs avantages, ce qui en fait un outil largement utilisé des logiciels et des bases de données, voici quelques-uns de ses avantages :

Les données sont stockées de manière structurée grâce à une base de données, ce qui facilite leur stockage et leur récupération.

Pour garder vos données en sécurité, les bases de données offrent des fonctions de protection comme la gestion des utilisateurs et des permissions d'accès.

En utilisant des requêtes SQL, il offre la possibilité de faire des commandes et d'analyser des données de façon efficace, ce qui rend plus simple l'extraction des informations de base de données.

L'intégration des bases de données avec d'autres systèmes et d'application facilite le partage et l'utilisation des informations.

2. MySQL Workbench :

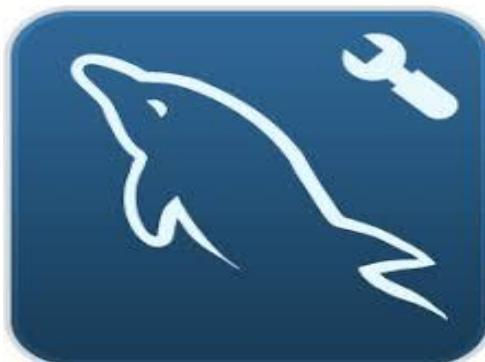


Figure 54 : Logo MySQL Workbench

MySQL Workbench est une application qui permet de concevoir, de modéliser et de gérer des bases de données MySQL. Il met à disposition une interface graphique permettant de créer, éditer et exécuter des requêtes SQL, ainsi que de gérer les utilisateurs, les permissions et les schémas de base de données comme des diagrammes qui représentent la structure logique d'une base de données.

VII. Bibliothèques de python utilisés :

Les bibliothèques python sont des modules et de fonction prédéfinies qui permettent aux développeurs d'effectuer des tâches spécifiques sans avoir à réécrire le code à zéro.

Les bibliothèques python se présentent sous différentes formes, certaines bibliothèques se concentrent sur les domaines spécifiques, tels que l'analyse de données, l'apprentissage automatique, le développement web ou le calcul scientifique. Parmi les bibliothèques python les plus populaires, on trouve NumPy pour le calcul scientifique, Flask et Django pour le développement web, OpenCV est plus axée sur le traitement d'images et la vision par ordinateur, Dlib pour l'apprentissage automatique et la reconnaissance des formes.



Figure 55 : Les 10 meilleures bibliothèques de python

1. La bibliothèque OpenCV :

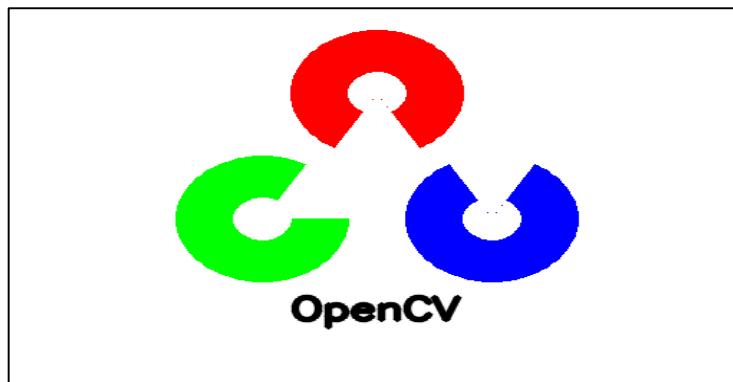


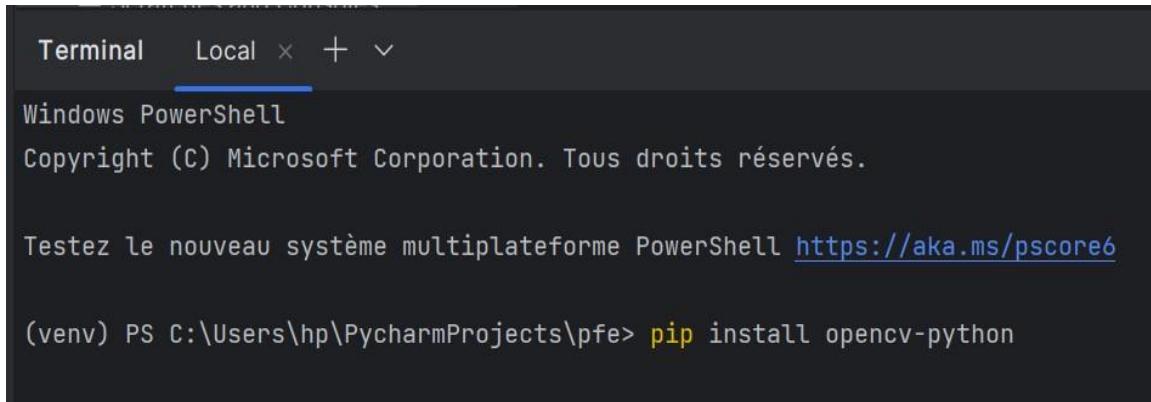
Figure 56 : Logo de OpenCV

a. Définition de OpenCV :

OpenCV (open source computer vision library) est une bibliothèque open source populaire pour les tâches de vision par ordinateur et traitement d'images. OpenCV prend en charge une grande variété de langages de programmation python, C++, Java, etc. Elle offre une vaste collection d'algorithme et de fonction qui permettent aux développeurs d'effectuer diverses tâches de vision par ordinateur, notamment le traitement d'images et de vidéo, la détection et le suivi d'objet, la reconnaissance faciale, l'extraction de caractéristiques, ...

b. Installation :

- Pour installer :



```
Terminal Local × + ▾
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

(venv) PS C:\Users\hp\PycharmProjects\pfe> pip install opencv-python
```

Figure 57: Installation OpenCV

➤ Pour l'importer :

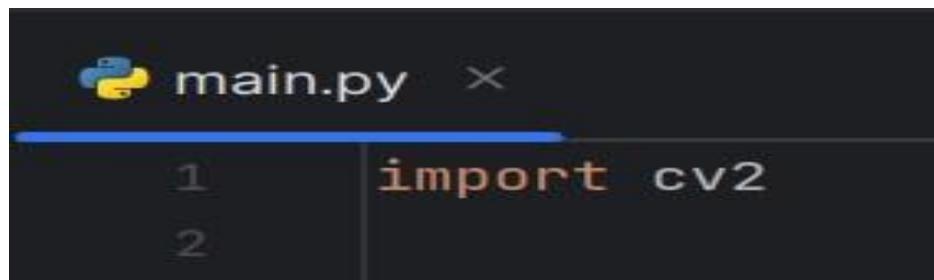


Figure 58 : Importation de OpenCV

2. La bibliothèque Dlib :



Figure 59 : Logo de Dlib

a. Définition de Dlib :

La bibliothèque Dlib est une bibliothèque logicielle open-source écrite en langage C++ avec des liaisons pour Python. Elle est principalement utilisée pour la vision par ordinateur, essentiellement pour la détection de visages, la reconnaissance faciale.

La bibliothèque Dlib est un choix populaire pour de nombreuses tâches de vision par ordinateur, offrant à la fois des performances élevées et une facilité d'utilisation grâce à son intégration avec Python.

b. Installer :

```
C:\Users\          >pip install dlib
Defaulting to user installation because normal site-packages is not writeable
Collecting dlib
  Using cached dlib-19.24.0.tar.gz (3.2 MB)
    Preparing metadata (setup.py) ... done
  Using legacy 'setup.py install' for dlib, since package 'wheel' is not installed.
  Installing collected packages: dlib
    Running setup.py install for dlib ... done
  Successfully installed dlib-19.24.0
WARNING: You are using pip version 22.0.4; however, version 22.1.2 is available.
You should consider upgrading via the 'C:\Program Files\Python310\python.exe -m pip install

```

Figure 60 : Installation de Dlib

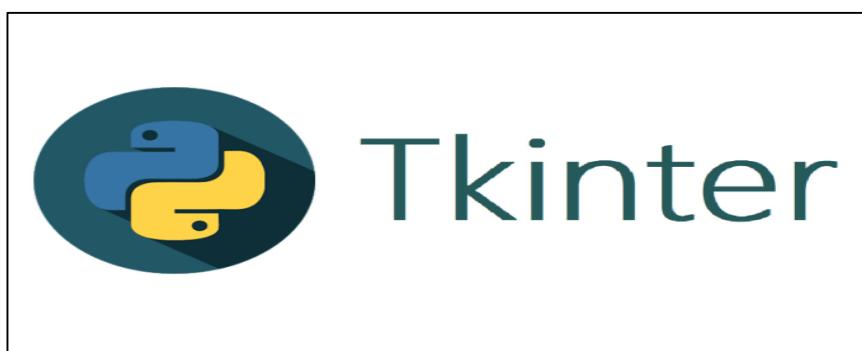
3. La bibliothèque Tkinter :

Figure 61 : Logo de Tkinter

Tkinter est une bibliothèque de Python classique qui permet de créer des interfaces graphiques. Offre la possibilité de concevoir des applications comprenant des fenêtres, des boutons, des menus et d'autre éléments d'interface utilisateur graphique. Tkinter est inclus avec la distribution standard de Python, ce qui signifie qu'il est disponible par défaut, sans nécessiter de téléchargement ou d'installation supplémentaire.

4. La bibliothèque NumPy :



Figure 62 : Logo de NumPy

a. Définition de NumPy

NumPy (numerical python) est une bibliothèque fondamentale pour le calcul scientifique en python. Il prend en charge les grands tableaux et matrices multidimensionnels, ainsi qu'un ensemble de fonctions mathématiques pour opérer efficacement sur ces tableaux.

NumPy est largement utilisé dans plusieurs domaines de l'informatique scientifique et numérique, l'apprentissage automatique, le traitement d'images. Il fournit une base solide pour travailler avec des données numériques et permet aux développeurs d'écrire un code efficace pour des opérations mathématiques complexes.

b. Installation :

- Pour installer :

A screenshot of a Windows PowerShell window. The title bar says "Terminal Local x +". The window shows the following text:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

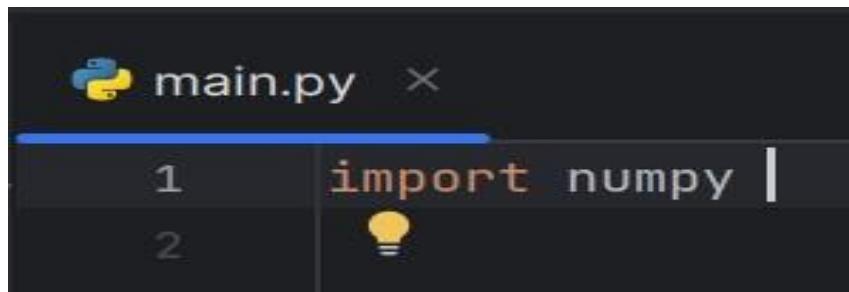
Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

(venv) PS C:\Users\hp\PycharmProjects\pfe> pip install numpy
```

The command "pip install numpy" is highlighted with a cursor.

Figure 63 : Installation de NumPy

- Pour l'importer :



A screenshot of a code editor window titled "main.py". The code editor shows two lines of Python code: "1 import numpy |" and "2". A yellow lightbulb icon is positioned next to the second line, indicating a potential error or warning.

Figure 64 : Importation de NumPy

5. Pandas :

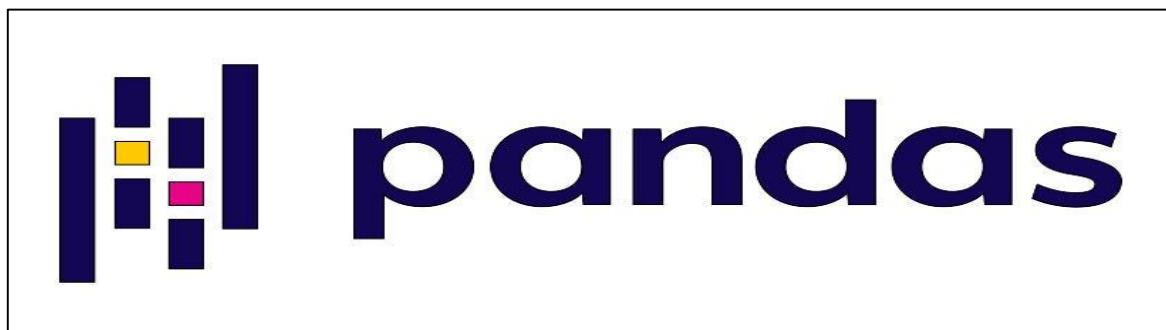


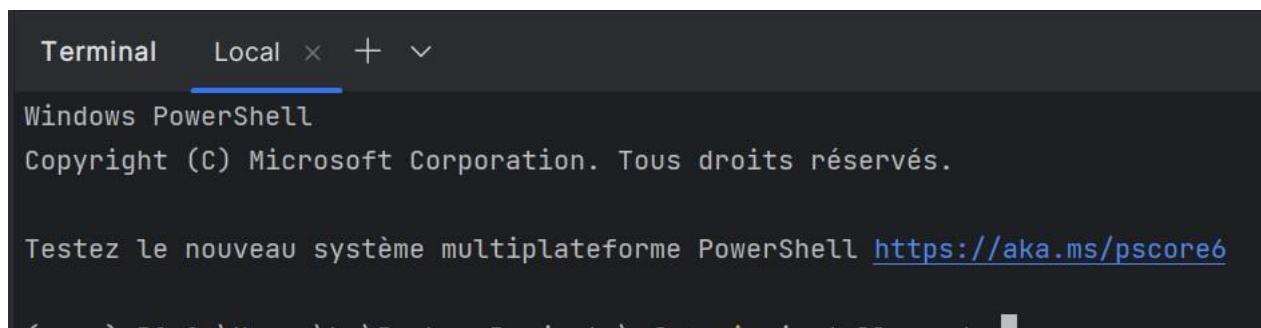
Figure 65: Logo de pandas

a. Définition de Pandas :

La bibliothèque logiciel open-source Pandas est un outil informatique libre très utilisé pour trier et étudier des informations en python. Il offre des façons efficaces de stocker des données, comme les dataframes, et propose de nombreuses fonction pour travailler avec des informations organisées. Très prisé dans divers domaines comme l'apprentissage automatique et la science des données, Pandas rend plus simple et rapide le traitement et l'analyse de données en fournissant une interface pratique. Avec pandas, on peut gérer des données du monde réel, les nettoyer, les explorer et les transformer facilement.

b. Installation :

- Pour installer :

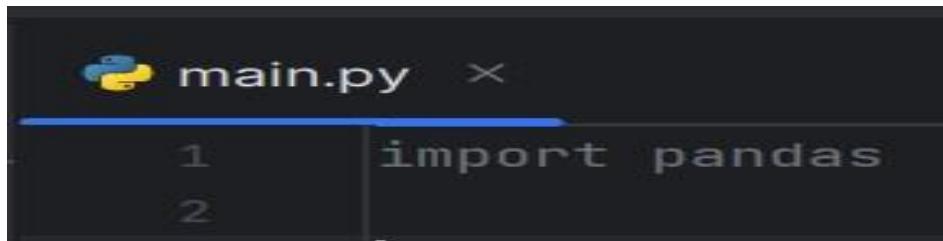


A screenshot of a Windows PowerShell terminal window. The title bar says "Terminal Local x + ▾" and "Windows PowerShell". The content of the terminal shows the following text:
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell <https://aka.ms/pscore6>

Figure 66 : Installation de pandas

➤ Pour l'importer :



```
main.py ×
1 import pandas
2
```

Figure 67 : Importation de pandas

VIII. Conclusion :

Dans cette partie de notre projet, nous avons mentionné les outils, les moyens et les applications que nous avons utilisés, ainsi que les étapes d'installations et les avantages qui nous ont incités à les choisir par rapport à d'autres, afin d'obtenir une application facile à utiliser

CHAPITRE V : IMPLÉMENTATION

I. Introduction :

Dans le cadre de ce chapitre, nous essayons de montrer comment notre application fonctionne où on va présenter ses différentes interfaces. Ces dernières sont les points d'entrée essentiels où l'utilisateur peut effectuer ses tâches, naviguer dans l'application et accéder aux fonctionnalités principales. Nous avons veillé à ce que ces interfaces soient faciles, simples et pratiques à utiliser. Cela permet à l'utilisateur de gérer notre application d'une manière fluide et productive.

II. Les interfaces de l'application :

1. Interface principale :

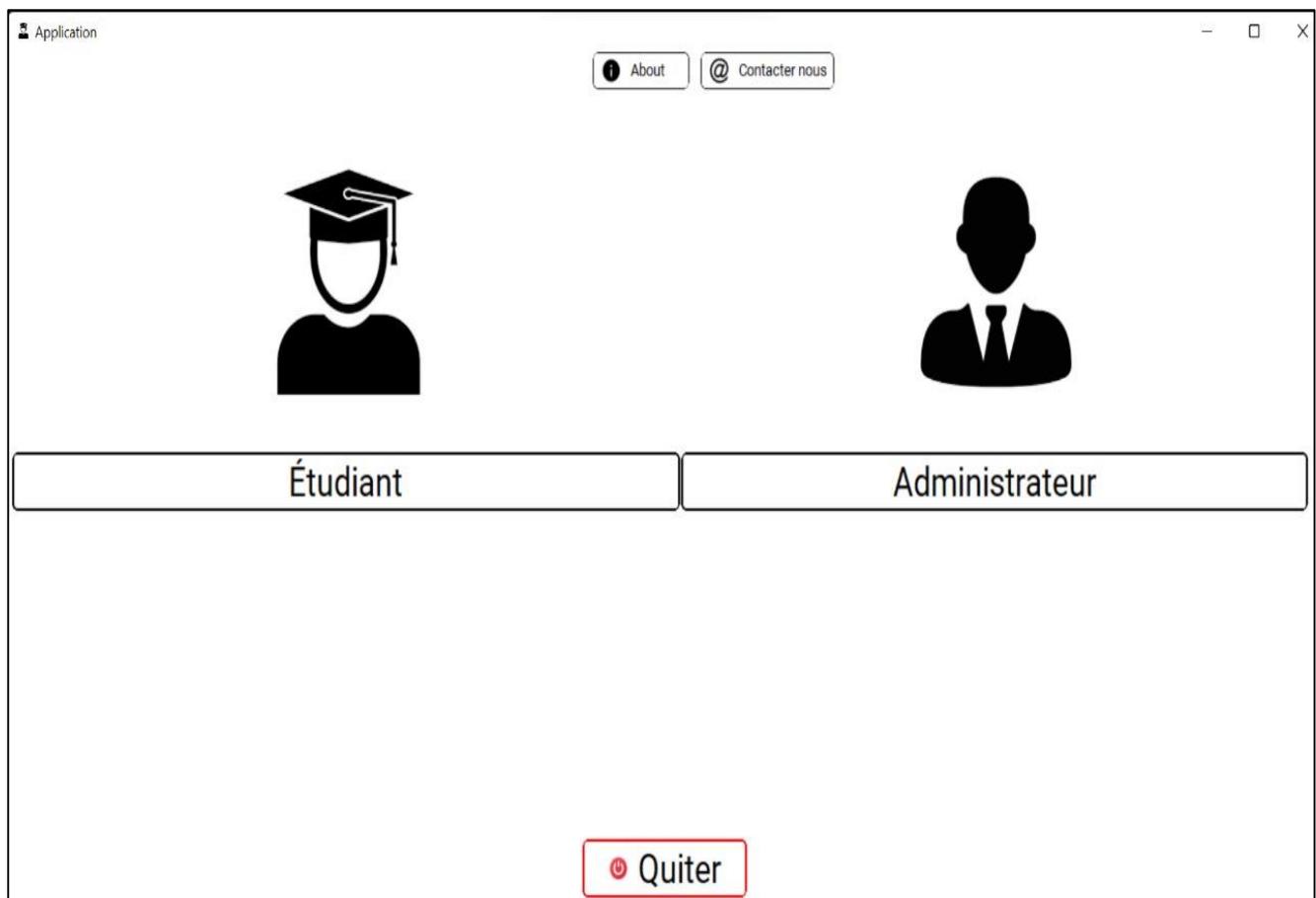


Figure 68 : Interface principal

2. Interface d'administrateur

a. Interface de connexion :

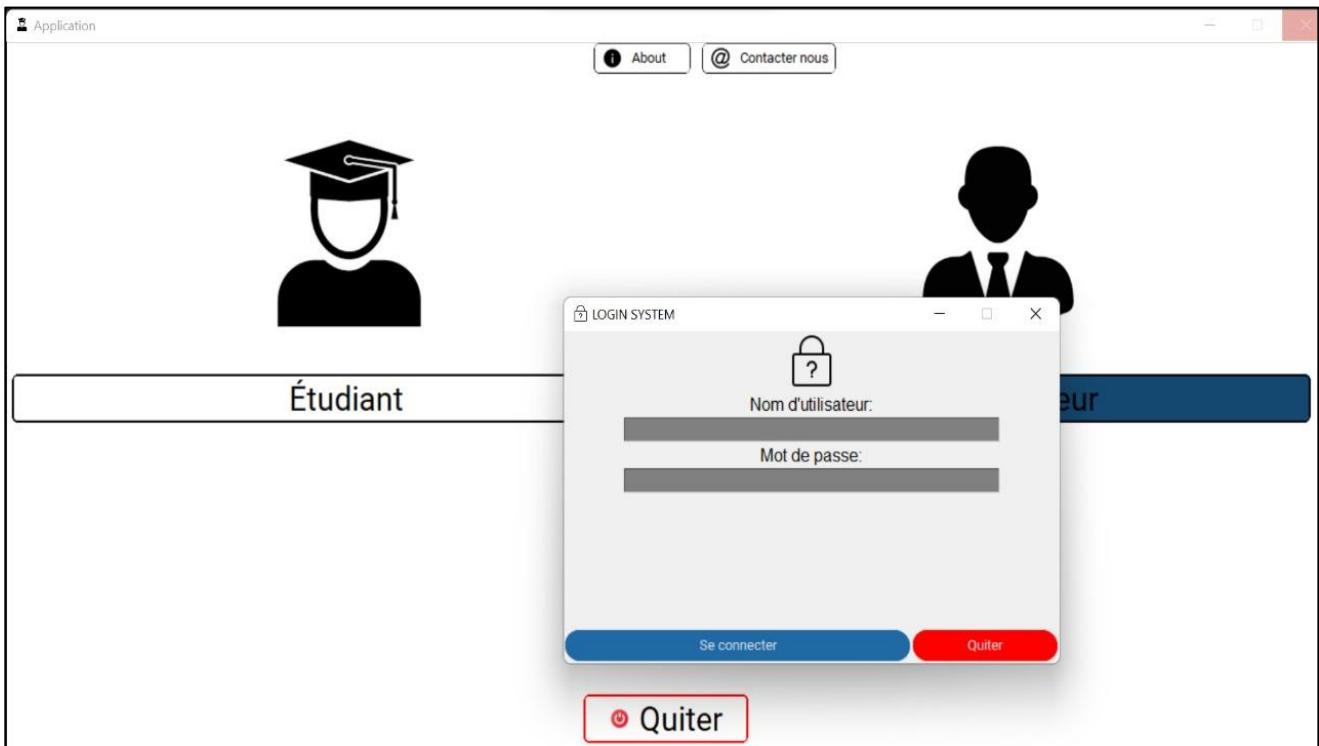


Figure 69 : Interface de connexion

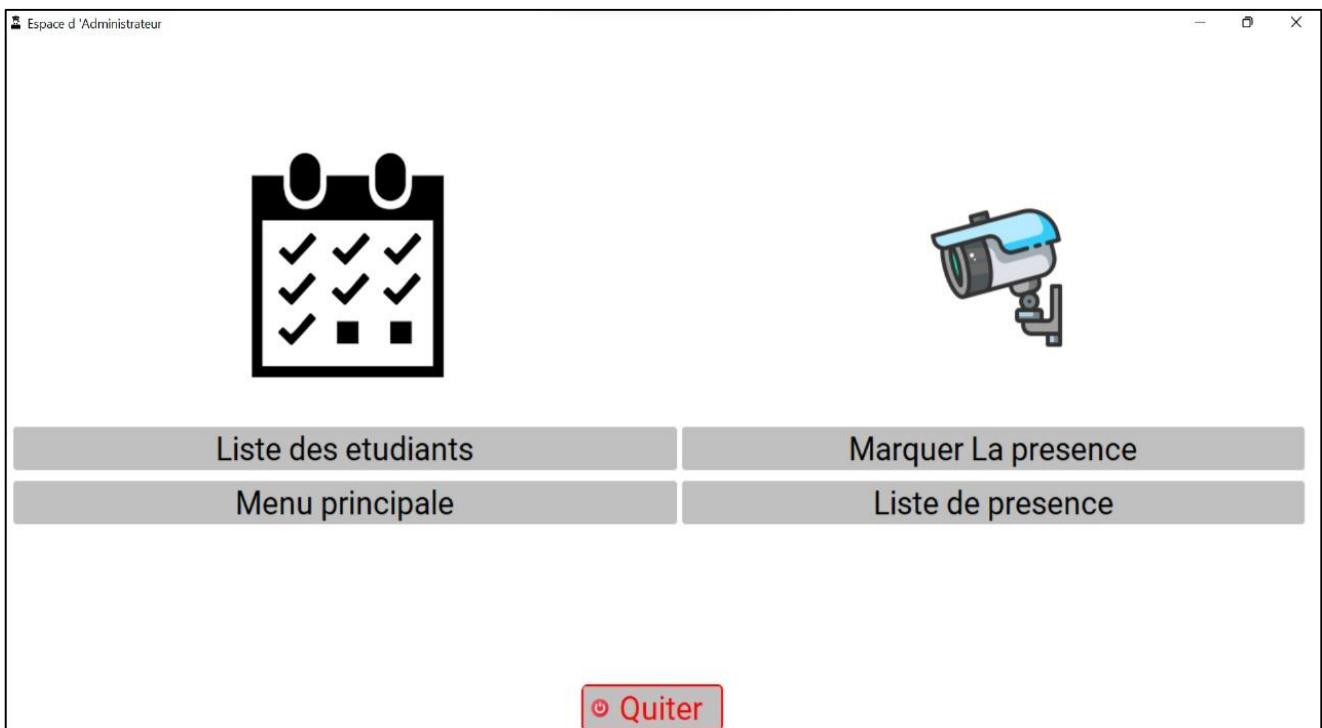
b. Interface principale d'administrateur :

Figure 70 : Interface principale d'administrateur

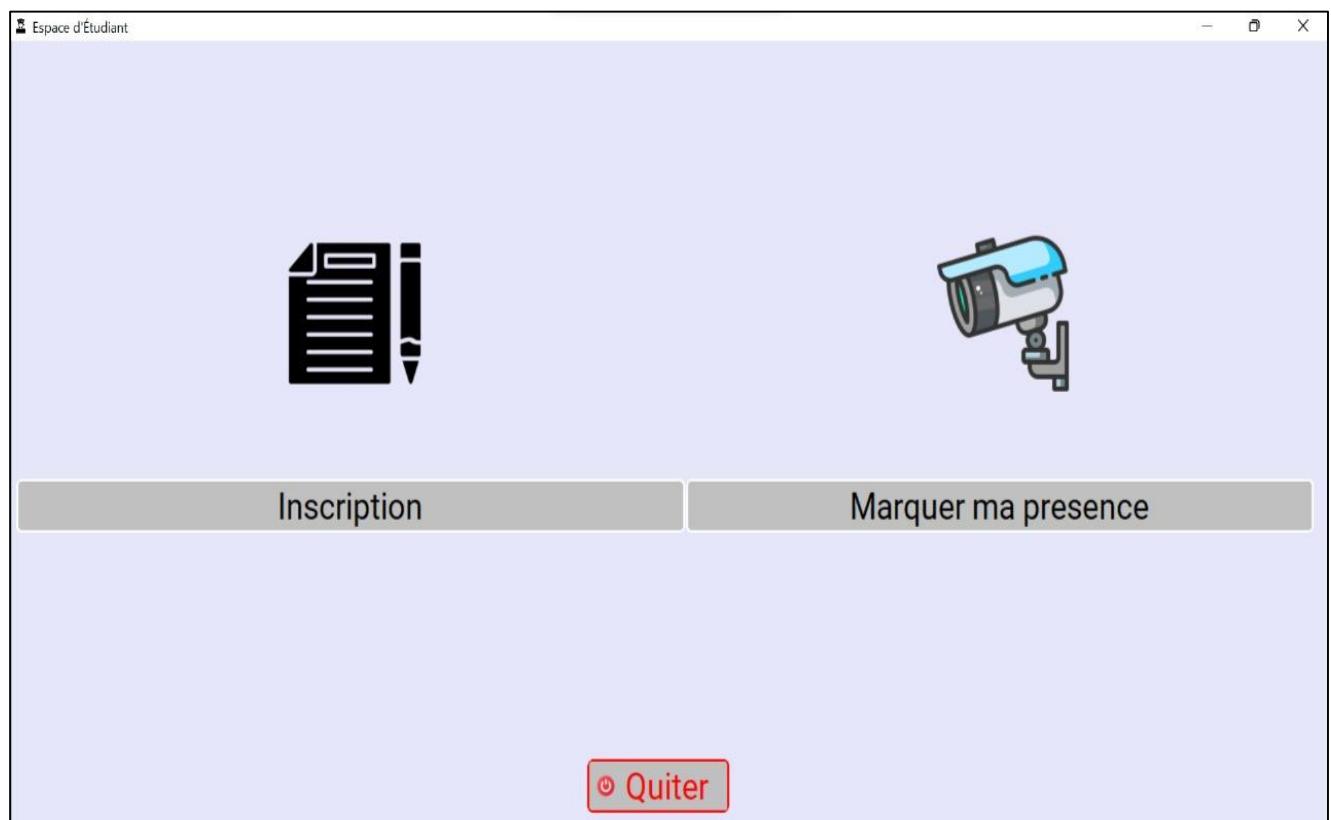
3. Interface d'étudiant :

Figure 71 : Interface d'étudiant

III. Démonstration :

Dans cette partie, nous allons présenter une démonstration de notre application, au cours de laquelle seront découvertes les différentes fonctionnalités qu'elle propose.

1. Interface principal :

Notre interface principale propose plusieurs fonctionnalités utiles. Telles que vous trouvez un résumé générale de notre application disponible dans le bouton **About**, aussi vous pouvez facilement nous contacter par notre Email qui est disponible dans le bouton **Contacter nous**. De plus, cette interface contient deux options qui ont important, espace d'étudiant et d'administrateur. Et en fin, cette interface vous offre la possibilité de quitter l'application en appuyant sur le bouton **Quitter**.

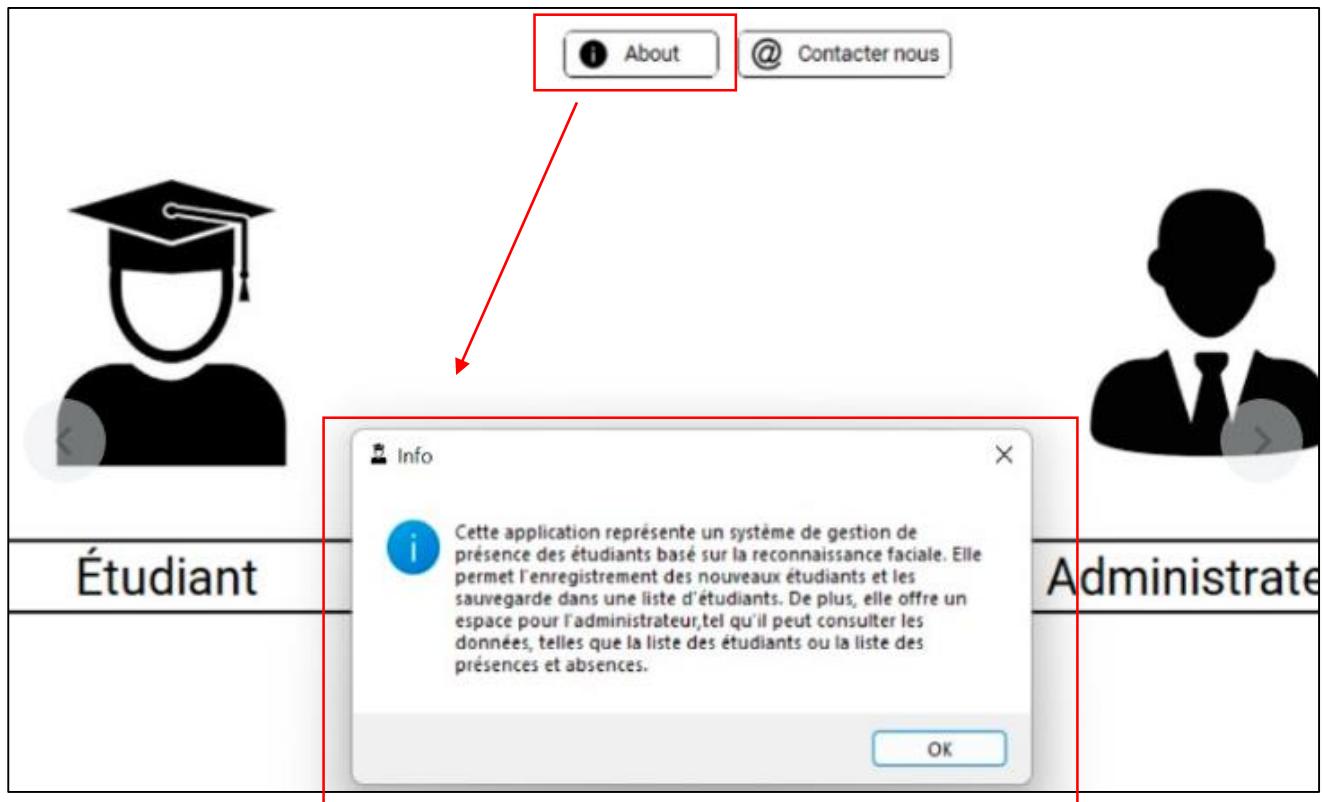


Figure 72 : Fonctionnement du bouton "about"

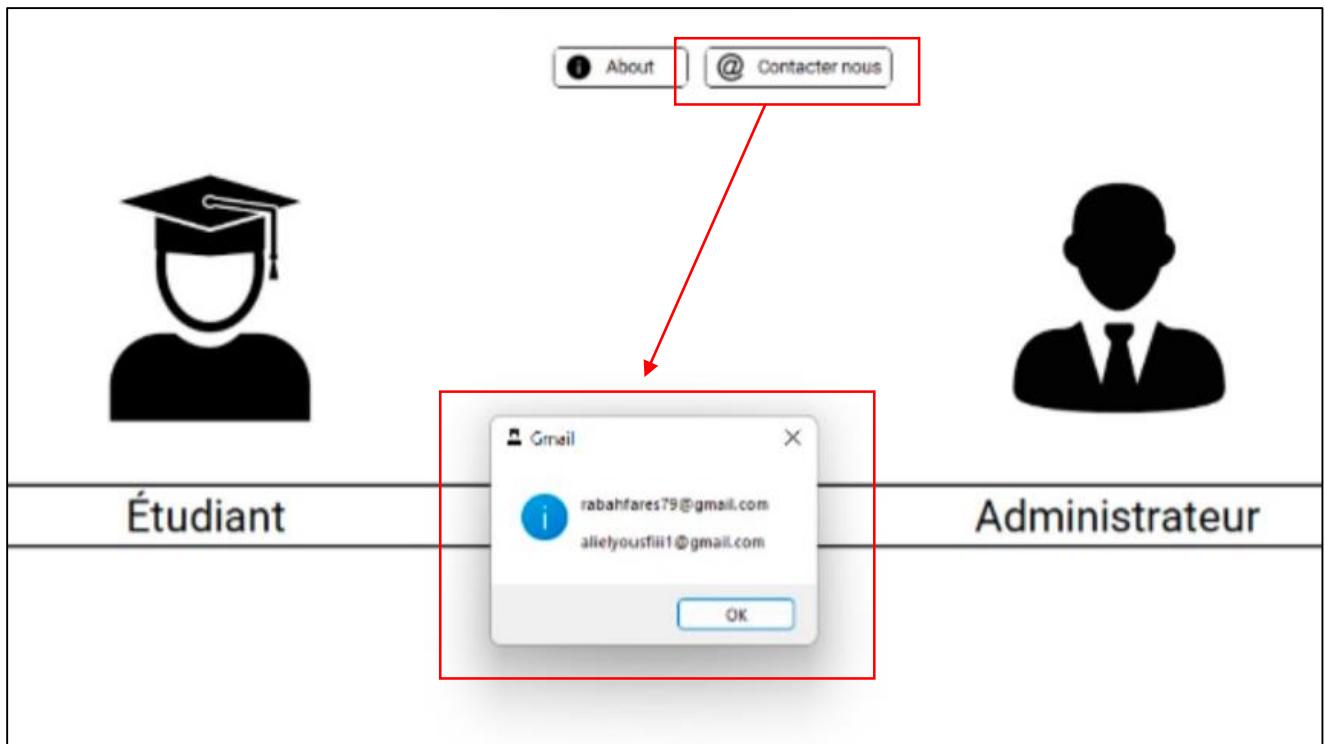


Figure 73 : Fonctionnement du bouton "contacter nous"

2. Espace d'étudiant :

Dans cette partie, nous traiterons des situations spécifiques des étudiants. Nous expliquerons comment un étudiant peut s'inscrire et marquer sa présence facilement. Elle propose un instruction étape par étape pour aider les étudiants pour faire l'inscription et marquer leurs présences.

a. Inscription :

➤ Etape 1 :

Premièrement, il faut cliquer sur le bouton **Étudiant** pour accéder à l'espace étudiant.

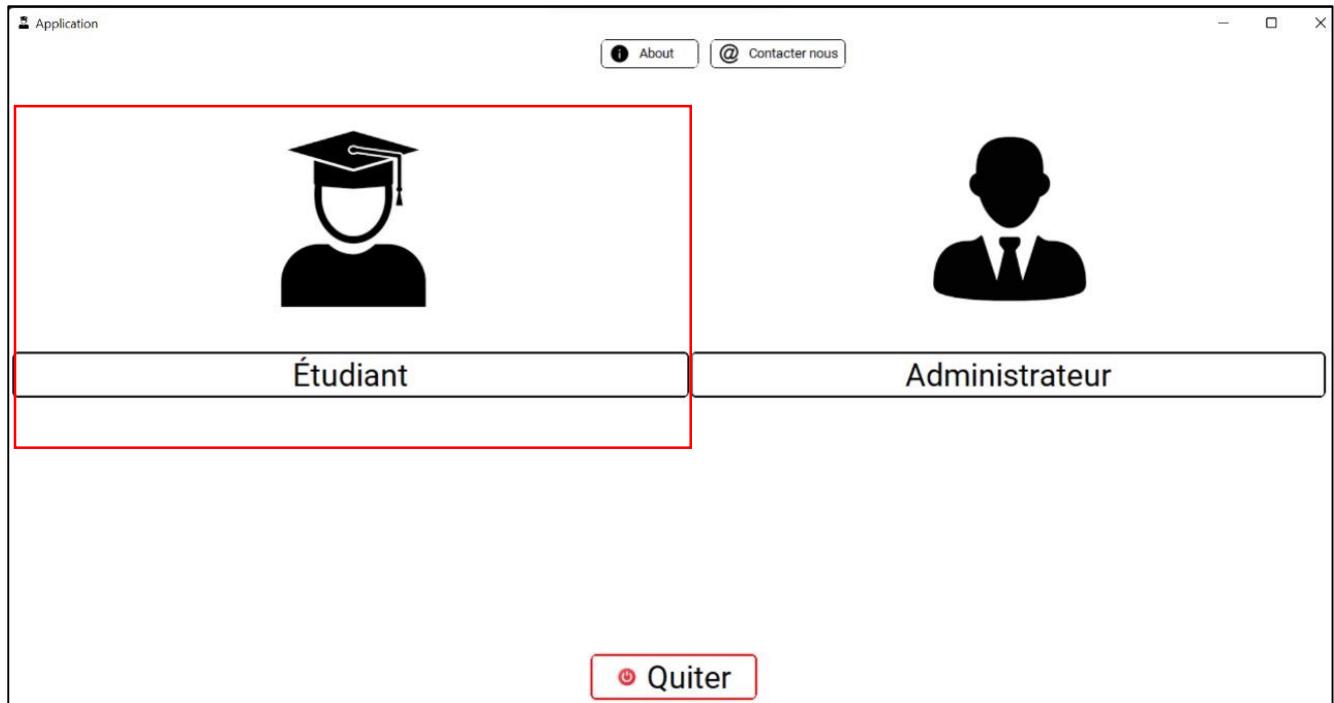


Figure 74 : L'interface principale

➤ **Etape 2 :**

Pour faire votre inscription il faut cliquer sur le bouton **Inscription**. Veuillez noter qu'il vous sera demandé de saisir votre informations personnelles (Nom, Prénom, CNE, ...) lors de cette étape, ainsi que de prendre des photos de votre visage.

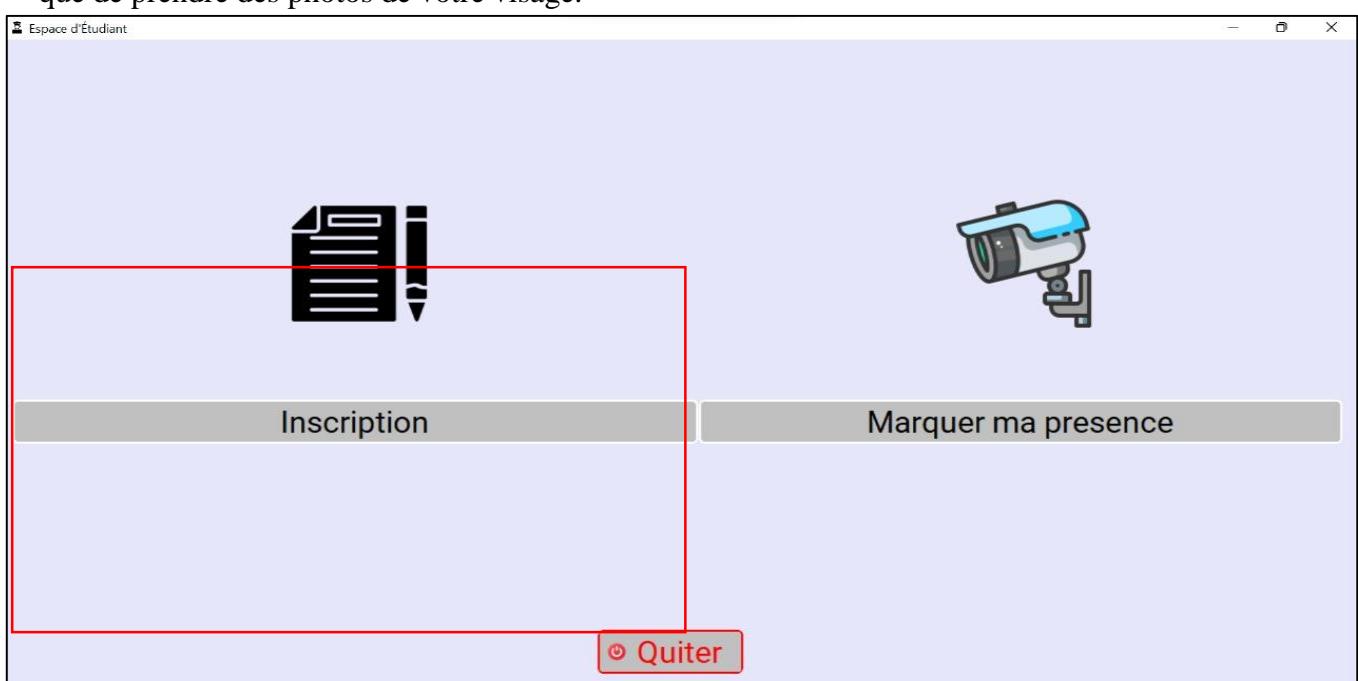


Figure 75 : Espace étudiant

➤ Etape 3 :

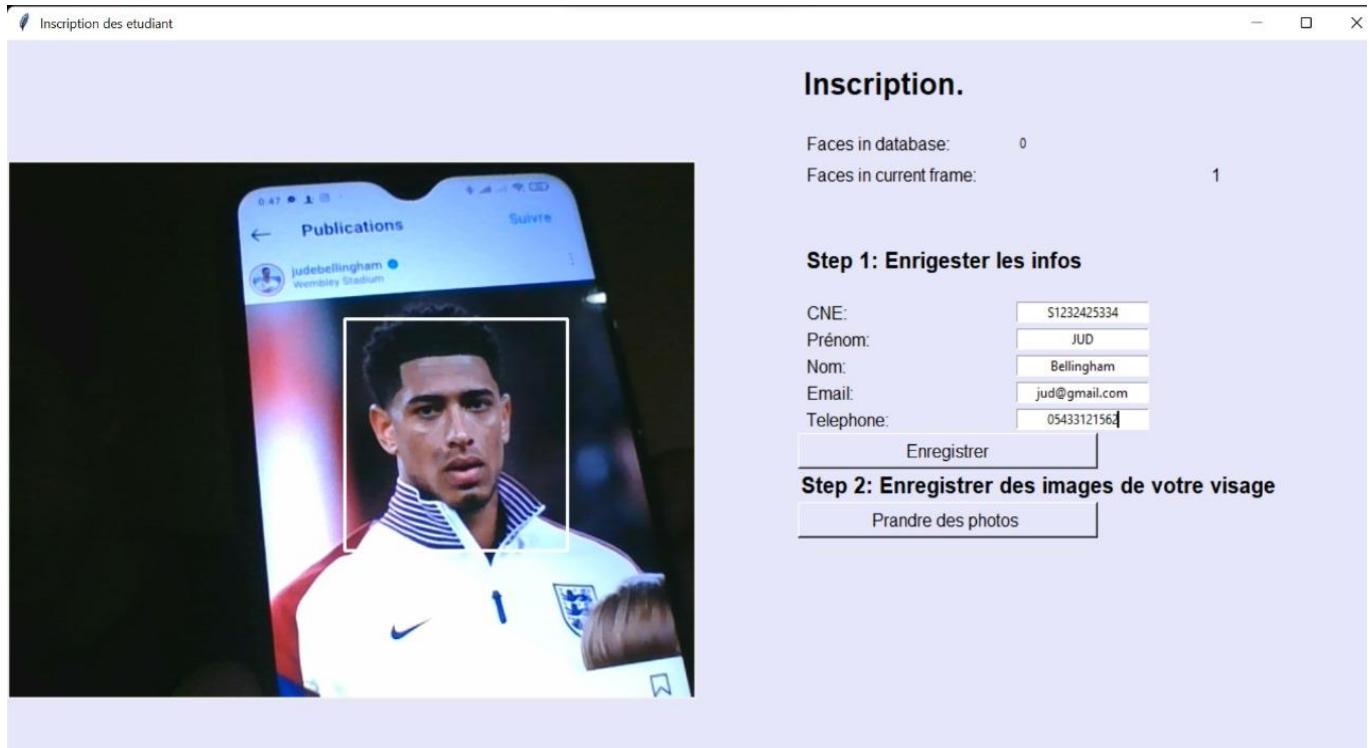


Figure 77 : Espace d'inscription

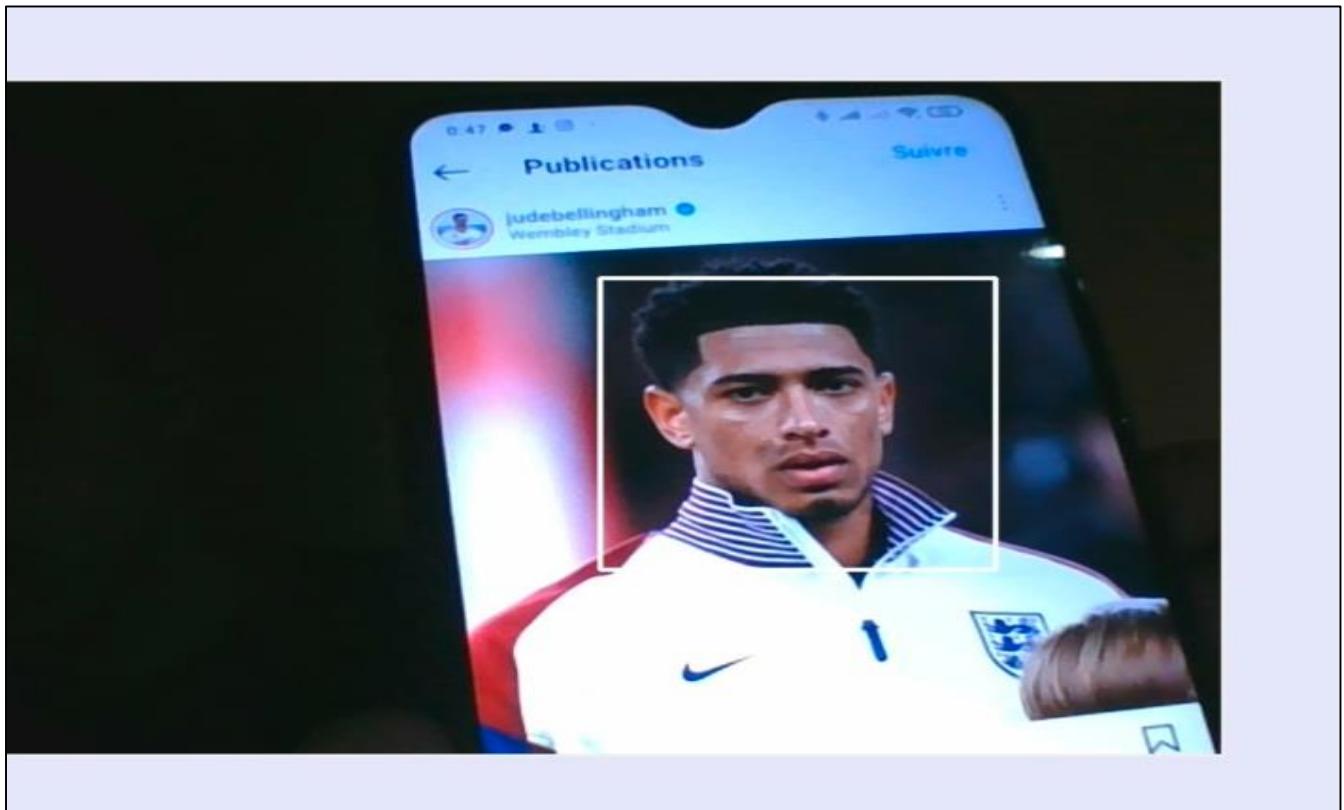


Figure 76 : Zone de détection visage et capture des images

Dans cette étape, vous devez saisir votre Nom, Prénom, CNE, Email, Téléphone, ainsi que de prendre des photos de votre visage. Il faut d'abord remplir vos informations personnelles, puis appuyer sur le bouton **Enregistrer**. Ensuite, présenter votre visage directement devant la caméra, le sorte que le cadre entourant votre visage soit blanc et n'est pas rouge, puis appuyer sur le bouton **Prendre des photos** pour prendre des images de visage et les enregistrés dans un dossier spécifique nommé par son CNE. Après cette opération, vous êtes considéré inscrit, et le système vous ajoutez à la liste des étudiants.

Inscription.

Faces in database: 0
Faces in current frame: 1

Step 1: Enregister les infos

CNE:	S1232425334
Prénom:	JUD
Nom:	Bellingham
Email:	jud@gmail.com
Téléphone:	05433121564

Enregister

Step 2: Enregister des images de votre visage

Prandre des photos

Figure 78 : Zone de saisir des informations personnelles

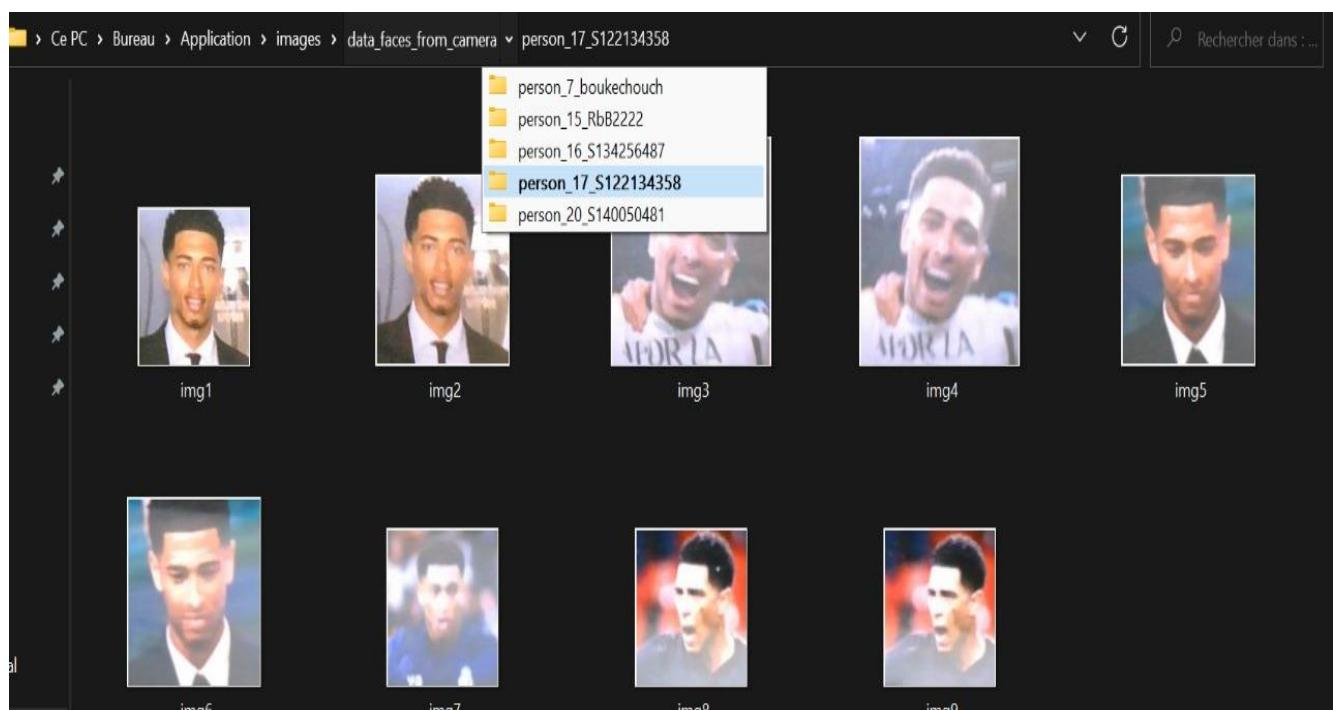


Figure 79 : Dossier des images captures

b. Marquage de présence :➤ **Etape 1 :**

Ensuite pour marquer votre présence, vous devez cliquer sur le bouton **Marquer ma présence**. Après le système ouvre immédiatement la caméra, pour commencer le marquage de présence. Il est obligatoire de répéter cette fonction chaque fois que vous assistez à une séance pour vous inscrire sur la liste de présence.

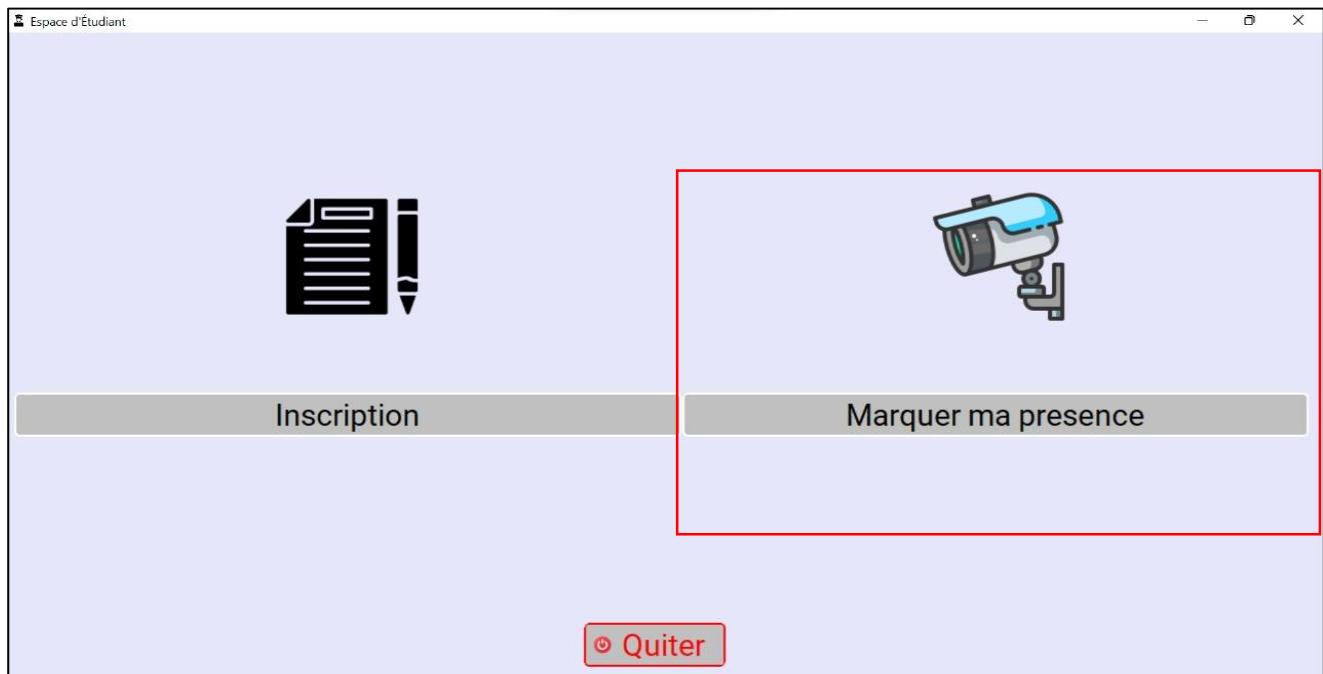


Figure 80 : Espace d'étudiant

➤ **Etape 2 :**

Dans cette étape, vous devrez présenter votre visage devant la caméra. Si vous étiez déjà inscrit et que vous avez assisté à la séance, vous serez marqué présente et vos informations personnelles seront enregistrées dans la liste de présence. Cependant, si l'étudiant est inscrit mais ne participe pas à la séance, ses informations seront enregistrées dans la liste d'absence. Dans le cas où l'étudiant n'est pas déjà inscrit dans la liste des étudiants, il sera considéré inconnu, et le système affichera **Inconnu** sous le cadre entourant son visage.

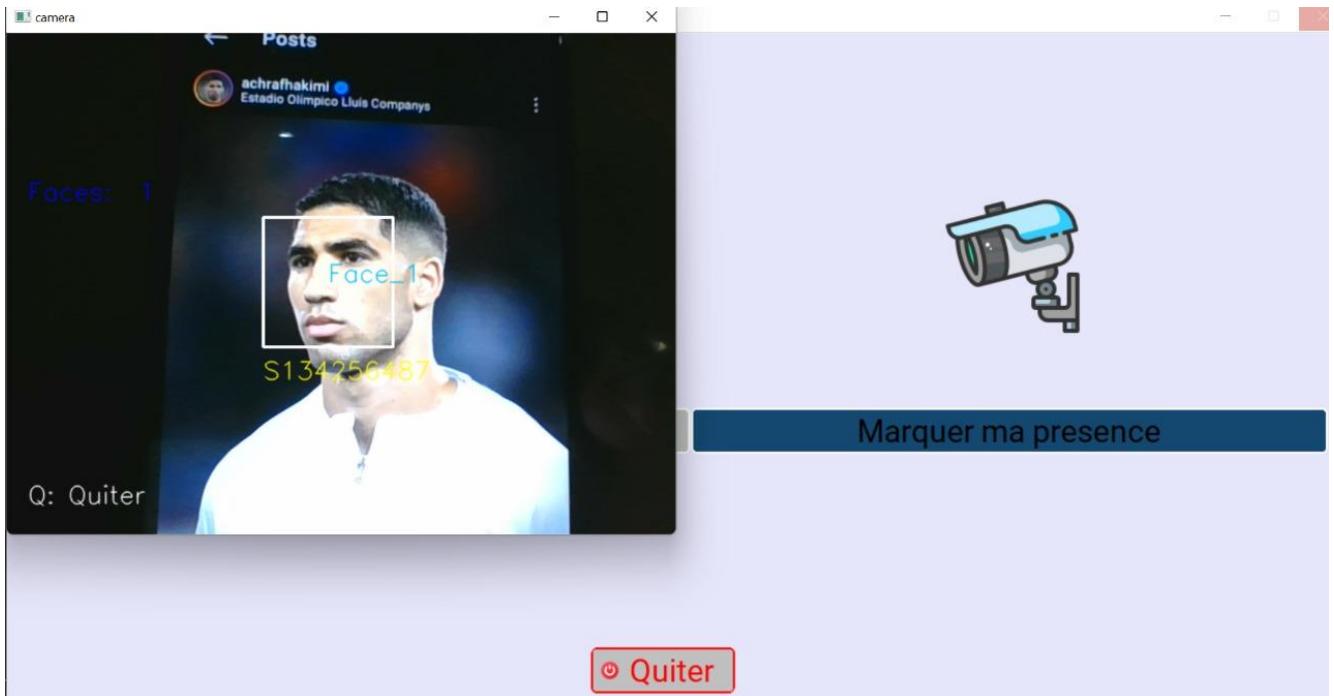


Figure 81 : Détection et reconnaissance de visage

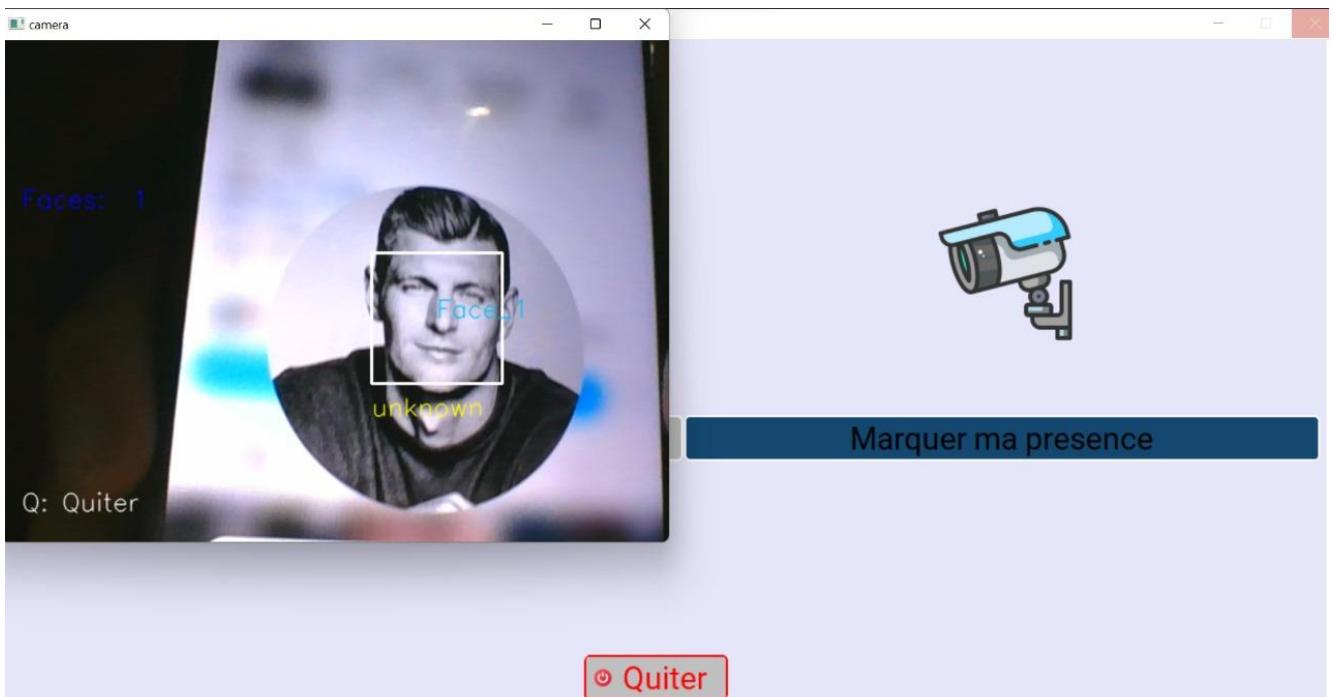


Figure 82 : Le cas où le visage n'est pas reconnu

3. Espace d'administrateur :

Dans cette section, nous réservons de l'espace pour les choses administratives. Nous allons expliquer comment un administrateur peut accéder facilement à la liste des étudiant qui sont inscrit, la liste de présence et d'absence, en plus de cela, l'administrateur peut aussi gérer lui-même le marquage de présence des étudiants. Nous comprenons l'importance d'une gestion efficace des données pour les administrateurs, et nous avons créé cette partie pour vous donner des instruction détaillées sur la façon de gérer ces tâches étape par étape.

a. Connexion :

➤ Etape 1 :

Premièrement, il faut cliquer sur le bouton **Administrateur** pour accéder à l'espace d'administrateur.

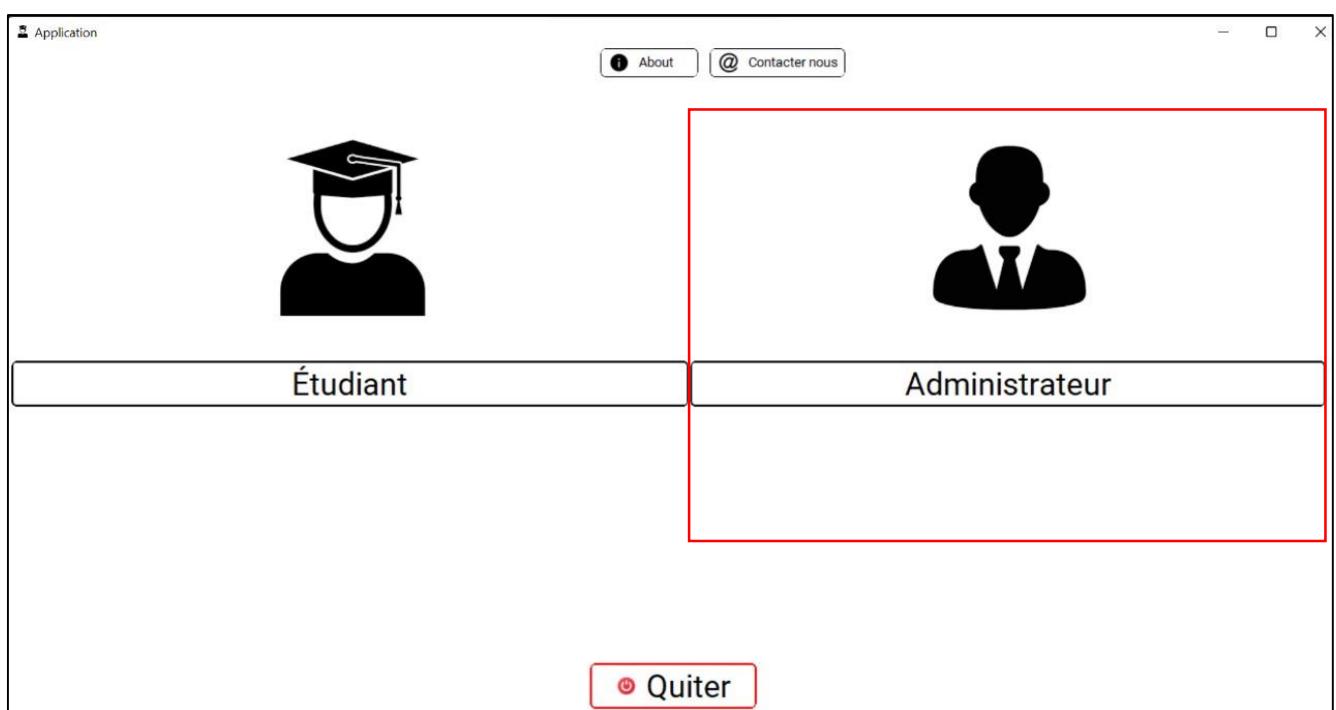


Figure 83 : Interface principal

➤ Etape 2 :

Ensuite, l'administrateur doit suivre cette étape ou il doit saisir son nom d'utilisateur et son mot de passe pour s'assurer qu'il est vraiment un administrateur. La mise en place de cette mesure de sécurité permet de vérifier la personne de l'administrateur et d'empêcher les personnes non autorisées d'accéder aux espace spécifique de l'administration.

Nous avons déjà créé une table dans base de données contenant les noms autorisés à cette section avec leurs mots de passe respectifs. Si le nom et le mot de passe correspondent à ceux de la table dans base de données, le système vous permettra d'accéder. Cependant, si le nom ou le mot de passe est incorrect, le système ne vous permettre pas d'accéder.

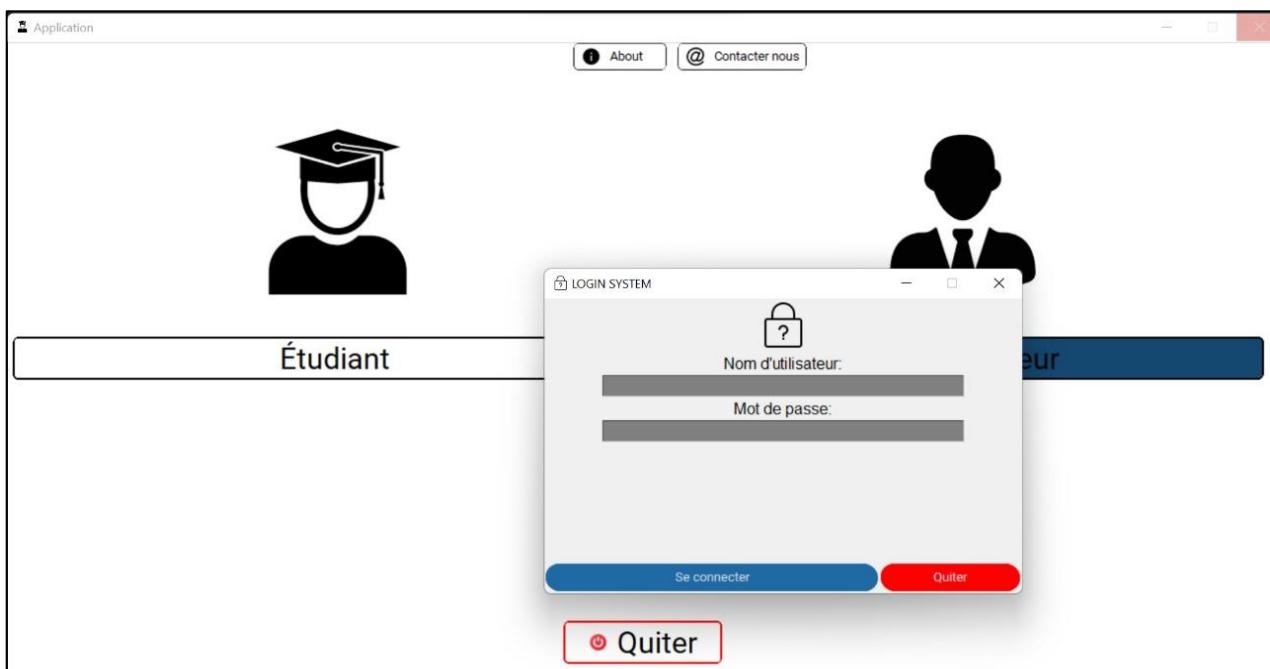


Figure 84 : Interface de connexion

b. La liste des étudiants :

Etape 1 :

Après la connexion, si l'administrateur souhaite d'accéder à la liste des étudiants, il doit simplement cliquer sur le bouton Liste des étudiants. Ce bouton permettra d'accéder à la liste complète des étudiants enregistrés

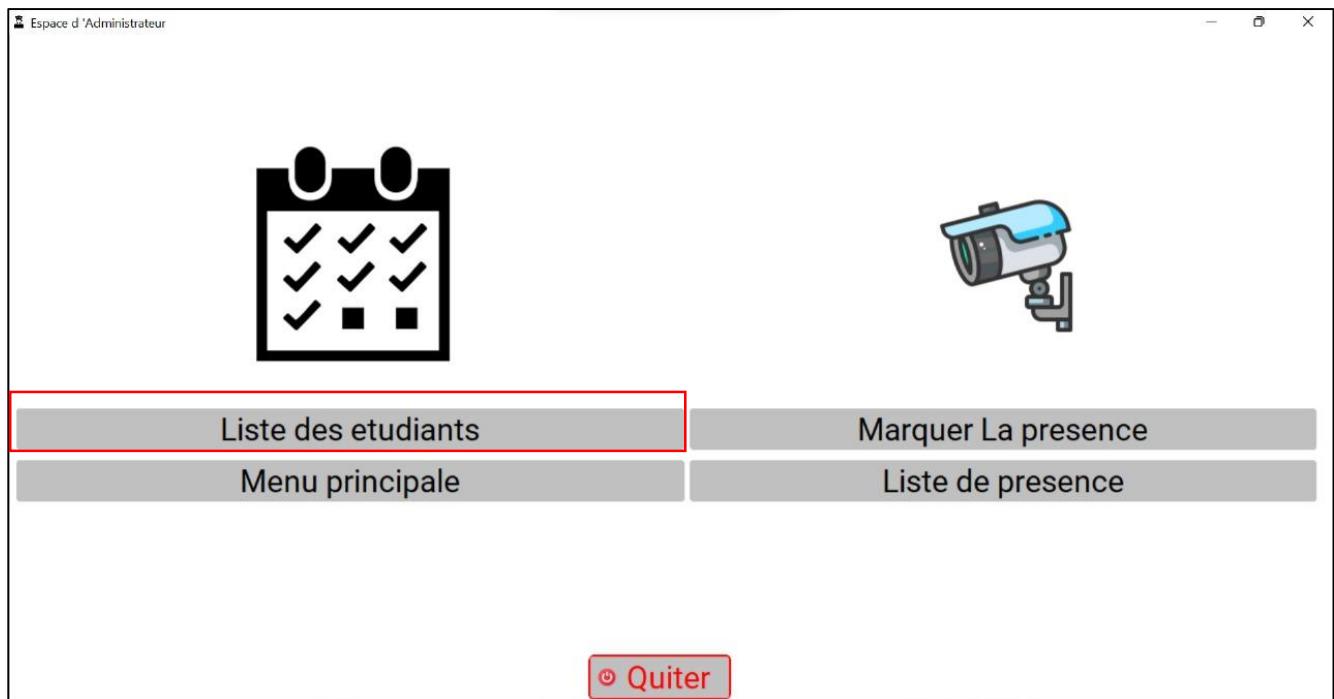


Figure 85 : Espace d'administrateur

➤ **Etape 2 :**

Après, le système affiché une interface administrateur facile à contrôler et à utiliser, lui permettant d'identifier tous les étudiants inscrits. De plus, il lui permet de supprimer des étudiants, de modifier leur information, ou de rechercher des étudiants soit par son CNE, nom ou prénom.

Gestion des étudiants					
Chercher un etudiant :		CNE	Recherche		
Rb82222 S122134358 s134157334 s13420410 S134256487 S140050481	CNE	Nom	Prenom	Email	Telephon
		Mohammed JUD boukechouch ali Achraf FARES	Elkhalifioui Bellingham mohamed elyousfi HAKIMI Rabah	med.khalfioui@gmail.com jud@gmail.com mohamedboukechouch@gmail.com alij hakimi@gmail.com rabah@gmail.com	0623114515 0543213465 16482439051233333 0676453645 0567543287 212606897391
<input type="text" value="CNE:"/> <input type="text" value="Prénom:"/> <input type="text" value="Nom:"/> <input type="text" value="Email:"/> <input type="text" value="Telephone:"/> <input type="button" value="Supprimer par (CNE)"/>					
<input type="button" value="Supprimer"/> <input type="button" value="Modifier"/> <input type="button" value="Clear les champs"/> <input type="button" value="Creer compte"/> <input type="button" value="Quitter"/>					

Figure 86 : Liste des étudiants

Cette section permet à l'administrateur d'ajouter d'autres utilisateurs pour qu'il devienne également un administrateur autorisé à accéder à cet espace, en appuyant sur le bouton **Créer compte**. Le système vous demande de saisir le nom de nouvel administrateur ainsi que son mot de passe, puis à appuyer sur le bouton **enregistrer**. Une fois cela fait, le système enregistre ces informations dans la base de données et reconnaît cet utilisateur comme un nouvel administrateur.

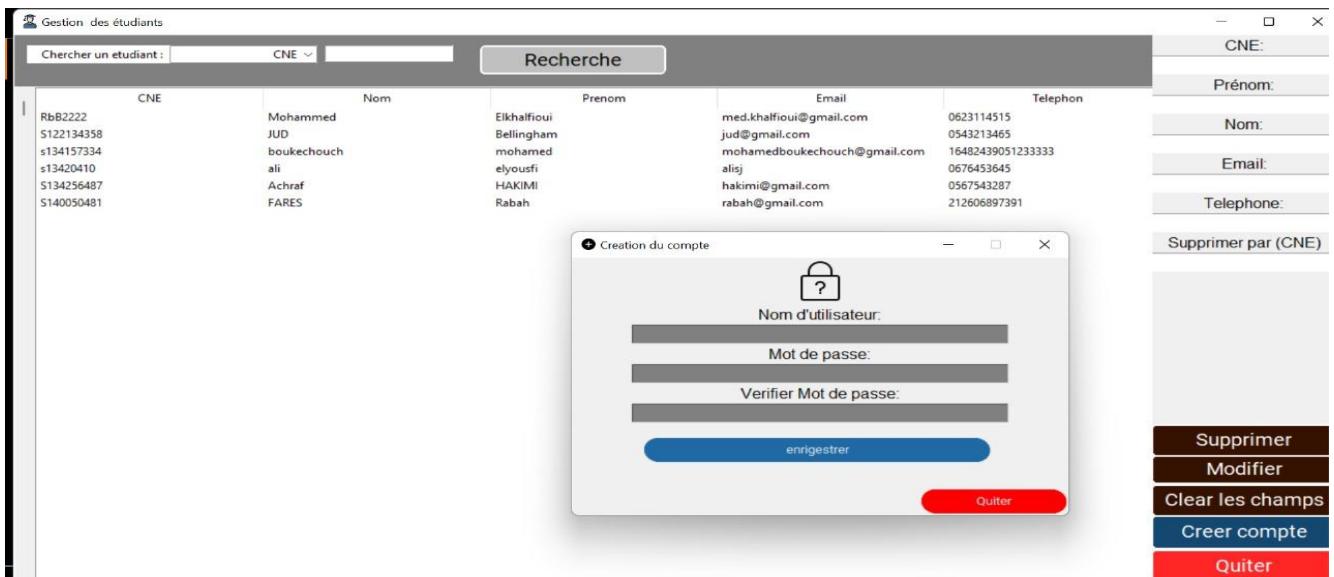


Figure 87 : Créer compte

c. Liste de présence

➤ Etape 1 :

Après la connexion, si administrateur souhaite d'accéder à la liste de présence, il doit simplement cliquer sur le bouton **Liste de présence**.

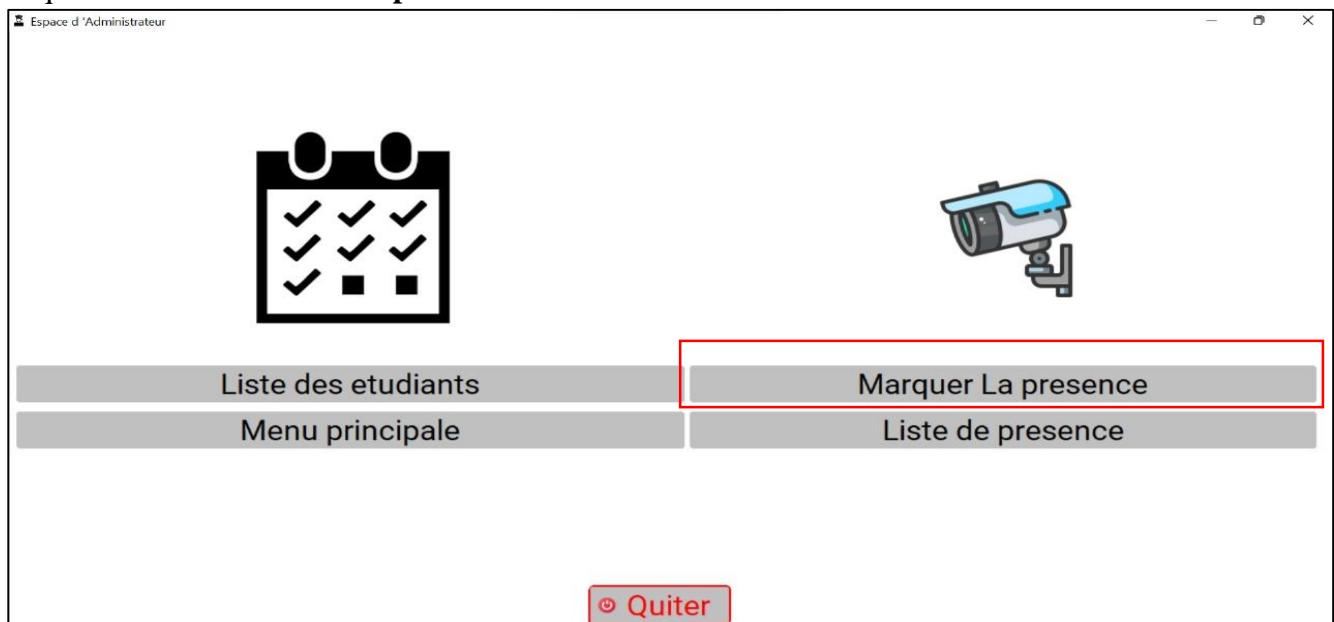


Figure 88 : Espace d'administrateur

➤ Etape 2 :

Après avoir appuyé sur le bouton **Liste de présence**, le système affiché une interface offrant deux options : soit diffuser la liste des étudiants présents en appuyant sur le bouton **Liste de présence**, soit diffuser la liste des absences en appuyant sur le bouton **liste d'absence**. Tout cela est conçu pour simplifier la gestion de l'administrateur en lui permettant de savoir facilement qui est présent et qui est absent.

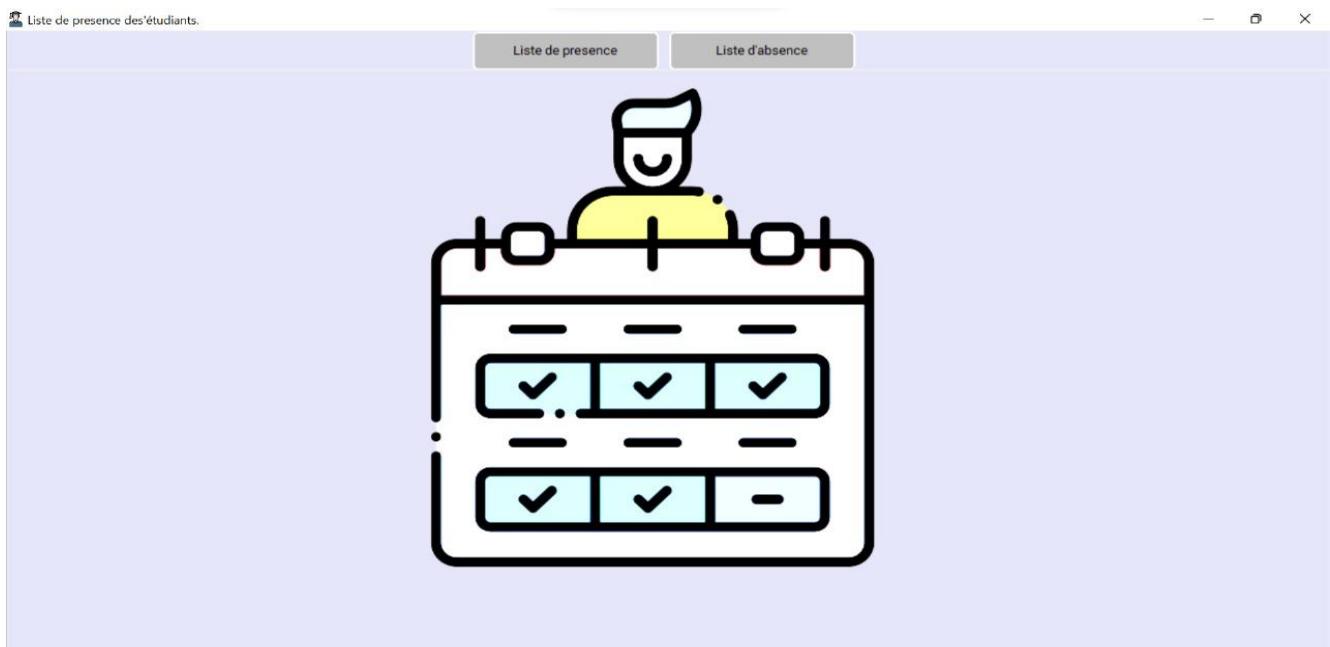


Figure 89 : Interface de liste de présence et d'absence

CNE	Nom	Prenom	Email	Telephon	Date	Time
S134256487	Achraf	HAKIMI	hakimi@gmail.com	0567543287	2024-05-26	23:21:15
S122134358	JUD	Bellingham	jud@gmail.com	0543213465	2024-05-26	23:33:07
S140050481	FARES	Rabah	rabah@gmail.com	212606897391	2024-05-26	23:42:33

Figure 90 : Liste de présence

CNE	Nom	Prenom	Email	Téléphon
RbB2222	Mohammed	Elkhalifioui	med.khalifioui@gmail.com	0623114515
s134157334	boukechouch	mohamed	mohamedboukechouch@gmail.com	16482439051233333
s13420410	ali	elyousfi	alisj	0676453645

Figure 91 : Liste d'absence

d. Marquage de présence :

Dans cette étape de marquage de présence par l'administrateur, les mêmes étapes sont suivies dans le cas du marquage de présence par l'étudiant. Nous avons également donné à l'administrateur la possibilité de marquer la présence. En effet, lorsque la séance commence, l'administrateur peut facilement cliquer sur le bouton "Marquer la présence". Ainsi, lorsque la caméra détecte un étudiant, sa présence sera automatiquement enregistrée

IV. Conclusion :

Dans ce passage, nous avons présenté l'ensemble des interfaces de notre application, en fournissant une explication détaillée de chacune, nous avons pris soin d'expliquer en détail comment chaque partie de notre application fonctionne

Conclusion générales

L'objectif de notre projet de fin d'études est de concevoir et développer une application de bureau en utilisant le langage python. Cette application permet la gestion de la présence des étudiants en intégrant la reconnaissance faciale. Basée sur l'apprentissage automatique en temps réel. Elle utilisera des techniques de machine learning pour identifier et suivre les visages des personnes dans un environnement spécifique, ce qui rend le processus rapide et précis. Grâce à l'utilisation de cette technologie notre application sera capable de fournir des informations précises sur la présence des individus. Ce système pourrait utiliser dans différents domaines telles que la gestion des effectifs, la gestion du personnel et le contrôle d'accès...etc. Pour garantir la performance et la fiabilité, notre application sera fournie des résultats rapides, exacte et robustes, permettant ainsi une utilisation efficace dans divers contextes et pour une gamme variée d'applications potentielles.

Dans la première partie de notre application, nous créons une interface utilisateur par la bibliothèque Tkinter permet aux étudiants de saisir leurs informations (Nom, Prénom, CNE, Email, Téléphone), puis capturer et d'enregistrer automatique des images de leur visage. Après, nous utilisons l'algorithme de détection frontal de Dlib utilise la méthode HOG pour analyser les caractéristiques des gradients orientés dans les images captures, qui permettant de localiser efficacement dans des dossiers appropriés.

Puis dans la deuxième partie en utilise une la bibliothèque Dlib pour extraire les caractéristiques des visages à l'aide de modèle ResNet-50 qui permet de représenter chaque visage en vecteurs numériques, et luis stockés dans un fichier CSV pour utilisation dans l'identification des personnes dans des autres images.

Enfin, notre application vise à réaliser la reconnaissance faciale en temps réel à partir d'un flux vidéo de la caméra, telle qu'en utilise la bibliothèque Dlib pour détecter les visages et en extraire les nouvelles caractéristiques et en les comparés avec les caractéristiques des visages stockées dans le fichier CSV.

Durant notre travail, nous avons obtenu des résultats excellents. Ces résultats ont été obtenus en utilisant une base de données riche et avec une variante de donnée nécessaire pour tester et évaluer une telle approche. Le succès que nous avons eu durant cette expérience nous donne la possibilité de contribuer au développement des applications de reconnaissance de visage et principalement celles basées sur l'algorithme qui on a utilisé dans notre projet.

Bibliographie

- [1] Wikipédia, «Wikipedia: L'encyclopédie libre,» dernière modification le 30 mai 2024.
https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_reconnaissance_faciale.
- [2] «"Reconnaissance faciale." La biométrie,» 12 février 2017. [En ligne]. Available:
<https://labiometrie.wordpress.com/2017/02/12/reconnaissance-faciale/>.
- [3] Wikipedia, «Histogramme de gradient orienté,» faite le 15 février 2022 . [En ligne]. Available:
https://fr.wikipedia.org/wiki/Histogramme_de_gradient_orient%C3%A9.
- [4] N.Marref, «"Apprentissage Incrémental &Machines à Vecteurs Supports",» université Batna 2013.
- [5] T. Gaspard, «Histoire du langage Python,» (2021, 22 mars). [En ligne]. Available:
<https://upway.io/blog/histoire-du-langage-python/>.
- [6] Microsoft, « "Configurez Visual Studio Code pour Go." Microsoft Learn,,» [En ligne]. Available:
<https://learn.microsoft.com/fr-fr/azure/developer/go/configure-visual-studio-code..>
- [7] Libertés, «"La reconnaissance faciale : avantages et inconvénients.",» Liberties.eu, 2021. . [En ligne]. Available: <https://www.liberties.eu/fr/stories/pros-and-cons-of-facial-recognition/43708>.