

الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي والبحث العلمي
Ministry of Higher Education and Scientific Research
جامعة سعد دحلب البلدية
SAAD DAHLAB University of BLIDA
كلية التكنولوجيا
Faculty of Technology
قسم الإلكترونيك
Electronics Department



MASTER THESIS

Specialty : Electronics of Embedded systems

Presented by :

MEDJAHED Imane

Artificial Intelligence, Facial Recognition Technology for Digitalization of University Residences: Security and Management.

Supervised by:

Mrs. BOUGHERIRA Hamida

ACKNOWLEDGEMENTS

I would like to begin by expressing my sincere gratitude to Allah, the most Merciful and Compassionate, for granting me the strength, Tewfik, and continuity to complete my master's thesis. Without His blessings and divine assistance, this achievement would not have been.

I am immensely grateful to my supervisor, Mrs. BOUGHRIRA Hamida, for her support, and continuous encouragement throughout this journey. Her expertise, dedication, and constructive feedback have played a pivotal role in shaping this thesis. I am indebted to her for her guidance and mentorship.

I would like to thank all the teachers of the Faculty of Electronics and their dedication in teaching us during the Master's course.

My thanks also go to Mr. GUESSOUM teacher at the University of Blida, who made me chair the jury of this thesis. I also thank Mr. BOUNEMRI teacher at BLIDA University, who agreed to judge my thesis.

I would also like to extend my heartfelt appreciation to the MLS Company, where I had the privilege to conduct my internship. I am grateful for their cooperation. The experience gained from working with the MLS Company has greatly enriched my academic endeavors and expanded my horizons.

Furthermore, I would like to thank the director of the University Residence "Zoubida Hamadouch SOUMAA 4", which welcomed our project idea and introduced me to the MLS Company.

I would like to express my deepest gratitude to my parents. Their unwavering love, encouragement, and sacrifices have been the cornerstone of my academic journey. They have been my pillars of strength, providing unwavering support and motivation throughout the ups and downs. I am forever indebted to them for their boundless belief in me and their constant prayers for my success.

All words remain weak to express my deep gratitude to all those who have helped me from near or far to achieve this work.

DEDICATION

I dedicate this master's thesis firstly, to myself as a reminder of the perseverance, resilience, and growth that I have experienced throughout the five years, its defiance of all obstacles without giving up. I am proud of the knowledge I have acquired, the skills I have developed, and the person I have become.

To my family, ONE BY ONE, starting from my grandparents until my little sister ASSIL, for their love, encouragement, and belief in my abilities. Their support has been my pillar of strength, and I am grateful for the sacrifices they have made to see me succeed.

To my best friends, for their constant motivation. Nour El-houda, Soultana, Yasmine, Your presence has brought joy and a sense of balance to my life, reminding me that there is more to this journey than academics.

To my study participants, Nour el houda, Mohamed, Chaimaa, and DJalal whose assistance made this research take a breath. Your contributions have given depth to my findings, and I hope that this work can make a positive impact on your lives.

This work is dedicated to all those who have played a part in my academic and personal development. Your belief in me has been the driving force behind my accomplishments. May this thesis serve as a testament to our collective efforts and inspire future generations to embrace the pursuit of knowledge and excellence.

ملخص:

تهدف أطروحة الماستر هذه الى إنشاء وتطوير نظام الحضور في الاقامات الجامعية باستخدام تقنية التعرف على الوجه. لتعزيز أمن الطلاب وتسهيل التخطيط الفعال لتحضير الوجبات من خلال احصاء عدد الطلبة بداخل المجمع لتجنب تبذير الطعام و تحسين نوعية الوجبة المقدمة. قمنا بالتعرف على خوارزميات تحديد و التعرف على الوجه مثل HAAR و HOG و FaceNet و مدى فاعلية كل نوع. و في الاخير اخترنا خوارزمية HOG لتحديد الوجه و FaceNet للتعرف على الوجه لتنفيذ الكود المبرمج. قمنا بإنشاء واجهة مستخدم GUI لتسهيل استخدام البرنامج من قبل المسير. حضينا بتربص عند شركة MLS سمحت لنا بالتعرف على التكنولوجيا المتطورة في هذا المجال و شاركنا معهم في تثبيت نظام التعرف على الوجه الخاص بهم.

الكلمات المفتاحية: نظام الحضور، امن الطلاب، تجنب تبذير الطعام، تحسين نوعية الوجبة، خوارزمية تحديد الوجه و خوارزمية التعرف على الوجه.

Résumé :

Ce mémoire de master vise à établir et développer un système de gestion des présences dans les résidences universitaires en utilisant la technologie de reconnaissance faciale pour améliorer la sécurité des étudiants et faciliter la planification une préparation efficace des repas en comptant le nombre des étudiants à l'intérieur du complexe, pour éviter le gaspillage alimentaire et améliorer la qualité des repas. Nous avons découvert les algorithmes de reconnaissance et de détection faciale tels que HAAR, HOG et FaceNet, ainsi que l'efficacité de chaque type.

Enfin, nous avons choisi l'algorithme HOG pour la détection des visages et FaceNet pour la reconnaissance faciale pour implémenter le code programmé. Nous avons créé une interface utilisateur graphique (GUI) pour faciliter son utilisation par l'administrateur.

Nous avons fait un stage chez l'entreprise MLS, ce qui nous a permis de nous familiariser avec la technologie de pointe dans ce domaine, et nous avons participé avec eux à l'installation de leur système de reconnaissance faciale.

Mots clés : la sécurité, éviter le gaspillage, améliorer la qualité des repas, détection facila, reconnaissance facial, HOG et FaceNet.

Abstract :

This master thesis aims to establish and develop an attendance system in university residences using facial recognition technology. To enhance student security and facilitate

effective meal preparation planning by counting the number of students inside the campus, to avoid food waste and improve meal quality. We got acquainted with face detection and recognition algorithms, such as Haar, HOG, and FaceNet, and the effectiveness of each type. Finally, we chose the HOG algorithms for face detection and FaceNet for facial recognition to implement the code programmed. We have created a Graphical User Interface (GUI) to make it easier for the administrator to use.

We had an internship at MLS Company, which allowed us to get acquainted with the advanced technology in this field, and we participated with them in installing their facial recognition system.

Keywords : security, meal quality, meal preparation planning, face detection, face recognition, HOG and FaceNet

Abbreviation List

AI : Artificial Intelligence

CNNs: Convolutional Neural Networks

DHCP: Dynamic Host Configuration Protocol

DL : Deep Learning

GUI : Graphical User Interface

HD : High-definition

HOG : Histogram of Oriented Gradients

ML : Machine Learning

MP : Megapixel

NC : Normal Close

NAT: Network Address Translation

NO : Normal Open

RFID : Radio Frequency Identification

RGB: Red Green Blue

RFID : Radio Frequency Identification

RNNs : Recursive Neural Networks

SDK : Software Development Kit

VS : Visual Studio

WEP : Watchdog parabolic antenna.

WPA: Wi-fi Protected Access

Contents table

Aknowledgements

Dedication

Abstract

Abbreviation List

Chapter 01: BIBLIOGRAPHY Studying 4

1.1. Introduction 4

1.2. Biometric recognition..... 4

1.3. Facial Recognition 5

1.3.1 Definition 5

1.3.2 Types of Facila Recognition..... 6

1.3.3 Facial recognition process..... 7

1.4. Facial recognition methods..... 9

1.4.1 Classical methods..... 9

1.4.2 Artificial Intelligence methods 11

1.5. Conclusion..... 15

Chapter 02: System Design 17

2.1. Introduction 17

2.2. The Objective of the system 17

2.3. The Security System Architecture 17

2.3.1 Student Registration 18

2.3.2 Tracking Exits 18

2.3.3 Information Display..... 19

2.3.4 Meal Preparation Optimization 19

2.3.5 Data Analysis and Reporting 19

2.3.6 Human-system Interaction 19

2.4. Security System Design 19

2.5. Creating the Graphical User Interface..... 20

2.5.1	Designing the Interface for Collecting Student's Pictures"	23
2.5.2	Data base creation	25
2.5.3	Facial recognition	27
2.5.4	Creating attendance lists	28
2.6.	Conclusion	30
Chapter 03: Implementation and results		32
3.1.	Introduction	32
3.2.	Work Environment	32
3.2.1	Materials	32
3.2.2	PYTHON	32
3.2.3	Visual Studio	33
3.2.4	The webcam used	33
3.2.5	The libraries	33
3.3.	Software development	35
3.3.1	The Steps for Creating the Code	35
3.4.	The camera Dahua	47
3.5.	RESULTS	48
3.6.	Obstacles Discussion and Solutions Proposition	51
3.7.	Conclusion	52
Chapter 04: Report of the internship at the mls company		54
4.1.	Introduction	54
4.2.	The MLS Company Security system architecture	54
4.3.	ZKTecho Company	54
4.4.	The MLS Company systems architecture	55
4.4.1	The Security System Design	55
4.4.2	The System Design of Feeding Service	56
4.5.	Materials	58
4.5.1	Tripod turnstiles	58

4.5.2	SpeedFace-V5L.....	59
4.5.3	The airMax LiteBeam M5.....	60
4.5.4	The printer “SMART”	60
4.6.	Work Environment.....	61
4.7.	Installation of Equipment.....	61
4.7.1	Installation of Security System.....	62
4.7.2	Installation of Feeding Service System	63
4.7.3	The installation of the Software OnouAPP	65
4.8.	Functioning Procedures	66
4.8.1	User Registration	66
4.8.2	Communication Settings	68
4.8.3	Access Control.....	69
4.8.4	Functioning procedures	69
4.8.5	RESULTS	72
4.9.	CONCLUSION.....	72
	General Conclusion	73

Figures table

- Figure 1.1:** Exemples of biometric modalities..... 5
- Figure 1.2 :** The difference between facial detection and recognition..... 6
- Figure 1.3 :** Surface’s temperature distribution detected by thermal camera..... 7
- Figure 1.4 :** Face recognition by thermal camera. 7
- Figure 1.5 :** Facial recognition process..... 9
- Figure 1.6:** The steps of determination of magnitude and direction. 10
- Figure 1.7 :** the image divided into many cells of size (8×8). 10
- Figure 1.8 :** The result of the HOG. 10
- Figure 1.9:** AI, ML and DL 12
- Figure 1.10:** DL architecture..... 13
- Figure 1.11:** CNN structure used for feature extraction. 14
- Figure 1.12:** FaceNet architecture. 14
- Figure 2.1 :** First process diagram. 17
- Figure 2.2:** Second process diagram. 18
- Figure 2.3:** Security System diagram..... 18
- Figure 2.4:** functional flowchart of our system..... 20
- Figure 2.5:** Step-by-Step Flowchart for Creating the Main Window with Functions..... 21
- Figure 2.6:** The flowchart of Building the User Interface Components..... 22
- Figure 2.7:** The flowchart of creating GUI for Data collecting. 23
- Figure 2.8:** Function A flowchart..... 24
- Figure 2.9:** Function B flowchart..... 25
- Figure 2.10:** FaceNet, Deep learning Flowchart..... 26
- Figure 2.11:** Data base creation Flowchart.. 26
- Figure 2.12:** Flowchart of Face Signatures Creation for Real-Time Face Detection and Recognition..... 27
- Figure 2.13:** Flowchart of Attendance List Creation Function and Conditions 29
- Figure 3.1:** Importing libraries..... 35
- Figure 3.2:** Defining column creation function. 36
- Figure 3.3:** Defining manipulation function. 36
- Figure 3.4:**Creating the widgets of GUI..... 37

Figure 3.5: Creating buttons.	37
Figure 3.6: Interactive GUI function.....	38
Figure 3.7: the Graphical User Interface.....	38
Figure 3.8: Importing libraries.	39
Figure 3.9: Taking pictures code.	39
Figure 3.10: Video capture interface.	39
Figure 3.11: the message will appear.	40
Figure 3.12: Image processing.	40
Figure 3.13: video capture window.	41
Figure 3.14 : the data collected.	41
Figure 3.15 : Load the known images and encode them.....	41
Figure 3.16: FaceNet, Deep Learning Architecture.....	42
Figure 3.17 : Feature extraction.....	43
Figure 3.18: Image Embedding Vector.....	43
Figure 3.19: Images pre-processing.....	44
Figure 3.20: Images Addition for detection.....	45
Figure 3.21: Saving names in lists.....	45
Figure 3.22: Attendance List Creation: Implementation conditional functions.	46
Figure 3.23: Excel file saved.	47
Figure 3.24: The Dahua camera.	48
Figure 3.25: the Graphical User Interface.....	48
Figure 3.26: Video capture interface.	49
Figure 3.27: the message after collecting the picture.	49
Figure 3.28: Identification and recognition.....	49
Figure 3.29: The list of entered Students.....	50
Figure 3.30: The list of exited Students.	50
Figure 3.31: The window of the final number.	51
Figure 4.1: Flowchart of working MLS security system.....	56
Figure 4.2: Feeding Service Conception System Architecture.....	57
Figure 4.3: The flowchart of working the Feeding service system.....	58
Figure 4.4: Tripod turnstiles.....	59
Figure 4.5 : SpeedFace-V5L.....	59

Figure 4.6: AirMax LiteBeam	60
Figure 4.7: The printer SMART	61
Figure 4.8: Linking the systems together	62
Figure 4.9: The installation of turnstiles with the V5L	63
Figure 4.10: the installation of feeding service system.	64
Figure 4.11: Diagram represents the role of the Drive	65
Figure 4.12 : The software	65
Figure 4.13: Steps of registration in the data by V5L.....	66
Figure 4.14: the final step of registration in the data by V5L.	67
Figure 4.15 : A successful verification by V5L.	67
Figure 4.16: The communication method.	68
Figure 4.17: The access controle method.....	69
Figure 4.18: the principle interface of Software “OnouAPP”	70
Figure 4.19: The first interface of OnouAPP	70
Figure 4.20: The window of managing information students	71

General Introduction

With the increasing number of students residing on campus, ensuring their safety and streamlining management processes have presented significant challenges. Many murders were recorded in university dormitories due to the negligence of security agents. Therefore, the entry of strangers threatens the security of students [28]. Despite the huge sums allocated to university services, many university dormitories across the country have known the deterioration of conditions, especially feeding services [29]. It is the cause of bad management, instead of preparing good meals for the student, we find that they prepare huge amounts of food with disastrous quality, and then it all goes to waste.

In recent years, artificial intelligence (AI) has revolutionized in various industries, offering innovative solutions to enhance efficiency and security [27]. One area that can significantly benefit from AI technology is security and management of university residences.

Our project addresses two issues; security and food, it aims to explore the potential of artificial intelligence using facial recognition systems to digitally transform the security and management of university residences. Facial recognition technology has advanced significantly, leveraging AI algorithms to accurately identify and authenticate individuals based on their facial features. By integrating this technology into the existing university residence security infrastructure, several advantages can be achieved.

Traditional security measures, such as cards, have limitations in terms of security breaches. Contrary to Facial recognition systems that provide a safer and more convenient alternative, allowing students seamless access to entry. By applying AI-powered facial recognition, will significantly reduce the risks associated with strangers entering, theft, and potential threats and crimes to student safety.

To achieve these goals, this research will include a comprehensive analysis of AI based facial recognition systems, their underlying technologies, and their application to university residence security and management.

By the end of this study, it is expected that the results of the research will contribute to a better understanding of how to use AI facial recognition technology effectively to

General Introduction

enhance the security and management of university residences while improving the efficiency and quality of services provided.

The dissertation will provide insights and recommendations for institutions interested in adopting this technology, paving the way for the digital transformation of security and management systems in the higher education sector.

Overview of the project:

This document is structured into four comprehensive chapters, each contributing to a deeper understanding of the techniques used.

1. The first chapter is a bibliography studying AI facial recognition methods, classic face recognition algorithms, and modern algorithms using deep learning facial recognition models such as YOLO and FaceNet for recognition.
2. The second chapter contains all the diagrams design and flowcharts which will explain the system and its processes, Along with the algorithm adopted.
3. The third chapter covered the Python code we created in the attendance system, which will achieve the idea we aspire to reach. All the parts of the code with a detailed explanation and all the steps that the process goes through. Together with creating a graphical interface to make it easier for use by non-programmers.
4. The final phase of this project is shown in Chapter 4, where a detailed internship report is provided; about my activities and knowledge acquired during my internship with the MLS Company. When I proposed my project idea to my campus director, he found it very interesting because he had received directives from the government to digitalize the campus management. A company “MLS” had just began installing a security system based on face recognition in Blida, and had a government contract to do so in all Algerian university campuses. When I contacted them, they enrolled me for an internship. The report describes the steps of installing the MLS system and the software adopted. Finally, we will compare our system with their system.

CHAPTRE 01
BIBLIOGRAPHY STUDYING

Chapter 01: **BIBLIOGRAPHY Studying**

1.1. Introduction

This thesis aims to evaluate algorithms in face detection and recognition to make informed decisions regarding options for implementing a computer vision system in residence.

To reach this goal, we relied in this chapter on bibliography that studies this concept and the results researchers have achieved in recent years. We describe face detection and recognition through their definitions, evaluation methods, and brief reviews of techniques deployed to solve these problems, including of deep learning algorithms, which appear to be very efficient.

1.2. Biometric recognition

Biometric recognition is a technology that utilizes unique physical or behavioral characteristics of an individual to verify their identity. Unlike traditional authentication methods like passwords, ID cards, and PINs, biometric traits cannot be easily lost, forgotten, or stolen since they are inherent to a person's body or behavior (Figure 1-1). This technology offers several advantages over conventional methods.

The primary advantage is that biometric traits are inherently tied to an individual and cannot be easily replicated or imitated. This ensures a higher level of security and reduces the risk of identity fraud. Additionally, biometric recognition eliminates the need for individuals to remember or carry physical credentials, enhancing convenience and usability.

There are two common tasks associated with identity authentication using biometrics: verification and identification. Verification involves comparing a user's claimed identity with a template stored in the system, typically belonging to an enrolled client. This process results in a binary output, where the system either accepts or rejects the identity claim.

On the other hand, identification involves comparing a client's biometric information with multiple templates stored in the system. This one-to-many comparison helps in determining the client's identity by searching through the entire database. The system provides a potential match or a ranked list of potential matches based on the similarity of biometric traits.

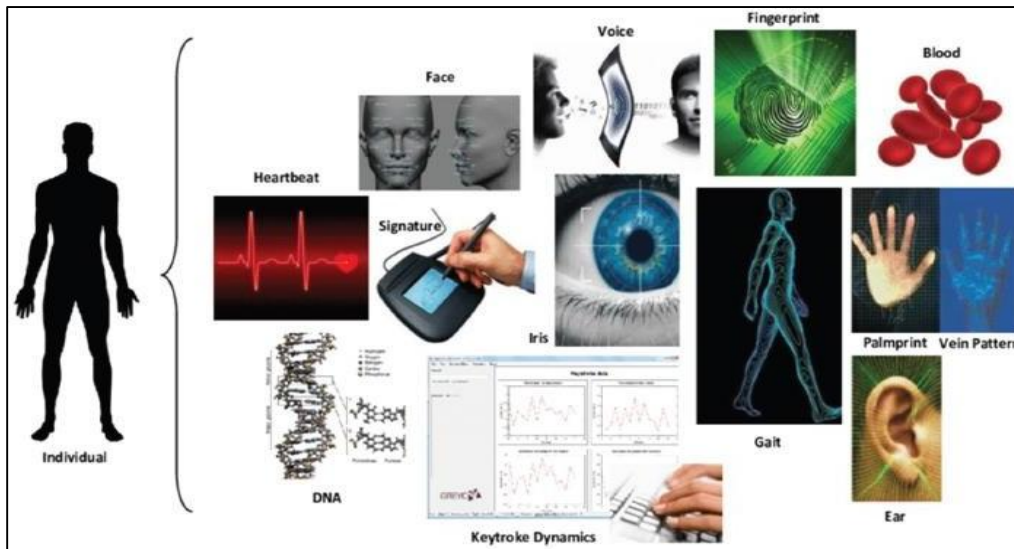


Figure 1.1: Exemples of biometric modalities.

1.3. Facial Recognition

1.3.1 Definition

Facial recognition is the method of identifying or verifying their identity. The concept of face involves a comprehensive examination, evaluation, and correlation of facial characteristics [2]. It involves the identification, study, and differentiation of patterns based on the unique features of an individual's face. The process of detecting and recognizing faces is dependent on an algorithm that includes facial recognition as a crucial component. Experts in the field categorize algorithms related to software or systems into two primary approaches. The approach of geometry concentrates on identifying unique attributes, while the methods of photo-metric statistics are utilized. Values can be extracted from an image using the methods illustrated in (Figure 2.2).

The latter views the human face as a whole unit, whereas the former concentrates on facial landmarks and examines their spatial dimensions and link to other traits [1].

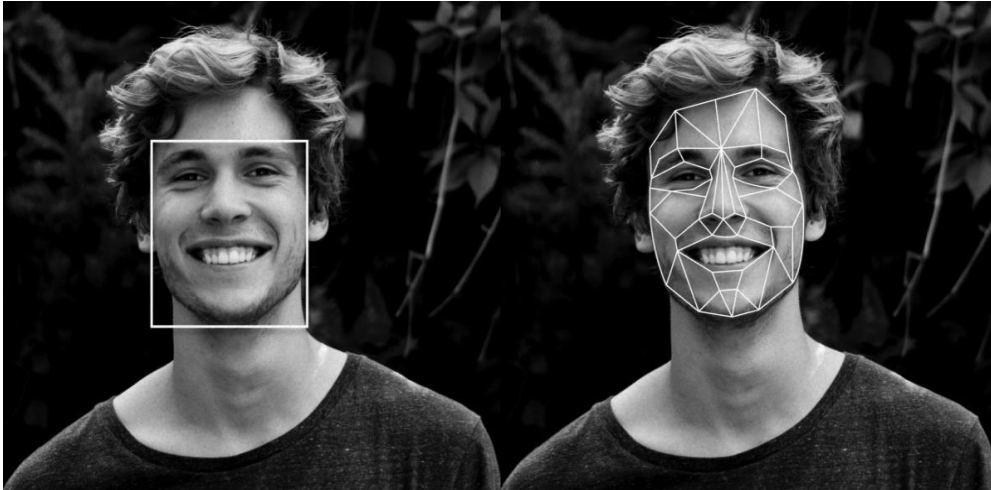


Figure 1.2 : The difference between facial detection and recognition.

1.3.2 Types of Facial Recognition

1.3.2.1. Three-Dimensional Recognition

3D recognition is a highly sophisticated method that uses sensors to scan a face's 3D elements, giving a unique data set. It is highly effective for viewing angles, given it has a 3D model to compare and analyze.

1.3.2.2. Skin Texture Analysis

Skin recognition technology has many applications — face detection algorithms, objectionable image filtering, hand gesture analysis, etc. It usually uses high-resolution images. Particular cases of skin texture analysis use different unique parameters like moles, skin color, skin tones, and many others. Recent research based on texture features and skin color showed exciting results. The researchers used a neural network to develop and test a skin recognition system. The feed-forward neural networks used in the project classified input texture images as "skin" and "non-skin" and showed an impressive performance [1].

1.3.2.3. Thermal Camera

A thermal camera is a device used to monitor the examined surface's temperature distribution. The temperature distribution is displayed as a colored picture with different colors corresponding to temperatures (Figure 3.3).

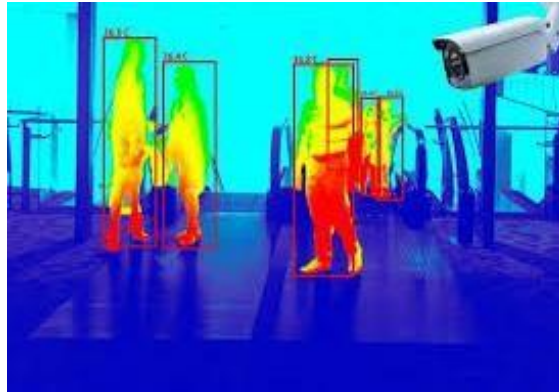


Figure 1.3 : Surface's temperature distribution detected by thermal camera.



Figure 1.4 : Face recognition by thermal camera.

1.3.3 Facial recognition process

Facial recognition is typically executed through Python since this approach is arguably the quickest. The process of face recognition using machine learning methods follows these five main steps:

1- Detecting a face

It is a vital first step in the facial recognition process. The machine first needs to detect the face(s) in an image to establish, if there are any faces at all, which will be the source material for the later steps.

2- Face Alignment

After detecting and recognizing a face, the next step is to determine the orientation of the face. One of the disadvantages of current face recognition technology is that the face must be clearly presented (not blocked by clothes or other objects), and must be facing the

camera to have a higher accuracy rate. You can teach a machine learning model to recognize key points on a face (jaw, eyes, mouth, etc.) and tilt the image slightly to center it and get a frontal orientation.

3- Face measurement & Extraction

Once the face is recognized and the main features of the face can be seen in the image, the next step is to extract the features needed for face recognition. This includes, but is not limited to measurements such as eyes, nose, mouth, etc. This step helps to find similar matches to the features extracted from the database.

4- Recognition

Only then can we perform the actual face recognition process. The final algorithm then compares the measurements extracted from the signature to a database for possible matches.

5. Verification

Finally, facial features can be compared between images until an exact match is found. If no exact match is found in the database, the face remains unconfirmed. As with any ML system, you must remember that the algorithm's functionality depends on the amount and quality of the training data. You can train a model on your data or using an open source dataset [4].

All these steps of facial recognition process resume in (Figure 1.5).

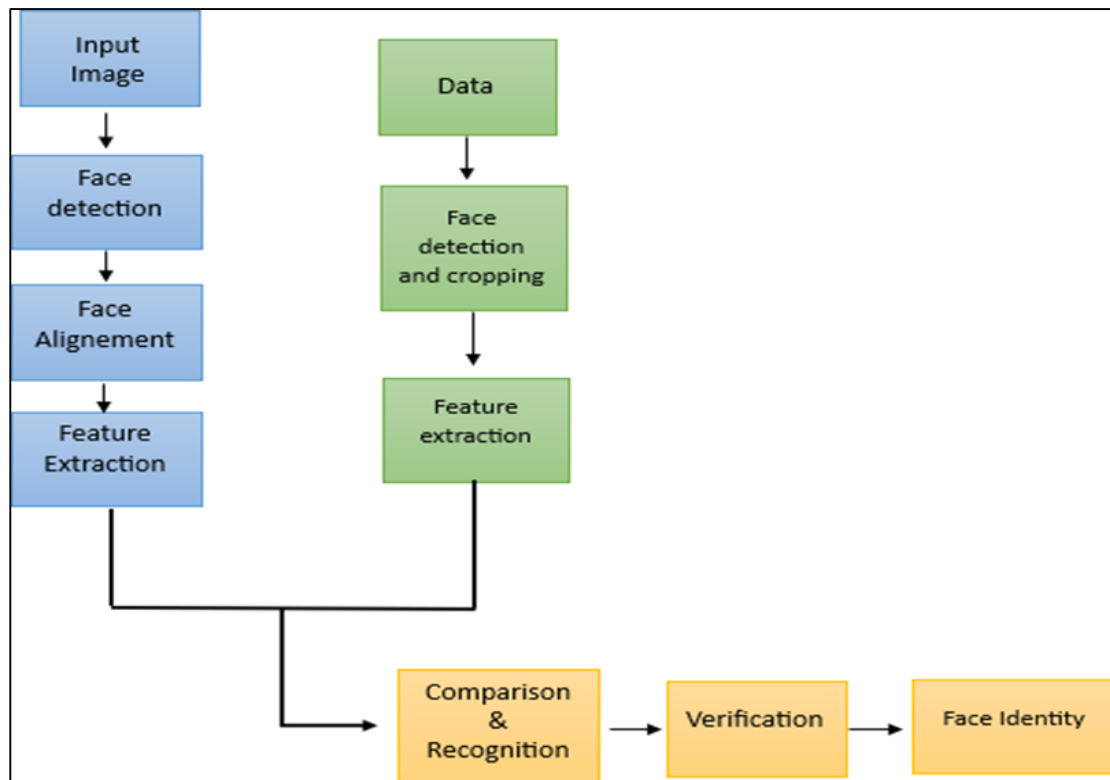


Figure 1.5 : Facial recognition process.

1.4. Facial recognition methods

1.4.1 Classical methods

1.4.1.1. The Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients or HOG is a feature descriptor. It is used in computer vision and image processing for object recognition. This technique computes the direction of gradients that occur in local parts of an image. HOG descriptors focus on the structure or shape of an object. It outperforms any edge descriptor because it uses the magnitude and angle of the gradient to compute features. Generate histograms for image regions using the volume and direction of the slope [10].

The Process of calculating Histogram of Oriented Gradients (HOG):

- Resizing the image into an image of 128x64 pixels (128 pixels height and 64 width).
- Calculating the gradient of the image. The gradients involve gradients by x axis (indicating horizontal change) and gradients by y axis (vertical change)

- Determine the magnitude and direction for each pixel value (Figure 1.6).

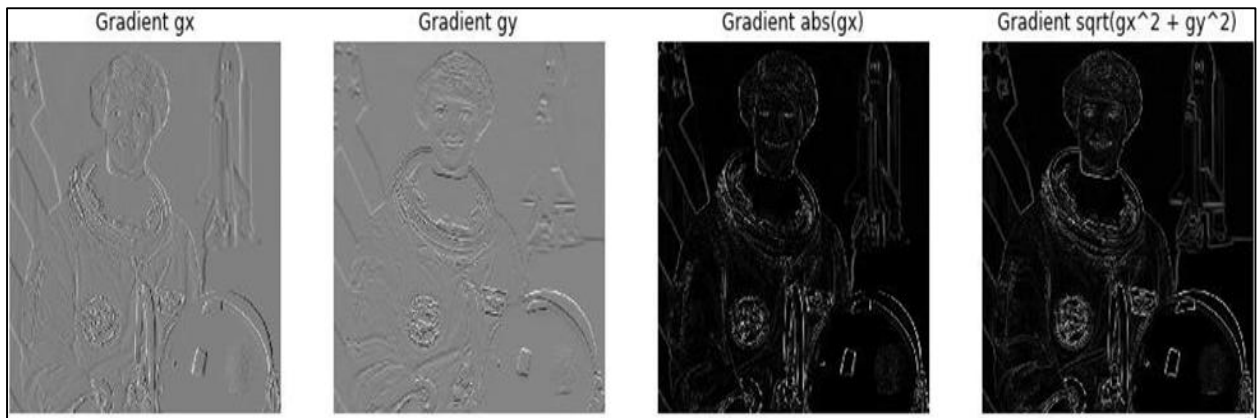


Figure 1.6: The steps of determination of magnitude and direction.

- Generate the histogram from the magnitude and direction of each pixel value (Figure 1.7).

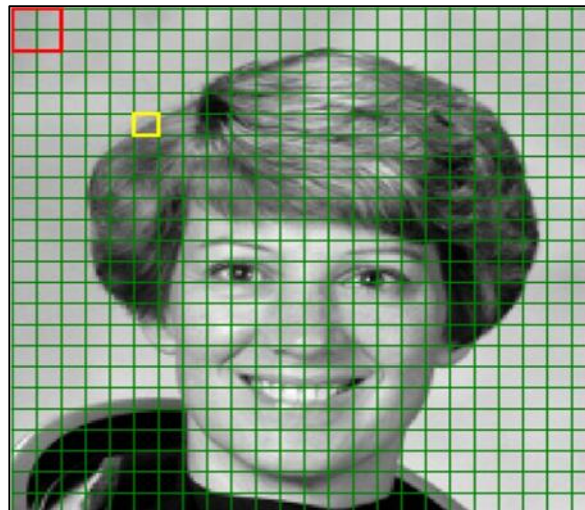


Figure 1.7 : the image divided into many cells of size (8×8).

- The final step is visualizing the HOG picture (Figure 1.8).

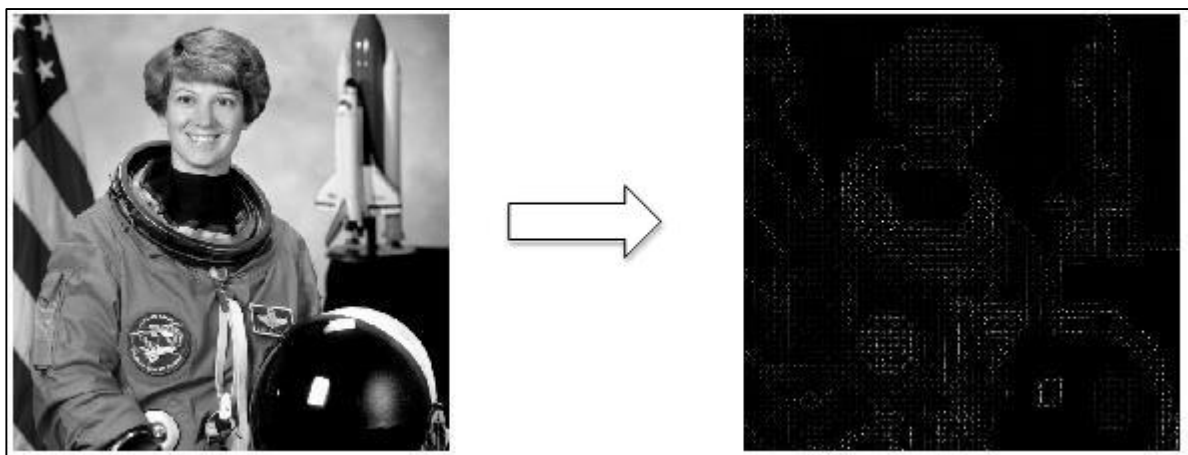


Figure 1.8 : The result of the HOG.

1.4.1.2. HAAR

Haar-Cascadian is a method for object detection using Haar feature-based cascade classifiers, which was proposed by Paul Viola and Michael Jones [16]. It is used for face detection in images or real-time videos [17]. Haar-like features are used to categorize subsections of an image, and a large number of these features are necessary to describe an object with sufficient accuracy [18].

1.4.2 Artificial Intelligence methods

1.4.2.1. Artificial Intelligence Definition

Artificial intelligence (AI) is the capacity of machines to exhibit intelligence, including perceiving, synthesizing, and inferring information. AI relies on imitating human intelligence and performing tasks ranging from the simplest to the most complex [18].

To facilitate problem-solving, AI combines computer science and substantial databases.

Complex functions that traditionally required human input, such as conversing, spotting patterns, and addressing issues, are now carried out by AI. Consumer apps and even embedded firmware use AI [14].

1.4.2.2. Machine Learning Method

Machine learning is a branch of artificial intelligence focused on developing ways for machines to learn from data and improve their performance on tasks, even if they are not explicitly programmed [14]. He has four basic approaches to machine learning: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning [15].

1.4.2.3. Deep Learning Method

Deep learning is a subset of machine learning that uses neural networks with three or more layers to extract high-level features from raw input (Figure 1.9) [14]. It is an artificial intelligence that mimics the way humans acquire knowledge and is used in various applications such as image processing and speech recognition [14]. Deep learning algorithms

are stacked in an increasingly complex and abstract hierarchy, making them more efficient and user-friendly than traditional machine learning algorithms [15].

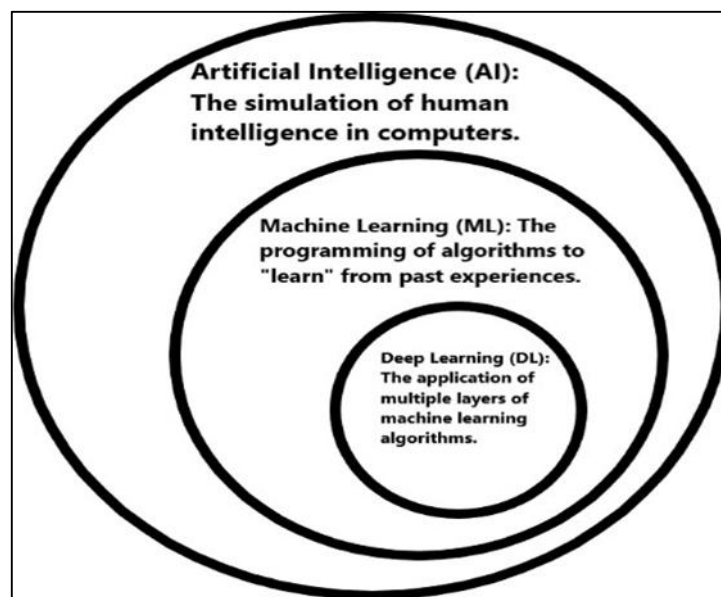


Figure 1.9: AI, ML and DL

1.4.2.4. The deep learning architecture

Deep learning architecture refers to the various types of neural networks that can be used to build solutions for a range of problem areas [14]. The most famous types of deep learning networks include recursive neural networks RNNs, and CNNs (Figure 1.10) [14] [11]. The two major architectures that are commonly used are CNNs for image modeling and RNNs for sequence modeling [19].

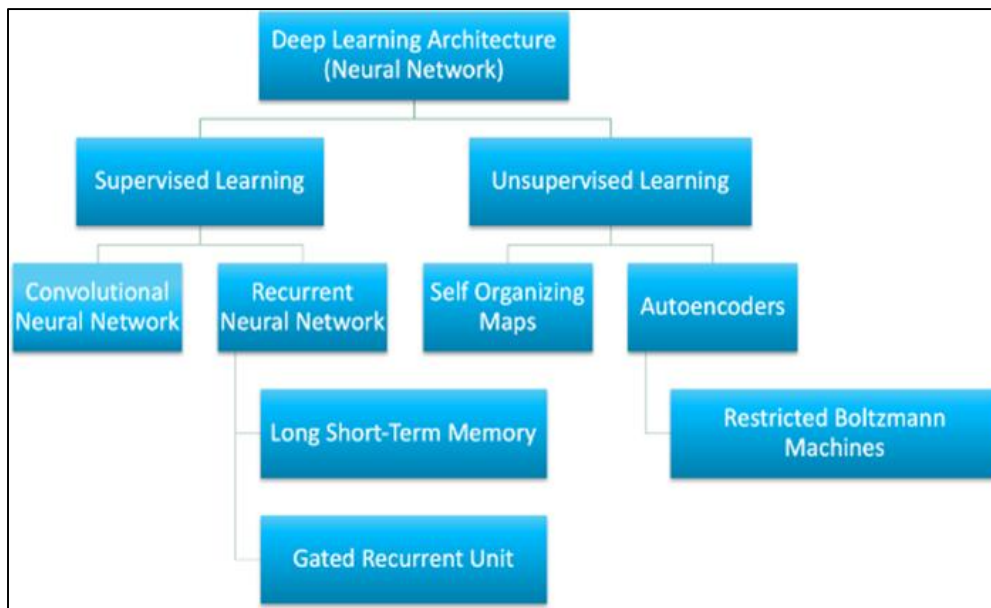


Figure 1.10: DL architecture.

1.4.2.5. Convolutional Neural Network Method

In deep learning, a convolutional neural network (CNN) is a special type of neural network designed to process data from multilayer arrays. CNNs are well suited for facial recognition software applications [12].

A CNN has multiple layers (Figure 1.11) that contain various filters (or commonly called kernels) that are responsible for detecting specific features of the object we are trying to capture. Early layers focus on general features, while the second layer tries to identify specific features. In CNNs, the values of individual filters in each convolutional layer are obtained by training on a specific training set. At the end of training, a unique set of filter values can be used to identify specific features in the dataset. These filter values are used to apply these filter values to a new image so that face recognition can be done [12].

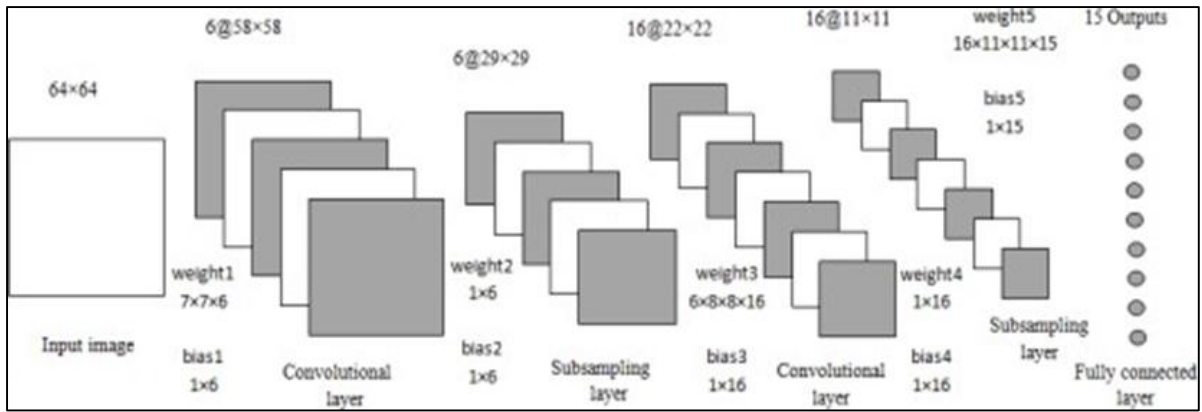


Figure 1.11: CNN structure used for feature extraction.

1.4.2.6. FaceNet Method

FaceNet is a face recognition system proposed by Google researchers in 2015 [20]. It directly learns a mapping of facial images to a compact Euclidean space, where distances correspond to measures of facial similarity [21].

FaceNet uses deep learning architectures such as ZF-Net and Inception Network and uses end-to-end learning in its architecture [20]. The architecture is then trained using a method called triplet loss as a loss function (Figure 3.12) [20].

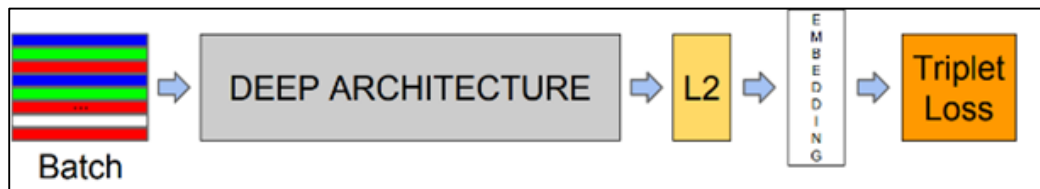


Figure 1.12: FaceNet architecture.

1.5. Conclusion

In this chapter, we had a bibliography studying about face recognition and its types, its process executed, and the methods it used: machine learning, deep learning, and the convolutional neural network.

In the next chapter, we are going to expose our system design and steps for creating an attendance system code programming in the university residence-based face recognition.

CHAPTRE 02
SYSTEM DESIGN

Chapter 02: System Design

2.1. Introduction

In this chapter, we describe the system's architecture, through the description of steps its design. We will first state our security systems objectives then we will proceed to the design of the student registration in and out list generation. We will also create the registered students database, and develop the algorithms of face recognition program, to control the student's identities at the entrance.

2.2. The Objective of the system

The system aims to verify the student identity with a digital photo, it will allow students to enter, prevent strangers from entering, and provide the number of students that are inside the campus. It will increase security, and the list will help the feeding service to prepare meals within the limits of the appropriate final number in order to avoid the problem of waste, and consequently contribute to improving the quality of the meals provided and reducing the meals costs.

2.3. The Security System Architecture

The system performs third tasks:

- The first task is allowing students to enter through facial recognition technology verification, and adding their names to a datetime list (Figure 2.1).

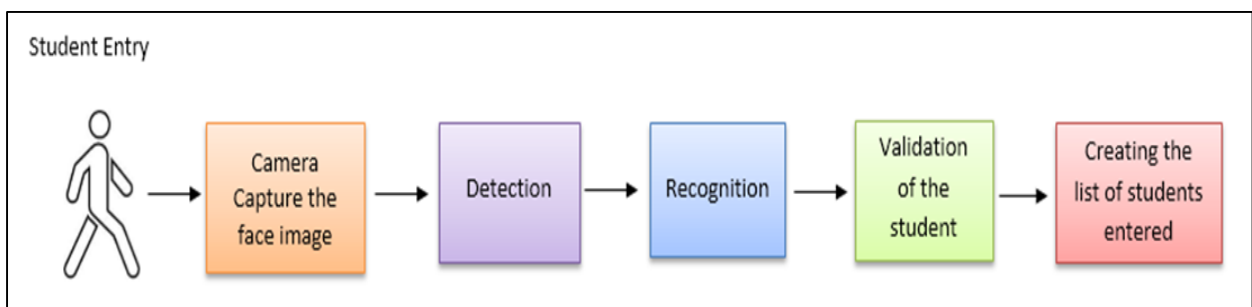


Figure 2.1 : First process diagram.

- The second task is creating the datetime list of the exited students also through facial recognition (Figure 2.2).

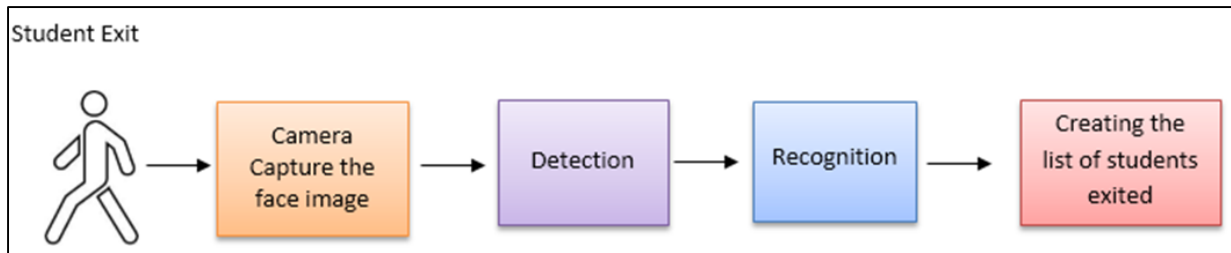


Figure 2.2: Second process diagram.

- The third task is the extraction of the final number of students inside the residency, by the subtraction of the number of students exited from the number of students entered to get the specific number (Figure 2.3).

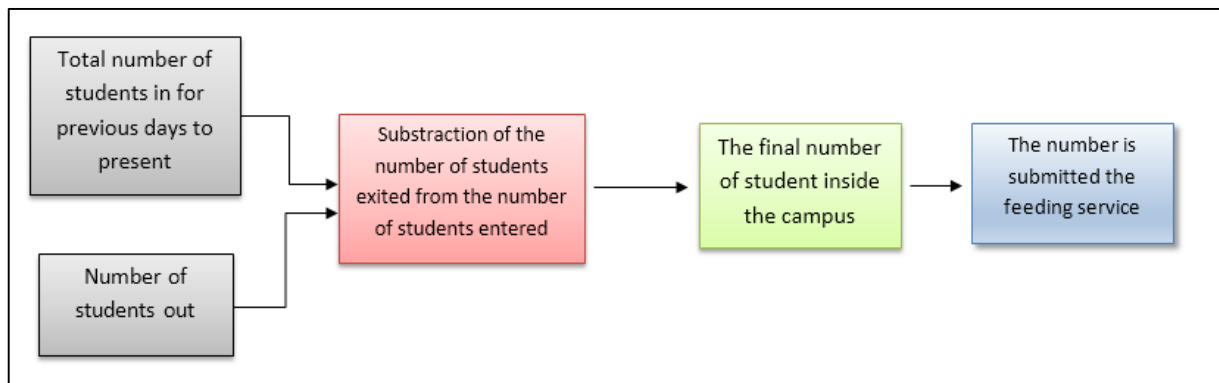


Figure 2.3: Security System diagram.

Our aims are Students entry and exit tracking, campus security and meals optimization

To implement this system effectively, we would need to consider several components and processes. As we explain next

2.3.1 Student Registration

Each student is required to register his or her entry to the campus. This is done through facial recognition techniques using a camera.

2.3.2 Tracking Exits

When a student decides to leave the campus, they must notify the system. This is done also through facial recognition. The system records the time of the exit.

2.3.3 Information Display

The system will have a central display or dashboard where authorized personnel, such as security officers or administrative staff, can access the information. It would show a real-time log of student entries and exits. This video surveillance display helps keep track of students' movements throughout the day.

2.3.4 Meal Preparation Optimization

By keeping track of student exits, the system provides an estimation of the number of students who have left the campus and are likely to skip dinner. This information can be used by the dining services to adjust the quantity of food prepared for dinner approximately. It helps reduce waste and ensures that an appropriate amount of meals are available for the students who remain on campus.

2.3.5 Data Analysis and Reporting

The system can also generate reports based on the recorded exit data. These reports could include statistics on the frequency and timing of student exits, which can be valuable for analysing patterns and making informed decisions related to campus operations, security planning, and resource allocation to improve the meals, avoid waste and corruption.

Overall, the proposed system would enhance security by tracking student exits and providing valuable information for optimizing campus operations and services, particularly in the context of meal preparation for dinner.

2.3.6 Human-system Interaction

The system is used by either the personal or the students. We need to design a simple graphic user interface (GUI), which will be an important component in the functioning of the system at different levels.

2.4. Security System Design

In the following sections, we will delve into the specific components, processes, of our system (Figure 2.4). An overview of how it can be implemented to maximize its effectiveness and impact.

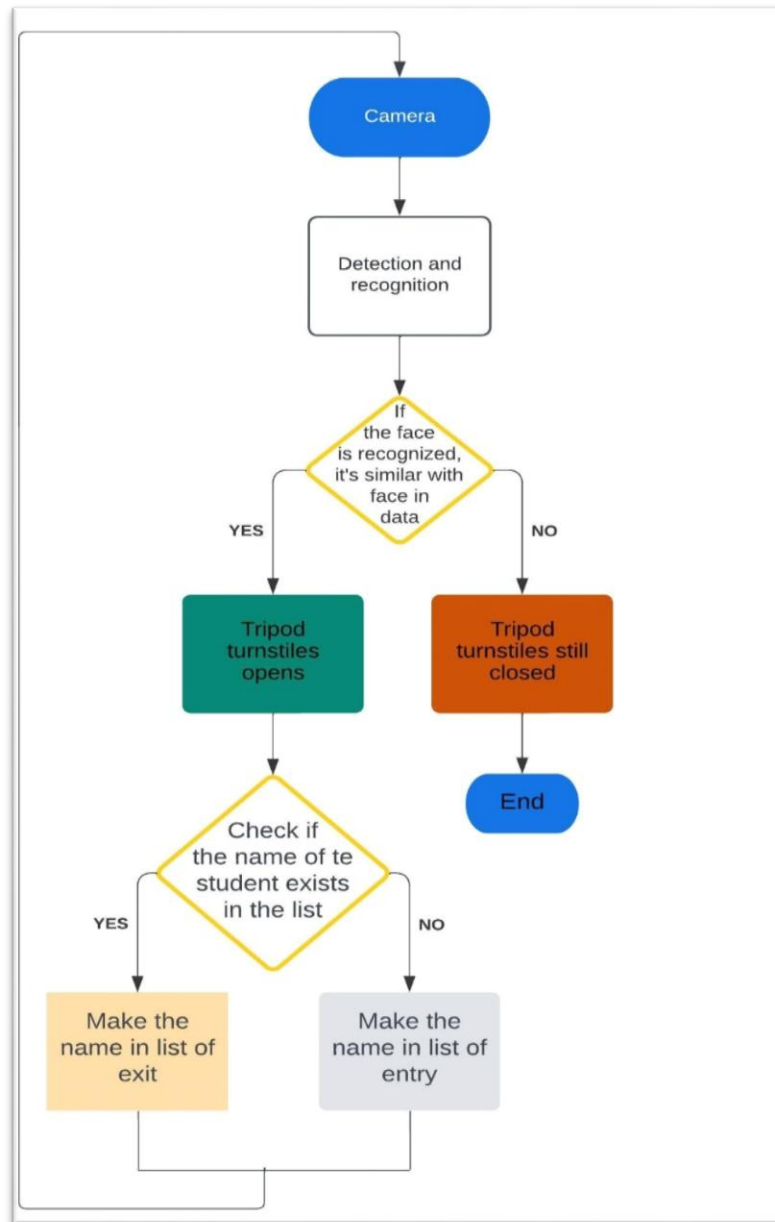


Figure 2.4: functional flowchart of our system.

2.5. Creating the Graphical User Interface

Creating a graphical user interface (GUI) for our system offers an easy way for users to interact with the system, reducing the complexity curve. GUIs enhance the administrator experience and improve the efficiency of system interactions. It improves present information in a visually meaningful way, making it easier to analyze data and make informed reports.

We create the main window (Figure 2.5), then the user interface components (Figure 2.6).

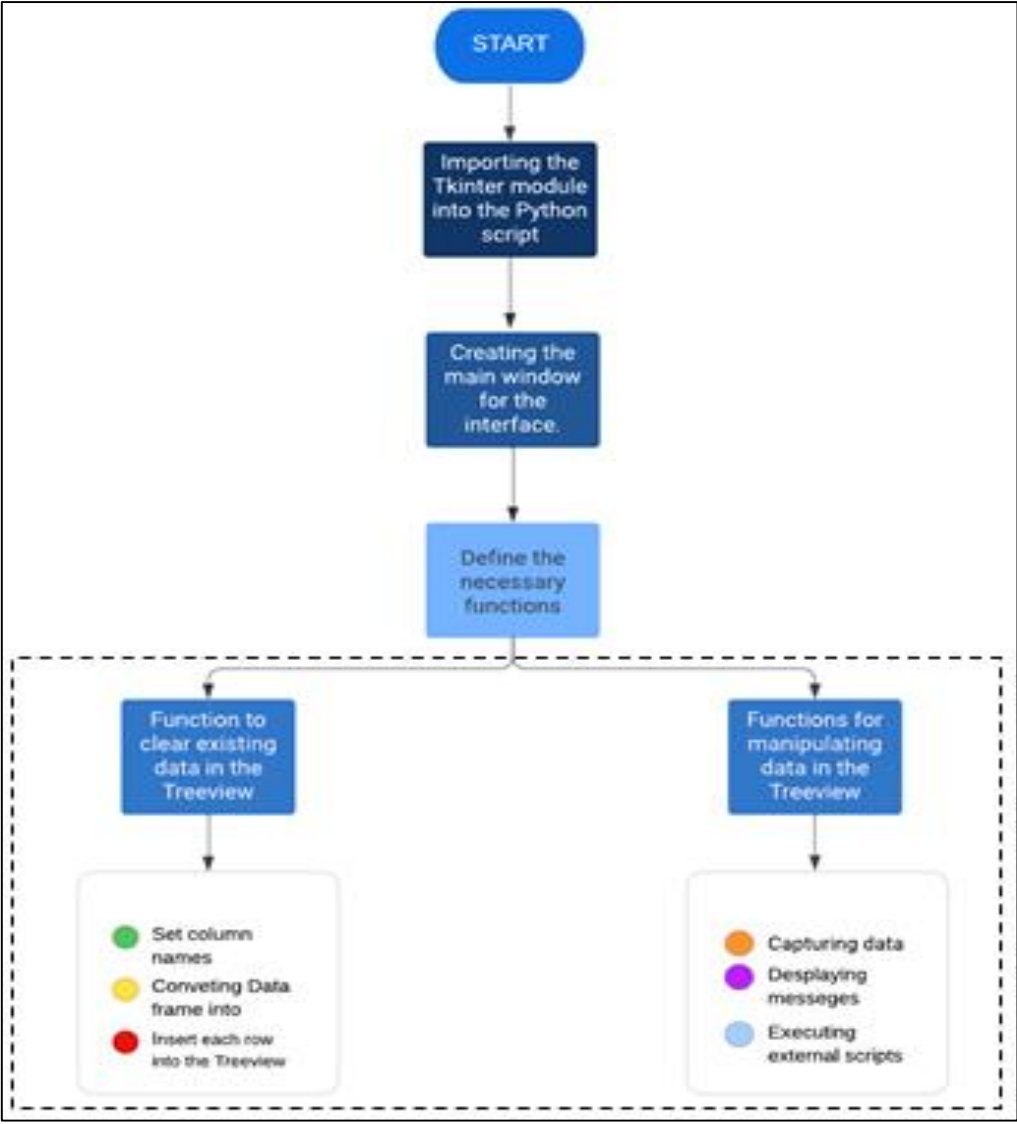


Figure 2.5: Step-by-Step Flowchart for Creating the Main Window with Functions

The next steps are completion of the precedent flowchart (Figure 2.6).

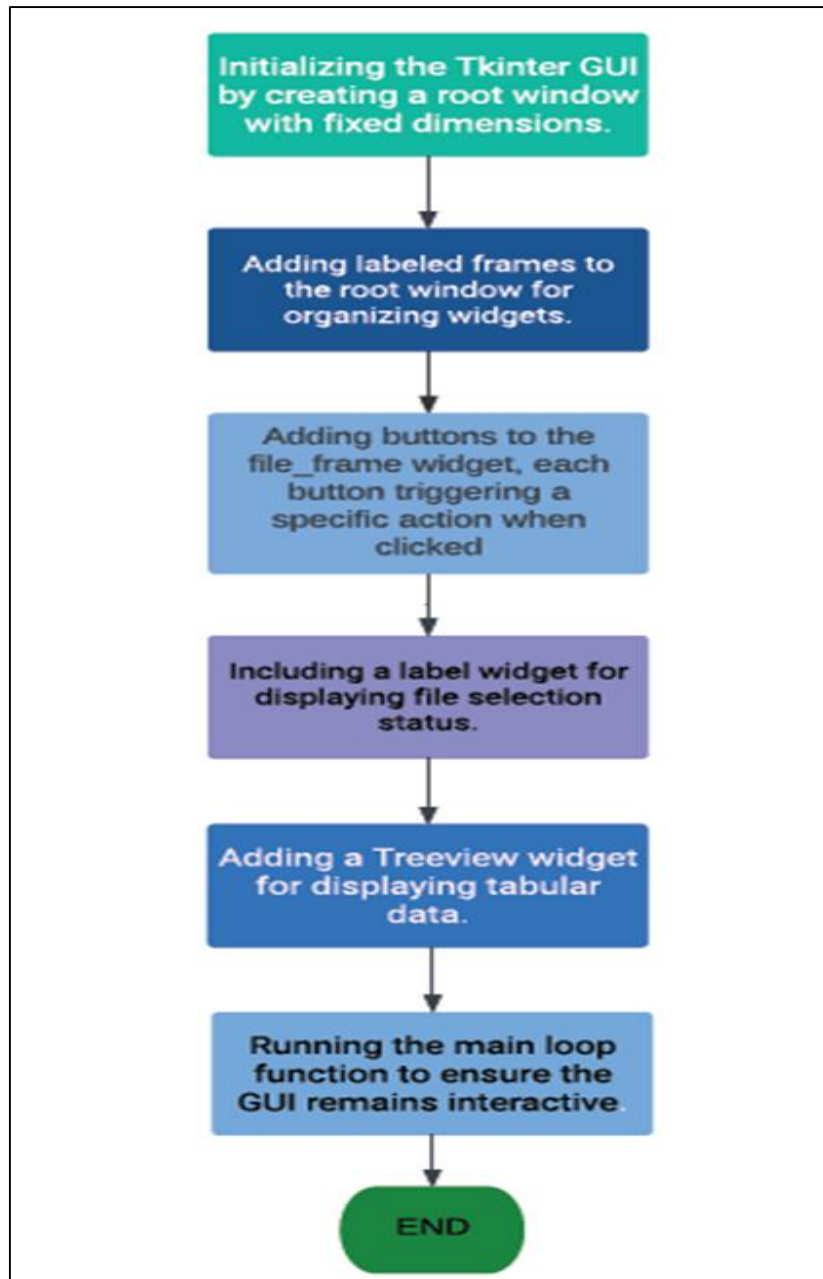


Figure 2.6: The flowchart of Building the User Interface Components

We create button for actions requested by students and by the administrator workers:

For the students 2 buttons: take a picture button and attendance button.

For security or administrator, three buttons: entered button, exited button and number in button.

2.5.1 Designing the Interface for Collecting Student's Pictures"

One such important aspect of our system is capturing and managing student pictures. To simplify this process and enhance user interaction, we create a graphical interface. This interface enables the capture of student pictures (Figure 2.7).

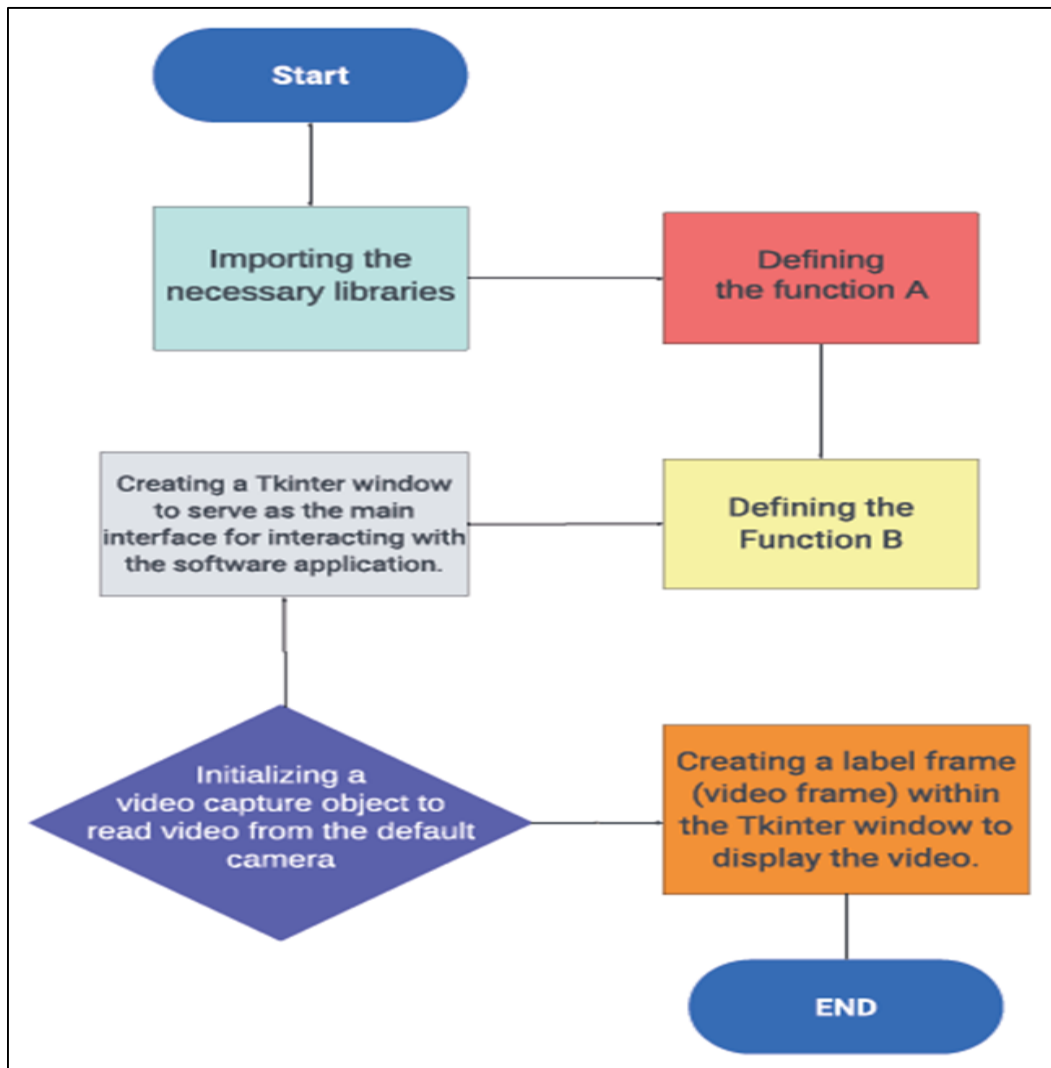


Figure 2.7: The flowchart of creating GUI for Data collecting.

- Function A is performed through the following steps (Figure 2.8):
 - a. Read a frame from the video source using OpenCV function.
 - b. Retrieve the photo name entered in a textbox.
 - c. Create a save path for the captured photo by appending the name with a specified directory path.
 - d. Save the captured frame as a JPEG image at the designated path using `cv2.imwrite()`.

- e. Display a message box to indicate successful addition.
- f. Update the status label with a success message.

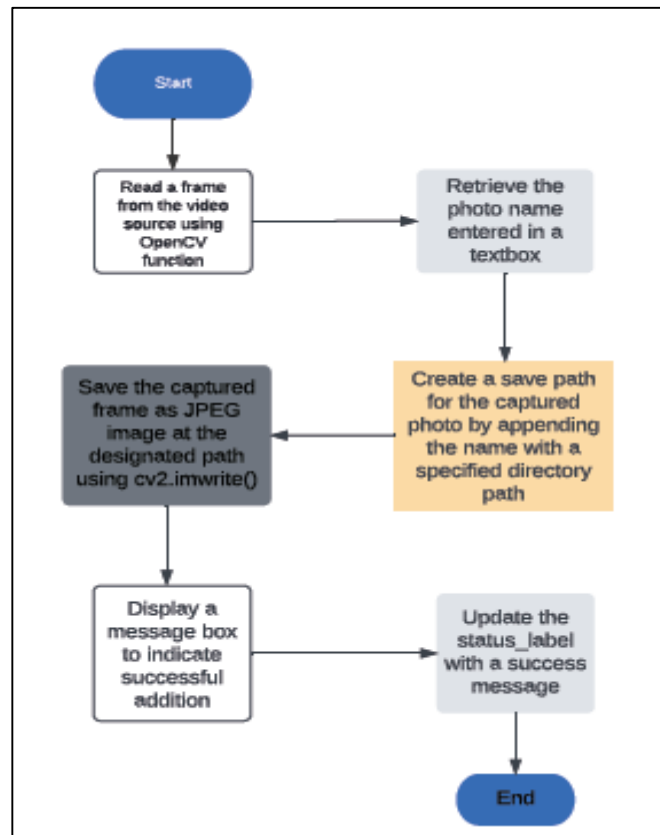


Figure 2.8: Function A flowchart.

- The function B should perform the following actions (Figure 2.9):
 - a. Read a frame from the video source using `video capture.read()` from OpenCV.
 - b. Convert the frame from the BGR format to the RGB format using `cv2.cvtColor()`.
 - c. Resize the frame to fit within the dimensions of the label widget.
 - d. Convert the resized frame into a PIL ImageTk object using `ImageTk.PhotoImage()` from the PIL library.
 - e. Update the label widget by configuring its image attribute with the PIL ImageTk object.
 - f. Schedule the function to be called again after a delay of 30 milliseconds.

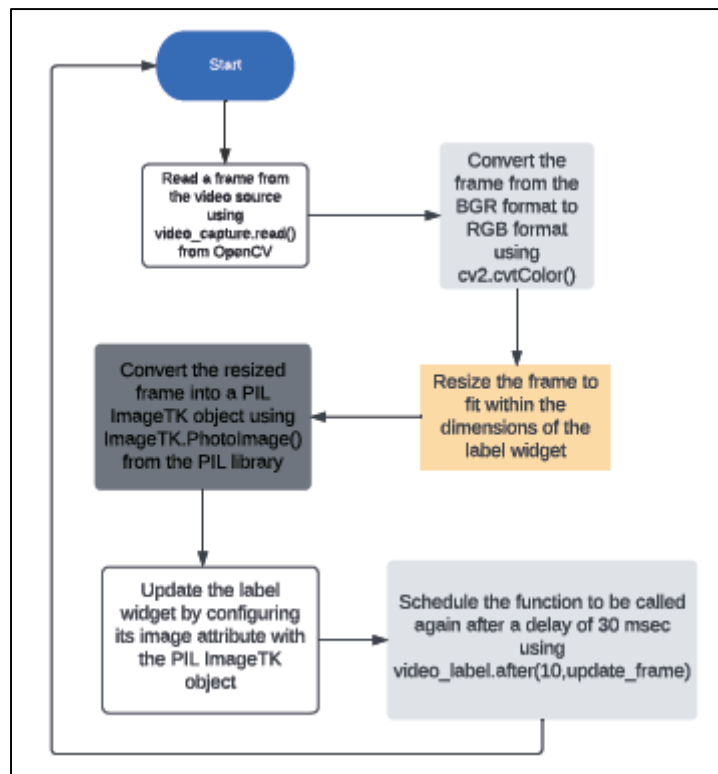


Figure 2.9: Function B flowchart.

2.5.2 Data base creation

The face recognition process compares newly captured faces with a database of dormitory resident facial characteristics therefore

To create face signatures, we start by importing libraries in Python, including face recognition, OpenCV, NumPy, and Os. These libraries provide essential functionality for face detection and recognition. Next, we define a list of variables that will be used throughout the process.

We then enter a loop to read the pictures using OpenCV from the list. This loop allows us to process an unlimited number of images of different individuals. Each image is associated with a specific name for identification.

Using pre-trained FaceNet algorithms, we detect the presence of faces in each image and locate bounding boxes around them. The code also identifies key facial landmarks such as the eyes, nose, and mouth. These landmarks serve as reference points for precise tracking of facial features.

By analyzing the positions of facial landmarks, we perform feature extraction. This involves calculating various features, such as the distances between the eyes, the angle of the eyebrows, or the shape of the mouth. These features help distinguish and characterize each

face (Figure 2.10).

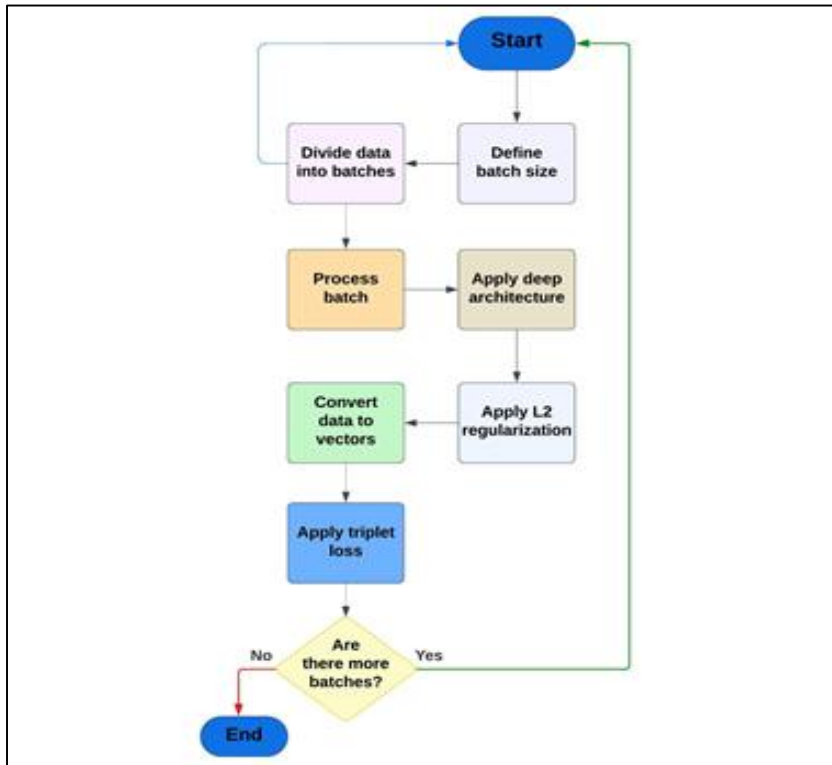


Figure 2.10: FaceNet, Deep learning Flowchart.

To obtain face embeddings, we resize the images and utilize FaceNet, which generates a vector of 128 numbers for each image. These embeddings represent unique signatures of the faces, capturing their distinguishing features (Figure 2.11).

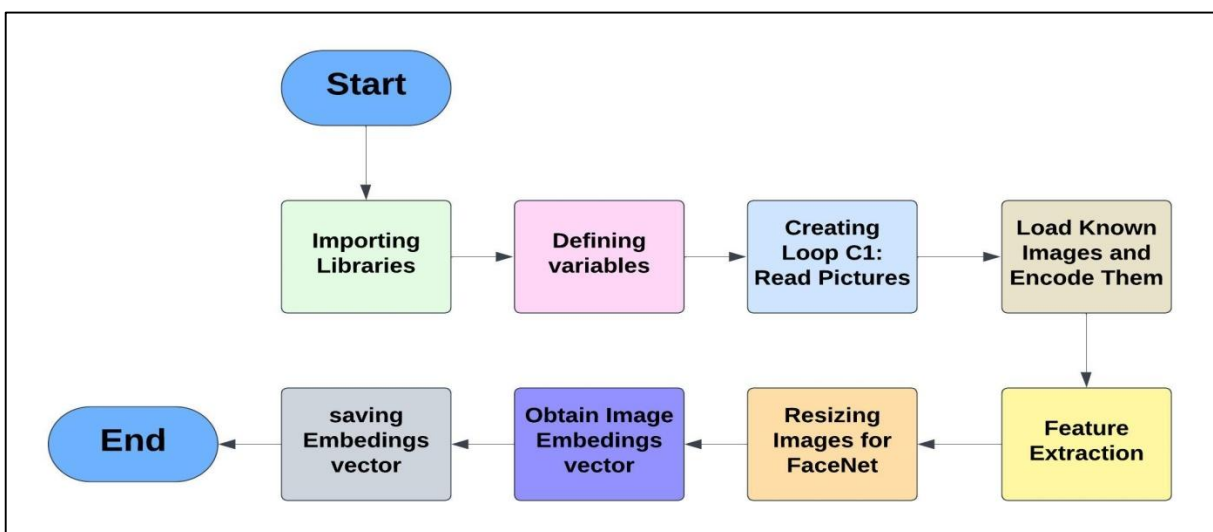


Figure 2.11: Data base creation Flowchart..

2.5.3 Facial recognition

Moving to real-time face detection and recognition, we open the camera using OpenCV library. We can specify either the laptop camera (index 0) or an external camera (index 1). The pre-processing steps for each captured frame involve resizing the frame for faster processing, converting the color format, and finding all the faces and their encodings.

After obtaining the face encodings, we compare them with the known faces in our database. We look for the closest match based on the smallest distance between the new face and the known faces.

For visualization, we draw a rectangle around the detected face and display a label with the name below it. Additionally, we can add the recognized name to an entry list using a "markattendance" function, which helps keep track of individuals.

Finally, the process ends, and we have successfully created face signatures, enabling us to detect and recognize faces in real-time scenarios. All these steps summarized in (Figure 2.12).

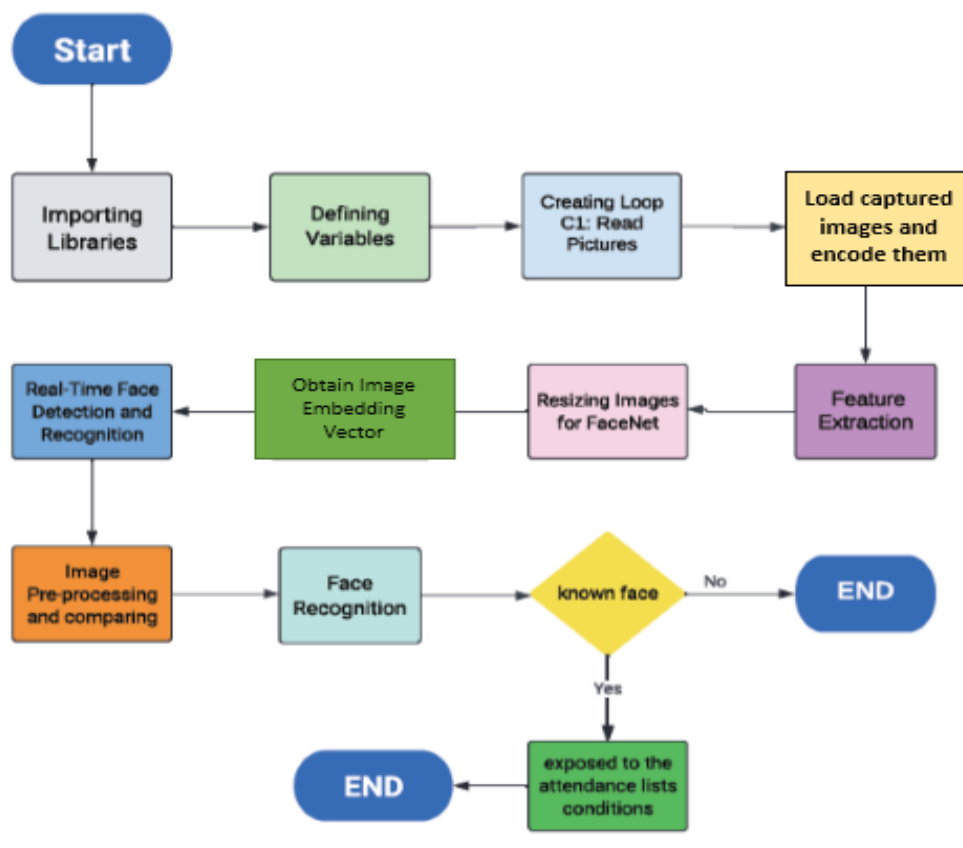


Figure 2.12: Flowchart of Face Signatures Creation for Real-Time Face Detection and Recognition.

2.5.4 Creating attendance lists

The system utilizes the pandas and datetime libraries in Python to create and maintain the three lists:

- Entry list: to store the names of students who entered.
- Exit list: to store the names of students who exited.
- Status list: to store the status of each student (in/out).

These lists help keep track of students who enter and exit the complex and their current status (in or out). The system follows a set of conditions to ensure accurate recording of attendance, taking into consideration previous entries and exits. The generated attendance list can be saved to a data interface and exported to an Excel file for further analysis and record-keeping. The flowchart illustrates the process of creating the attendance list and implementing the necessary conditions to accurately mark student attendance (Figure 2.13).

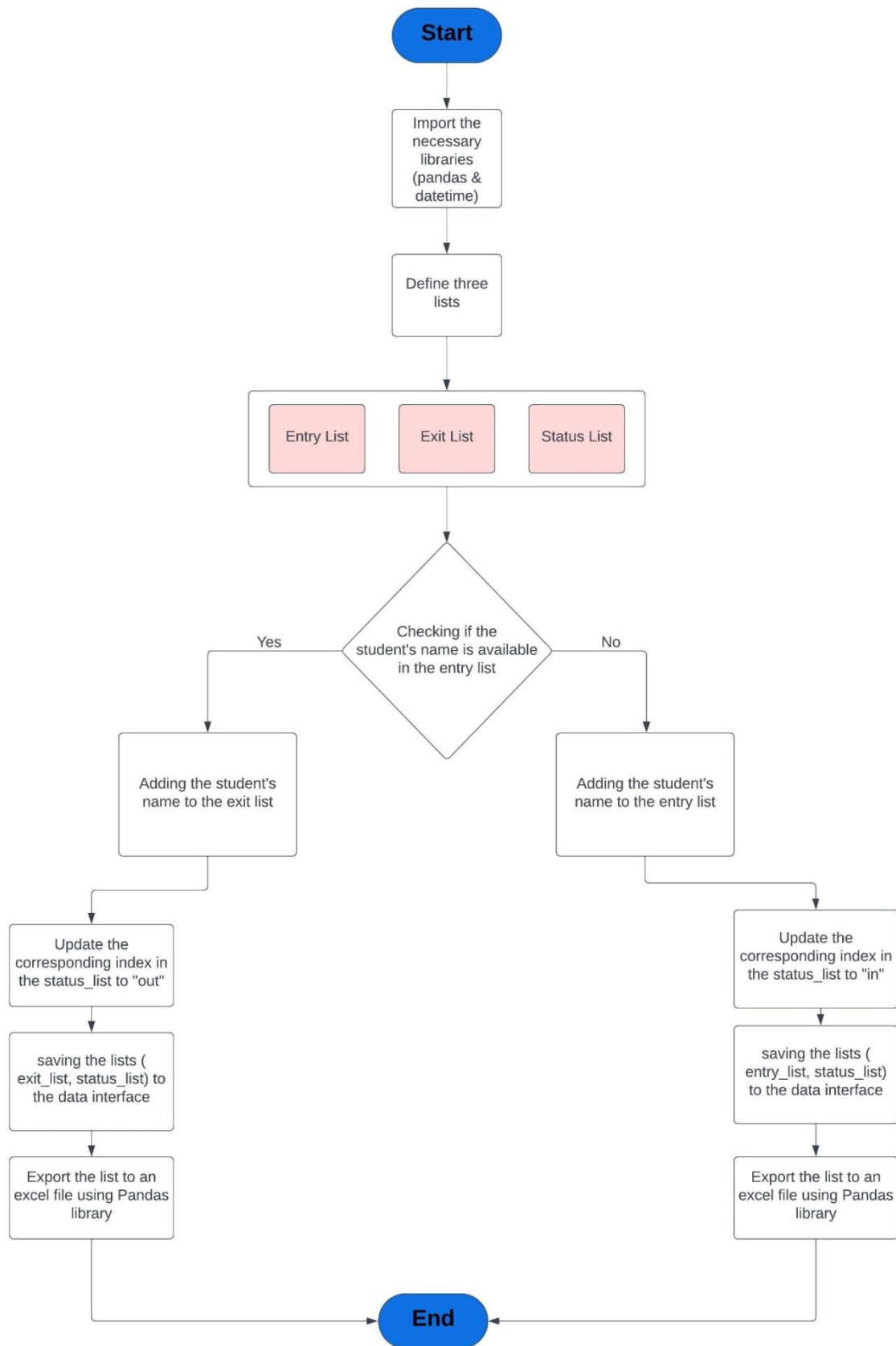


Figure 2.13: Flowchart of Attendance List Creation Function and Conditions

2.6. Conclusion

This chapter provides the details of the system design. How we can create and prepare the resident database that will be used for students face recognition. Moreover, the steps and details of creating the Graphical User Interface.

The next chapter will focus on the implementation and give experimental results to investigate the performance and feasibility of the proposed system. We will create a GUI with commands that activated the facial recognition programs based FaceNet pre-trained algorithms

CHAPTRE 03
IMPLEMENTATION AND
RESULTS

Chapter 03: Implementation and results

3.1. Introduction

In this chapter, we will explore the steps of creating and programming our codes for the attendance system using Dlib's HOG algorithm for face detection and the FaceNet model for recognition by python language.

We will focus on counting the final number of students inside the campus. We will create a graphical user interface (GUI) to provide a user-friendly interface for accessing lists.

The GUI simplifies the interaction and facilitates system management.

3.2. Work Environment

3.2.1 Materials

- Laptop hp Latitude E5440.
- Processor Intel Core i3-5010U @ 2.10 GHZ.
- Graphical Card GT-720M.
- Installed memory (RAM) 8.00 GB.
- Windows 10 pro 64 bit.
- Visual Studio Code
- Webcam

3.2.2 PYTHON

Python is a high-level object-oriented programming language. It has dynamic semantics and built-in data structures that facilitate rapid application development. Due to its dynamic binding and typing, Python acts as a glue or scripting language that helps to integrate components within an application or between applications [26].

Python code that integrates also with the chosen computer vision or facial recognition library. By leveraging these libraries, developers can easily access the underlying algorithms and models, saving time and effort in implementing complex image-processing tasks from scratch.

3.2.3 Visual Studio

VS Code, or Visual Studio Code, is an editor for source code created by Microsoft with the Electron Framework and is available on Windows, Linux, and macOS. It offers a range of features such as debugging support, syntax highlighting, intelligent code completion, and code refactoring, and embedded Git, among others.

3.2.4 The webcam used

A webcam is a compact digital video camera that functions as a real-time input device, capturing digital images. These images are then transferred to a computer, which subsequently sends them to a server. From there, the images can be transmitted to the hosting page. The primary purpose of a webcam is to provide a clear and bright image, offering a satisfactory level of image quality.

The HP webcam, for instance, boasts a resolution of 1280x720 pixels, which has needed specifications for achieving high-definition (HD) quality [31].

We use the HP webcam for test during the development phase of our project.

3.2.5 The libraries

3.2.5.1. Dlib

The dlib library is a C++ toolkit that features machine learning tools and algorithms. It is important to install it to use the `face_recognition` library.

3.2.5.2. OpenCV

OpenCV is an open-source computer vision and machine learning software library that provides a common infrastructure for computer vision applications and accelerates the use of machine perception.

It is a comprehensive set of classic and state-of-the-art computer vision and machine learning algorithms. OpenCV can be used to detect and recognize faces [22].

3.2.5.3. NumPy

NumPy is a Python library used for working with arrays. It is particularly useful for performing mathematical operations on arrays. NumPy arrays are used to represent images in OpenCV when using Python [23].

3.2.5.4. Face_Recognition

Face_recognition is a Python library that wraps dlib's face recognition functionality, making it easy to work with. It is one of the additional libraries that need to be installed to perform facial recognition with Python and OpenCV. The library contains an implementation of "deep metric learning" which is used to build face embeddings used for the actual recognition process [23].

3.2.5.5. Datetime

Datetime is a module in Python that provides classes for working with dates and times.

3.2.5.6. Pandas

Pandas are a Python library for data analysis and manipulation. It provides fast, flexible and expressive data structures that make working with "relational" or "tagged" data simple and intuitive [25]. It allows Python programs to read and modify Excel files.

3.2.5.7. Os

This module is responsible for using operating system dependent functionality, as creating folder, files, opening, copying, cutting and pasting.

3.2.5.8. Tkinter

Tkinter is a Python module used to create Graphical User Interface (GUI) applications. It is one of the most used modules for creating GUI applications in Python as it is simple and easy to work with.

3.3. Software development

In our Python code, we relied on a trained model and updatable, which means one picture in our data is enough for the program to recognize the face, and with the continuous exposure to the camera, the program enriches the data and elevates the accuracy.

3.3.1 The Steps for Creating the Code

3.3.1.1. Installing the Libraries

First, we use the instruction “pip install “in the command window to install libraries in the computer:

- Pip install cmake
- Pip install dlib
- Pip install opencv-python
- Pip install face_recognition
- Pip install numpy
- Pip install datetime
- Pip install pandas

3.3.1.2. Creating the Graphical User Interface (GUI)

To create a user interface for our system, we used the Tkinter library, so first of all, importing the Tkinter module into our Python script (Figure 3.1).

Create the main window for our interface. This window will contain all the elements of our interface as the function and the conditions.

```
import tkinter as tk
from tkinter import filedialog, messagebox, ttk
import main
from main import capture, load_entry, load_sortie, clac
import pandas as pd
from subprocess import call
```

Figure 3.1: Importing libraries.

The next step is to define all the functions we need to create:

- The first function clears the existing data in the Treeview, sets the column names as headings, converts the DataFrame rows into a list of lists, and inserts each row into the Treeview widget with its corresponding column values. The function helps populate the Treeview with data from the provided DataFrame (Figure 3.2).

```
def load_data(df):
    clear_data()
    tv1["column"] = list(df.columns)
    tv1["show"] = "headings"
    for column in tv1["columns"]:
        tv1.heading(column, text=column)

    df_rows = df.to_numpy().tolist()
    for row in df_rows:
        tv1.insert("", "end", values=row)
    return None
```

Figure 3.2: Defining column creation function.

- The (Figure 3.3) consists of functions that perform various tasks related to manipulating data in a Treeview widget, capturing data, displaying messages, and executing external scripts.

```
def clear_data():
    tv1.delete(*tv1.get_children())
    return None

def entry():
    df=load_entry()
    load_data(df)

def sortie():
    df=load_sortie()
    load_data(df)

def capt():
    clear_data()
    capture(True)

def affiche():
    messagebox.showinfo("In", "number of student inside:"+clac())

def impor():
    call(["python", "photo.py"])
```

Figure 3.3: Defining manipulation function.

- The provided code initializes a Tkinter GUI by creating a root window with fixed dimensions. The root window is set to a size of 900x500 pixels. Two labeled frames, frame1

and `file_frame`, are added to the root window. `Frame1` is positioned at the top of the window and `file_frame` is positioned at 65% of the window's height on the left side. These frames serve as containers for other widgets in the GUI (Figure 3.4).

```
root = tk.Tk()
root.geometry("900x500")
root.pack_propagate(False)
root.resizable(0, 0)
frame1 = tk.LabelFrame(root, text="Data")
frame1.place(height=300, width=900)
file_frame = tk.LabelFrame(root, text="Options")
file_frame.place(height=100, width=900, rely=0.65, relx=0)
```

Figure 3.4: Creating the widgets of GUI.

- Add multiple buttons to the `file_frame` widget in the Tkinter GUI. Each button has a specific role and triggers a corresponding action when clicked. The "Entred" button calls the `entry()` function, the "Exited" button calls the `sortie()` function, the "Attendance" button calls the `capt()` function, the "Take a picture" button calls the `impor()` function, and the "number in" button calls the `affiche()` function. These buttons are positioned within the `file_frame` widget and provide convenient interactions for various functionalities in the GUI (Figure 3.5).

```
button1 = tk.Button(file_frame, width=20, text="Entred", command=lambda: entry())
button1.place(rely=0.65, relx=0.20)
button2 = tk.Button(file_frame, width=20, text="Exited", command=lambda: sortie())
button2.place(rely=0.65, relx=0.0)
button1 = tk.Button(file_frame, width=20, text="Attendance", command=lambda: capt())
button1.place(rely=0.65, relx=0.40)
button1 = tk.Button(file_frame, width=20, text="Take a picture", command=lambda: impor())
button1.place(rely=0.65, relx=0.60)
button1 = tk.Button(file_frame, width=20, text="number in", command=lambda: affiche() )
button1.place(rely=0.65, relx=0.80)
```

Figure 3.5: Creating buttons.

- Includes a label widget to display file selection status, a Treeview widget to display tabular data, and scrollbars for vertical and horizontal scrolling within the Treeview. The `mainloop()` function ensures the GUI remains interactive (Figure 3.6).

```

label_file = ttk.Label(file_frame, text="No File Selected")
label_file.place(rely=0, relx=0)

tv1 = ttk.Treeview(frame1)
tv1.place(relheight=1, relwidth=1)

treescrollx = tk.Scrollbar(frame1, orient="horizontal", command=tv1.xview)
treescrollx.pack(side="bottom", fill="x")
treescrollx.pack(side="right", fill="y")
tk.mainloop()

```

Figure 3.6: Interactive GUI function.

- As a result of running the code, the GUI (Graphical User Interface) will be displayed as shown in (Figure 3.7).

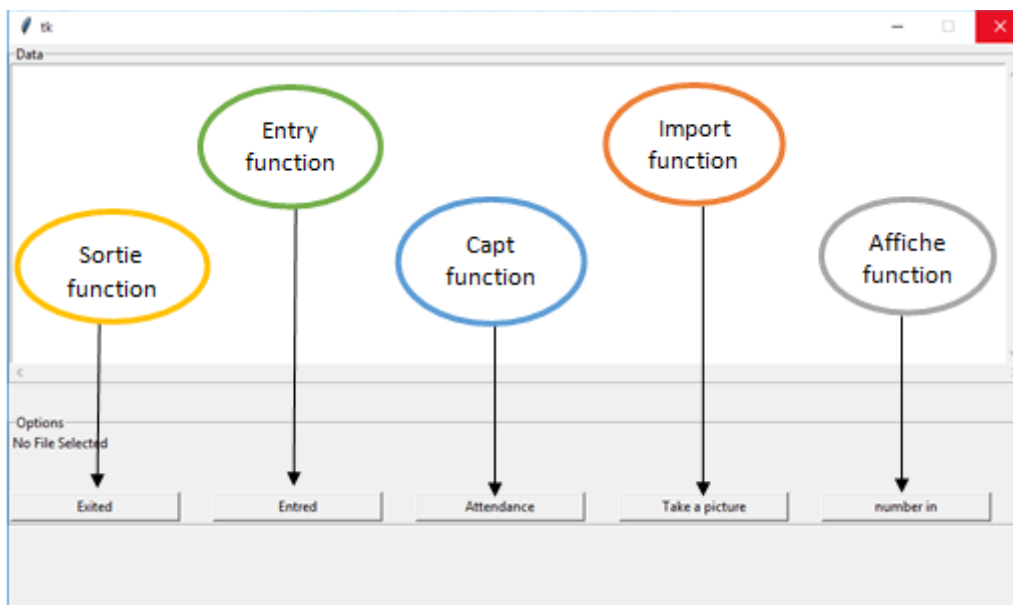


Figure 3.7: the Graphical User Interface

3.3.1.3. Building a Students Picture Collection Interface

For making collecting data easier for administrator and students to interact with software applications, we created a graphical interface user also for taking pictures of students.

- Starting by importing the Tkinter library for GUI and CV2 library to capture the faces (Figure 3-8).

```
import tkinter as tk
import cv2
from PIL import ImageTk, Image
from tkinter import messagebox
```

Figure 3.8: Importing libraries.

- Define the `capture_photo()` function captures a frame from a video source using OpenCV's `video_capture.read()` function. It retrieves the photo name entered in a textbox and creates a save path for the captured photo by appending the name with a specified directory path. The captured frame is then saved as a JPG image at the designated path using `cv2.imwrite()`. A message box is displayed to indicate successful addition, and the `status_label` is updated with a success message. This function essentially captures a photo from a video stream, saves it with the provided name, and provides feedback to the user about the successful capture and save operation (Figure 3-9).

```
save_path = "C:\Users\Hp Pc\Desktop\test\persons" + photo_name + ".jpg"
cv2.imwrite(save_path, frame)
messagebox.showinfo("info", "added succes")
```

Figure 3.9: Taking pictures code.

- As a result of running the code (Figure 3.10) and (Figure3.11).

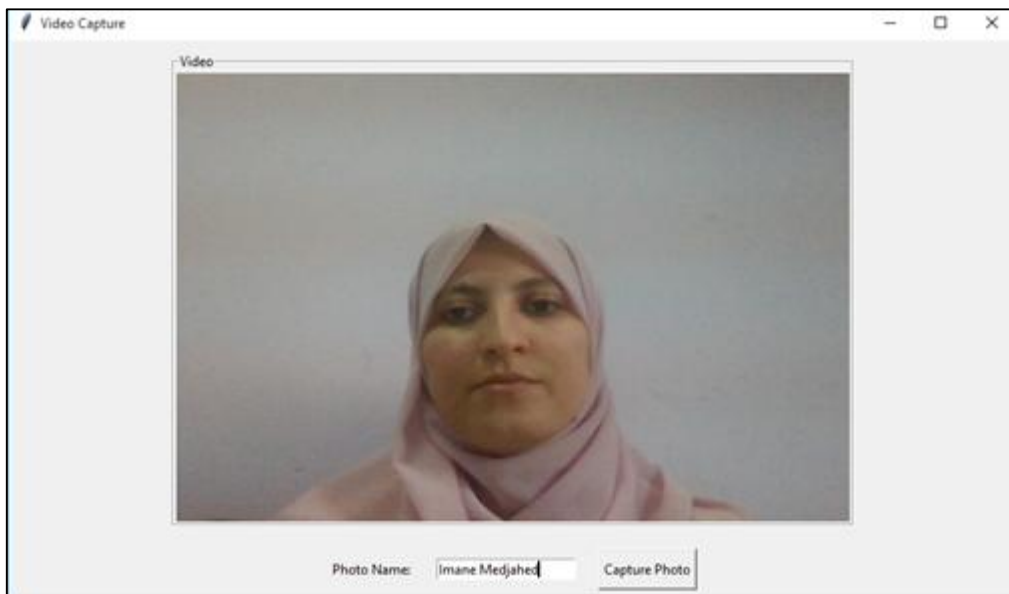


Figure 3.10: Video capture interface.

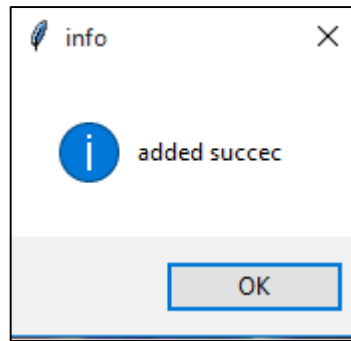


Figure 3.11: the message will appear.

- The `update_frame()` function is responsible for continuously updating the video frame displayed in a label widget. It reads a frame from a video source using `video_capture.read()` from OpenCV. The frame is then converted from the BGR format to the RGB format using `cv2.cvtColor()`. Next, the frame is resized to fit within the dimensions of the label widget. The resized frame is converted into a PIL ImageTk object using `ImageTk.PhotoImage()` from the PIL library. The label widget is then updated with the new frame by configuring its image attribute with the PIL ImageTk object. Finally, the function schedules itself to be called again after a delay of 10 milliseconds using `video_label.after(10, update_frame)`. This creates a continuous loop, resulting in a live video feed being displayed in the label widget (Figure 3.12).

```
def update_frame():  
  
    ret, frame = video_capture.read()  
    frame_rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)  
    frame_resized = cv2.resize(frame_rgb, (label_width, label_height))  
    img = ImageTk.PhotoImage(image=Image.fromarray(frame_resized))  
    video_label.configure(image=img)  
    video_label.image = img  
    video_label.after(10, update_frame)
```

Figure 3.12: Image processing.

- Tkinter window with a specific dimensions. It initializes a video capture object to read video from the default camera. The video will be displayed within a labeled frame (`video_frame`) placed within the Tkinter window.

```

video_capture = cv2.VideoCapture(0)
window = tk.Tk()
window.title("Video Capture")
window.geometry("900x500")
video_frame = tk.LabelFrame(window, text="Video")
video_frame.pack(padx=10, pady=10)

```

Figure 3.13: video capture window.

3.3.1.4. Creating the database of Face Signatures

To create the faces signatures program, we call some important libraries in Python: face recognition, OpenCV, numpy, and Os. After creating the list of variables.

The next step is creating loop C1 for reading the pictures by the opencv from the data collected (Figure 3.14), there isn't a limited number for adding pictures of persons, and the instruction "className" read the name of each image (Figure 3.15).

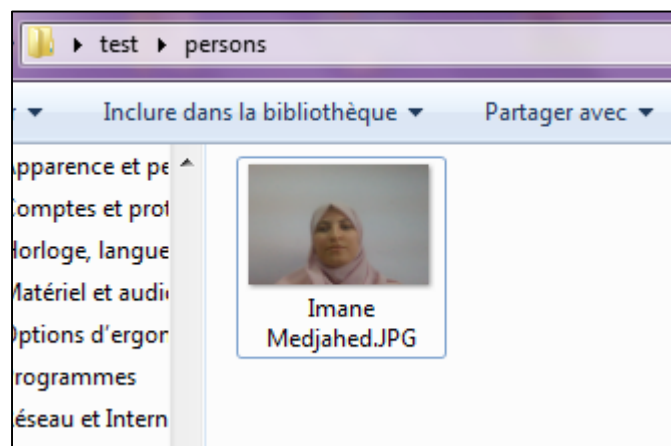


Figure 3.14 : the data collected.

```

for c1 in personsList:
    curPersons = cv2.imread(f'{path}/{c1}')
    images.append(curPersons)
    classNames.append(os.path.splitext(c1)[0])
print(classNames)

```

Figure 3.15 : Load the known images and encode them

Using pre-trained FaceNet algorithms (Figure 3.16), the code detects the presence of faces in an image or video frame. It locates the bounding box around each detected face.

Once the faces are detected, the code identifies key facial landmarks, such as the eyes, nose, and mouth. These landmarks serve as reference points to locate and track facial features precisely.

Feature extraction by analyzing the positions of the facial landmarks, the code can extract various features. For example, the distances between the eyes, the eyebrows' angle, or the mouth's shape can be calculated (Figure 3.17).

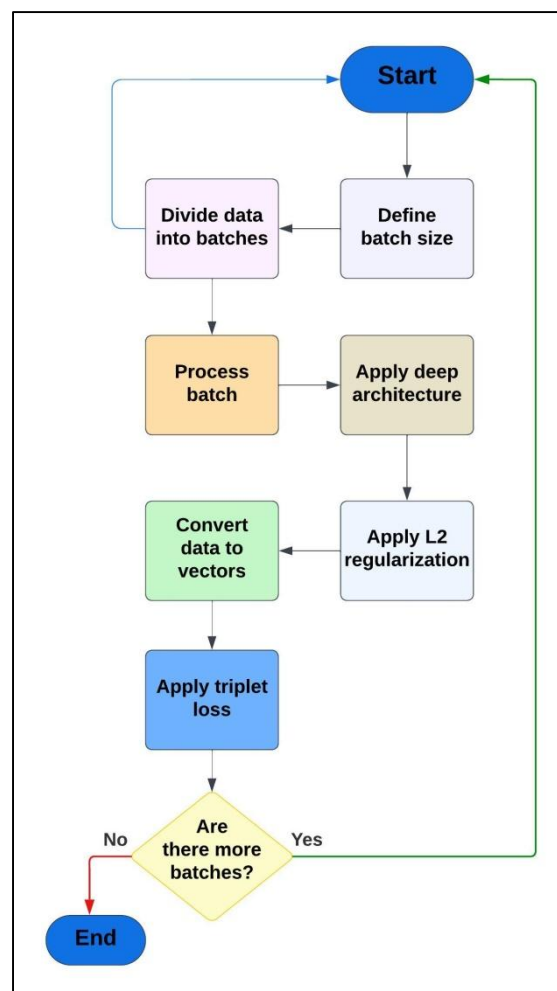


Figure 3.16: FaceNet, Deep Learning Architecture.

```

def findEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList

encodeListKnown = findEncodings(images)
print('Encoding Complete.')

```

Figure 3.17 : Feature extraction.

It depends on resized images, which will give us a vector of 128 numbers for each image “embedding” (Figure 3.18).

```

['Imen']
[array([-0.09033273,  0.072135 ,  0.01142237, -0.15468268, -0.13195851,
        -0.02958804, -0.04001168, -0.10417762,  0.19095084, -0.2195515 ,
         0.11692587, -0.11032686, -0.21852693,  0.06142154, -0.06986907,
         0.23608756, -0.14884086, -0.14246953, -0.09559368, -0.07765993,
         0.06127493,  0.07171903,  0.02354544,  0.10196082, -0.04822369,
        -0.31682837, -0.10426553, -0.06088898,  0.005429 , -0.04957286,
         0.0079825 ,  0.08143029, -0.10077191, -0.00498023,  0.03321079,
         0.10772241,  0.04518942, -0.08544032,  0.21036676,  0.01403544,
        -0.26431125,  0.00220962,  0.14310183,  0.24572133,  0.12928286,
         0.06197201,  0.01830485, -0.1369905 ,  0.10879688, -0.26883709,
         0.0297863 ,  0.03930248,  0.01361453,  0.02265805,  0.04791982,
        -0.21481372,  0.05103078,  0.16446909, -0.21008879,  0.07425629,
         0.09430848, -0.13608888, -0.02468558, -0.04533651,  0.31031892,
         0.1272061 , -0.13864557, -0.16134027,  0.23531099, -0.14323118,
        -0.06506482,  0.07280885, -0.10517276, -0.13044272, -0.29061142,
        -0.00757861,  0.47642583,  0.16892715, -0.09743071,  0.05392373,
        -0.10229231,  0.01150568,  0.08889668,  0.1331376 , -0.03753785,
        -0.0740613 , -0.10196372,  0.00911235,  0.23306949, -0.03740004,
        -0.04828491,  0.20939837,  0.03975929,  0.07150587, -0.06164193,
         0.08032266, -0.17672551, -0.00686728, -0.19291078, -0.04678009,
        -0.00555965, -0.007628 ,  0.00178167,  0.17280507, -0.21608102,
         0.12939821, -0.03957316, -0.10130354, -0.03531248,  0.10098694,
        -0.06796245, -0.03423727,  0.14471315, -0.21784864,  0.10694793,
         0.22099987,  0.12040986,  0.17690292,  0.13349128,  0.08515773,
         0.07373254, -0.06861115, -0.24780284, -0.05339889,  0.09406544,
        -0.01196914,  0.07046232,  0.11067271]])]
Encoding Complete.

```

Figure 3.18: Image Embedding Vector.

In the next part the code will moving to the real-time face detection and recognition by opening the camera by OpenCV library, using the index (0) for the camera of the laptop or the index (1) for an external camera (Figure) by opening the interface of collecting pictures .

The pre-processing steps of image (Figure 3.19):

- Capture a single frame from the video stream
- Resize the frame to 1/4 size for faster face recognition processing
- Convert the image from BGR color (OpenCV) to RGB color (face_recognition)
- Find all the faces and face encodings in the current frame
- See if the face matches any known faces
- Use the known face with the smallest distance to the new face recognition.

```
def capture(cont:bool):
    cap = cv2.VideoCapture(0)

    while True and cont:
        _, img = cap.read()

        imgS = cv2.resize(img, (0,0), None, 0.25, 0.25)
        imgS = cv2.cvtColor(imgS, cv2.COLOR_BGR2RGB)

        faceCurentFrame = face_recognition.face_locations(imgS)
        encodeCurentFrame = face_recognition.face_encodings(imgS, faceCurentFrame)

        for encodeface, faceLoc in zip(encodeCurentFrame, faceCurentFrame):
            matches = face_recognition.compare_faces(encodeListKnown, encodeface)
            faceDis = face_recognition.face_distance(encodeListKnown, encodeface)
            matchIndex = np.argmin(faceDis)
```

Figure 3.19: Images pre-processing

- Draw a rectangle around the face
- Draw a label with the name below the face (Figure 3.20).

```

if matches[matchIndex]:
    name = classNames[matchIndex].upper()
    print(name)
    Find=True
    y1, x2, y2, x1 = faceLoc
    y1, x2, y2, x1 = y1*4, x2*4, y2*4, x1*4
    cv2.rectangle(img, (x1, y1), (x2, y2), (0,0,255), 2)
    cv2.rectangle(img, (x1,y2-35), (x2,y2), (0,0,255), cv2.FILLED)
    cv2.putText(img, name, (x1+6, y2-6), cv2.FONT_HERSHEY_COMPLEX, 1, (255,255,255), 2)
    df_1,df_2,df_state=data_frames()
    markAttendance(name,df_1,df_2,df_state)
    cont=show()
cv2.imshow('Face Recognition',img)
cv2.waitKey(1)
if not cont :
    cv2.destroyAllWindows()

```

Figure 3.20: Images Addition for detection

- The last option is adding the recognized name to the entry list by 'markattendance' function

3.3.1.5. Creation the Attendance List Function and Conditions

The important part in our programming code is the function we used that marks the attendance of students with three lists, the first one for entering students, the second list for exiting from the complex, and the third one the status of students still inside the complex.

```

def load_entry():
    try:
        df_1 = pd.read_excel('Attendance {0}.xlsx'.format(str(datetime.date.today())),"entry")
    except:
        try:
            names = pd.read_excel('Attendance {0}.xlsx'.format(str(datetime.date.today() - datetime.timedelta(days=1))), "state", index_col=None)
            df_1 = pd.DataFrame({'Index':[1 for i in range(0,len(names['Name']))], 'Name': names['Name'], 'Time': [0] * len(names['Name']), 'Date':
            except:
                df={"Index":[],"Name":[],"Time":[],"Date":[]}
                df_1=pd.DataFrame(df)
    return df_1

def load_sortie():
    try:
        df=pd.read_excel(('Attendance {0}.xlsx'.format(str(datetime.date.today()))),"sortie")
    except:
        df={"Index":[],"Name":[],"Time":[],"Date":[]}
        df=pd.DataFrame(df)
    return df

```

Figure 3.21: Saving names in lists

This is achieved by:

- Saving the names of students in the lists depends on conditions, which is marking the attendance in the entry list first, if the name is available in the entry list, the name will be added in the exiting list and taking the last days in consideration.
- Saving the lists to the data interface and Excel file (Figure 3.22).

```
def data_frames():
    # Load entry
    try:
        df_1 = pd.read_excel('Attendance {0}.xlsx'.format(str(datetime.date.today())),"entry", index_col=None)
    except:
        try:
            names = pd.read_excel('Attendance {0}.xlsx'.format(str(datetime.date.today()) - datetime.timedelta(days=1)))
            df_1 = pd.DataFrame({'Index':[1 for i in range(1,len(names['Name']))], 'Name': names['Name'], 'Time': [0]})
        except:
            df={"Index":[],"Name":[],"Time":[],"Date":[]}
            df_1=pd.DataFrame(df)
    # Load sortie
    try:
        df_2 = pd.read_excel('Attendance {0}.xlsx'.format(str(datetime.date.today())),"sortie", index_col=None)
    except:
        df_2 = pd.DataFrame({'Index':[], 'Name': [], 'Time': [], 'Date': []})
    # Load state
    try:
        df_state = pd.read_excel('Attendance {0}.xlsx'.format(str(datetime.date.today())),"state", index_col=None)
    except:
        try:
            df_state = pd.read_excel('Attendance {0}.xlsx'.format(str(datetime.date.today()) - datetime.timedelta(days=1)))
        except:
            df_state = pd.DataFrame({'Name': [], 'State': []})
    return df_1,df_2,df_state
```

Figure 3.22: Attendance List Creation: Implementation conditional functions.

The result of testing in (Figure 3.23) one file contains the three lists.

Index	Name	Time	Date
0	NOUR SELMANE	21:42:49	20/06/23
1	IMANE MEDJAHED	21:44:34	20/06/23

Name	State
NOUR SELMANE	in
IMANE MEDJAHED	out

Figure 3.23: Excel file saved.

3.4. The camera Dahua

The university residence utilizes Dahua cameras for surveillance purposes (Figure 3.24). These cameras come in 12-megapixel (12 MP) and 15-megapixel (15 MP) resolutions, enabling them to capture highly detailed and clear images. The 12 MP camera produces images at approximately 4000x3000 pixels, while the 15 MP camera captures images at around 4608x3456 pixels [32]. The cameras offer enhanced image quality, making them suitable for applications where capturing fine details is essential, such as surveillance. Their higher megapixel resolutions make them potentially effective for face recognition, as they can capture intricate facial features. However, it is important to consider other factors such as lighting, camera placement, and the quality of face recognition software, as they also significantly impact the accuracy of face recognition outcomes.

We were allowed to use this camera, to test our security in a real working environment.



Figure 3.24: The Dahua camera.

3.5. RESULTS

As a result of running the code, the GUI (Graphical User Interface) will be displayed as shown in (Figure 3.25).

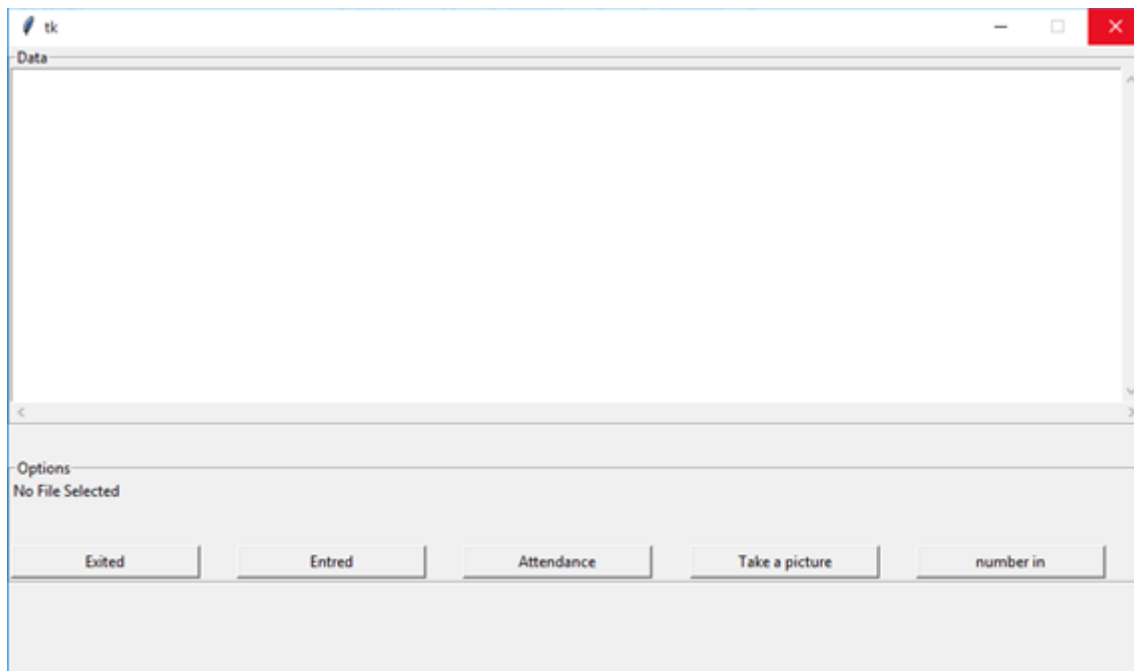


Figure 3.25: the Graphical User Interface.

The first step we take a picture by clicking on “Take a picture” button, the interface will appear the video capture interface (Figure 3.26) after writing the name and clicking on “capture photo” button. The system will save the picture taken with the name written in the database and show the message (Figure 3.27).

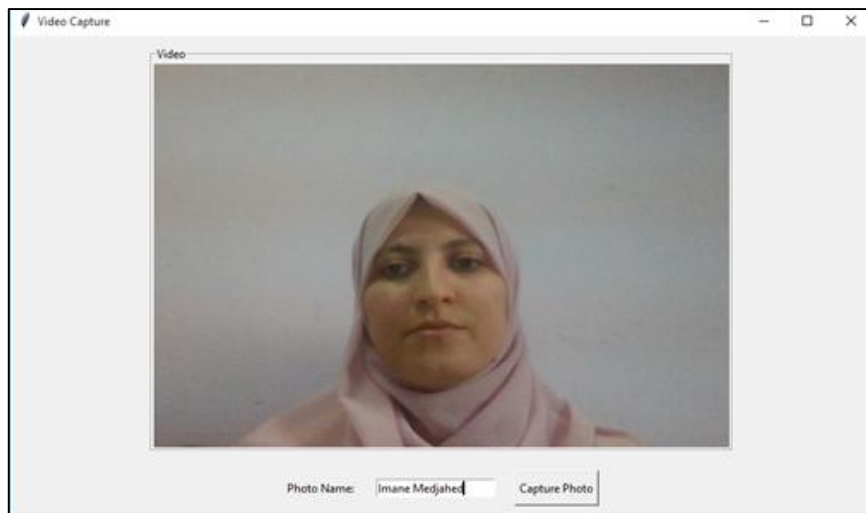


Figure 3.26: Video capture interface.

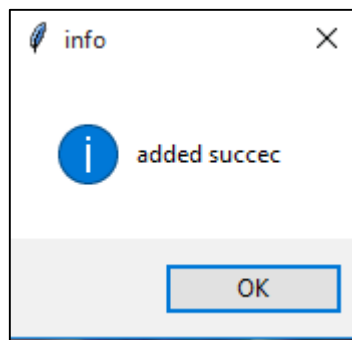


Figure 3.27: the message after collecting the picture.

Next, I did the attendance by clicking in the attendance button, if the detected face is recognized, the name of the student is displayed below the identified face (Figure 3.28).



Figure 3.28: Identification and recognition

The list will be created automatically with the name of the person, the date and the time.

The Lists will be created in interface data and Excel at the same time, and attendance will be updated on every entry or exit. I did the experiment with my friend Nour Selman, and I repeated the process several times for 3 days, as the numbers show (Figure 3.29) and (Figure 3.30).

Index	Name	Time	Date
0	IMANE MEDIAHED	10:18:41	22/06/23
1	NOUR SELMANE	14:31:40	22/06/23
2	IMANE MEDIAHED	14:33:12	22/06/23
3	NOUR SELMANE	14:39:04	22/06/23
4	IMANE MEDIAHED	19:48:13	22/06/23

Figure 3.29: The list of entered Students.

Index	Name	Time	Date
0	IMANE MEDIAHED	10:14:58	22/06/23
1	NOUR SELMANE	14:31:17	22/06/23
2	IMANE MEDIAHED	14:31:32	22/06/23
3	NOUR SELMANE	14:32:43	22/06/23
4	IMANE MEDIAHED	14:33:47	22/06/23
5	NOUR SELMANE	14:39:24	22/06/23

Figure 3.30: The list of exited Students.

Every day has its own lists of the entered and exited students with the date.

The last and most important option that our project aims at, is counting the number of students in residence, which we get by clicking on the button “number in” on the interface. The widow that contains the number it will appear (Figure 3.31).

This number can now be given to the dining hall manager or feeding personnel to determine the number of meals to be prepared to limit wasting goods.

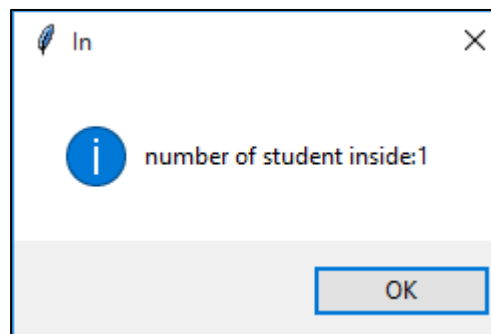


Figure 3.31: The widow of the final number.

3.6. Obstacles Discussion and Solutions Proposition

In the framework of discussing the obstacles and challenges facing the proposed system, the first obstacle is that the preparation of dinner begins at 4 pm, as our system counts the number of students in the residence and presents it to the committee responsible for preparing dinner. The number presented at that time is not final, because there are still students who will enter the campus and others who will leave, so it is approximate. In the pursuit of a solution, two proposals have been suggested:

- The first is the pointing pendulum inside the residence, where students register themselves in the morning, confirming attendance for lunch, dinner, or both. However, we saw that it is not practical, as it is not suitable for students who are not in the residence all day and come after 4 pm.
- The second proposal is the most appropriate and is represented by a mobile application that will be accessible to all students and in which several features and services can be added to students that contribute to improving living conditions on campus in the future.

3.7. Conclusion

In conclusion, this chapter has provided us with a comprehensive overview of the steps involved in creating a code programming for an attendance system, implemented in Python.

The focus was on creating lists and accurately counting the final number of students present inside the campus. Additionally, a graphical user interface (GUI) was developed to enhance the user experience and simplify the interaction with the system. The GUI not only provides a user-friendly interface for accessing lists but also streamlines system management.

This chapter has demonstrated the potential for creating an efficient and effective attendance system.

CHAPTRE 04

REPORT OF THE INTERNSHIP AT THE MLS COMPANY

Chapter 04: Report of the internship at the mls company

4.1. Introduction

In order to implement the studied system, I initiated contact with the director of the university residence “ZOUBIDA HAMADOUCHE SOUMAA 4” where I currently reside. The director expressed enthusiasm towards the idea, particularly due to a ministerial directive emphasizing the need to digitize the university services sector. He informed me that there is a company responsible for this matter; this company has permission from the Supreme Council of Security in Algeria. I contacted them, and they agreed to enrol in an internship with them, after signing a pledge stipulating that I would not release any private information except within the framework of my studies, also we can't implement our system because we are not a company and we don't have any authority.

This final chapter contains the essence of the internship and what I learned from them. The materials used for the realization of their system.

4.2. The MLS Company Security system architecture

Eurl Reseaux MLS is an Algerian company providing services of installation telephonic reseau and informatic, we cooperated with her to provide the necessary machines and provide assistance from her previous experiences in installation.

4.3. ZKTecho Company

ZKTecho is one of the world's largest manufacturers of RFID and biometric readers (fingerprint, face, and finger vein readers). Products include consumer products such as access control readers and panels, short- and long-range facial recognition cameras, and battery-powered door locks with fingerprint and facial recognition [33].

The company's security solutions are multi-lingual and localized in over 18 languages.

The founder of Supcon Technology has devoted himself to the original research and development of biometric authentication methods and the biometric SDK that was originally widely used in his PC security and his identity authentication fields. With continuous progress in development and numerous market applications and years of experience, the team has gradually built an identity authentication ecosystem and an intelligent security ecosystem based on biometric technology [33].

4.4. The MLS Company systems architecture

The MLS Company adopts two systems; the first one is to solve security issues while the second one to event doubling meals problem in the feeding service.

The second system will help the administration in conducting Data analysis by creating lists of students who took their meals, The administrator will compare the records of students who are physically present on campus with those of students who consumed meals in an effort to spot any notable variations that might be explained by factors like poor food quality or other underlying causes.

4.4.1 The Security System Design

The security system bases on a SpeedFace-V5L technology from ZKTecho a world's company and a tripod turnstiles working by NO (normal open) and NC (normal close) system.

The NONC system works with simple procedures, which opens, the tripod turnstile when a known student is recognized by V5L and keep the tripod turnstiles close for unknown persons. At the end, the system creates the list of students entered (Figure 4.1).

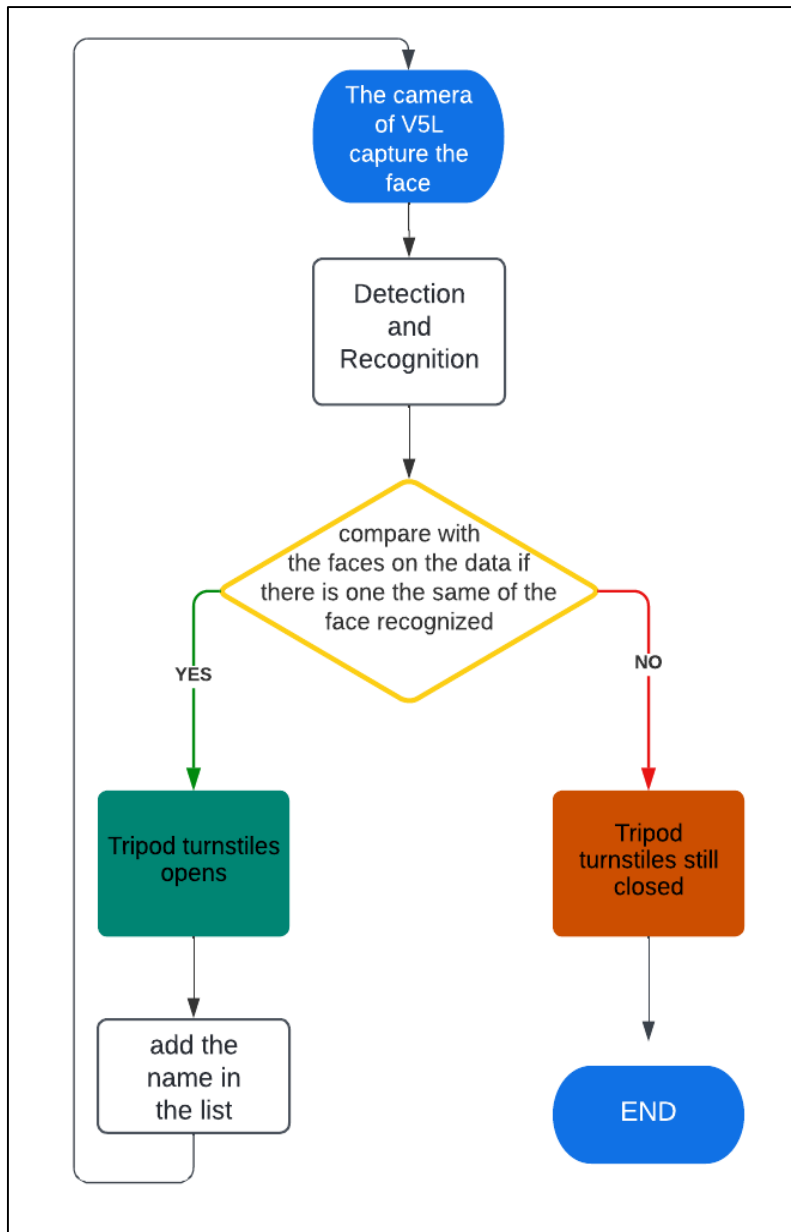


Figure 4.1: Flowchart of working MLS security system

4.4.2 The System Design of Feeding Service

For solving the problem of duplicated meals, they have created a system consisting of a central unit in which the software is installed, which contains the database, SpeedFace-V5L for facial recognition devices, and a ticket printer (Figure 4.2).

The process of this system resume in (Figure 4.3): when the student comes to the camera (he must be registered in the database previously), it recognizes him, the system gives an order to print a ticket containing his name, time and meal number, and the system will not print a ticket for him again until after the period of serving meals end.

Finally, the software will give the administration the list of students was taken the meal.

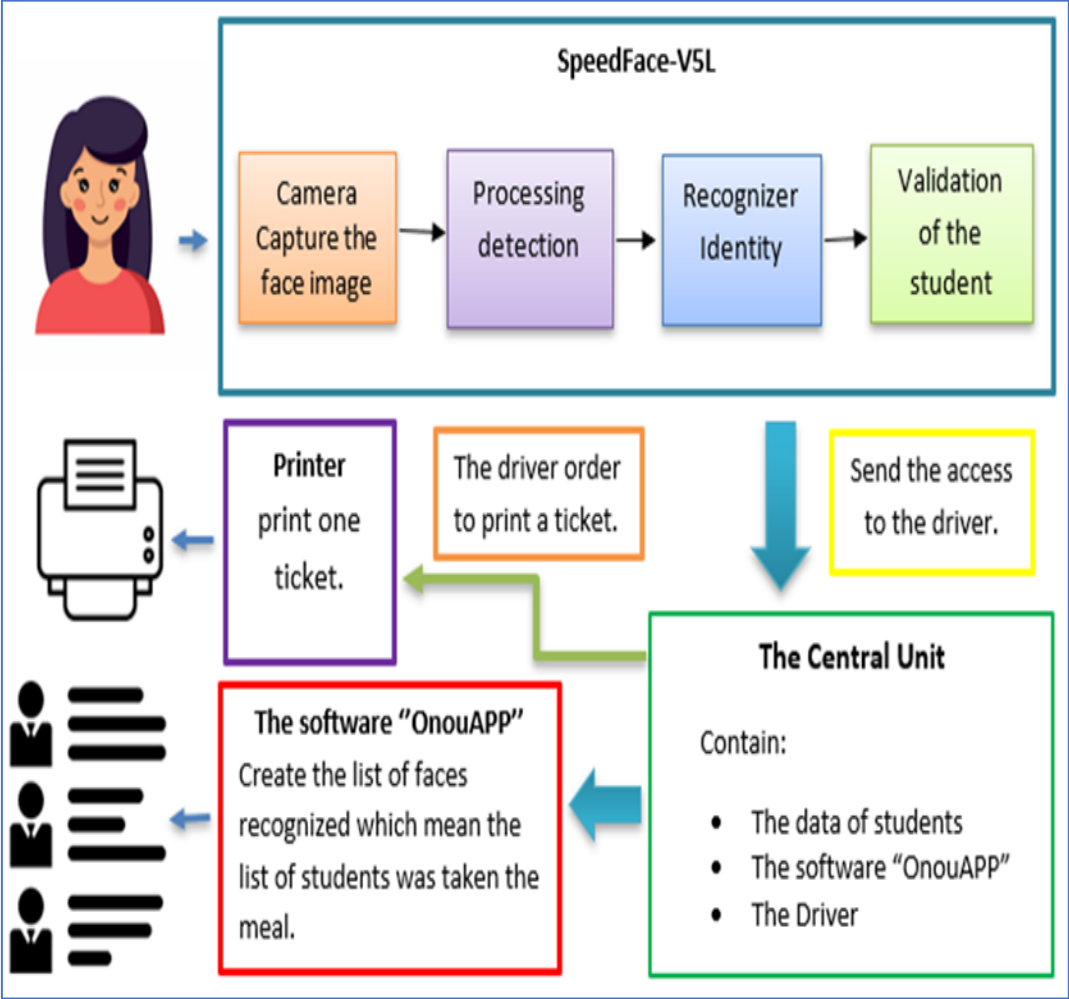


Figure 4.2: Feeding Service Conception System Architecture

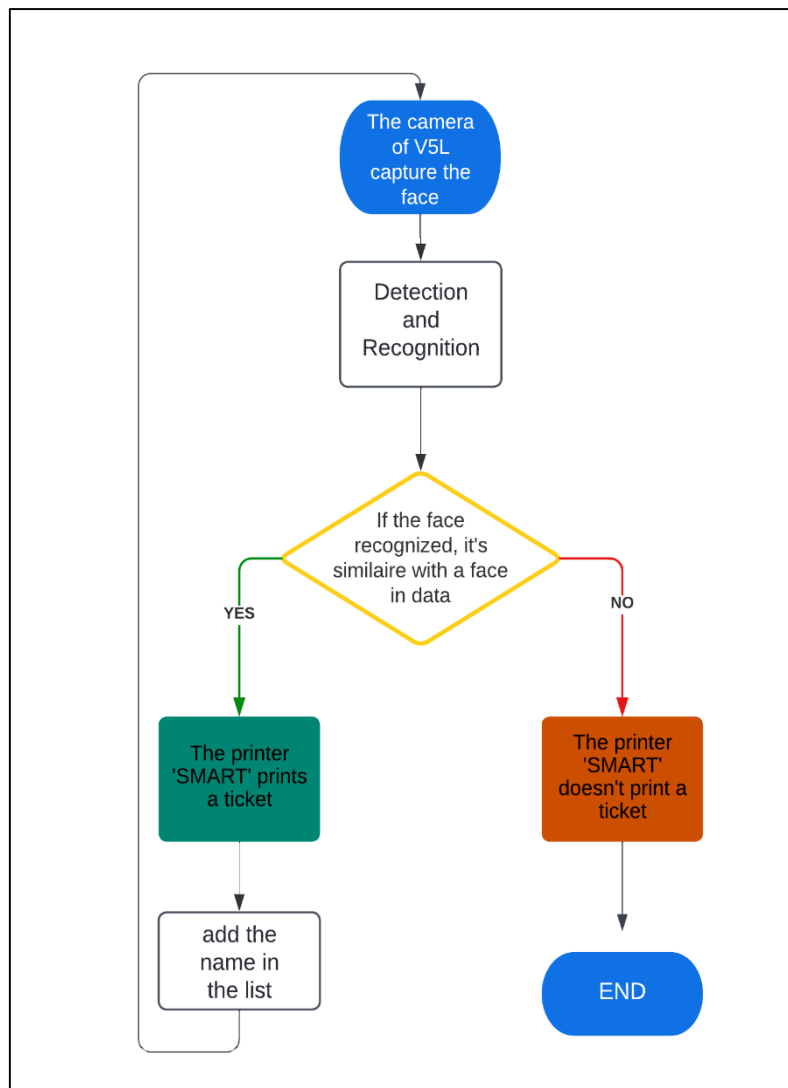


Figure 4.3: The flowchart of working the Feeding service system

4.5. Materials

This part elucidates the necessary materials and the characteristic of each component for the realization of the two systems.

4.5.1 Tripod turnstiles

The turnstiles are equipped with a sturdy tripod hub and encased in durable stainless steel. The electro-magnetic locks are designed to conserve power by remaining in standby mode until the turnstile is activated (Figure 4.4). In situations of emergencies or power loss, the arms of the tripod can be easily unlocked and dropped down. A visual cue to indicate whether access has been granted or denied. Users can easily and quickly authenticate using card and fingerprint due to the ergonomic design [33].



Figure 4.4: Tripod turnstiles

4.5.2 SpeedFace-V5L

The SpeedFace-V5L is a cutting-edge access control system that utilizes biometric technology, with the ability to recognize both facial and palm vein patterns. The SpeedFace-V5L Series is an advanced iteration of the SpeedFace-V5L Visible Light Facial Recognition Terminal, featuring intelligent facial recognition algorithms that are engineered for maximum efficiency, as well as the latest in computer vision technology with an embedded Deep Learning function. This device supports various verification methods, including facial and palm recognition fingerprint scanning, card swiping, and password entry, with rapid recognition and a large capacity, which enhances security performance across the board (Figure 4.5) [33].



Figure 4.5: SpeedFace-V5L

4.5.3 The airMax LiteBeam M5

The Ubiquiti AirMax LiteBeam M5-23 with parabolic antenna (Figure 4.6) is appropriate for creating a point-to-point connection the two system with the accommodation office.

It consists with a UV-resistant Cat 5e UTP network cable that is particularly built for outdoor use. In the case of an internal fault, oversees automatically managing the passing band Watchdog parabolic antenna. WEP, WPA, and WPA2 encryption offer the highest level of wireless security. Its service modes include router, NAT, client PPPoE, QoS, and server DHCP [33]



Figure 4.6: AirMax LiteBeam

4.5.4 The printer “SMART”

SMART is a thermic printer with a high-performance for the restaurant service system. It has a speed of printing a ticket in 0.9 seconds. Its type of interface Serial port + USB + network port it is connected to the unit (Figure 4.7).



Figure 4.7: The printer SMART

4.6. Work Environment

- Desktop module Oid.
- Processor Intel(R) Core(TM) i3-2120 CPU @ 3.30 GHZ.
- Installed memory (RAM) 4.00 GB.
- Windows 7
- Exploitation system 64 bit.

4.7. Installation of Equipment

The security system and the feeding system were installed separately. The feeding service system included backup servers and data storage devices. These systems were then linked using a parabolic antenna “AirMax LiteBeam”, which was installed to ensure a clear line of sight between the two systems. A desktop application called OnouAPP was installed on a computer to monitor and control both the security and restoration systems. The communication between the systems was established through the parabolic antenna, allowing seamless integration and management of the systems through the OnouAPP desktop application for data recovery (Figure 4.8).

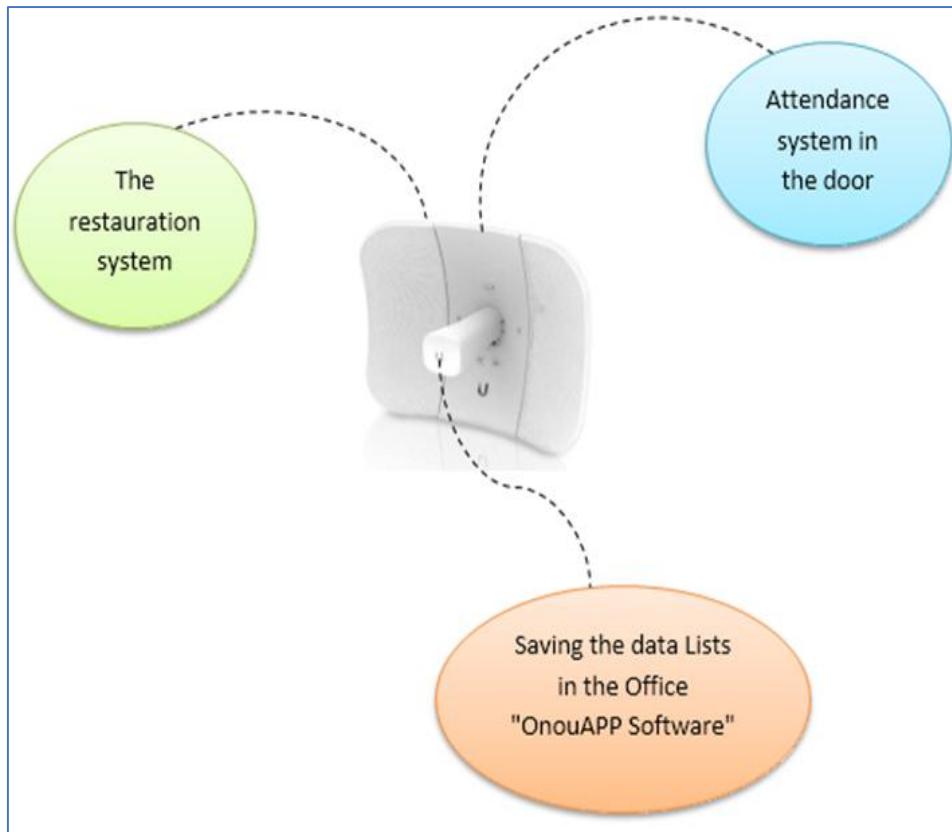


Figure 4.8: Linking the systems together

4.7.1 Installation of Security System

First of all, we started by Installing the tripod turnstile on the mounting plate and tightening the screws, and Applying screw-securing glue before using and putting gaskets on the screws to adjust the direction of the turnstile (Figure 4.9).

Next, we installed the SpeedFace-V5L and we tied threads related to the tripod turnstile.

Then we linked the systems with the parabolic antenna "AirMax LiteBeam" by network cable.



Figure 4.9: The installation of turnstiles with the V5L.

4.7.2 Installation of Feeding Service System

The installation of the feeding service system aims to address the problem of duplicated taking meals. This system incorporates various components, including a central unit, facial V5L recognition technology, a ticket printer, and a 220V power supply (Figure 4.10).

At the core of the system is the central unit, which serves as the central control hub. It manages and coordinates the entire process, ensuring smooth operation and counting the meals distributed. The central unit is equipped with the software “OnouAPP” that handles student data and the overall management of the system.

The facial V5L recognition technology is a key component that enhances the efficiency and security of the system. It utilizes advanced facial recognition algorithms to identify individual students as they approach the food distribution area.

To streamline the process, a ticket printer is incorporated into the system. This printer generates tickets for each student, containing relevant information such as the name, the date and time, and the number of the meal. These tickets serve as a means of authentication and help facilitate a smooth and organized distribution process.

The system ensures accurate meal distribution, reduces waste, improves the overall dining experience for students, and fights corruption.

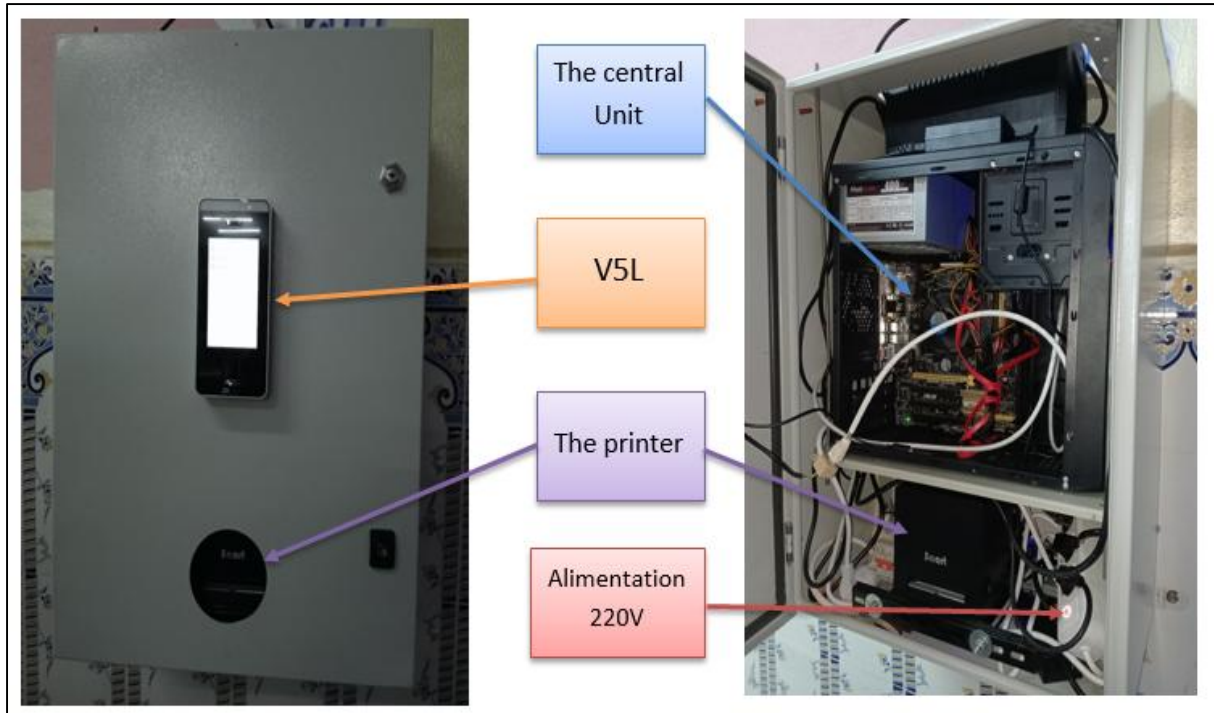


Figure 4.10: the installation of feeding service system.

4.7.2.1. The SDK Driver

The SDK is a Software Development Kit specifically designed for creating drivers. In the context of computer systems, a driver is a software component that allows the operating system to communicate and interact with hardware devices such as printers in our case. A driver SDK provides developers with the necessary tools, libraries, documentation, and resources to build, modify, and maintain drivers for various hardware devices. It simplifies the process of driver development by offering a standardized framework and programming interfaces that handle low-level hardware interactions (Figure 4.11). Ultimately, a driver SDK helps accelerate the development process and enhances the overall quality and compatibility of hardware drivers.

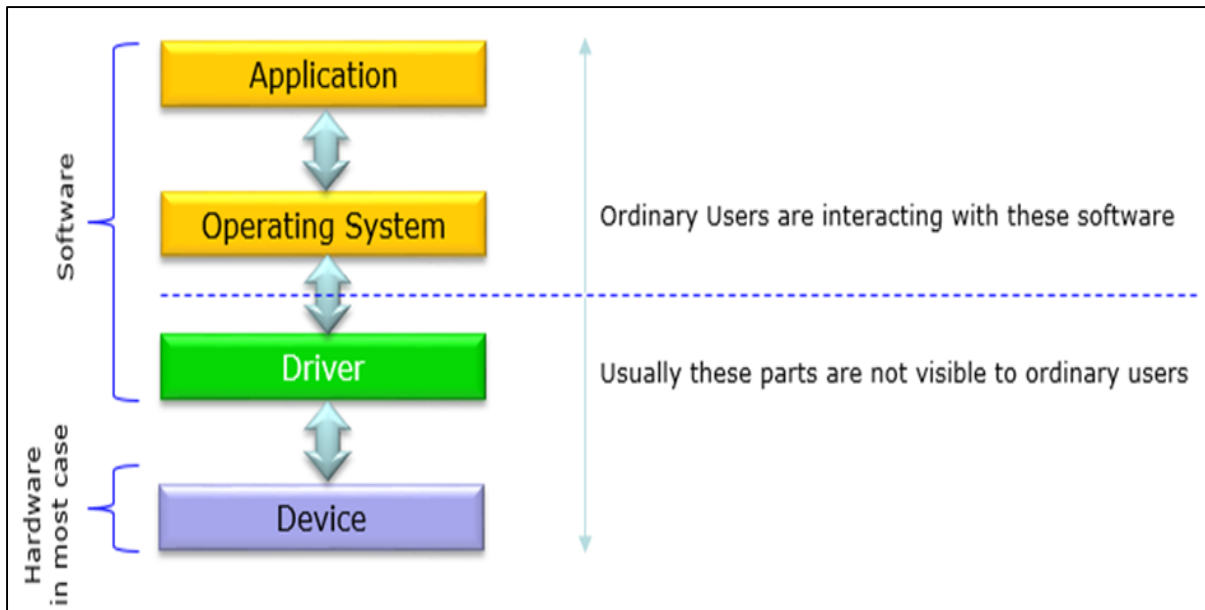


Figure 4.11: Diagram represents the role of the Drive

4.7.3 The installation of the Software OnouAPP

The software “OnouAPP” (Figure 4.12), we adopted in our system it was created by the informatics engineer Ismail who works in the MLS Company.

The functionality of the software is the same what I aspire to realise just it needs an addition which is creating the list of exits.

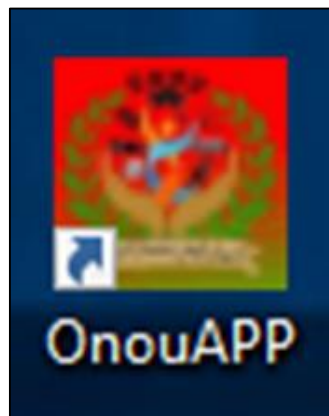


Figure 4.12 : The software

4.7.3.1. PostgreSQL

PostgreSQL is an advanced enterprise-grade open source relational database system. It is used as the primary database for web, mobile and analytics applications. PostgreSQL supports the most popular programming languages: Python, Java, C#, C/C+, Ruby, and

JavaScript [26]. PostgreSQL has many advanced features offered by other enterprise; the most needed one is database management systems between the software and the systems installed.

4.8. Functioning Procedures

4.8.1 User Registration

We collected student's faces by SpeedFace V5L following the steps below:

1. First, we click on User Mgt On the main menu (Figure 4.13).
2. We tap New User, and then we enter the User ID and Name.

Note:

- ✚ A name can take up to 34 characters.
- ✚ The user ID may contain 1-9 digits by default.

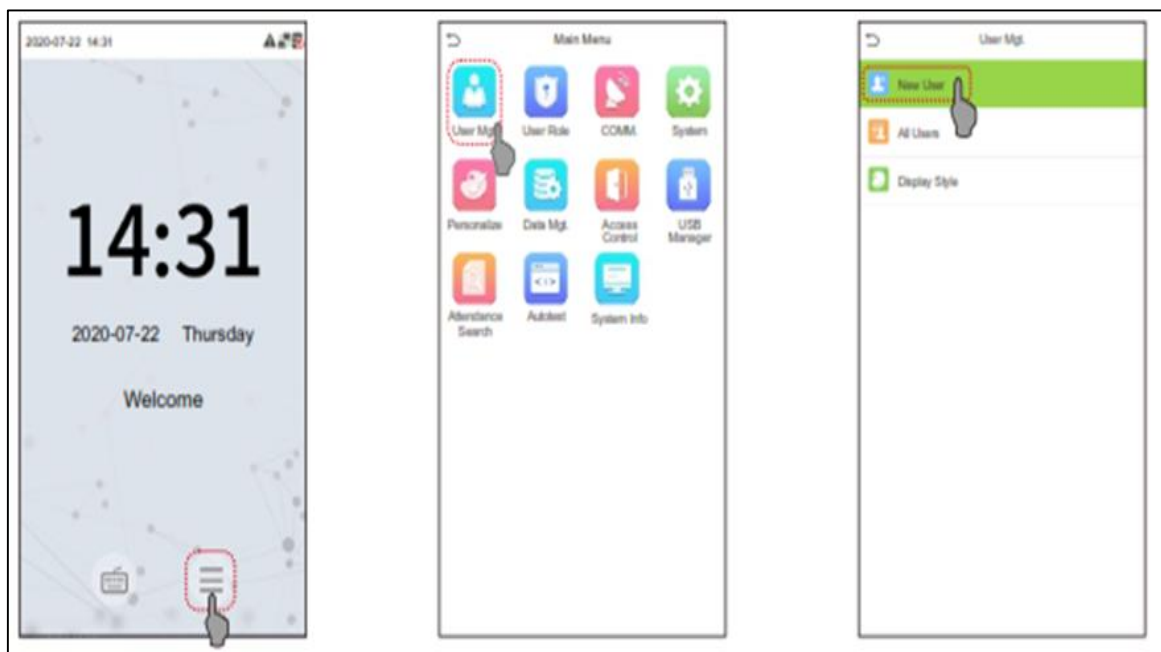


Figure 4.13: Steps of registration in the data by V5L

3. Entering the ID students names and takes a picture (Figure 4.14).

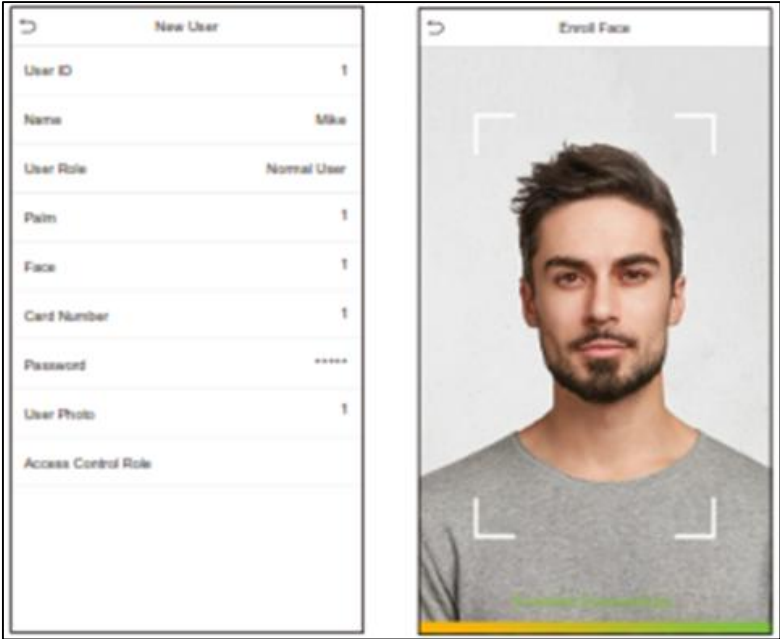


Figure 4.14: the final step of registration in the data by V5L.

- 4. Facial verification: In this verification mode, the device compares the collected facial images with all face data registered in the device. The following is the pop-up prompt of a successful comparison result (Figure 4.15).



Figure 4.15: A successful verification by V5L.

4.8.2 Communication Settings

Tapping COMM on the Main Menu to set the relevant parameters of Network, Serial Comm, PC connection, and Network Diagnosis.

Comm Key facilitates improving the security of data by setting the communication between the device and the desktop. Once the Comm Key is set, its connection password must be provided before the device gets connected to the desktop software.

Tapping PC Connection on the Comm. Settings interface to configure the communication settings. The identity number of the device ranges between 1 and 254.

If the communication method is RS232/RS485, it needs to input this device ID in the software communication interface (Figure 4.16).

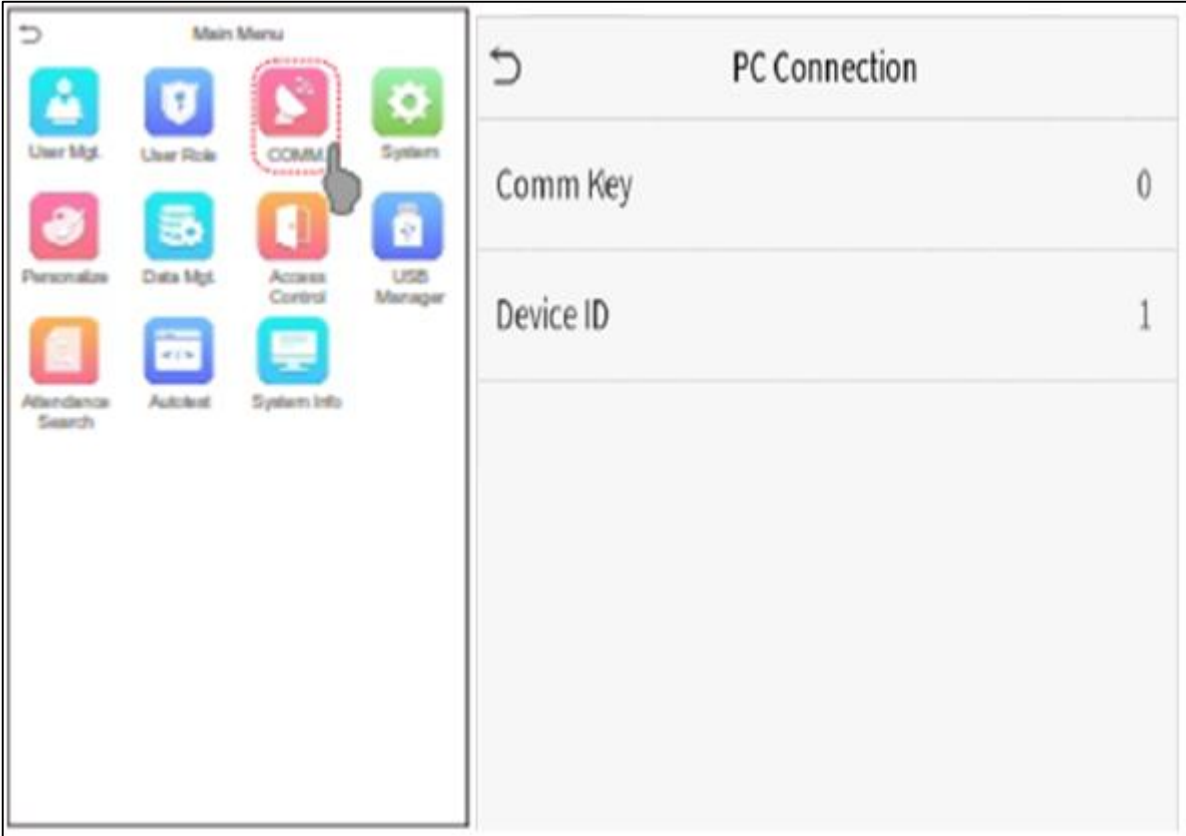


Figure 4.16: The communication method.

4.8.3 Access Control

The Access Control Role sets the door access privilege for each user. This includes the access group, verification mode and also facilitates to set the group access time-period.

- Tap Access Control Role > Access Group, to assign the registered users to different groups for better management. New users belong to Group 1 by default and can be reassigned to other groups. The device supports up to 99 Access Control groups (Figure 4.17).
- Tap Time Period, to select the time period to use.

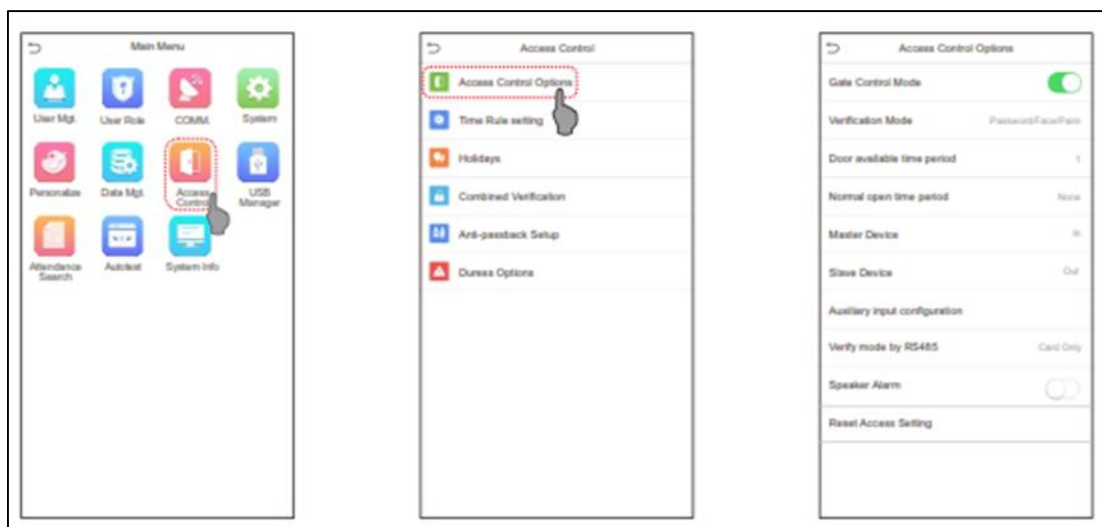


Figure 4.17: The access control method.

This option access to see the lists and control them from V5L on OnouAPP software.

4.8.4 Functioning procedures

After installing the software "OnouAPP" and PostgreSQL, then entering the IP address of the parabolic antenna "AirMax LiteBeam on the administration computer, connexion the software by login using the password (Figure 4.18).

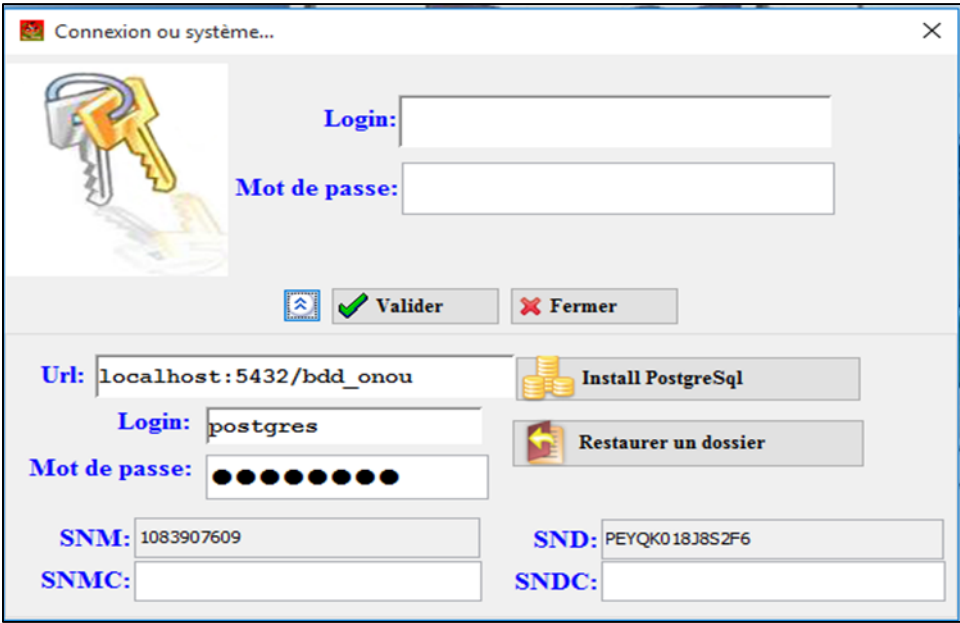


Figure 4.18: the principle interface of Software “OnouAPP”

Immediately after logging in, a window interface will appear (Figure 4 .19)

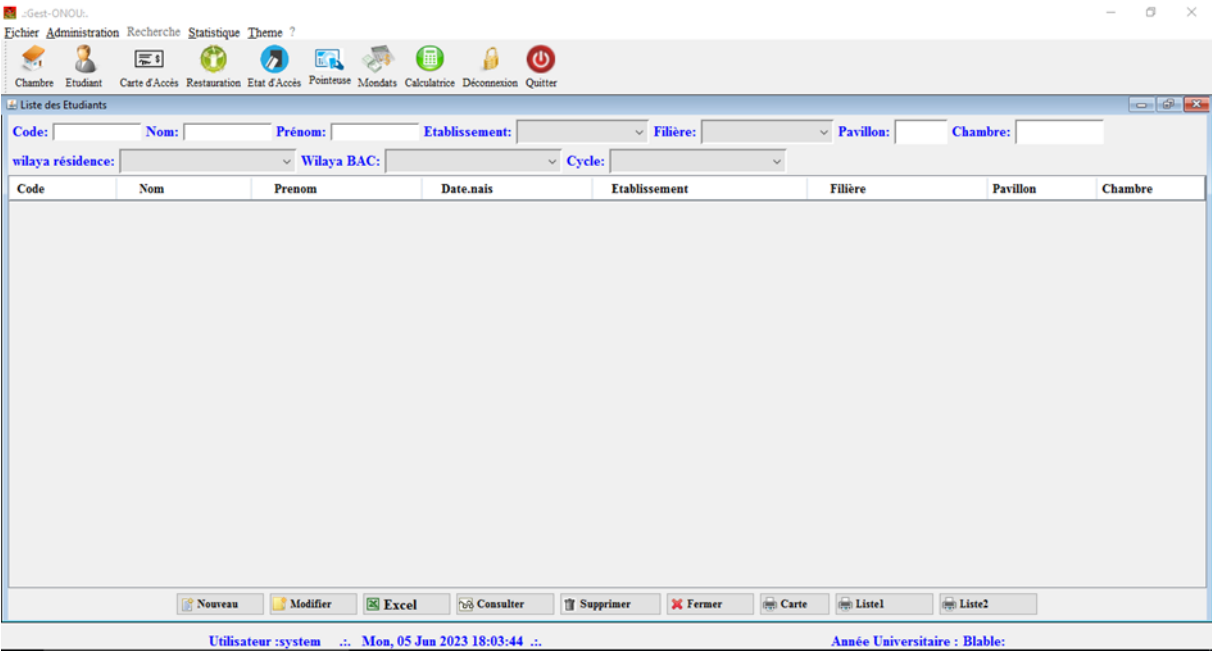


Figure 4.19: The first interface of OnouAPP

The window interface presents an intuitive layout with an organized design to manage student records efficiently. It offers a wide range of functions and features that facilitate tasks related to student management.

CHAPTER 04 REPORT OF THE INTERNSHIP AT THE MLS COMPANY

- The top bar contains various options such as "Student List," "Meal Records," "Statistics" and more.
- The down bar also contains various options such as "New Student," "Excel Lists," and some settings options.
- The "New Student" function allows the administrator to enter the details of the student. By clicking on this option, a form box appears, prompting the user to input information such as the student's name, age, birth date, registration number, room, and other relevant data required for record-keeping purposes (Figure 4-20) the picture of student selected from the users list registered by V5L.
- Modifying student information is easy through the interface. By selecting a specific student from the student list, the administrator can access and edit their details. Moreover, the interface allows administrators to delete student records when necessary, ensuring the database remains up-to-date and accurate.

The screenshot shows a web-based form titled "Fiche Etudiant" for managing student information. The form is organized into several sections:

- Etudiant:** Fields for Code, Nom, Prénom, Date naissance, and Lieu de naissance. Includes a "Carte d'accès" section with a "Code-barre" field and a "Rétablir" button.
- Info Général:** Fields for Nationalité, adresse, Tél, Etablissement, Sexe, wilaya de naissance, wilaya, and Filière.
- Info BAC:** Fields for N° BAC, Wilaya BAC, Année BAC, and Mention BAC.
- Affectation:** Fields for Date affectation, Cycle, Niveau, Position, Motif, Pavillon, Chambre, and Date sortie. Includes buttons for "Renouveler" and "Libérer".
- Mondat:** Fields for N°, Montant, and Date.

At the bottom of the form, there are buttons for "Valider", "Nouveau", "Carte", and "Fermer".

Figure 4.20: The window of managing information students

The "Student List" feature enables the user to view a comprehensive list of all the students who entered the campus by the V5L.

CHAPTER 04 REPORT OF THE INTERNSHIP AT THE MLS COMPANY

Additionally, the window provides a dedicated section for "Meal Records." This section displays a list of students who have taken meals, along with details such as the date, and meal type. It serves as a convenient reference for tracking students and data analysis for improving meals.

4.8.5 RESULTS

The installed system achieved good consistency, everything is going according to what was planned by the company, and because I pledged not to release any information related to university residence, I cannot present the results obtained in the lists because they contain information related to resident students. I will be satisfied with what I have provided so far.

4.9. CONCLUSION

During my internship with the company, I gained valuable insights and hands-on experience in various aspects of entrepreneurialism. I also learned about the importance of adapting to evolving technologies. The company introduced me to innovative software and tools used in the management of university services. I acquired proficiency in utilizing these technologies to streamline processes, automate tasks, and improve overall efficiency.

Overall, my internship with the company provided me with a comprehensive understanding of the university services sector and the practical skills required effectively managing and enhancing such services. I am grateful for the opportunity to learn from experienced professionals and contribute to the continuous improvement of the residence management system.

GENERAL CONCLUSION

General Conclusion

This master thesis seeks to provide a thorough grasp of the design and implementation of an attendance system in university housing, based on face detection and recognition by computer vision, for improving security service and help manage feeding services better while lowering randomness and food waste.

We reviewed the existing face detection and recognition algorithms and emphasized the value of deep learning algorithms and how effective they are at resolving these issues.

We discussed the system's architecture and operations. We also highlighted the methods used to guarantee the effectiveness, efficiency, and general success of the system.

In this research, we presented the procedures taken to write and program the code, and the creation of a graphical user interface (GUI) to facilitate system management and improve user experience.

However, the system faces obstacles and challenges. The first obstacle is that the preparation of dinner begins at 4 pm, and the number of students presented at that time is not final. This is because there are still students who will enter the campus and others who will leave, so it is approximate. To solve this issue, two proposals have been suggested. The first proposal was the pointing pendulum inside the residence, where students register themselves in the morning, confirming attendance for lunch, dinner, or both. However, we saw that it is not practical, as it is not suitable for students who are not in the residence all day and come after 4 pm. The second proposal is the most appropriate and is represented by a mobile application that will be accessible to all students and in which several features and services can be added to students that contribute to improving living conditions on campus in the future. Among these features and services, we can find:

- Push notifications: The application can provide push notifications, allowing students to receive important updates and reminders, Events etc.
- Access to amenities: The application can provide information about campus amenities such as dorms, dining options, the menu and community and study spaces
- Activities, sport competition...
- Social media platforms integration: The application can be integrated with social media platforms, allowing students to connect with each other and share their experiences.

GENERAL CONCLUSION

- Yearly room renewal.

As mentioned earlier, we worked with Eurl Reseaux MLS, which is an Algerian company that excels in providing a wide range of services, including installation of telephonic networks, surveillance camera and informatics. We have identified a number of important areas where we can improve our services as part of our ongoing efforts to increase system efficiency. By working closely with Eurl Reseaux MLS, they not only provide the necessary equipment for installations but also take use of their wealth of knowledge to offer thorough support throughout the whole installation process. To further optimize the system state, we can introduce the following additional services:

- System Integration: integrating telephonic networks and informatics with existing systems for smooth communication and data flow.
- Security Solutions: implementing robust security measures, including firewalls and encryption protocols, to protect network and data.
- Maintenance and Support: providing regular maintenance, troubleshooting, and timely upgrades to ensure smooth system operation.

Overall, this research paves the way for future developments in the University services sector.

In conclusion, this master's thesis has allowed us to discover the revolution of AI and take an experience in Entrepreneurship in terms of technology and digitization.

REFERENCES

- [1] RecFaces. (2023). Understanding facial recognition algorithms. *RecFaces*. <https://recfaces.com/articles/facial-recognition-algorithms>
- [2] *Facial recognition: top 7 trends (tech, vendors, use cases)*. (2023, June 12). Thales Group. <https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/biometrics/facial-recognition>
- [3] Gavin. (2023). 5 most common uses of facial recognition. *NEC*. <https://www.nec.co.nz/market-leadership/publications-media/5-most-common-uses-of-facial-recognition/>
- [4] SuperAnnotate AI Inc. (n.d.). *Essential guide to face recognition: How it works | SuperAnnotate*. <https://www.superannotate.com/blog/guide-to-face-recognition>
- [5] Atenco, J. C., Moreno, J. C., & Ramirez-Cortes, J. M. (2023). Audiovisual Biometric Network with Deep Feature Fusion for Identification and Text Prompted Verification. *Algorithms*, 16(2), 66. <https://doi.org/10.3390/a16020066>
- [6] Waleed Ahmed Ali Dahea, Doctor of Philosophy Researcher at Swami Ramanand Teerth Marathwada University https://www.researchgate.net/profile/Waleed-Dahea/publication/328769278_Multimodal_biometric_system_A_review/links/5be46fb4a6fdcc3a8dc77f44/Multimodal-biometric-system-A-review.pdf
- [7] TALEB Imene, Thesis of doctorate " *Le système biométrique : détection et reconnaissance de visage* ", USTO 2018/2019
- [8] Atenco, J. C., Moreno, J. C., & Ramirez-Cortes, J. M. (2023c). Audiovisual Biometric Network with Deep Feature Fusion for Identification and Text Prompted Verification. *Algorithms*, 16(2), 66. <https://doi.org/10.3390/a16020066>
- [9] Master's Thesis PRESENTED BY BOUABDALLAH Aimene & AMRANE Mohammed Ridha " *Face Recognition based attendance system* ", USDB 2021/2022
- [10] Singh, A. (2023). Feature Engineering for Images: A Valuable Introduction to the HOG Feature Descriptor. *Analytics Vidhya*.

REFERENCES

<https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/>

[11] Singh, A., Bhatt, S., Nayak, V. *et al.* Automation of surveillance systems using deep learning and facial recognition. *Int J Syst Assur Eng Manag* **14** (Suppl 1), 236–245 (2023). <https://doi.org/10.1007/s13198-022-01844-6>

[12] “Implementing Face Recognition Using Deep Learning and Support Vector Machines.” *Www.codemag.com*, www.codemag.com/Article/2205081/Implementing-Face-Recognition-Using-Deep-Learning-and-Support-Vector-Machines.

[13] Rawat, Ayush Singh. “Top 10 Deep Learning Algorithms You Should Know | Analytics Steps.” *Www.analyticssteps.com*, www.analyticssteps.com/blogs/top-deep-learning-algorithms. Accessed 2 July 2023.

[14] IBM. “What Is Deep Learning? | IBM.” *Www.ibm.com*, 2023, www.ibm.com/topics/deep-learning.

[15] Burns, E. (2021, March). *What is deep learning and how does it work?* SearchEnterpriseAI. <https://www.techtarget.com/searchenterpriseai/definition/deep-learning-deep-neural-network>[16] OpenCV. “OpenCV: Cascade Classifier.” *Docs.opencv.org*, docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html.

[17] Behera, Girija Shankar. “Face Detection with Haar Cascade.” *Medium*, 29 Dec. 2020, towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08.

[18] Wikipedia Contributors. “Haar-like Feature.” *Wikipedia*, Wikimedia Foundation, 20 May 2019, en.wikipedia.org/wiki/Haar-like_feature.

[19] Architectures, M. (2013). *Deep Learning*. O’Reilly | Safari. <https://www.oreilly.com/library/view/deep-learning/9781491924570/ch04.html>

[20] “FaceNet - Using Facial Recognition System.” *GeeksforGeeks*, 2 May 2020, www.geeksforgeeks.org/facenet-using-facial-recognition-system/.

REFERENCES

- [21] Schroff, Florian, et al. "FaceNet: A Unified Embedding for Face Recognition and Clustering." *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, arxiv.org/abs/1503.03832, <https://doi.org/10.1109/cvpr.2015.7298682>.
- [22] OpenCV. "About." *OpenCV*, 2018, opencv.org/about/.
- [23] "Face Recognition with OpenCV, Python, and Deep Learning." *PyImageSearch*, 18 June 2018, pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/.
- [24] Python Software Foundation. "Datetime — Basic Date and Time Types — Python 3.7.2 Documentation." *Python.org*, 2002, docs.python.org/3/library/datetime.html.
- [25] Pandas. "Python Data Analysis Library — Pandas: Python Data Analysis Library." *Pydata.org*, 2018, pandas.pydata.org/.
- [26] *What is PostgreSQL*. (n.d.). <https://www.postgresqltutorial.com/postgresql-getting-started/what-is-postgresql/>
- [27] PhD, D. R. (2023, May 4). *The Future of Security: Using AI to Enhance Monitoring and Recognition*. Coruzant Technologies. <https://coruzant.com/ai/the-future-of-security-using-ai-to-enhance-monitoring-and-recognition/>
- [28] Saada, Hana. *Algiers: Student Slaughtered in His Dormitory at University Campus in Ben Aknoun- Video | DZ Breaking*. 11 Feb. 2019, www.dzbreaking.com/2019/02/11/algiers-student-slaughtered-dormitory-university-campus-ben-aknoun/. Accessed 2 July 2023.
- [29] [https://www.ennaharonline.com/Corruption, mismanagement, and contractors in senior positions in the university services sector](https://www.ennaharonline.com/Corruption,mismanagement,andcontractorsintheseniorpositionsintheuniversityservicessector)
- [30] "Introduction to Tkinter." *GeeksforGeeks*, 15 May 2020, www.geeksforgeeks.org/introduction-to-tkinter/.
- [31] *What is a Webcam? - Definition from Techopedia*. (2019). [Techopedia.com. https://www.techopedia.com/definition/5333/webcam](https://www.techopedia.com/definition/5333/webcam).

REFERENCES

[32] "Dahua Technology - Leading Video Surveillance Solution Provider with CCTV Produc." *Www.dahuasecurity.com, www.dahuasecurity.com/fr/*. Accessed 2 July 2023.

[33] "ZKTeco | Home Page." *Zkteco.com, 2022, www.zkteco.com/en/*.