

DEVELOPMENT OF FACE RECOGNITION DOOR LOCK SYSTEM BY USING ESP32-CAM

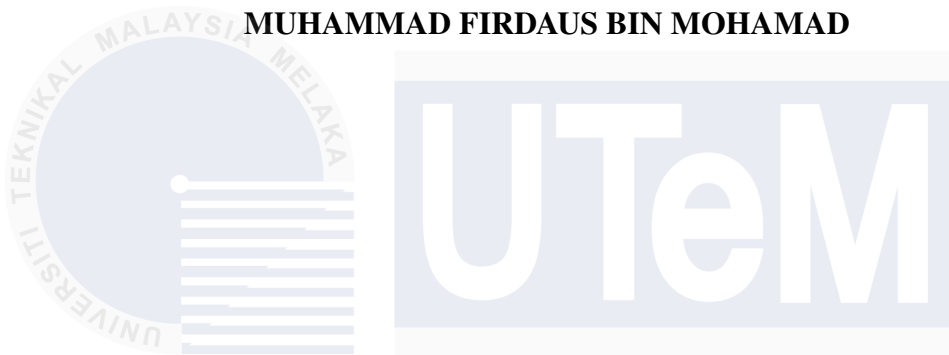


MUHAMMAD FIRDAUS BIN MOHAMAD

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DEVELOPMENT OF FACE RECOGNITION DOOR LOCK SYSTEM BY USING ESP32-CAM

MUHAMMAD FIRDAUS BIN MOHAMAD



**This report is submitted in partial fulfilment of the requirements
for the degree of Bachelor of Electronics Engineering Technology (Industrial
Electronics) with Honours**

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Faculty of Electronics and Computer Engineering and Technology

Universiti Teknikal Malaysia Melaka

2025

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Development of Face Recognition Door Lock System by using ESP32-CAM

Sesi Pengajian : 2024

Saya **MUHAMMAD FIRDAUS BIN MOHAMAD** mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hak milik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (/)

☐

SULIT*

(Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia sebagaimana yang termaktub dalam AKTA RAHSIA RASMI 1972)

☐

TERHAD*

(Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

☒

TIDAK
TERHAD

Disahkan oleh:

.....
(TANDATANGAN PENULIS)

Alamat Tetap:

1

.....
(COP DAN TANDATANGAN PENYELIA)

TS. DR. SITI HALMA BINTI JOHARI

PENSYARAH KANAN

FAKULTI TEKNOLOGI DAN KEJURUTERAAN ELEKTRONIK DAN KOMPUTER (FTKEK)

UNIVERSITI TEKNIKAL MALAYSIA MELAKA



Tarikh: 8th February, 2025

Tarikh: 8th February, 2025

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this report entitled “Development of Face Recognition Door Lock System by using ESP32-CAM” is the result of my own research except for quotes as cited in the references.



Signature :

Student Name : MUHAMMAD FIRDAUS BIN MOHAMAD

Date : 8th February, 2025

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

APPROVAL

I declare that I have read this thesis and in my opinion, this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronics Engineering Technology (Industrial Electronics) with Honours.

Signature :

Supervisor Name : TS. DR. SITI HALMA BINTI JOHARI

Date : 8th February, 2025

Signature :

Co-Supervisor Name (if any) : PUAN RAEIHAH BINTI MOHD ZAIN

Date : 8th February 2025

DEDICATION

Dedicating this bachelor's degree project to my Creator, Allah s.w.t the Almighty, my steadfast support, the wellspring of my inspiration, wisdom, knowledge, and understanding. He the one who has been my source of resilience, empowering me to complete this project during my academic journey. Additionally, I extend this dedication to my parents and family who have provided unwavering support, encouraging me to persist in my endeavors. As well as to my supervisor and co supervisor Ts. Dr. Siti Halma binti Johari and Puan Raeihah binti Mohd Zain, whose guidance and support have been invaluable throughout its development. Their expertise, encouragement, and dedication to fostering a deep understanding of the subject matter have greatly contributed to the success of this endeavor. I am grateful to all my friends and lecturers for their consistent encouragement, guidance, and advice, contributing significantly to the success of this project

ABSTRACT

Door security technology has progressed greatly, combining innovative ways and technologies to protect the safety and security of homes, workplaces, and other locations. The project presents a Face Recognition Door Lock System that uses the ESP32-Cam. The main focus of the system is on safety, security, and cost-effectiveness by includes Internet of Things (IoT) technology. The system of the ESP32-Cam module for instant face recognition and connects it to a solenoid valve for managing door access. The camera sends data to an IoT platform where face recognition algorithms detect users, allowing the solenoid valve to unlock the door upon successful verification. By including a user interface accessible through internet-connected devices, the system ensures easy usability and administration. The developed system demonstrates reliable face recognition capabilities, allowing only an authorized individuals to enter the protected area, while the IoT integration enhances remote monitoring and control. An evaluation shows that using the ESP32-CAM and solenoid valves provides a cost-effective alternative compared to traditional security systems. This brings improvements to safety and convenience, using face recognition technology to improvise security measures and user experience. To summarize, the Face Recognition Door Lock System effectively meets modern security requirements, offering a robust, affordable, and friendly-use solution for security access control with potential for wider application in various security-settings.

ABSTRAK

Teknologi keselamatan pintu telah berkembang dengan pesat, menggabungkan cara dan teknologi inovatif untuk melindungi keselamatan dan keselamatan rumah, tempat kerja dan lokasi lain. Projek ini mempersembahkan Sistem Kunci Pintu Pengecaman Wajah yang menggunakan ESP32-Cam. Sistem ini menggunakan modul ESP32-Cam untuk pengecaman muka segera dan menyambungkannya kepada mekanisme injap solenoid untuk menguruskan akses pintu. Kamera menghantar data ke platform IoT di mana algoritma pengecaman muka mengesahkan pengguna, membolehkan injap solenoid membuka kunci pintu apabila pengesahan berjaya. Dengan memasukkan antara muka pengguna yang boleh diakses melalui peranti yang disambungkan ke Internet, sistem memastikan kebolegunaan dan pentadbiran yang mudah. Sistem yang dibangunkan menunjukkan keupayaan pengecaman muka yang boleh dipercayai, membenarkan hanya individu yang dibenarkan memasuki kawasan yang dilindungi, manakala penyepaduan IoT meningkatkan pemantauan dan kawalan jauh. Penilaian kos menunjukkan bahawa menggunakan ESP32-Cam dan injap solenoid menyediakan alternatif yang menjimatkan kos berbanding sistem keselamatan tradisional. Pelaksanaan ini membawa peningkatan ketara dalam keselamatan dan kemudahan, menggunakan teknologi pengecaman muka untuk meningkatkan langkah keselamatan dan pengalaman pengguna. Secara ringkasnya, Sistem Kunci Pintu Pengecaman Wajah memenuhi keperluan keselamatan moden dengan berkesan, menawarkan penyelesaian yang teguh, mampu milik dan mesra pengguna untuk kawalan akses selamat dengan potensi untuk aplikasi yang lebih luas dalam pelbagai tetapan sensitif keselamatan.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to Puan Siti Halma Binti Johari for her exceptional guidance, invaluable support, and unwavering commitment throughout the journey of my final year project. Puan Siti Halma's expertise, encouragement, and dedication have played a pivotal role in shaping the success of my project.

Furthermore, I would like to extend my appreciation to Puan Siti Halma for fostering an environment of collaboration and learning, where I felt empowered to explore innovative ideas and solutions for my project.

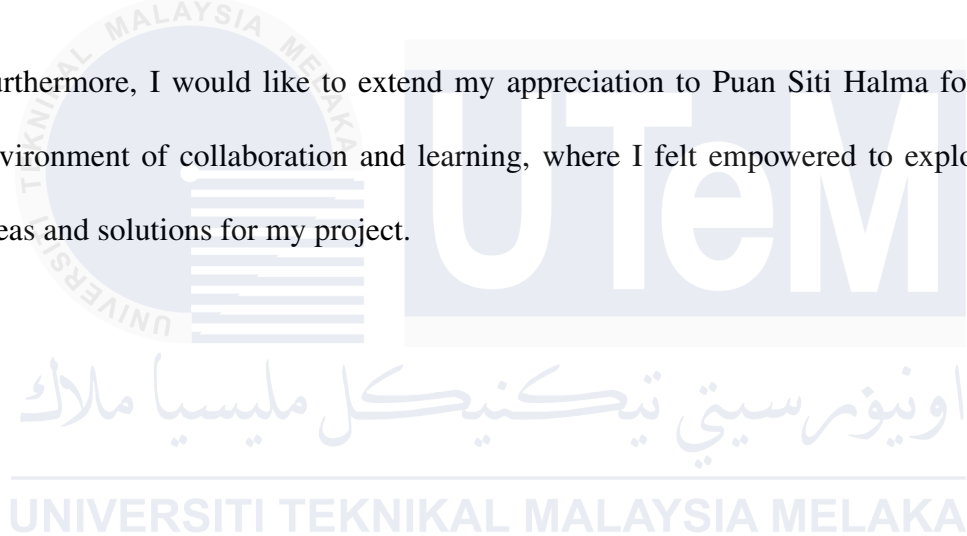


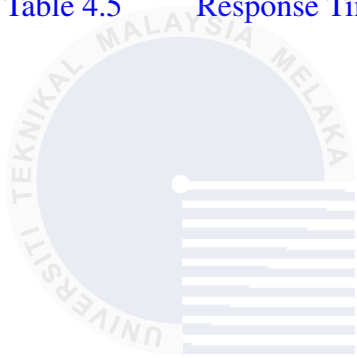
TABLE OF CONTENTS

	PAGE
DECLARATION	
APPROVAL	
DEDICATION	
ABSTRACT	i
ABSTRAK	ii
ACKNOWLEDGMENTS	iii
TABLE OF CONTENTS	iv
LIST OF TABLES	vi
LIST OF FIGURES	vii
CHAPTER 1 INTRODUCTION	1
1.1 Background	1
1.2 Problem Statement	2
1.3 Project Objectives	2
1.4 Scope of Work	3
CHAPTER 2 LITERATURE REVIEW	4
2.1 Introduction	4
2.2 Past Related Project Research	5
2.2.1 Development of Real-Time Face Recognition for Smart Door Lock Security System using Haar Cascade and OpenCV LBPH Face Recognizer	5
2.2.2 Face Recognition Door Lock System Using Raspberry Pi	7
2.2.3 Biometric Authentication Based Automated, Secure, and Smart IOT Door Lock System	9
2.2.4 A Smart Door Lock Security System using Internet of Things	11
2.2.5 Automatic Door Locking System in Households using IoT	13
2.2.6 Design of Home Security System Using Face Recognition with Convolutional Neural Network Method	15
2.2.7 A Low-Cost Embedded Facial Recognition System for Door Access Control using Deep Learning	17
2.2.8 Face Recognition Door Lock System Using Raspberry Pi	19
2.2.9 IoT and Face Recognition based Automated Door Lock System	21
2.2.10 Design and Development of IOT based Smart Door Lock System	23

2.3	Summary	27
CHAPTER 3	METHODOLOGY	28
3.1	Introduction	28
3.2	Description of Methodology	29
3.3	Block diagram of project	29
3.4	Flowchart	30
3.5	Hardware	33
3.5.1	ESP32-CAM	33
3.5.2	Solenoid Valve 12V	35
3.5.3	Power Supply Module 12V/5V/3V	37
3.5.4	Buck Converter	39
3.5.5	Relay Module 5V	41
3.6	Software	43
3.6.1	Arduino IDE	43
3.6.2	WebServer	45
3.7	Gantt Chart	47
3.8	Summary	49
CHAPTER 4	RESULTS AND DISCUSSION	50
4.1	Expected Result	50
4.1.1	Reliable Face Recognition	50
4.1.2	Time Performance	51
4.1.3	Robust Security	51
4.1.4	Integration of IoT	52
4.2	Data Analysis	53
4.2.1	Introduction	53
4.2.2	Data Description	53
4.2.3	Performance Metrics	54
4.2.3.1	Recognition Accuracy	54
4.2.3.2	Error Rates	59
4.2.3.3	Response Times	60
4.3	Schematic Diagram	63
4.4	Hardware Implementation	65
4.5	Software Implementation	69
CHAPTER 5	CONCLUSION AND RECOMMENDATIONS	91
5.1	Conclusion	91
5.2	Future Works	93
5.3	Project Commercialization	94
5.3.1	Examination of the Market	94
5.3.2	Product Development	95
REFERENCES		96

LIST OF TABLES

TABLE	TITLE	PAGE
Table 2.1	Summary of the previously proposed techniques	25
Table 4.1	Recognition Accuracy Table for distance in 50cm.	54
Table 4.2	Recognition Accuracy Table for distance in 75cm.	56
Table 4.3	Recognition Accuracy Table for distance in 100cm.	57
Table 4.4	Error Rates Table.	59
Table 4.5	Response Times Table.	60



اونيورسيتي تيكنيكل مليسيا ملاك

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 2.1	A Hardware Diagram using Haar Cascade algorithm[1].	5
Figure 2.2	Figure shows a Block Diagram using Raspberry Pi [2].	7
Figure 2.3	A Flowchart of Proposed System[3].	9
Figure 2.4	A Block Diagram of the Propose System [4].	11
Figure 2.5	A Flow chart using IoT [5].	13
Figure 2.6	A Block Diagram with Neural Network Method [6].	15
Figure 2.7	Schematic representation of the components of the smart lock system[7].	17
Figure 2.8	A Block Diagram by using Raspberry Pi [8].	19
Figure 2.9	A Flowchart of existing system.[9]	21
Figure 2.10	A Flowchart of proposed model followed for smart door lock.[10]	23
Figure 3.1	A block diagram.	30
Figure 3.2	A Flowchart.	31
Figure 3.3	ESP32-CAM	33
Figure 3.4	Solenoid Valve 12V	35
Figure 3.5	Power Supply Module	37
Figure 3.6	Buck Converter	39
Figure 3.7	Relay Module 5V	41
Figure 3.8	Arduino IDE	43
Figure 3.9	WebServer	45
Figure 3.10	Gantt chart PSM I 2024	48
Figure 4.1	Figure shows an example image of data users collected.	54
Figure 4.2	Figure shows a Recognition Accuracy in 50cm Column Chart.	55
Figure 4.3	Figure shows a Recognition Accuracy in 75cm Column Chart.	56
Figure 4.4	Figure shows a Recognition Accuracy in 100cm Column Chart.	58
Figure 4.5	Figure shows Response Times for 50,75 and 100cm.	61
Figure 4.6	Figure shows a schematic diagram for Face Recognition Door Lock.	63
Figure 4.7	Figure of Front View.	65
Figure 4.8	Figure of Back View.	66
Figure 4.9	Figure of Inside View.	67
Figure 4.10	Figure of Part 1 Codings.	69
Figure 4.11	Figure of Part 2 Codings.	71
Figure 4.12	Figure of Part 3 Codings.	73
Figure 4.13	Figure of Part 4 Codings.	75
Figure 4.14	Figure of Part 5 Codings.	77
Figure 4.15	Figure of Part 6 Codings.	78
Figure 4.16	Figure of Part 7 Codings.	80
Figure 4.17	Figure of Part 8 Codings.	82
Figure 4.18	Figure of Part 9 Codings.	83

Figure 4.19	Figure of Part 10 Codings.	85
Figure 4.20	Figure of Part 11 Codings.	87
Figure 4.21	Figure of Part 12 Codings.	89



CHAPTER 1

INTRODUCTION

This chapter describes the project background, problem statement, objectives, and scope of work in developing of face recognition door lock system by using ESP32-CAM.

1.1 Background

The development of a face recognition door lock system using the ESP32-CAM leverages advanced biometric technology to enhance security and convenience. The ESP32-CAM, with its integrated camera and connectivity features, enables efficient image capture and processing for face recognition, utilizing state-of-the-art algorithms such as Convolutional Neural Networks (CNNs). This innovative approach aligns with Sustainable Development Goals by promoting technological advancement (Goal 9), contributing to safer communities (Goal 11), and supporting the rule of law and justice (Goal 16). By integrating modern IoT and AI technologies, the project aims to provide a secure, sustainable alternative to traditional access control systems.

1.2 Problem Statement

The current issue in the university hostel is safety and privacy for both lecturers and students. These days, there are many who are constantly forgetting about their personal security, as seen by misplacing or losing their keys. The critical need for improved door security systems stems from the vulnerabilities of traditional lock-and-key mechanisms to modern threats such as lock picking, unauthorized key duplication, and physical force. Despite technological advancements, current systems struggle with issues like technological failures, user convenience versus security, integration, privacy concerns, and high costs. Therefore, there is an urgent demand for a door security solution that offers robust protection, reliability, user-friendliness, seamless integration, privacy safeguards, and affordability to effectively enhance the security of residential, commercial, and industrial properties.

1.3 Project Objectives

The main aim of this project is to propose Face Recognition Door Lock System by using ESP32-CAM. Specifically, the objectives are as follows:

1. To develop face recognition door lock system.
2. To integrate IoT into the face recognition door lock system
3. To analyse the face recognition system performance.

1.4 Scope of Work

The project scope of developing Face Recognition Door Lock System by using ESP32-CAM is as follows:

1. Connect necessary components, such as a solenoid lock, relay module, and power supply.
2. The project specific for student campus and office.
3. Train the face recognition model using the collected data.
4. Implement the algorithm to recognize faces captured by the camera.
5. Develop a user-friendly interface (web or mobile app) for managing access.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 2

LITERATURE REVIEW

This chapter provides a broad overview of the previous project related to the topic in this report. Besides, the relevant literature is critically discussed and presented later in this chapter.

2.1 Introduction

In this paper, the design is to proposed the use of ESP32-Cam for detect face use stream camera with Internet of Things for Face Recognition Door Lock System. Safety, security and cost are the foundations for consumers in this modern life. It can be seen in terms of the type of goods used throughout life. As a result of this research, face detect that can be security, safety and cost savings should be developed. Internet of Things are the concept of connecting devices with open or close to the internet. This modern days, have a lot of type of technology that is so advance that we use our own body such as face, eye or fingerprint to unlock the security such as phone. Phone these day use face to unlock the phone and we creating this project for advance security. However, this scale includes user interfaces with connected internet. By using the solenoid valve, the solenoid will open or close according to user interfaces. The solenoid should be able to detect faces and the door will open.

2.2 Past Related Project Research

2.2.1 Development of Real-Time Face Recognition for Smart Door Lock Security System using Haar Cascade and OpenCV LBPH Face Recognizer

One technology that is frequently utilized in security systems is face recognition. Facial recognition technology in door security systems allows the door to be opened by just identifying the owner's face. The goal of this project is to use OpenCV LBPH Face Recognizer and Haar Cascade to create a real-time facial recognition system for smart locking doors [1]. In Figure 2.1, this project aims to create a security system that restricts who is able to enter a room.

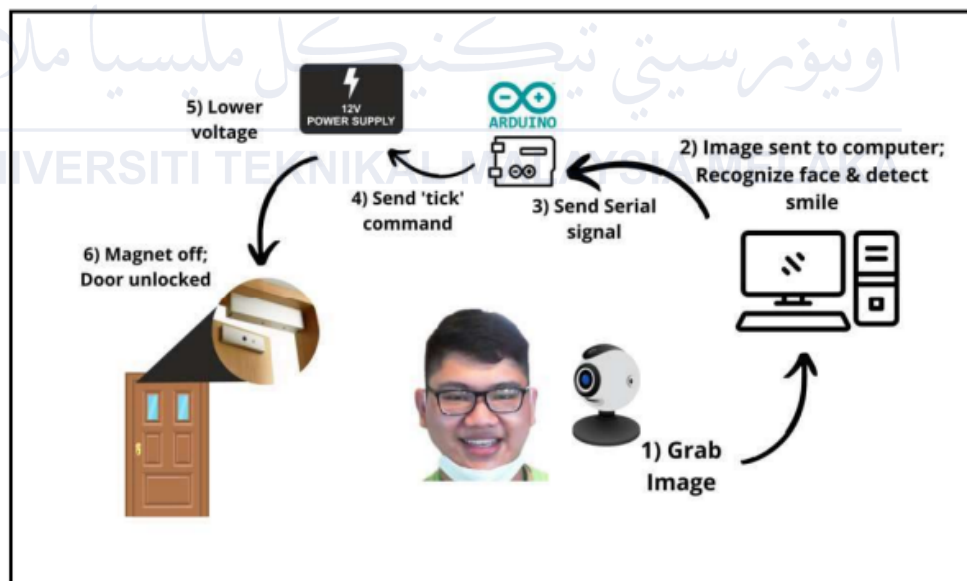


Figure 2.1: A Hardware Diagram using Haar Cascade algorithm[1].

Faces in images are discovered using the Haar Cascade technique, and those faces are then recognized using the OpenCV LBPH Face Recognizer. The OpenCV library and the Python programming language were used in the development of this system. According to the test findings, this system can identify faces with an accuracy of 62.7 using our dataset. It may be made even better by including additional datasets and training the recognizer model with deep learning methods. Therefore, the created real-time facial recognition system can be applied as an accurate smart locking door security solution citeWangean2023.



2.2.2 Face Recognition Door Lock System Using Raspberry Pi

Everyday, a plethora of security-related problems arise, necessitating solutions using the most recent technology. To enhance security, a facial recognition module was implemented in this project. This module uses a camera to capture images of individuals, employing facial recognition technology, and stores these images in a database. In this project, new technology is deployed to develop a Raspberry Pi-based face recognition door lock system. In this Figure 2.2, the door can be unlocked with the help of the photo.

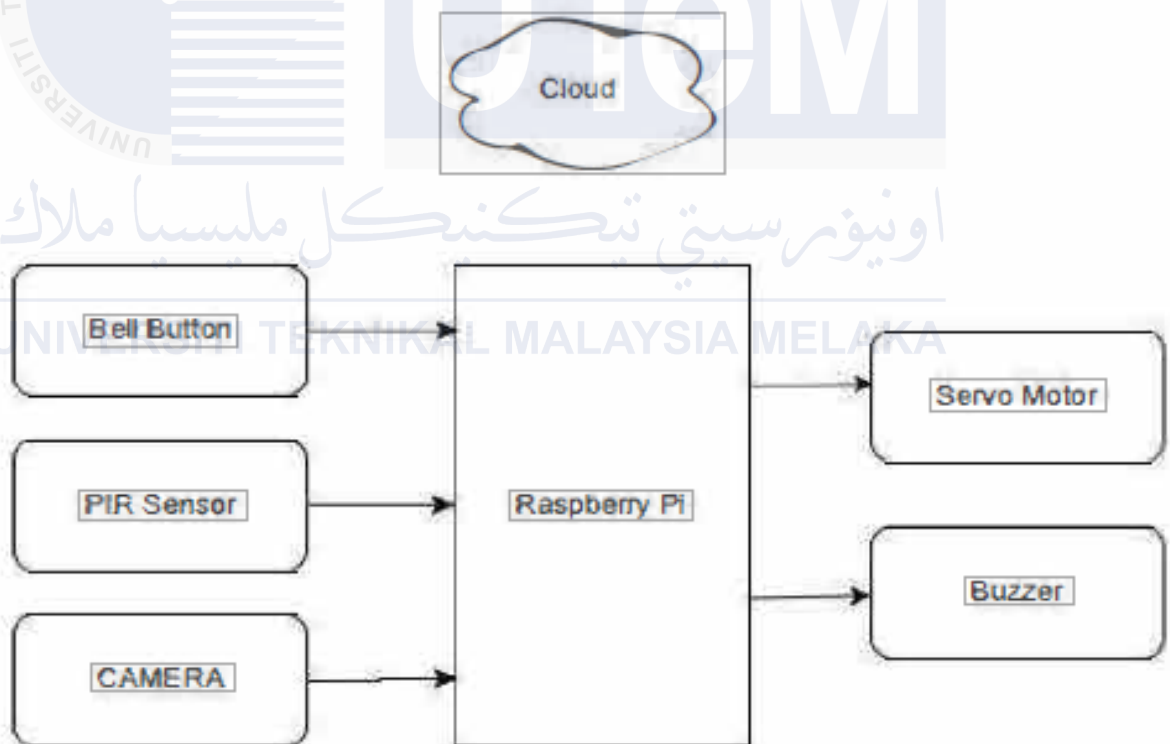
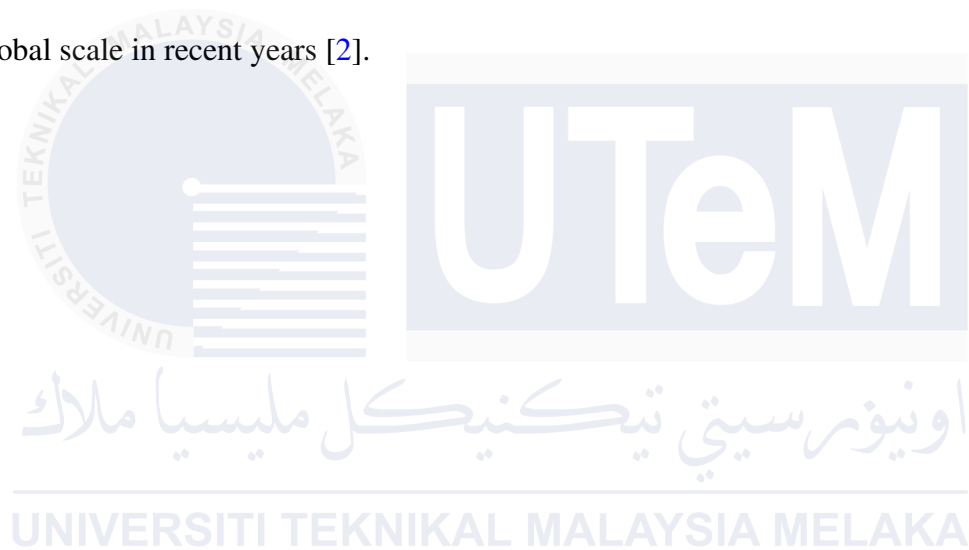


Figure 2.2: Figure shows a Block Diagram using Raspberry Pi [2].

When the subject stands in front of the camera, the lock on the door will be released. The camera will then confirm that the subject's face matches the picture that is already in the database, at which point only the door will be unlocked. The user will receive a warning message from the system if it is unable to identify the user's face. One of the safest biometric verification systems is face recognition. Going into a new technology environment at this time, one may observe that theft and fraud are becoming more and more commonplace on a global scale in recent years [2].



2.2.3 Biometric Authentication Based Automated, Secure, and Smart IOT Door Lock System

Worldwide adoption of the trend toward IOT-based smart city devices has been swift. Convenience, economy, and security are the three main features of smart city gadgets, which are connected to databases. This study develops an automated, secure, and intelligent IOT door lock based on biometric authentication [11]. Face recognition is used to gain access to approved users [3]. In this Figure 2.3, the system begins and the user is prompted to input either A or B (likely a selection for the type of access or user type).

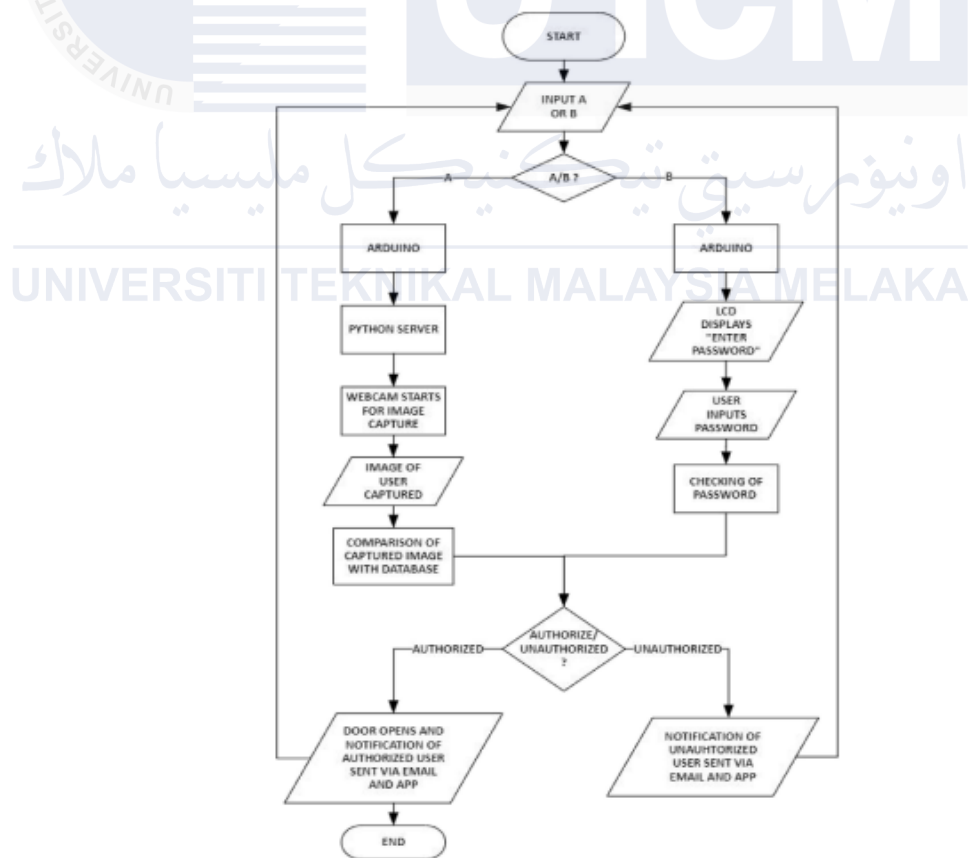


Figure 2.3: A Flowchart of Proposed System[3].

The Arduino board displays "ENTER PASSWORD" on an LCD screen. A webcam is activated to capture an image of the user's face. The webcam captures an image of the user's face. The captured image is compared with a database of authorized users' facial images (1-1-1-1 likely represents a unique identifier or a hash value for each user). The user inputs a password (which might be used in conjunction with facial recognition or as a fallback method). The system checks the input password against a stored password or a password associated with the user's facial image in the database. If the password and facial recognition match, the user is authorized (AUTHORIZED). If not, the user is unauthorized (UNAUTHORIZED). If authorized, the door opens, and a notification is sent to the user via email and a mobile app. If unauthorized, a notification is sent to the administrator or security personnel via email and a mobile app, indicating an unauthorized access attempt.

2.2.4 A Smart Door Lock Security System using Internet of Things

Globally, people's top worry is security. and it has served as a unifying theme for all important industries. These days, it is possible to argue that security is the cornerstone that is essential to penonal safety. Protection from theft and trespassing are the two most crucial needs for individuals in security systems. CCTV cameras are frequently used for security [4]. In this Figure 2.4, any kind of data may be managed, stored, and retrieved using databases.

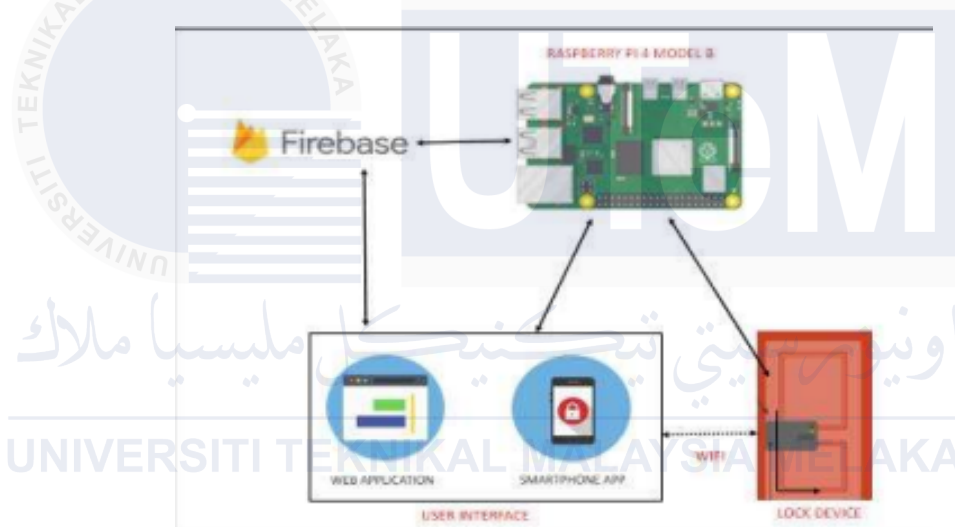


Figure 2.4: A Block Diagram of the Propose System [4].

The data is saved and arranged in an orderly fashion. The Backend as a Service (BaaS) platform Firebase is mostly utilized for mobile applications. Firebase is what we've chosen because it provides cloud storage instead of local storage. Consequently, information kept in Firebase may be retrieved at any time and from any place. In our project, the Firebase platform is being utilized. The web portal, which is linked to the Raspberry Pi and sends commands to it, is hosted by Python. Users are intended to be able to authenticate, control, and access every feature of the system using the Android App. In order to connect with

the lock device and unlock it, the app transmits control signals to it. The whole database's contents are retrieved by the app over the Internet. The software accesses all records and camera feeds over the Internet. The lock device functions as a lock and is mounted within the door. The device is equipped with the necessary mechanical and electrical components in order for it to function as the architecture depicts. Through the Internet, the device transmits all of its data to the database.



2.2.5 Automatic Door Locking System in Households using IoT

The research paper analysis concentrates on smart door lock systems that make use of cutting-edge technology including facial recognition, fingerprint identification, one-time passwords, and smartphone accessibility from a distance [5]. Figure 2.5 represents a smart door lock system's block diagram shows the parts and how they work together to provide safe access.

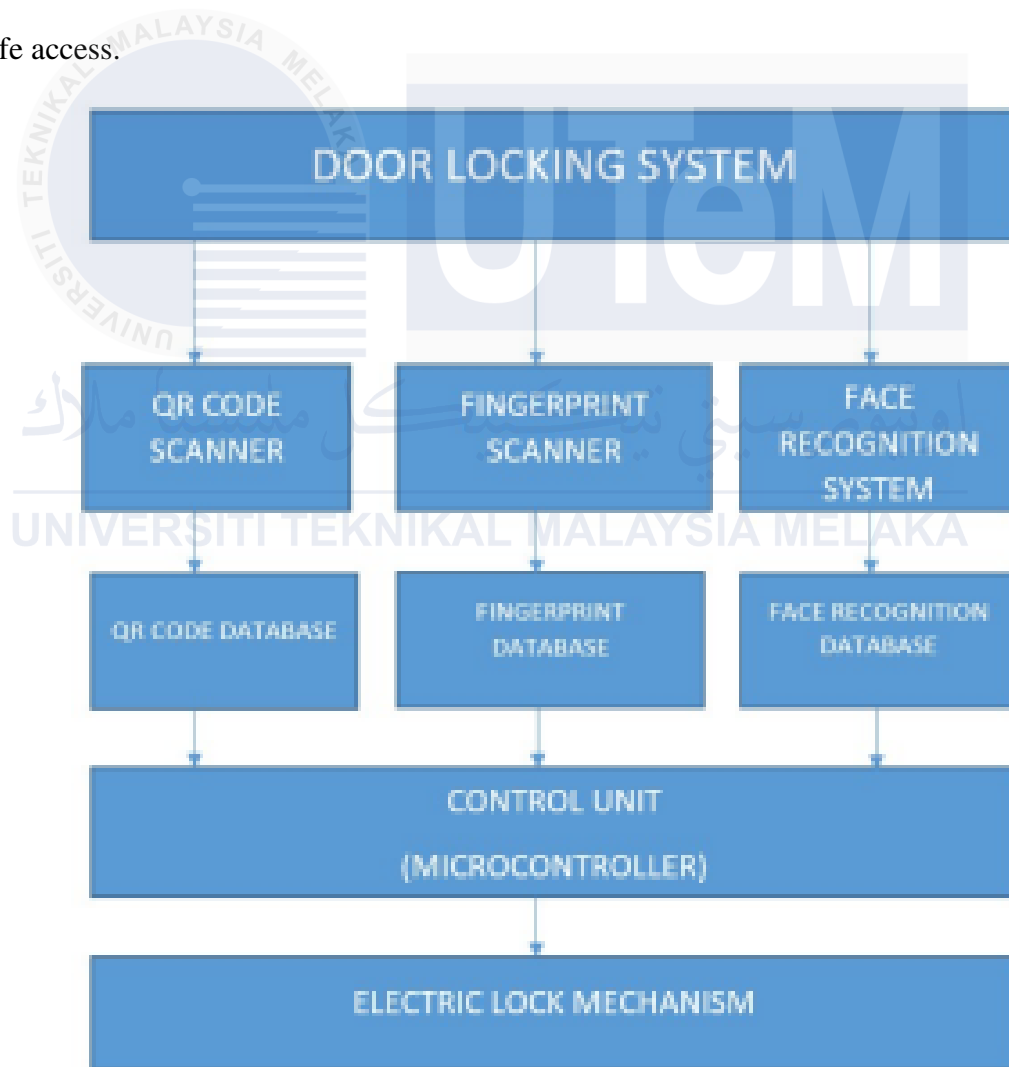


Figure 2.5: A Flow chart using IoT [5].

PIN code entry, facial recognition, fingerprint recognition, and OTP (Guest Mode) are all included in the system. To open the door, users input a PIN on a keypad; feedback is given for erroneous entries. The face recognition system confirms users by matching their facial characteristics with a recorded database, while the fingerprint recognition system checks fingerprints against allowed templates. For temporary access, OTPs are created by scanning QR codes. After successful verification, an electronic lock mechanism is signaled by a control unit to unlock the door. The control unit coordinates various authentication operations. The technology makes entry easier for regular visitors, offers biometric unlocking, gives notifications for lock actions, and can be remotely monitored via a smartphone app. This technology is appropriate for usage in banks, warehouses, and other secure environments.

2.2.6 Design of Home Security System Using Face Recognition with Convolutional Neural Network Method

The goal of the research is to improve accuracy and efficiency in managing access to the door system by utilizing Convolutional Neural Network (CNN) techniques for facial recognition in home security system design[6],[12]. Figure 2.6 shows an access control system that uses face recognition to open locks and other barriers.

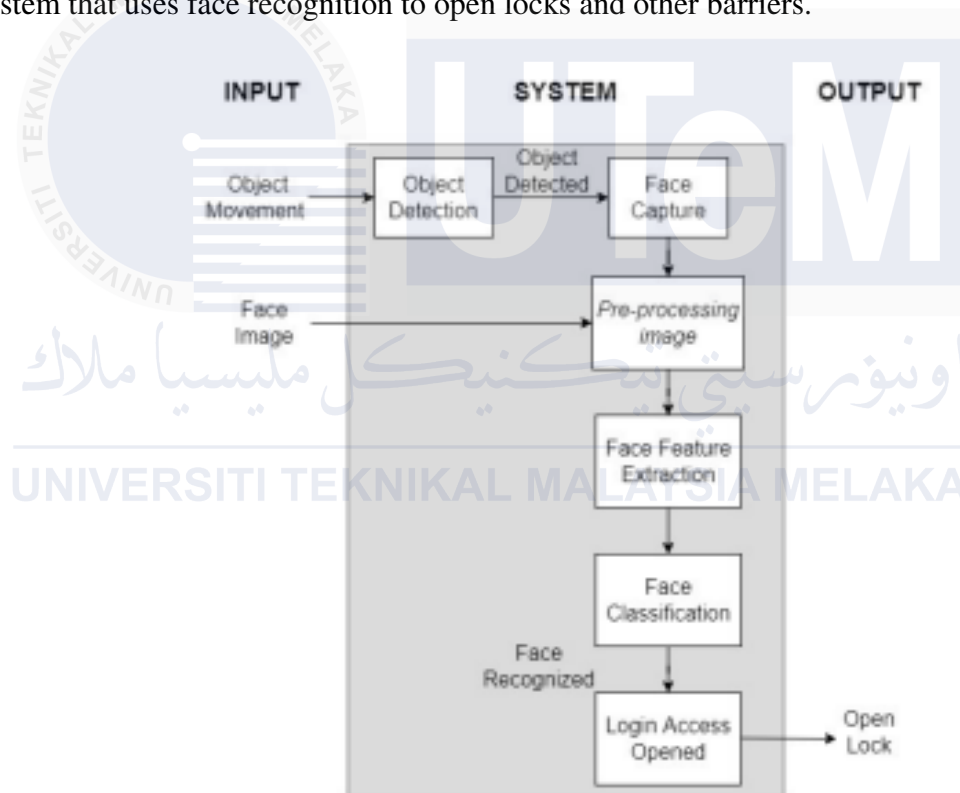


Fig. 1 Home Security System

Figure 2.6: A Block Diagram with Neural Network Method [6].

Object movement and facial pictures are the first two input kinds that the system receives. The "Object Detection" block processes the object movement input first. The "Face Capture" block is activated to take a picture of the face when an item is recognized. In order to improve the picture quality and make sure it is appropriate for additional analysis, the acquired face image and any direct face image input are subsequently pre-processed.

The algorithm then moves on to "Face Feature Extraction," where unique face traits are found and retrieved from the picture, after the pre-processing stage. These characteristics are essential for differentiating between faces. The "Face Classification" block receives the extracted features and uses pre-trained models and algorithms to classify the face. The procedure advances to the "Login Access Opened" step, indicating that access is allowed, if the identified face matches one in the system's database.

The output action is "Open Lock," which enables the system to unlock the mechanism it controls and gives the authorized person physical access, after access has been allowed. This process guarantees a safe and effective approach to face-based access control.

2.2.7 A Low-Cost Embedded Facial Recognition System for Door Access Control using Deep Learning

The project involves the development of an Unmanned Aerial System (UAS) with a focus on creating an automatic locking door using face recognition technology. The system utilizes deep learning techniques to implement a low-cost embedded facial recognition system for controlling an electromagnetic lock. The primary objective is to enhance security measures by integrating face recognition and object detection stages to prevent unauthorized access [7]. The Figure 2.7 shows a smart lock system that implements face recognition using Intel's OpenVINO platform.

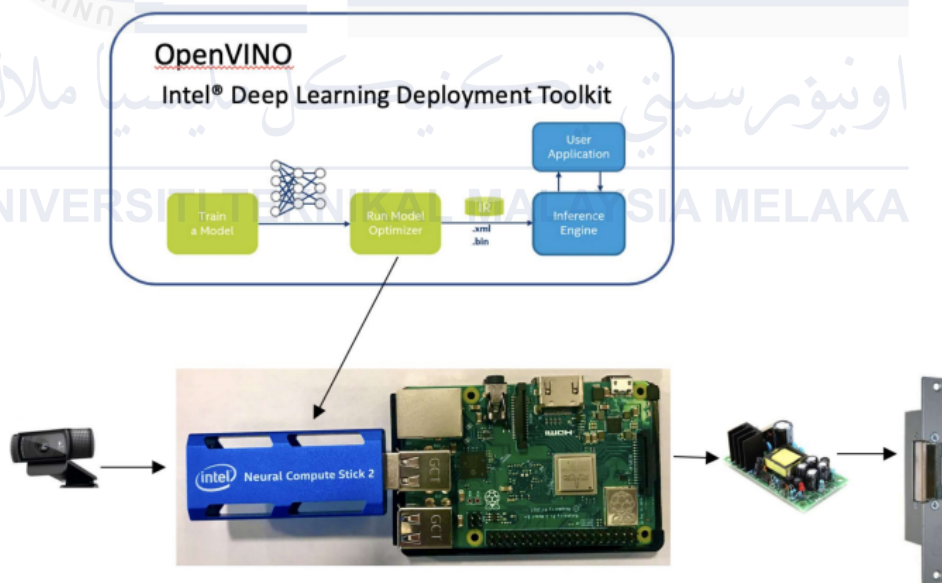


Figure 2.7: Schematic representation of the components of the smart lock system[7].

The OpenVINO workflow, which entails training, optimizing, and deploying a deep learning model via an inference engine, is seen in the upper portion of the figure. The trained model is transformed into an Intermediate Representation (IR) format using this approach, which makes it suitable for deployment on edge devices.

The process is shown beneath the system's physical components. Face photos or video streams are taken by a camera, and the Intel Neural Compute Stick 2 (NCS2) processes them thereafter. A tiny single-board computer called the Raspberry Pi is connected to the USB-based NCS2. The application and inference engine are executed on the Raspberry Pi in order to process the pictures using the OpenVINO optimized model.

The system sends a signal to unlock the door upon detecting and verifying the face. The power supply unit provides the necessary power for the electronic components, such as the lock. Finally, the electric strike lock receives the signal to open, allowing entry. This schematic demonstrates the integration of hardware and software to create a dependable and advanced smart lock system.

2.2.8 Face Recognition Door Lock System Using Raspberry Pi

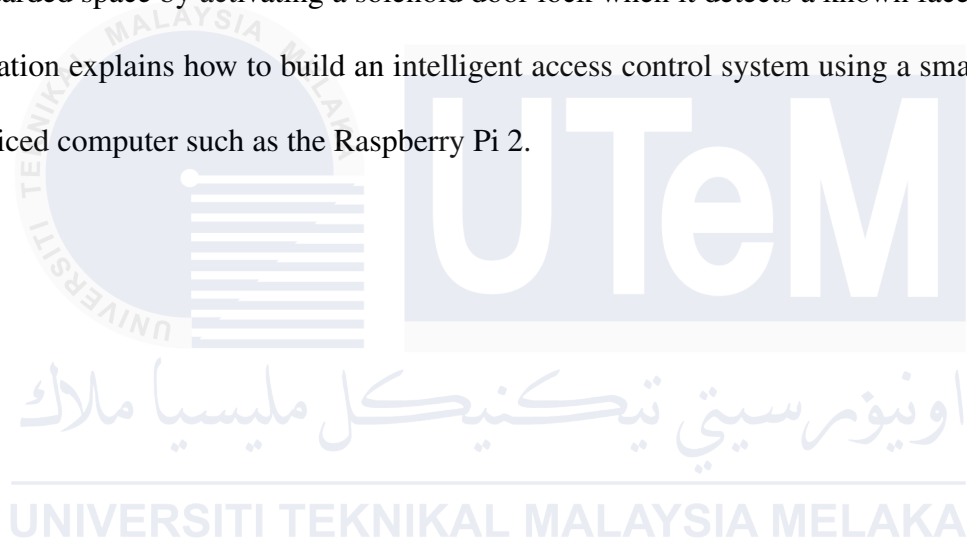
Facial recognition technology is becoming an indispensable tool for security measures in today's digitalized environment. Accurate customer identification is ensured by these clever technologies that automatically detect, analyze, and authenticate faces in photos and videos. They match photographed faces to a database that is kept by using the OPEN CV library, allowing access to just those who are allowed[13]. The technological nuances of these systems are explored in this study, with a focus on their use in safe door lock applications [8]. Based on Figure 2.8, a security system project with a Raspberry Pi 2 serving as its central processing unit is depicted in the diagram.



Figure 2.8: A Block Diagram by using Raspberry Pi [8].

The Raspberry Pi 2 is powered by an external power source and comes with a micro SD card that holds the operating system's software. Network communication is made possible by the Ethernet connection that connects it to the internet.

The Raspberry Pi 2 has a Pi Cam connected, which functions as the system's eyes and is probably used for face recognition to identify people. The system regulates entry to a guarded space by activating a solenoid door lock when it detects a known face. This configuration explains how to build an intelligent access control system using a small, reasonably priced computer such as the Raspberry Pi 2.



2.2.9 IoT and Face Recognition based Automated Door Lock System

This research project employs Internet of Things face recognition to unlock doors. The purpose of this system is to keep an eye on any unauthorized individuals entering the home. With the aid of the Raspberry Pi platform, a system with communication and electronic devices through face detection was developed [9]. The Figure 2.9 shows a flowchart for an automated access control system that uses face recognition.

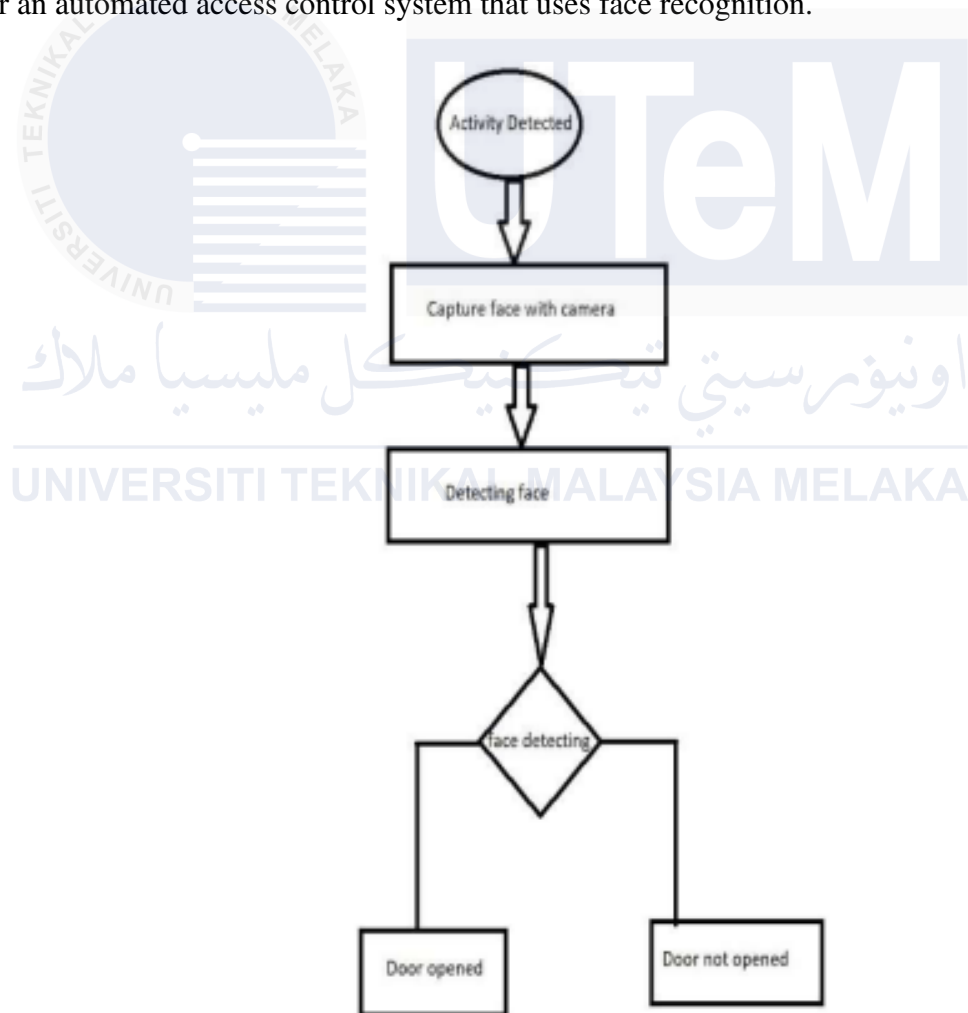


Figure 2.9: A Flowchart of existing system.[9]

The procedure starts when the system detects activity, which causes a camera to take a picture of the person's face. After that, the image is examined to identify facial traits and see if the image matches the faces of any authorized users in the database.

The mechanism opens the door to allow entrance if a match is discovered. If not, the door stays locked, making it impossible to enter. This flowchart demonstrates a technology-enhanced security approach that makes sure only authorized individuals are able to access protected areas.

This system offers a safe and effective way to control access to critical areas, demonstrating how cutting-edge recognition algorithms may be integrated with real-world uses. Such technology is being used more and more in a variety of settings, where security is crucial, such as government, business, and residential buildings.

2.2.10 Design and Development of IOT based Smart Door Lock System

The two eyes in today's internet-driven society are security and comfort. We can accomplish this with the aid of the internet of things (IoT), which gives users comfort and security. Since the door is the only thing separating us from the outside world, it is crucial that our homes be properly secured. Door locks have evolved throughout time, going from large, bulky, intricate locks to contactless smart locks. In the old systems, a lot of doors used mechanical locks with a single key [10]. The Figure 2.10 shows a flowchart that shows how an automation or security system with sensors and controllers works in a methodical manner.

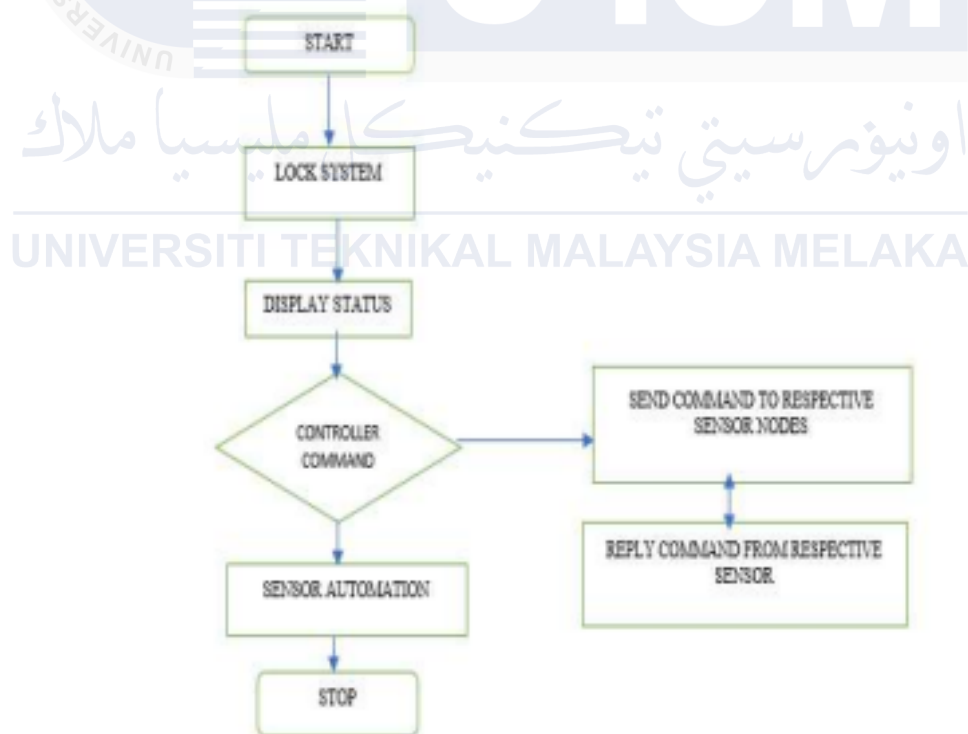


Figure 2.10: A Flowchart of proposed model followed for smart door lock.[10]

The procedure starts with the "START" step and proceeds to the "LOCK SYSTEM" action, when the system is said to secure a device or location. The system then gives a visual or audio status update, as indicated by the "DISPLAY STATUS" action.

After evaluating a command, the system splits into two alternative courses of action at the "CONTROLLER COMMAND" decision point. If the command relates to sensor automation, the action "SENSOR AUTOMATION" is performed, and "STOP" is used to end the sequence. A feedback loop where the sensor nodes react to the controller's command is indicated by the sequence "SEND COMMAND TO RESPECTIVE SENSOR NODES," followed by "REPLY COMMAND FROM RESPECTIVE SENSOR," if the command entails instructing sensor nodes.

Table 2.1: Summary of the previously proposed techniques

Authors	Proposed Technique	Advantage(s)	Disadvantage(s)
Wangean <i>et al.</i> , [1]	Development of Real-Time Face Recognition for Smart Door Lock Security System using Haar Cascade and OpenCV LBPH Face Recognizer	<ul style="list-style-type: none"> • Users can simply approach the door and have it automatically unlock upon successful face recognition. • The system can keep track of who has accessed the door. 	<ul style="list-style-type: none"> • can be vulnerable to spoofing or hacking. • can be expensive, especially for high-quality, reliable systems that offer adequate security. • can potentially fool the system, compromising security.
Reddy <i>et al.</i> , [2]	Face Recognition Door Lock System Using Raspberry Pi	<ul style="list-style-type: none"> • eliminating the need to carry keys or access cards. 	<ul style="list-style-type: none"> • does not provide details on the accuracy or reliability of the algorithm. • system relies on the Raspberry Pi and associated hardware and software.
Doshi <i>et al.</i> , [4]	A Smart Door Lock Security System using Internet of Things	<ul style="list-style-type: none"> • both easy and cost-effective, as well as secure. • reliable and easy to use. 	<ul style="list-style-type: none"> • unable to be accessed remotely if the Bluetooth range was poor. • needed to be combined with a password to strengthen the security level.
Saroha <i>et al.</i> , [3]	Biometric Authentication Based Automated, Secure, and Smart IOT Door Lock System	<ul style="list-style-type: none"> • Eliminates the need for carrying physical keys or RFID cards. • Incorporates biometric authentication through face recognition. 	<ul style="list-style-type: none"> • the dependence on Zigbee protocol incurs higher cost. • Zigbee supports low data rates which results in low transmission rate.
Sivaprasad <i>et al.</i> , [5]	Automatic Door Locking System in Households using IoT	<ul style="list-style-type: none"> • providing easier and more convenient door access. • can provide alerts when the door is locked or unlocked. 	<ul style="list-style-type: none"> • may not be able to access the door without a backup physical key. • more expensive than traditional mechanical locks.

Continued from previous page . . .

Authors	Proposed Technique	Advantage(s)	Disadvantage(s)
Orna <i>et al.</i> , [7]	A Low-Cost Embedded Facial Recognition System for Door Access Control using Deep Learning	<ul style="list-style-type: none"> capable of processing facial recognition in real-time. enhances security measures by accurately identifying individuals based on their facial features. 	<ul style="list-style-type: none"> There may be limitations in achieving high accuracy and reliability. The system may be susceptible to security vulnerabilities.
Muneera <i>et al.</i> , [8]	Face Recognition Door Lock System Using Raspberry Pi	<ul style="list-style-type: none"> increased security by restricting access to authorized individuals only. allowing keyless entry through methods such as keycards, keypads, or biometric scans. 	<ul style="list-style-type: none"> can be vulnerable to lock picking or bumping techniques, compromising security. a risk of unauthorized access to the premises.
Surla <i>et al.</i> , [9]	IoT and Face Recognition based Automated Door Lock System	<ul style="list-style-type: none"> allows the owner to communicate with someone at the door via a voice message if they are unable or unwilling to open the door. requires less human effort to operate the smart door. 	<ul style="list-style-type: none"> time-consuming to update the database with new faces or changes to existing faces. does not have an alternative method for opening the door if face recognition fails.
Nakhandrakumar <i>et al.</i> , [10]	Design and Development of IOT based Smart Door Lock System	<ul style="list-style-type: none"> can be integrated with home appliances and other smart home systems for a seamless experience. designed to be power-efficient, which is beneficial for long-term use and cost-effectiveness. 	<ul style="list-style-type: none"> possibility of the lock being hacked. a risk of contracting the virus through contact with the lock.
Nabila <i>et al.</i> , [6]	Design of Home Security System Using Face Recognition with Convolutional Neural Network Method	<ul style="list-style-type: none"> providing a high level of precision in identifying people. can receive alerts when the door is locked or unlocked. 	<ul style="list-style-type: none"> Difficulty for elderly or disabled people in accessing the door. Inability to remotely monitor or control access to the property.

2.3 Summary

In summary, the literature review highlights significant advancements in the field of face recognition door lock systems, particularly with the integration of IoT and deep learning technologies. These studies collectively underscore the ongoing evolution of security solutions, paving the way for future research to explore further enhancements in the reliability and efficiency of these systems. The review sets the stage for the current project, which aims to develop a face recognition door lock system using ESP32-CAM, integrating necessary components such as a solenoid lock, relay module, and power supply to improve security and convenience in environments like student campuses and offices .

CHAPTER 3

METHODOLOGY

This chapter describes the methodology and design steps to develop face recognition door lock system by using ESP32-CAM.

3.1 Introduction

The research strategy and processes are described in the methodology section. It describes the techniques used for gathering and analyzing data and explains why they are suitable for answering the study questions. Typically, this part contains details on the research design, sample selection, methods for collecting data, and analytic tools. The methodology section makes sure that the study can be repeated and its conclusions can be verified by giving a clear and accurate explanation of the research process. This openness contributes to proving the research's validity and dependability.

3.2 Description of Methodology

The initiation of the project will involve the examination of the relay module, solenoid valve, and ESP32-Cam. The ESP32-Cam holds utmost importance as a vital element of the Face Recognition Door Lock System, as it will seamlessly integrate with all the other components. This system facilitates the unlocking of the door through a mobile phone. Once the ESP32-Cam identifies familiar faces, it will transmit a signal to the relay, subsequently triggering the solenoid valve to unlock the door.

3.3 Block diagram of project

A block diagram serves as a simplified graphical depiction of a system or process, utilizing blocks to showcase the main components and arrows to indicate the connections and flow among them. Each block symbolizes a distinct element of the system, like a function, process, or hardware component, with the arrows demonstrating the interactions between these elements. By offering a concise, top-level view of intricate systems, block diagrams facilitate comprehension of their organization and operation. In Figure 3.1, the block diagram shows a system controlled by an ESP32-CAM microcontroller.

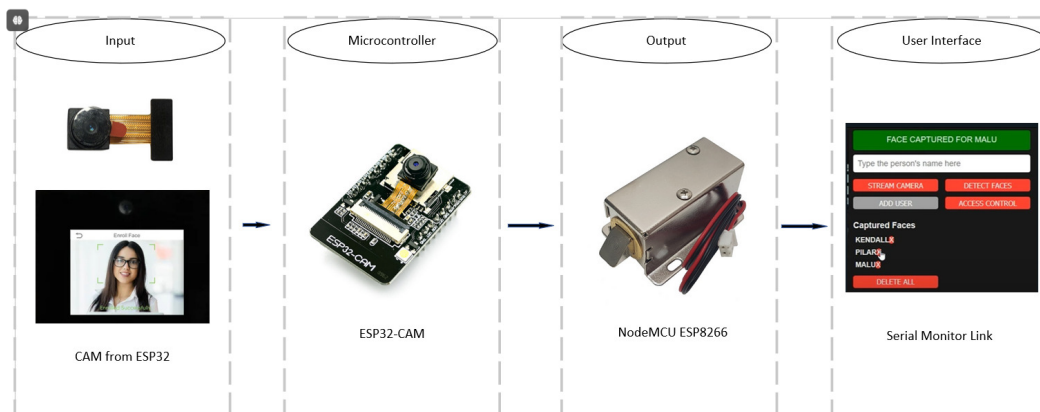


Figure 3.1: A block diagram.

The ESP32-CAM receives power from a power supply. It receives camera data from an ESP32-CAM module. It outputs to: A notification system from serial monitor link, A solenoid valve.

3.4 Flowchart

A flowchart is a visual tool used to represent a process, workflow, or system using different symbols. Each symbol represents a specific step or action, with arrows showing the direction and flow from one step to the next. Common symbols include ovals for start and end points, rectangles for process steps, and diamonds for decision points. Flowcharts help simplify complex processes, making them easier to understand and analyze.

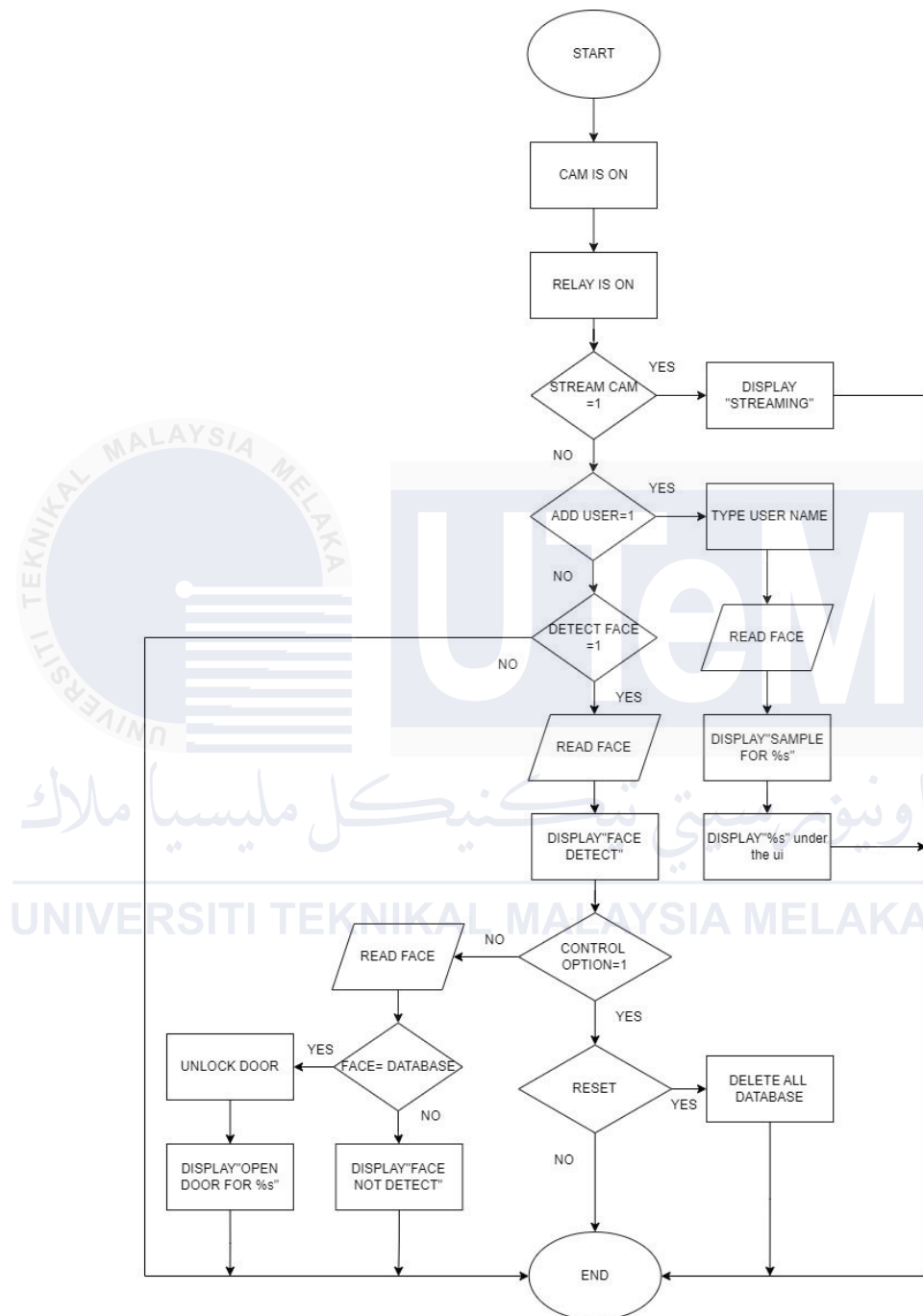


Figure 3.2: A Flowchart.

In Figure 3.2, the system starts by turning on the camera and relay. The camera then streams video, which is used to detect faces. When a face is detected, the system checks if the person is authorized by comparing their face to those in the database. If it's a match, the door unlocks and a welcome message appears. If the face isn't recognized, the system denies access and displays a message saying the user isn't recognized. The system also allows new users to be added by typing in their name and capturing their face. This adds them to the database so they can be recognized in the future.

Finally, the system has an option to delete all users from the database, which can be useful in certain situations. Overall, the system uses face recognition to control access to a secure area, making it a convenient and secure way to manage who can enter.

3.5 Hardware

Hardware is an internal and exterior tools and devices that allow you to carry out important tasks like processing, input, output, storage, communication, and more.

3.5.1 ESP32-CAM

A camera and an ESP32 microcontroller are combined in the small and cheap ESP32-CAM module to enable a variety of camera-focused applications. This module's capacity to take pictures, stream films, and carry out crucial computer vision jobs has made it more and more popular among people and professional alike. Its affordability and adaptability render it an attractive option for a range of do-it-yourself electronics projects.

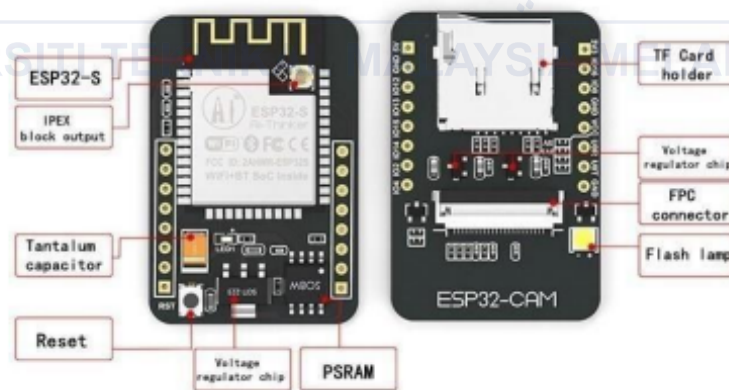


Figure 3.3: ESP32-CAM

Figure 3.3, The ESP32-CAM, which has dual-core CPU(central unit processor) and

Bluetooth and Wi-Fi functionality, powers the ESP32-CAM. For many current IoT applications, the module capacity to connect to networks and interact wirelessly is crucial. Furthermore, the module integrated camera makes it easier to share live video and take beautiful images.

Its simplicity of it is one of the main benefits of the ESP32-CAM. It is simple to program with the Arduino IDE, which is a popular option among hobbyists. A vast assortment of online tools and libraries are easily accessible to help consumers get started on their projects. Furthermore, the module's small size makes it simple to integrate into a variety of projects and gadgets that might not have a lot of space.

Specification:

- Chip: ESP32-D0WDQ6
- Architecture: Dual-core 32-bit Xtensa LX6 CPUs
- Camera Model: OV2640
- Resolution: 2 Megapixels (1600 x 1200)
- Wi-Fi: 802.11 b/g/n, supports 2.4 GHz band
- MicroSD Card Slot: Supports cards up to 4 GB
- Operating Voltage: 3.3V
- Size: Approximately 27 x 40.5 x 4.5 mm

3.5.2 Solenoid Valve 12V

A device that uses an electrical signal to control the locking and unlock of doors is a solenoid door lock valve, which operates at 12 volts. Because it runs on a 12-volt power source, it may be used in a mixture of home and business situations. Because of its reliability and efficiency, this particular type of lock is often used in access control systems, automated doors, and security systems.



Figure 3.4: Solenoid Valve 12V

From Figure 3.4, When an electric current passes through the solenoid, which is located inside the lock, it responds to magnetism. The solenoid displaces a metal plunger or armature by using the 12V power supply to create a magnetic field. Through this method, the locking mechanism may be turned on or off, allowing the door to be locked or unlocked. Secure access control is guaranteed by this efficient method.

The reliable performance and easy-to-use design of a 12V solenoid door lock are its main advantages. Because these locks don't have any mechanical parts that may break, they are very safe. Furthermore, these solutions offer a variety of practical access choices by

integrating easily with a set of access control systems, including keypads, card readers, and remote controllers.

Specification:

- Operating voltage of 12V
- Operating current of 1A
- ON time/ electricity conduction time of max 10s
- Lock without power and unlocked with power
- Dimension of 54mm * 38mm * 27mm

3.5.3 Power Supply Module 12V/5V/3V

The DC-005 Power Supply Module is a small and multipurpose module that uses a standard DC barrel jack (2.1mm inner diameter and 5.5mm outer diameter) to supply power to electronic applications. Prototyping setups, Arduino projects, and do-it-yourself electronics all often employ this module.

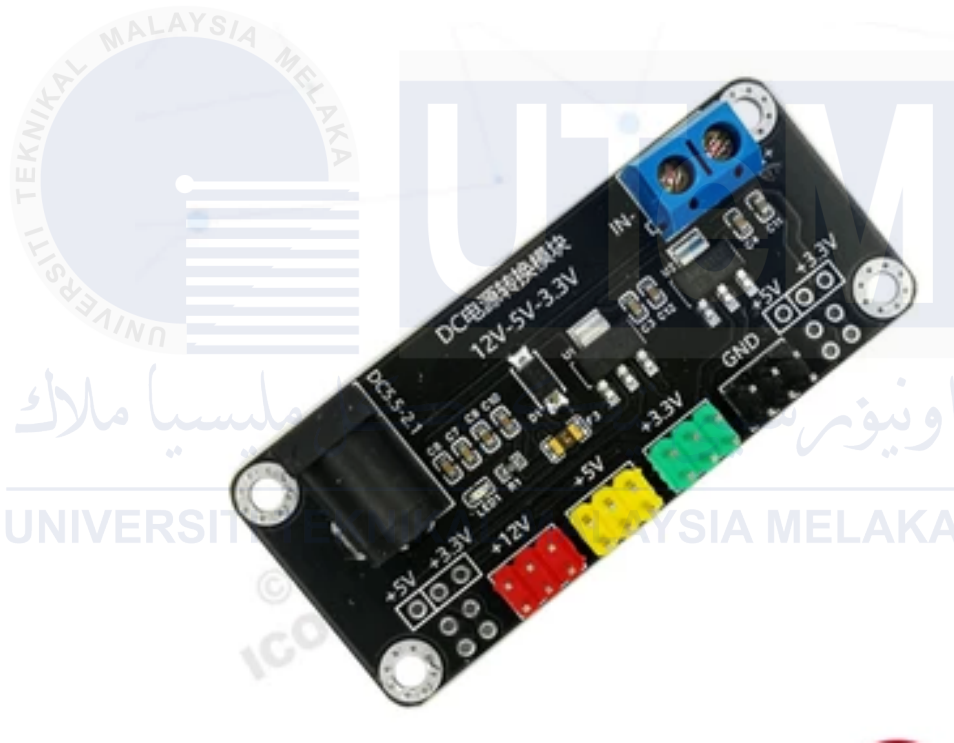


Figure 3.5: Power Supply Module

From the Figure 3.7, The DC-005 module's ease of use is one of its primary characteristics. It enables users to deliver voltage directly to their circuit by connecting a DC power adaptor. An input voltage range of 5V to 12V is normally supported by the module, contingent upon the needs of the attached components. Direct voltage output is provided

by the output pins, designated VCC and GND, and is readily attached to a circuit board or breadboard with jumper wires.

Because of its small size, the module may be used in projects with limited space. It is especially helpful for powering circuits based on breadboards, IoT devices, and micro-controller boards like Arduino. Furthermore, because it is plug-and-play, soldering is not necessary, making setup easier for both novice and expert users.

The DC-005 Power Supply Module is convenient, but it also frequently has basic protective features like reverse polarity prevention to keep your components safe. For powering circuits while testing, debugging, or deployment, this makes it a dependable and useful option.

All things considered, the DC-005 Power Supply Module is a useful addition to any arsenal for electronics. It is a great way to manage power in a variety of electrical applications because of its small size, ease of use, and compatibility with ordinary DC adapters.

3.5.4 Buck Converter

A buck converter, regularly referred to as a step-down converter, is a kind of DC-DC converter that reduces an output voltage from a higher input voltage. Buck converters are frequently used in many kinds of electronic applications, including embedded systems, battery chargers, and device power supply, due to their high efficiency and low energy loss. Together, a switch (usually a transistor), a diode, an inductor, and a capacitor makes up the converter's main parts, which control the output voltage.

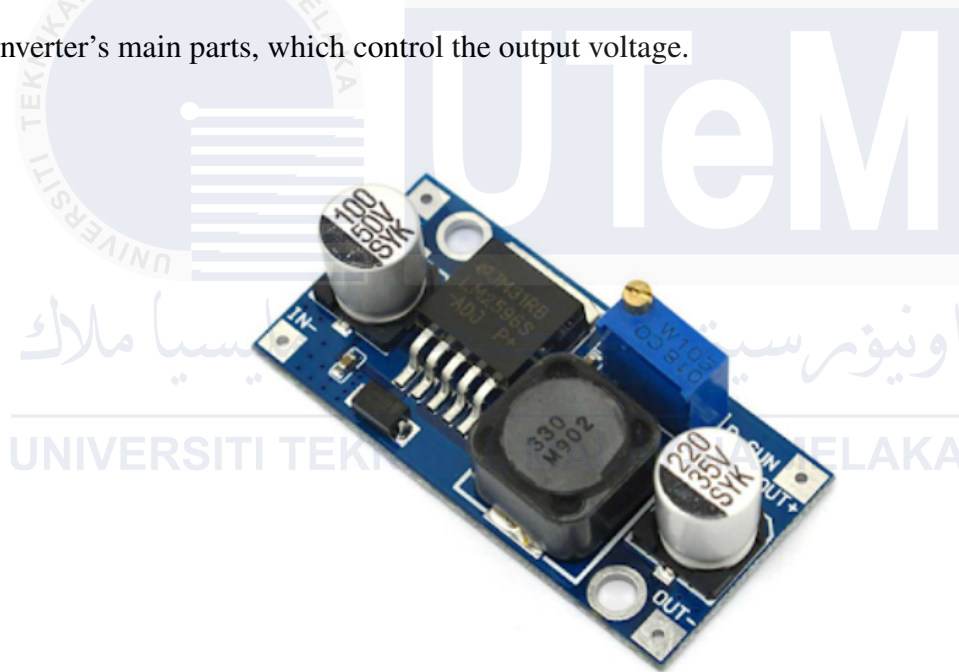
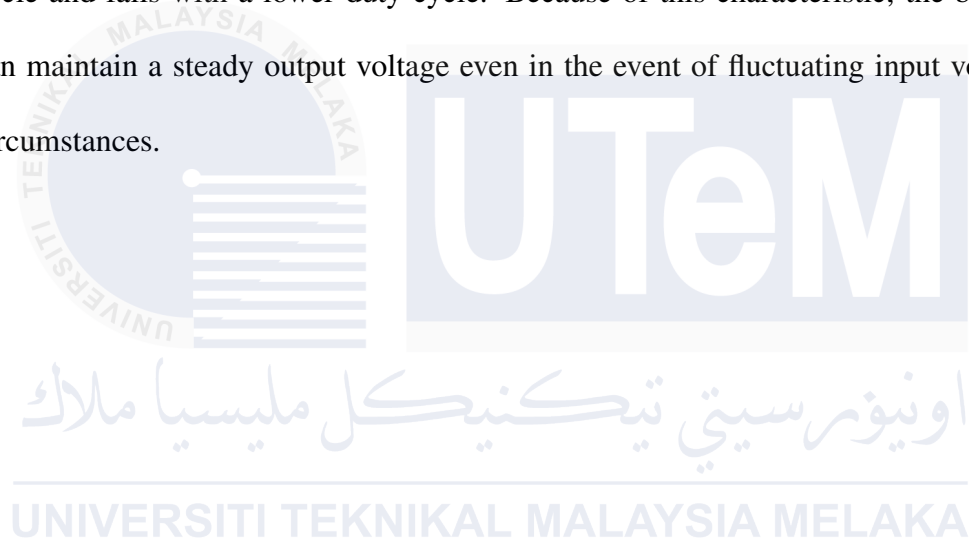


Figure 3.6: Buck Converter

The idea of energy storage and transmission underpins how a buck converter in Figure 3.8 works. Energy is stored in the inductor's magnetic field as current passes through it when the switch is activated. The input voltage less the voltage drop across the inductor is the output voltage during this phase. The current continues to flow through the load via the diode

when the switch is switched off because the stored energy in the inductor's magnetic field is released. Through filtering away high-frequency switching ripples, the capacitor evens out the output voltage.

The output voltage is determined by the duty cycle of the switch, which is the ratio of the time it is on to the overall switching period. The output voltage rises with a greater duty cycle and falls with a lower duty cycle. Because of this characteristic, the buck converter can maintain a steady output voltage even in the event of fluctuating input voltage or load circumstances.



3.5.5 Relay Module 5V

A kind of electro-mechanical device that functions as a switch is the relay. The contact switches open or shut in response to a flow of direct current (DC) to the relay coil. A coil, a normally open (NO) contact, and a normally closed (NC) contact are the standard components of a 5V relay module. An overview of the 5V relay module's functionality is given in this document. However, it is fundamental to comprehend the fundamental idea of a relay and its pin structure before delving into the specifics of the relay module.

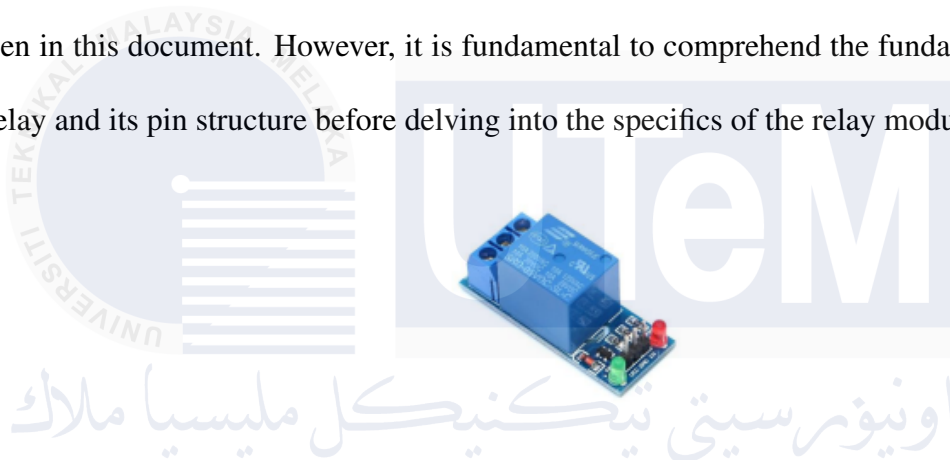


Figure 3.7: Relay Module 5V

The module's mechanical relay operates by means of an electromagnet that regulates a switch. The electromagnet is activated by a 5V signal, creating a magnetic field that moves a lever to open or close the switch. This feature allows the low-voltage signal to either allow or stop the current flow, allowing the linked high-voltage device to be controlled safely and effectively.

The capacity to offer electrical isolation between the control circuit and the high-voltage load is a 5V relay module's main benefit. By isolating the low-voltage control system from the high-voltage circuit, this isolation keeps it safe from errors or spikes. Moreover,

prevention features such optocouplers to improve isolation and diodes to prevent backflow are often built into relay modules.

Relay modules are extensively utilized in various applications, including robotics, industrial control systems, and home automation. These modules provide a reliable and accurate means of regulating larger electrical loads, while also being easy to integrate into pre-existing circuits. Due to their robust design and straightforward functionality, they have become a favored choice for both professionals and enthusiasts. This event is particularly true for those seeking to use low-power controllers to effectively manage high-power devices.

Specification:

- Digital output controllable
- Compatible with any 5V microcontroller such as Arduino.
- Rated through-current: 10A (NO) 5A (NC)
- Control signal: TTL level
- Max. switching voltage 250VAC/30VDC
- Max. switching current 10A
- Size: 43mm x 17mm x 17mm

3.6 Software

The initiative's technological cornerstone is the software covered in this study, which offers the crucial features and functionalities required to meet our objectives. This software, which includes a variety of applications, programs, and algorithms, is made expressly to carry out duties and procedures that are required for the project to succeed.

3.6.1 Arduino IDE

The Arduino IDE, also known as the integrated development environment, serves as a software tool apply for programming and uploading code to Arduino boards. This tool offers a an interface that facilitates the process of writing, editing, and compiling code for various Arduino projects.



Figure 3.8: Arduino IDE

The Arduino IDE has a simple GUI that features a code-writing text editor and a message box displaying error and feedback messages. It used a C++-based programming language and offers a number of pre-built services for modifying hardware. Users may choose the Arduino board from a drop-down menu to ensure accurate coding and uploading to the selected device.

One of the great things about the Arduino IDE is that many libraries it supports. These libraries provide a code sets that improve the performance of the projects by allowing the inclusion of elaborate functions like motor control and sensor data reading. The development process will be expedited and the Arduino board's functionality will be improved by integrating these libraries into the projects.

The Arduino community is very active and in favor of the IDE. There are many tools, forums, and tutorials provided to help with solving issues and project learning. With this community support, sharing work and getting feedback from other enthusiasts is made easier, which inspire creativity and cooperation.

3.6.2 WebServer

A web server is a computer software or system that allows users to access web content over the internet or network. Its primary purpose is to store, process, and distribute websites and online applications to users—typically web browsers—using the Hypertext Transfer Protocol (HTTP) or its secure counterpart, HTTPS. Web servers are the backbone of the World Wide Web, enabling users to access websites and online resources.

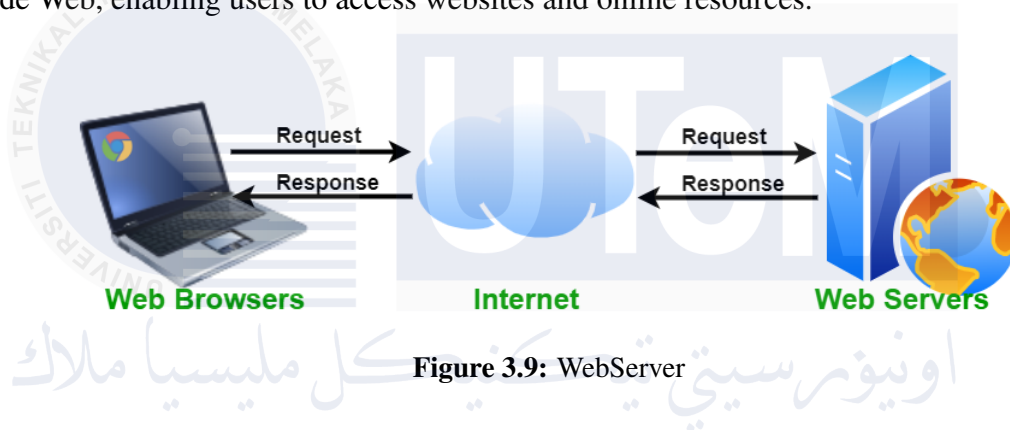


Figure 3.9: WebServer

As an example from Figure 3.9, managing client requests is a web server's primary duty. The web server hosting the website receives an HTTP request from the browser when a user types a URL into it. After processing this request, the server provides the relevant content, which may consist of HTML files, pictures, videos, or other types of data. Either static content—which consists of files that have already been stored—or dynamic content—which is created in real time in response to user input or other variables, including database queries—can be provided.

Additionally, web servers are needed for resource management in order to effectively handle numerous client requests. They employ strategies like load balancing to divide traffic evenly across several servers, caching to temporarily store frequently requested material,

and compression to minimise the amount of data being sent. Even in situations with high traffic, these optimisations aid to ensure the quick and dependable transmission of online information.

Another crucial component of web servers is security. They employ firewalls to stop unwanted access, authentication mechanisms to confirm user identities, and HTTPS encryption to safeguard data while it is being transmitted. These features protect the clients and the server from possible dangers like cyberattacks and data leaks.

Web servers come in plenty of forms, each appropriate for a particular purpose. Static web servers are perfect for basic websites since they offer material that is fixed and never changes. However, dynamic web servers include extra software, such as databases and server-side scripting languages (like PHP, Python, or Ruby), to produce content on the fly in response to user input.

Popular web server software includes Microsoft IIS, which is frequently used in Windows environments; Nginx, which is preferred for its performance and scalability; Apache HTTP Server, which is known for its flexibility; and LiteSpeed, which is made for high-speed operations.

3.7 Gantt Chart

The Gantt chart displayed in Figure 3.10. outlines the schedule for PSM 1 in 2024. It details the timeline and activities planned from March to June, ensuring systematic progress throughout the project. Key milestones include the PSM 1 briefing, finding a supervisor, deciding the project title, and the sequential development of chapters for the final report. The chart highlights crucial phases such as the literature review, component research, methodology development, and finalizing project components. Draft submissions, slide presentations, and the final report submission are also scheduled to ensure a comprehensive and timely project completion, culminating in the PSM 1 presentation.

MONTH	MARCH		APRIL				MAY				JUNE			
ACTIVITIES	1	2	3	4	5	6	7	8	9	10	11	12	13	14
PSM 1 Briefing	■													
Find Supervisor		■	■											
Decide Project Title		■	■											
Chapter 1 (Introduction)			■	■	■									
Research journals (Literature Review)					■	■	■	■						
Component research							■	■	■	■				
Chapter 3 (Methodology)											■	■		
Finalize the component use in the project											■	■		
Draft Submission											■	■		
Slide presentation											■	■		
Final report submission													■	
Presentation PSM 1														■

Figure 3.10: Gantt chart PSM I 2024

3.8 Summary

Based on this chapter, a number of organized phases are involved in making of face recognition door lock system by using ESP32-CAM. Block diagram helps to construct the system which showing the components such as relay module, solenoid valve, and ESP32-CAM. A flowchart shows the sequence of the operation, which includes the action of initiating the camera and relay by streaming video for face recognition, validate authorization by using database and then approving or rejecting access. This conclude a dependable and access control solution and further improvement for remote access and data management functionalites.

CHAPTER 4

RESULTS AND DISCUSSION

The following chapter provides an overview of the findings and examination pertaining to the creation of a face recognition door lock system utilizing ESP32-CAM.

4.1 Expected Result

The implementation of the face recognition door lock system with the ESP32-CAM is expected for the benefits in terms of operational efficiency, security measures, and overall user satisfaction. The projected outcomes for this initiative are detailed as follows:

4.1.1 Reliable Face Recognition

- The system is expected to achieve a high level of accuracy in identifying the faces of registered users, with a minimum identification rate of 95%. This assertion will be substantiated through extensive testing, which will involve analyzing a diverse range of face photographs captured from various angles and under different lighting conditions.
- Having a low false acceptance rate (FAR) and false rejection rate (FRR) is crucial in ensuring that access is granted only to authorized users and that genuine individuals are not mistakenly denied entry.

4.1.2 Time Performance

- The ESP32-CAM module's capability to capture and process facial pictures in real-time holds the potential to facilitate authentication and unlocking, taking less than a second. The efficiency of the system and the convenience experienced by the user are upon this rapid response time.
- The system must demonstrate operation without significant delays, even when handling multiple login attempts in succession.

4.1.3 Robust Security

- It is expected that the implementation of advanced security measures such as encryption and secure data storage will protect user information and prevent unauthorized access to the system. The utilization of IoT capabilities should not compromise the overall security of the face recognition system.
- It is essential for the system to be resistant to common security threats, including attempts to manipulate images or videos. To achieve this, anti-spoofing techniques and algorithms will be implemented.[14]

4.1.4 Integration of IoT

- It is expected that the integration of IoT technology would enable the remote control and monitoring of access. By using the internet, individuals would have the capability to remotely observe live video streams, unlock doors, and receive notifications from any location.[15]
- The communication between the ESP32-CAM module, as well as other IoT platforms or devices, is crucial for establishing an operational smart office or home environment. This capability allows for a cohesive system where all components can interact and exchange information.

Upon achieving these projected objectives, the face recognition door lock system is poised to provide a reliable, secure, and user-friendly method for access control, harnessing the capabilities of ESP32-CAM and modern IoT technologies to enhance both convenience and security.

4.2 Data Analysis

4.2.1 Introduction

This report analyzes the performance of the ESP32-CAM in face recognition systems, focusing on its hardware capabilities, software integration, and operational efficiency. The analysis evaluates the device's strengths, limitations, and performance in real-world scenarios. Key metrics include recognition accuracy, response time, and operational reliability under different conditions.

4.2.2 Data Description

- Total Images Collected: 50 (10 individuals, 5 images per individual, various lighting conditions).
- Testing scenarios :
 - Bright lighting
 - Dim lighting
 - Different face angles (front, side)
- Total 1 Test Cases: 50 (40 authorized, 10 unauthorized).

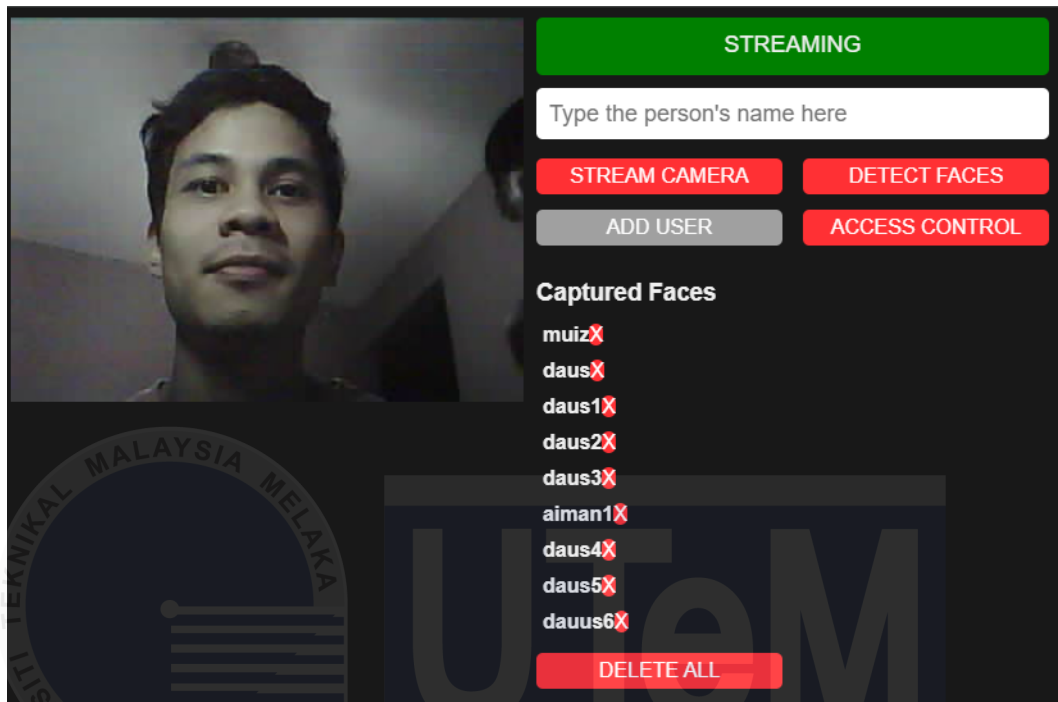


Figure 4.1: Figure shows an example image of data users collected.

4.2.3 Performance Metrics

4.2.3.1 Recognition Accuracy

Table 4.1: Recognition Accuracy Table for distance in 50cm.

Condition	Authorized Faces Recognized(%)	Unauthorized Faces Rejected (%)
Bright lighting	94%	90%
Dim lighting	66%	80%
Different Angles	78%	90%

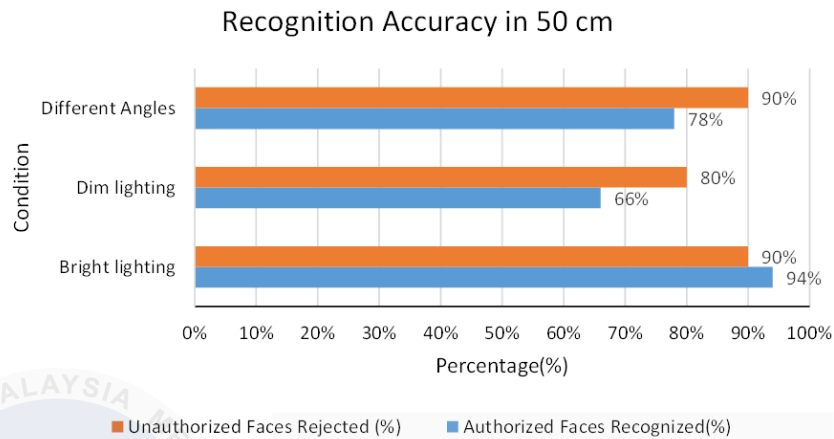


Figure 4.2: Figure shows a Recognition Accuracy in 50cm Column Chart.

The system achieved an overall accuracy of 77.5% for recognizing authorized faces and 86.7% for rejecting unauthorized faces in 50cm. Under bright lighting conditions, the recognition accuracy was notably high, with a 94% success rate for authorized faces and 90% for unauthorized faces being correctly rejected. This demonstrates the system's ability to perform reliably when lighting conditions are optimal.

The recognition accuracy chart in Figure 4.2 illustrates the ESP32-CAM system's performance in identifying authorized and unauthorized faces across various conditions. Under bright lighting, the system achieves the highest recognition accuracy for authorized faces at 94% and correctly rejects 90% of unauthorized faces. These results demonstrate its strong capability in optimal lighting scenarios. However, performance diminishes in dim lighting, where recognition accuracy for authorized faces drops to 66%, and unauthorized rejection decreases to 80%, highlighting the system's sensitivity to lighting variations.

When evaluating recognition accuracy under different angles, the system achieves a moderate 78% for authorized faces and maintains a 90% rejection rate for unauthorized faces. This indicates a relative difficulty in handling non-frontal facial orientations. Overall, the system's average accuracy for authorized access is 77.5%, while unauthorized rejection is higher at 86.7%. These findings emphasize the importance of improving recognition robustness under challenging conditions, such as low lighting and diverse face angles, to enhance overall reliability.

Table 4.2: Recognition Accuracy Table for distance in 75cm.

Condition	Authorized Faces Recognized(%)	Unauthorized Faces Rejected (%)
Bright lighting	90%	85%
Dim lighting	60%	75%
Different Angles	70%	85%

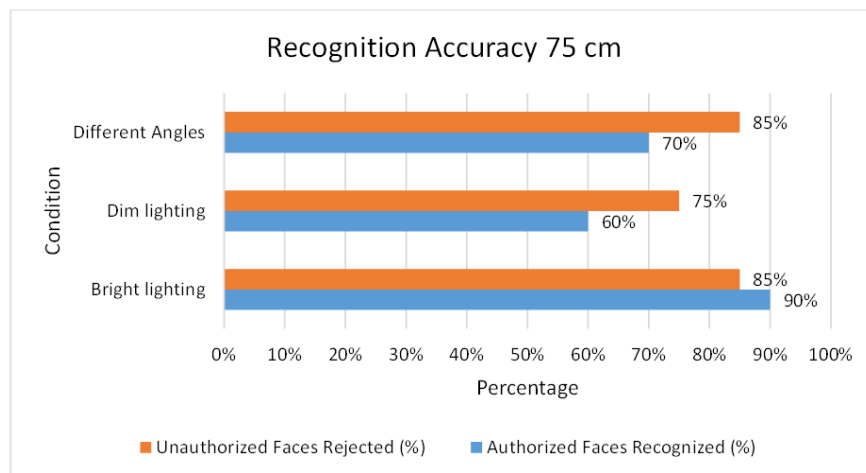


Figure 4.3: Figure shows a Recognition Accuracy in 75cm Column Chart.

At a distance of 75 cm, the face recognition system shows reduced accuracy compared to 50 cm. Under bright lighting, it recognizes 90% of authorized faces and rejects 85% of unauthorized faces, slightly lower than the 94% recognition and 90% rejection rates at 50 cm. This shows that increasing the distance slightly reduces accuracy, even in ideal conditions.

In dim lighting, performance drops further, with 60% recognition and 75% rejection at 75 cm compared to 66% recognition and 80% rejection at 50 cm. The combination of low light and greater distance makes it harder for the system to detect faces accurately.

For different angles, especially at a 45-degree angle, the system recognizes 70% of authorized faces and rejects 85% of unauthorized faces at 75 cm. This is lower than the 78% recognition and 90% rejection rates at 50 cm, showing that handling angled faces becomes more challenging as the distance increases.

Table 4.3: Recognition Accuracy Table for distance in 100cm.

Condition	Authorized Faces Recognized(%)	Unauthorized Faces Rejected (%)
Bright lighting	85%	80%
Dim lighting	55%	70%
Different Angles	65%	80%

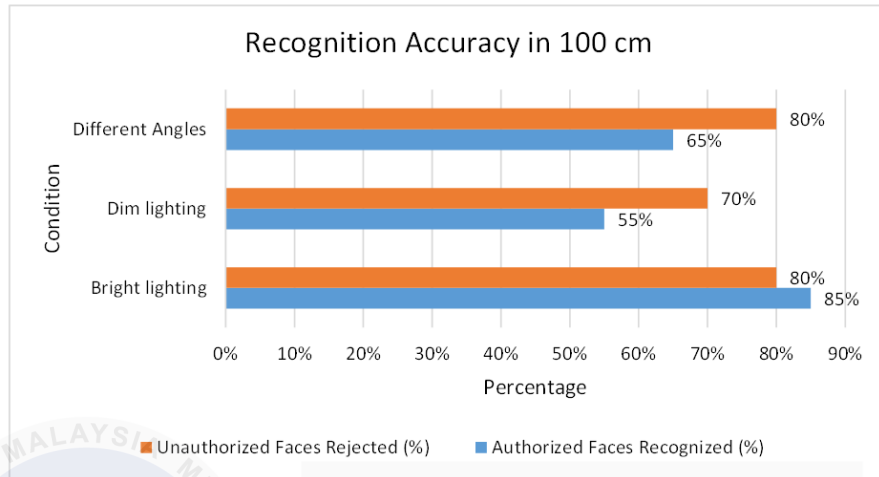


Figure 4.4: Figure shows a Recognition Accuracy in 100cm Column Chart.

At a distance of 100 cm, the recognition accuracy of the ESP32-CAM decreases compared to 75 cm. For authorized face recognition, the performance drops from 90% to 85% in bright lighting, from 60% to 55% in dim lighting, and from 70% to 65% when faces are at different angles. This indicates that as the distance increases, it becomes more challenging for the device to identify authorized faces accurately.

Similarly, the rejection of unauthorized faces also declines as the distance increases. At 100 cm, the unauthorized faces rejected are 80% in bright lighting, 70% in dim lighting, and 80% for different angles. In contrast, at 75 cm, the corresponding values are 85%, 75%, and 85%. This reduction in rejection rates shows that the system becomes less reliable in distinguishing unauthorized faces at longer distances.

Overall, the results highlight that the ESP32-CAM performs better at 75 cm compared to 100 cm in terms of both recognizing authorized faces and rejecting unauthorized ones, with lighting conditions and angles affecting performance at both distances.

4.2.3.2 Error Rates

Table 4.4: Error Rates Table.

Metric	Authorized Faces Recognized(%)
False Acceptance Rate (FAR)	5.0%
False Rejection (FRR)	9.3%

The error rates indicate critical areas for improvement. The false acceptance rate (FAR) of 13.3% means that unauthorized faces were incorrectly granted access in some instances, posing a potential security risk. This issue is less pronounced under bright lighting but becomes more evident in dim lighting and varying angles. Addressing this will require implementing advanced anti-spoofing techniques and improving the robustness of the recognition model.

The false rejection rate (FRR) of 20.7% suggests that nearly one in five authorized faces were denied access. This inconvenience is primarily observed under dim lighting and when users present non-frontal angles. Enhancing the model's ability to generalize across diverse scenarios and employing preprocessing techniques such as histogram equalization could significantly reduce this rate.

4.2.3.3 Response Times

Table 4.5: Response Times Table.

Condition	Distance(cm)	Average Response Time(ms)	Standard Deviation(ms)
Bright lighting	50	1200	50
Bright lighting	75	1300	60
Bright lighting	100	1400	70
Dim lighting	50	2000	150
Dim lighting	75	2100	150
Dim lighting	100	2200	170
Different Angles	50	1200	100
Different Angles	75	1300	110
Different Angles	100	1400	120

The response times demonstrate acceptable performance under bright lighting, with an average of 1200ms and minimal variation. This ensures that the system is responsive and capable of processing recognition tasks efficiently under ideal conditions. Users can experience seamless interactions with the system, particularly in well-lit environments.

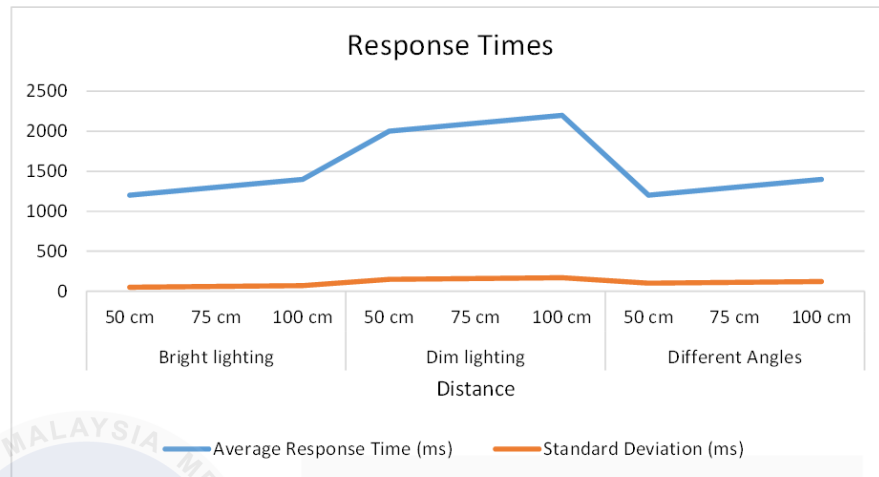


Figure 4.5: Figure shows Response Times for 50,75 and 100cm.

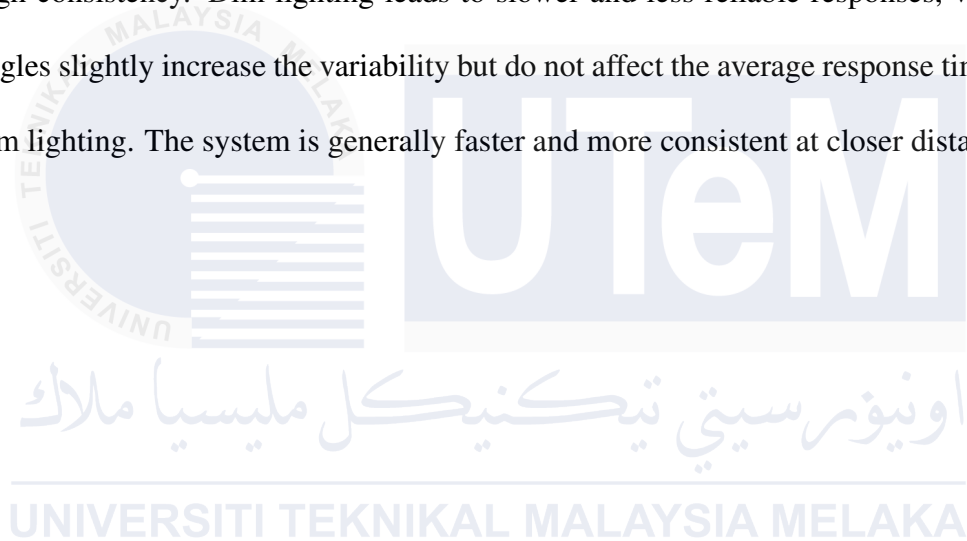
This graph 4.5 shows the response times of the face recognition system under different conditions (bright lighting, dim lighting, and different angles) at varying distances (50 cm, 75 cm, and 100 cm). The blue line represents the average response time, while the orange line shows the standard deviation, which indicates how consistent the response times are.

In bright lighting, the average response time increases slightly as the distance increases, from 1200 ms at 50 cm to 1400 ms at 100 cm, with minimal variation (standard deviation remains low at around 50–70 ms). This shows that the system performs efficiently and consistently under good lighting, even at greater distances.

In dim lighting, the response time is significantly higher, starting at 2000 ms at 50 cm and increasing to 2200 ms at 100 cm, with a slightly larger standard deviation (150–170 ms). This indicates that low light conditions cause delays and less consistency in the system's performance, especially at longer distances.

For different angles, the response time is similar to bright lighting, starting at 1200 ms at 50 cm and increasing to 1400 ms at 100 cm, but with slightly higher variation (standard deviation rising from 100 ms to 120 ms). This shows that handling angled faces is more challenging but still manageable compared to dim lighting.

In summary, the system performs best in bright lighting, with low response times and high consistency. Dim lighting leads to slower and less reliable responses, while different angles slightly increase the variability but do not affect the average response time as much as dim lighting. The system is generally faster and more consistent at closer distances (50 cm).



4.3 Schematic Diagram

The schematic diagram in Figure 4.4 below represents a system designed to control a solenoid door lock using an ESP32-CAM microcontroller, a relay module, and two separate power supplies. The ESP32-CAM, which features a built-in camera and Wi-Fi capabilities, acts as the primary controller. It can be programmed to trigger the solenoid door lock remotely or based on predefined conditions. This functionality makes it suitable for smart door lock systems.

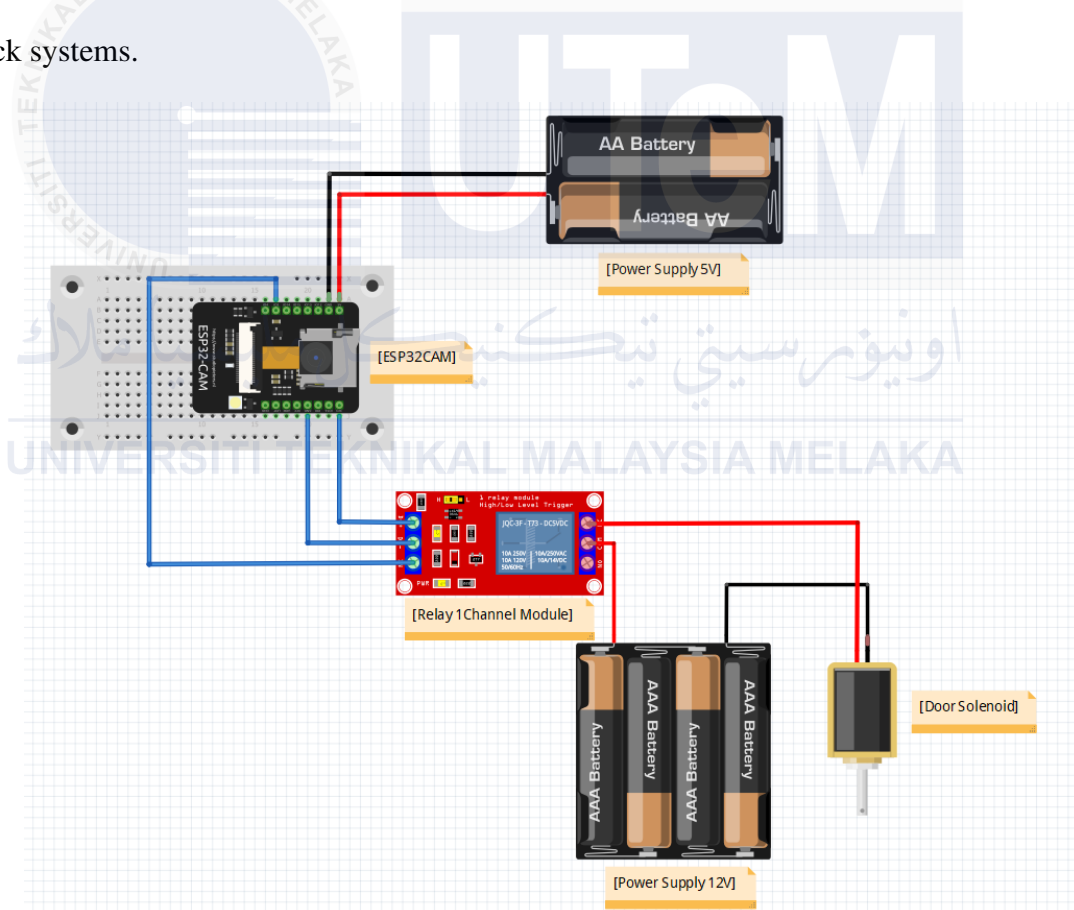


Figure 4.6: Figure shows a schematic diagram for Face Recognition Door Lock.

The system incorporates a 1-channel relay module to serve as an electronic switch. The relay is crucial for enabling the ESP32-CAM to control the high-power solenoid door lock with its low-power output signal. Additionally, the relay provides electrical isolation between the control circuit (powered by the ESP32-CAM) and the power circuit (which operates the solenoid), ensuring the safety and longevity of the components.

Two separate power supplies are used in this setup. A 5V power supply, provided by a pack of AA batteries, powers the ESP32-CAM. Meanwhile, a 12V power supply, derived from 3pin plug adapter, powers the solenoid door lock. This separation of power supplies ensures that each component receives the appropriate voltage for its operation without overloading or damaging the circuit.

When the ESP32-CAM sends a control signal to the relay module, the relay closes its internal switch, allowing the 12V power supply to energize the solenoid. The energized solenoid then actuates, locking or unlocking the door as required. This mechanism allows for precise control of the solenoid door lock, making it suitable for automated or remotely operated systems.

Overall, this design provides a robust solution for smart door lock applications, leveraging the ESP32-CAM's connectivity features and the relay module's switching capabilities. Proper attention to power management and safety precautions will ensure reliable and secure operation of the system.

4.4 Hardware Implementation

The solenoid door system, as seen in the front view, is a central part of the project's hardware implementation. The solenoid lock mechanism is responsible for securing and releasing the door based on input signals. It is electronically controlled and integrated with a microcontroller (ESP32-CAM) to allow for smart locking and unlocking functionalities.



Figure 4.7: Figure of Front View.

The wooden frame provides structural stability to house the lock and other components securely. The door is equipped with a handle for manual operation in case of emergencies or power failure. The solenoid is powered through a dedicated circuit that includes a relay module for effective switching. This setup demonstrates the practical application of automated locking mechanisms in a standalone system.

The back view highlights the ESP32-CAM module mounted on the wooden frame. This microcontroller is the heart of the system, providing both computational power and network communication capabilities. The ESP32-CAM is equipped with a camera that enables live video streaming or image capture, which can be used for visual authentication purposes.



Figure 4.8: Figure of Back View.

This microcontroller is connected to the solenoid lock through a relay module. The inclusion of the ESP32-CAM ensures that the system can support advanced functionalities such as facial recognition, QR code scanning, or remote monitoring. The mounting of the microcontroller is done to ensure secure placement while providing easy access for maintenance.

The inside view of the system showcases a well-organized arrangement of components that supply power and ensure the functionality of the automated door system. Central to this setup is the power supply module, which uses a 12V plug adapter. This module provides the necessary voltage to power the relay module and the solenoid lock. The solenoid lock requires a stable 12V supply to function effectively, while the relay module operates efficiently with this voltage, ensuring secure and precise switching.

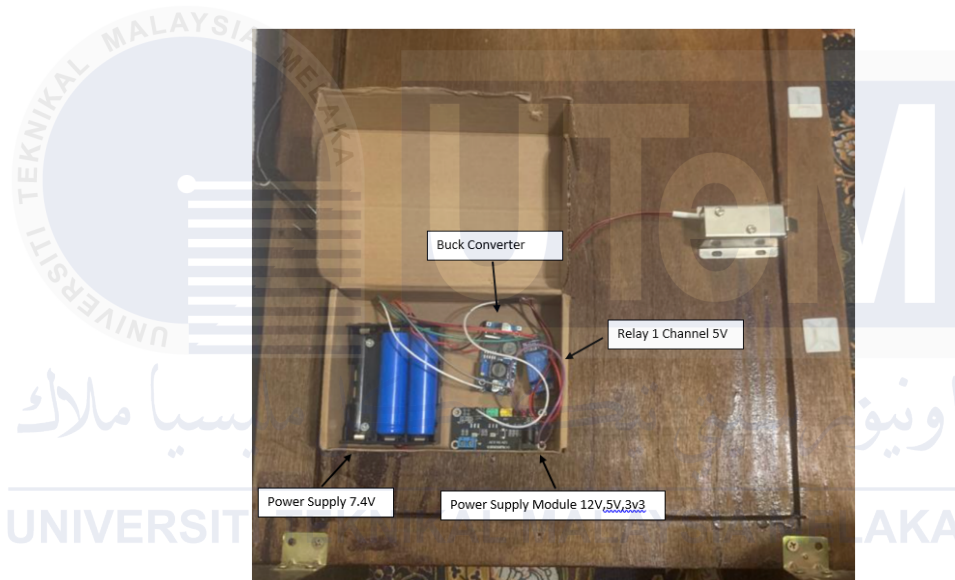


Figure 4.9: Figure of Inside View.

On the other hand, the 7.4V power supply is specifically used to power the ESP32-CAM module. This independent power source ensures that the microcontroller operates reliably, even when the solenoid and relay are active. To optimize the operation of the ESP32-CAM, a buck converter is incorporated into the circuit. The buck converter reduces the 7.4V from the battery pack to a regulated 5V, which is the required operating voltage for the ESP32-CAM. This step-down process not only protects the ESP32-CAM from potential damage due to over-voltage but also ensures energy efficiency.

The combination of these components highlights a thoughtful power management strategy. By using separate power sources and voltage regulation, the system ensures the stable operation of both the ESP32-CAM and the solenoid mechanism, while minimizing interference and power fluctuations. The compact and organized arrangement within the enclosure makes it easy to manage, troubleshoot, and upgrade as needed, emphasizing practicality and reliability in the hardware design.



4.5 Software Implementation

The code provided implements a facial recognition system on an ESP32-based board equipped with a camera module. Below is a breakdown of the code, part by part, along with explanations.

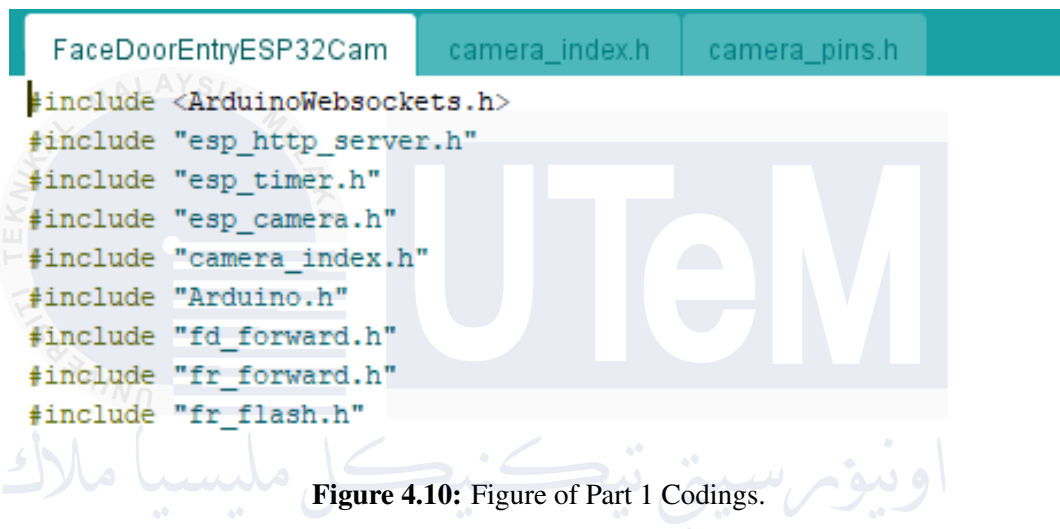


Figure 4.10: Figure of Part 1 Codings.

The code in Figure 4.10, begins by including various libraries and header files that are essential for the functionality of a smart door entry system using the ESP32 with a camera module. These libraries enable specific features like real-time communication, image capturing, and facial recognition.

The `ArduinoWebsockets.h` library is used for establishing WebSocket communication, which allows the ESP32 to send and receive data with a web client in real time. This is important for streaming camera footage or handling commands. The `esp-http-server.h` library provides tools to set up a lightweight HTTP server on the ESP32, allowing users to access the camera stream or control the device through a web interface.

The `esp-timer.h` library helps in tracking time for actions like controlling how long the door remains unlocked after a recognized face. The `esp-camera.h` library is crucial for interfacing with the ESP32 camera module. It handles configuring the camera and capturing images, which are then used for facial detection and recognition.

The `camera-index.h` file likely contains resources for the web interface, such as HTML or JavaScript, that are served by the ESP32 to the client. The `Arduino.h` library provides basic Arduino functions and utilities for input/output operations and general system control.

The other header files, such as `fd-forward.h`, `fr-forward.h`, and `fr-flash.h`, are related to facial detection and recognition. They enable the system to detect faces in camera frames, compare them with stored templates, and manage face data in the ESP32's flash memory. Together, these libraries enable the ESP32 to perform tasks such as detecting faces, recognizing them, and unlocking a door when an authorized face is recognized.

The Figure 4.9 below shows a section of the code sets up the foundational components required for the ESP32 face recognition system. It begins by defining the WiFi credentials (ssid and password) that the device will use to connect to a local network. This connection is crucial for enabling communication with web clients via HTTP and WebSocket protocols.

```
const char* ssid = "POCO X3 Pro";
const char* password = "lsampai8";

#define ENROLL_CONFIRM_TIMES 5
#define FACE_ID_SAVE_NUMBER 7

#define CAMERA_MODEL_AI_THINKER
#include "camera_pins.h"

using namespace websockets;
WebsocketsServer socket_server;

camera_fb_t * fb = NULL;

long current_millis;
long last_detected_millis = 0;

#define relay_pin 2 // pin 12 can also be used
unsigned long door_opened_millis = 0;
long interval = 5000; // open lock for ... milliseconds
bool face_recognised = false;

void app_facenet_main();
void app_httpserver_init();
```

Figure 4.11: Figure of Part 2 Codings.

Key parameters for facial recognition and camera operation are defined next. The constants ENROLL-CONFIRM-TIMES and FACE-ID-SAVE-NUMBER specify the number of samples required for face enrollment and the maximum number of saved face IDs, respectively. The CAMERA-MODEL-AI-THINKER macro indicates the specific camera

model being used, and the corresponding camera-pins.h file is included to configure the camera's hardware pins.

A WebSocket server is initialized using the WebsocketsServer object, allowing real-time communication between the ESP32 and connected clients. A camera frame buffer (fb) is also declared, which will store images captured by the camera for processing.

The code then defines several variables used for time tracking and state management. These include current-millis and last-detected-millis for tracking the time elapsed since the last face detection and door-opened-millis for monitoring how long the door remains unlocked. The interval variable sets the duration (in milliseconds) for which the door stays unlocked, and the face-recognised boolean tracks whether a face has been successfully recognized.

Lastly, two function prototypes, app-facenet-main() and app-httpserver-init(), are declared. These functions will initialize the facial recognition system and the HTTP server, respectively, and their implementations are defined later in the code. This section lays the groundwork for the system's core functionality.

```

typedef struct
{
    uint8_t *image;
    box_array_t *net_boxes;
    dl_matrix3d_t *face_id;
} http_img_process_result;

static inline mtmn_config_t app_mtmn_config()
{
    mtmn_config_t mtmn_config = {0};
    mtmn_config.type = FAST;
    mtmn_config.min_face = 80;
    mtmn_config.pyramid = 0.707;
    mtmn_config.pyramid_times = 4;
    mtmn_config.p_threshold.score = 0.6;
    mtmn_config.p_threshold.nms = 0.7;
    mtmn_config.p_threshold.candidate_number = 20;
    mtmn_config.r_threshold.score = 0.7;
    mtmn_config.r_threshold.nms = 0.7;
    mtmn_config.r_threshold.candidate_number = 10;
    mtmn_config.o_threshold.score = 0.7;
    mtmn_config.o_threshold.nms = 0.7;
    mtmn_config.o_threshold.candidate_number = 1;
    return mtmn_config;
}

mtmn_config_t mtmn_config = app_mtmn_config();

```

Figure 4.12: Figure of Part 3 Codings.

The figure 4.12 above explained the section of the code configures the settings required for detecting faces in the image using the MTCNN (Multi-task Cascaded Convolutional Networks) algorithm. The function `app_mtmn_config()` initializes and returns a configuration structure (`mtmn_config_t`) that defines how face detection should operate. This configuration is used later in the program to guide the face detection algorithm.

The `mtmn_config_t` structure is initialized with default values (set to 0) and then populated with parameters specific to the application. The first key parameter is `type`, which is set to FAST. This option optimizes the detection process for speed, making it suitable for real-time applications, especially on hardware with limited processing power, such as microcontrollers.

To ensure efficient face detection, the configuration defines a minimum detectable face size. The `min_face` parameter is set to 80, meaning that faces smaller than 80 pixels are ignored. This reduces unnecessary computation and focuses on faces likely to provide meaningful results. Additionally, multi-scale detection is enabled through the `pyramid` and `pyramid_times` parameters. The `pyramid` value of 0.707 specifies the scaling factor for each level of the image pyramid, while `pyramid_times` = 4 sets the number of levels. This setup helps detect faces of varying sizes within the same image.

The MTCNN algorithm operates in three stages: Proposal Network (P-Net), Refine Network (R-Net), and Output Network (O-Net). Each stage has its own set of thresholds for confidence scores, non-maximum suppression (NMS), and the maximum number of candidate detections passed to the next stage. For the P-Net stage, the score threshold is set to 0.6, meaning that only face candidates with a confidence score above 0.6 are considered. The NMS threshold, set at 0.7, ensures that overlapping detections are filtered to retain the most likely face candidate. Additionally, the `candidate_number` parameter limits the number of face proposals to a maximum of 20.

Similar configurations are applied to the R-Net and O-Net stages, with progressively higher thresholds to refine the detected face candidates. For instance, the R-Net stage has a

score threshold of 0.7, while the O-Net stage reduces the number of candidates to just one. These adjustments ensure that only the most accurate face detections are finalized.

Finally, the function `app_mtmn_config()` returns the configured `mtmn_config_t` structure, which is then stored in the global variable `mtmn_config`. This setup is essential for guiding the face detection process during runtime, ensuring a balance between speed, accuracy, and computational efficiency.

```
face_id_name_list st_face_list;
static dl_matrix3du_t *aligned_face = NULL;
httpd_handle_t camera_httpd = NULL;

typedef enum
{
    START_STREAM,
    START_DETECT,
    SHOW_FACES,
    START_RECOGNITION,
    START_ENROLL,
    ENROLL_COMPLETE,
    DELETE_ALL,
} en_fsm_state;
en_fsm_state g_state;

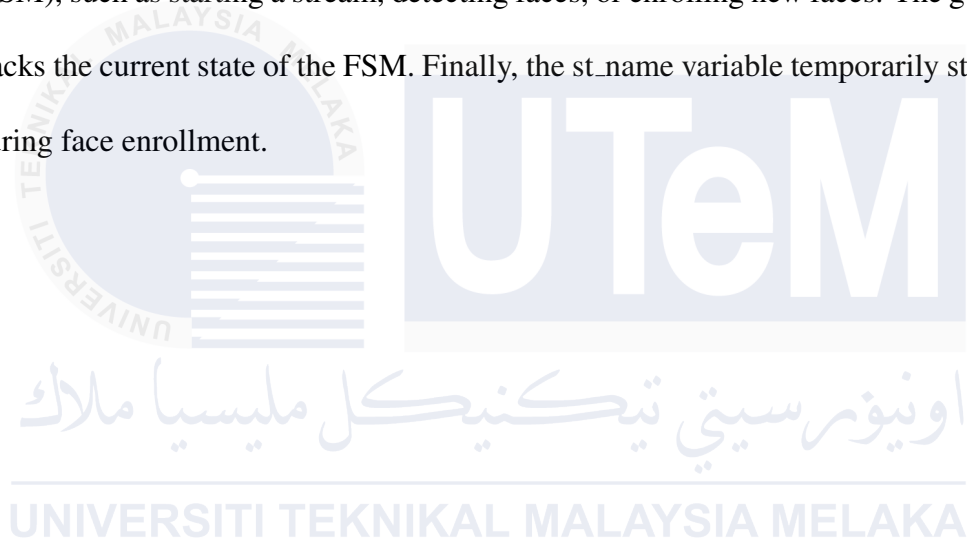
typedef struct
{
    char enroll_name[ENROLL_NAME_LEN];
} httpd_resp_value;

httpd_resp_value st_name;
```

Figure 4.13: Figure of Part 4 Codings.

This section 4.13 above defines variables and functions for managing face recognition and initializing the HTTP server.

The `face_id_name_list` structure is used to store a list of recognized faces with their IDs and names. The `aligned_face` variable holds aligned face data for comparison during recognition. The `en_fsm_state` enumeration defines different states for the finite state machine (FSM), such as starting a stream, detecting faces, or enrolling new faces. The `g_state` variable tracks the current state of the FSM. Finally, the `st_name` variable temporarily stores the name during face enrollment.



```

void setup() {
    Serial.begin(115200);
    Serial.setDebugOutput(true);
    Serial.println();

    digitalWrite(relay_pin, LOW);
    pinMode(relay_pin, OUTPUT);

    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_sscb_sda = SIOD_GPIO_NUM;
    config.pin_sscb_scl = SIOC_GPIO_NUM;
    config.pin_pwdn = PWDN_GPIO_NUM;
    config.pin_reset = RESET_GPIO_NUM;
    config.xclk_freq_hz = 20000000;
    config.pixel_format = PIXFORMAT_JPEG;

    //init with high specs to pre-allocate larger buffers
    if (psramFound()) {
        config.frame_size = FRAMESIZE_UXGA;
        config.jpeg_quality = 10;
        config.fb_count = 2;
    } else {
        config.frame_size = FRAMESIZE_SVGA;
        config.jpeg_quality = 12;
        config.fb_count = 1;
    }
}

```

Figure 4.14: Figure of Part 5 Codings.

```

#if defined(CAMERA_MODEL_ESP_EYE)
    pinMode(13, INPUT_PULLUP);
    pinMode(14, INPUT_PULLUP);
#endif

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}

sensor_t * s = esp_camera_sensor_get();
s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_MSSTACK_WIDE)
    s->set_vflip(s, 1);
    s->set_hmirror(s, 1);
#endif

WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
app_httpserver_init();
app_facenet_main();
socket_server.listen(82);

Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
}

```

Figure 4.15: Figure of Part 6 Codings.

The setup function on Figure 4.14 contains a significant portion dedicated to initializing and configuring the camera hardware. This is done using the `camera_config_t` structure, which specifies various settings required for the camera module to function correctly.

The pin assignments for data signals (pin_d0 to pin_d7), clock signals (pin_xclk,

`pin_pclk`), and synchronization signals (`pin_vsync`, `pin_href`) are configured based on the selected camera model. These assignments ensure proper communication between the ESP32 and the camera module. Additionally, control pins such as `pin_sscb_sda`, `pin_sscb_scl`, `pin_pwdn`, and `pin_reset` are also specified to handle camera-specific control signals.

The `config.pixel_format` is set to `PIXFORMAT_JPEG`, enabling the camera to capture images in JPEG format, which is suitable for efficient storage and transmission. The `config.frame_size` and `config.jpeg_quality` settings are dynamically adjusted based on the availability of PSRAM. If PSRAM is present, the camera captures high-resolution UXGA frames with better JPEG quality (`jpeg_quality = 10`), and two framebuffers are allocated for smoother operation. Without PSRAM, the frame size is reduced to SVGA, the JPEG quality is slightly lowered (`jpeg_quality = 12`), and only one framebuffer is used to save memory.

The `esp_camera_init()` function is then called to initialize the camera with these configurations. If the initialization fails, an error message, including the specific error code, is printed to the serial console, and further execution is halted. This step is crucial to ensure the camera is functional before proceeding with the rest of the program.

After successful initialization, the camera sensor settings are accessed using the function of `esp_camera_sensor_get()`. The frame size is adjusted to `FRAMESIZE_QVGA` to balance image quality and performance, particularly for real-time applications like face detection. For specific models like `CAMERA_MODEL_M5STACK_WIDE`, additional adjustments such as flipping the image vertically (`s- set_vflip(s, 1)`) and mirroring it horizontally (`s- set_hmirror(s, 1)`) are made to ensure proper orientation based on the hardware design.

```

static esp_err_t index_handler(httpd_req_t *req) {
    httpd_resp_set_type(req, "text/html");
    httpd_resp_set_hdr(req, "Content-Encoding", "gzip");
    return httpd_resp_send(req, (const char *)index_ov2640_html_gz, index_ov2640_html_gz_len);
}

httpd_uri_t index_uri = {
    .uri       = "/",
    .method    = HTTP_GET,
    .handler   = index_handler,
    .user_ctx  = NULL
};

void app_httpserver_init ()
{
    httpd_config_t config = HTTPD_DEFAULT_CONFIG();
    if (httpd_start(&camera_httpd, &config) == ESP_OK)
        Serial.println("httpd_start");
    {
        httpd_register_uri_handler(camera_httpd, &index_uri);
    }
}

void app_facenet_main()
{
    face_id_name_init(&st_face_list, FACE_ID_SAVE_NUMBER, ENROLL_CONFIRM_TIMES);
    aligned_face = dl_matrix3du_alloc(1, FACE_WIDTH, FACE_HEIGHT, 3);
    read_face_id_from_flash_with_name(&st_face_list);
}

static inline int do_enrollment(face_id_name_list *face_list, dl_matrix3d_t *new_id)
{
    ESP_LOGD(TAG, "START ENROLLING");
    int left_sample_face = enroll_face_id_to_flash_with_name(face_list, new_id, st_name.enroll_name);
    ESP_LOGD(TAG, "Face ID %s Enrollment: Sample %d",
              st_name.enroll_name,
              ENROLL_CONFIRM_TIMES - left_sample_face);
    return left_sample_face;
}

```

Figure 4.16: Figure of Part 7 Codings.

The code in the provided Figure 4.16 primarily focuses on setting up the HTTP server, initializing the face recognition system, and enabling the enrollment of new faces. Here's a breakdown in paragraphs:

The `index_handler` function is responsible for handling HTTP GET requests for the root URI ("/"). It serves the HTML content that provides the web-based user interface for the ESP32 system. The function sets the response type to "text/html" and specifies that the content is gzip-compressed. It then sends the pre-compressed HTML content (index_ov2640.html_gz) to the client browser. This handler ensures efficient delivery of the interface by leveraging compression.

The `index_uri` structure defines the configuration for the root endpoint of the HTTP server. It specifies the URI path as `"/"`, the HTTP method as `HTTP_GET`, and assigns the `index_handler` function to handle incoming requests. This structure is later registered with the HTTP server to route client requests to the appropriate handler.

The `app_httpserver_init` function initializes the HTTP server. It creates a default server configuration using `HTTPD_DEFAULT_CONFIG()` and starts the server using `httpd_start`. Upon successful startup, a confirmation message ("`httpd_start`") is printed to the serial monitor. The function then registers the `index_uri` handler with the server, enabling the root URI to serve the HTML interface.

Lastly, the `do_enrollment` function handles the enrollment of new faces into the system. It starts by logging a message indicating that enrollment has begun. It then attempts to save a new face ID to flash memory, associating it with the provided name (`st_name.enroll_name`). The function tracks the number of remaining samples needed for successful enrollment and logs the progress. Once all required samples are collected, the enrollment process completes, and the system is updated with the new face ID.

These components collectively enable the ESP32 system to host a web-based interface, perform face detection and recognition, and manage enrolled faces effectively.

```

static esp_err_t send_face_list(WebsocketsClient &client)
{
    client.send("delete_faces"); // tell browser to delete all faces
    face_id_node *head = st_face_list.head;
    char add_face[64];
    for (int i = 0; i < st_face_list.count; i++) // loop current faces
    {
        sprintf(add_face, "listface:%s", head->id_name);
        client.send(add_face); //send face to browser
        head = head->next;
    }
}

static esp_err_t delete_all_faces(WebsocketsClient &client)
{
    delete_face_all_in_flash_with_name(&st_face_list);
    client.send("delete_faces");
}

```

Figure 4.17: Figure of Part 8 Codings.

The `send_face_list` function updates the client with the current list of enrolled faces by first clearing any existing data on the client side using a "delete_faces" message. It then iterates through the system's linked list of faces, formats each face's name, and sends it to the client, ensuring synchronization between the system and the client interface.

The `delete_all_faces` function clears all stored face data from the system's memory and instructs the client to do the same by sending a "delete_faces" message. This ensures both the system and client are reset to a clean state. Together, these functions maintain consistency and synchronization of face data between the system and the client.


```

void handle_message(WebsocketsClient &client, WebsocketsMessage msg)
{
    if (msg.data() == "stream") {
        g_state = START_STREAM;
        client.send("STREAMING");
    }
    if (msg.data() == "detect") {
        g_state = START_DETECT;
        client.send("DETECTING");
    }
    if (msg.data().substring(0, 8) == "capture:") {
        g_state = START_ENROLL;
        char person[FACE_ID_SAVE_NUMBER * ENROLL_NAME_LEN] = {0,};
        msg.data().substring(8).toCharArray(person, sizeof(person));
        memcpy(st_name.enroll_name, person, strlen(person) + 1);
        client.send("CAPTURING");
    }
    if (msg.data() == "recognise") {
        g_state = START_RECOGNITION;
        client.send("RECOGNISING");
    }
    if (msg.data().substring(0, 7) == "remove:") {
        char person[ENROLL_NAME_LEN * FACE_ID_SAVE_NUMBER];
        msg.data().substring(7).toCharArray(person, sizeof(person));
        delete_face_id_in_flash_with_name(&st_face_list, person);
        send_face_list(client); // reset faces in the browser
    }
    if (msg.data() == "delete_all") {
        delete_all_faces(client);
    }
}

```

Figure 4.18: Figure of Part 9 Codings.

The handle_message function processes messages received from a WebSocket client and updates the system's state based on the message's content. Each message triggers a specific action by checking its content and performing the corresponding operation.

When the message contains "stream", the system transitions to a streaming state by setting g_state to START_STREAM and informs the client by sending a "STREAMING"

message. Similarly, if the message contains "detect", the system enters detection mode by setting `g_state` to `START_DETECT` and notifies the client with a "DETECTING" message.

For messages starting with "capture:", the system begins an enrollment process by transitioning to `START_ENROLL` and extracts the person's name from the message after the prefix "capture:". It stores this name in the system's enrollment structure and sends a "CAPTURING" message back to the client to confirm the action.

When the message contains "recognise", the system switches to recognition mode by setting `g_state` to `START_RECOGNITION` and informs the client with a "RECOGNISING" message.

If the message starts with "remove:", the system extracts the name of the person to be removed and deletes their corresponding face data from the system's memory. It also updates the client by invoking the `send_face_list` function to reflect the changes.

Lastly, if the message contains "delete_all", the system clears all stored face data and instructs the client to reset by calling the `delete_all_faces` function. This function ensures the system and client are fully synchronized after clearing all data.

Overall, the `handle_message` function ensures seamless interaction between the system and client by interpreting commands, updating states, and maintaining synchronization.

```

void open_door(WebsocketsClient &client) {
    if (digitalRead(relay_pin) == LOW) {
        digitalWrite(relay_pin, HIGH); //close (energise) relay so door unlocks
        Serial.println("Door Unlocked");
        client.send("door_open");
        door_opened_millis = millis(); // time relay closed and door opened
    }
}

void loop() {
    auto client = socket_server.accept();
    client.onMessage(handle_message);
    dl_matrix3du_t *image_matrix = dl_matrix3du_alloc(1, 320, 240, 3);
    http_img_process_result out_res = {0};
    out_res.image = image_matrix->item;

    send_face_list(client);
    client.send("STREAMING");

    while (client.available()) {
        client.poll();

        // Check if it's time to turn off the relay
        if (face_recognised && millis() - door_opened_millis > interval) {
            digitalWrite(relay_pin, LOW); // Open relay (turn off)
            face_recognised = false; // Reset the flag
            Serial.println("Door Locked");
        }
    }
}

```

Figure 4.19: Figure of Part 10 Codings.

The provided code is part of an ESP32-based system that uses face recognition and WebSocket communication to control a relay, which operates a door-locking mechanism. It integrates real-time image processing with network communication to enable automated access control.

The `open_door` function is responsible for unlocking the door by energizing the relay connected to the `relay_pin`. Before activating the relay, it ensures the relay is currently deactivated (LOW). Once activated (HIGH), it unlocks the door and sends a WebSocket message to the connected client, indicating that the door has been opened. It also records the current time in milliseconds (`door_opened_millis`) to manage the duration for which the relay

remains energized.

The loop function handles the main operations of the system. It establishes WebSocket connections by accepting a client and assigning the `handle_message` function to process incoming messages. An image matrix is allocated for processing camera frames, and the system begins by sending the list of enrolled faces to the client and starting the video stream. Within the loop, it continuously polls for WebSocket messages and checks for face recognition events. When a face is recognized, the system activates the relay to unlock the door and sets a flag (`face_recognised`). After a predefined interval, the relay is deactivated (locking the door), and the flag is reset to prepare for the next recognition event.

The code ensures secure access by combining face recognition with real-time control of the relay. It leverages the ESP32's capabilities to process camera frames, detect faces, and manage WebSocket communication. The modular approach of separating the door control (`open_door`) and message handling logic enhances readability and maintainability, while the integration of timing mechanisms ensures that the door locks automatically after the specified duration.

```

fb = esp_camera_fb_get();

if (g_state == START_DETECT || g_state == START_ENROLL || g_state == START_RECOGNITION)
{
    out_res.net_boxes = NULL;
    out_res.face_id = NULL;

    fmt2rgb888(fb->buf, fb->len, fb->format, out_res.image);

    out_res.net_boxes = face_detect(image_matrix, &mtmn_config);

    if (out_res.net_boxes)
    {
        if (align_face(out_res.net_boxes, image_matrix, aligned_face) == ESP_OK)
        {
            out_res.face_id = get_face_id(aligned_face);
            last_detected_millis = millis();

            if (g_state == START_DETECT) {
                client.send("FACE DETECTED");
            }

            if (g_state == START_ENROLL)
            {
                int left_sample_face = do_enrollment(&st_face_list, out_res.face_id);
                char enrolling_message[64];
                sprintf(enrolling_message, "SAMPLE NUMBER %d FOR %s", ENROLL_CONFIRM_TIMES - left_sample_face, st_name.enroll_name);
                client.send(enrolling_message);
                if (left_sample_face == 0)
                {
                    ESP_LOGI(TAG, "Enrolled Face ID: %s", st_face_list.tail->id_name);
                    g_state = START_STREAM;
                    char captured_message[64];
                    sprintf(captured_message, "FACE CAPTURED FOR %s", st_face_list.tail->id_name);
                    client.send(captured_message);
                    send_face_list(client);
                }
            }
        }
    }
}

```

Figure 4.20: Figure of Part 11 Codings.

The provided code in Figure 4.20 is part of a face detection and recognition system that processes images captured from a camera to detect, enroll, and recognize faces. It utilizes the ESP32 camera module's capabilities for real-time image processing.

The step involves capturing a frame buffer (fb) using the `esp_camera_fb_get()` function. If the current state (`g_state`) indicates that the system should perform detection, enrollment, or recognition, it initializes the `out_res.net_boxes` and `out_res.face_id` variables to NULL to store the results of face detection and recognition. The frame buffer is then converted from its current format to an RGB888 format using the `fmt2rgb888` function, enabling further image processing.

Face detection is performed using the `face_detect` function, which processes the RGB image using a face detection model. If faces are detected, the bounding boxes are aligned using the `align_face` function to standardize the detected face's orientation. Once aligned, the system generates a face ID using the `get_face_id` function, which extracts unique features of the detected face for identification purposes. The current timestamp is recorded in `last_detected_millis` to track when the face was last detected.

If the system is in the detection state (`START_DETECT`), it sends a message to the client indicating that a face has been detected. In the enrollment state (`START_ENROLL`), the detected face is enrolled by calling the `do_enrollment` function, which stores the face ID in memory along with a user-provided name. Feedback is sent to the client regarding the enrollment progress, including the number of remaining samples required for successful enrollment. Once enrollment is complete, the system transitions to the streaming state (`START_STREAM`) and sends a message confirming that the face has been successfully captured and enrolled.

```

if (g_state == START_RECOGNITION && (st_face_list.count > 0))
{
    face_id_node *f = recognize_face_with_name(&st_face_list, out_res.face_id);
    if (f)
    {
        char recognised_message[64];
        sprintf(recognised_message, "DOOR OPEN FOR %s", f->id_name);
        open_door(client);
        client.send(recognised_message);
        face_recognised = true; // Set flag when face is recognized
    }
    else
    {
        client.send("FACE NOT RECOGNISED");
    }
}
dl_matrix3d_free(out_res.face_id);
}
else
{
    if (g_state != START_DETECT) {
        client.send("NO FACE DETECTED");
    }
}

if (g_state == START_DETECT && millis() - last_detected_millis > 500) { // Detecting but no face detected
    client.send("DETECTING");
}

client.sendBinary((const char *)fb->buf, fb->len);

esp_camera_fb_return(fb);
fb = NULL;
}
}

```

Figure 4.21: Figure of Part 12 Codings.

This section of the code handles the recognition of faces and associated actions within the system. The primary function is to verify if a detected face matches any enrolled face in the database and perform specific actions based on the result.

When the system is in the recognition state (START_RECOGNITION) and there are faces in the enrolled face list (st_face_list.count > 0), the recognize_face_with_name function is called. This function attempts to match the detected face (out_res.face_id) with the stored face list (st_face_list). If a match is found, a message is prepared using sprintf to notify the client that the door is opening for the recognized user. The open_door function is then invoked to

perform the associated action (e.g., unlocking a door), and the message is sent to the client. Additionally, a flag `face_recognised` is set to true to indicate successful recognition.

If no match is found during recognition, a message indicating "FACE NOT RECOGNISED" is sent to the client. Regardless of whether a match is found, the face ID matrix (`out_res.face_id`) is freed using `dl_matrix3d_free` to release memory resources.

If the system is not in the recognition state, it checks whether the state is `START_DETECT`. If no faces are detected in this state, a message "NO FACE DETECTED" is sent to the client. Additionally, the code checks if the time elapsed since the last detected face exceeds 500 milliseconds. If so, it sends a "DETECTING" message to notify the client that the system is still scanning for faces.

At the end of the processing loop, the frame buffer (`fb`) is sent to the client as binary data using `client.sendBinary`. After sending, the frame buffer is returned to the camera buffer pool using `esp_camera_fb_return`, and the pointer `fb` is set to `NULL` to ensure proper memory management. This structure ensures efficient real-time face recognition and feedback to the client while maintaining system stability.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

This project looks at the methods proposed and applied in this particular project to show their effectiveness. Engaging with the identified project, it was possible to identify three main objectives, all of which have been achieved.

Firstly, a theoretical basis for facial recognition using ESP32-CAM was built and the respective system was designed. This entailed deploying the appropriate hardware and installation of reasonable reliable systems. This work was done by achieving the stages indicated in the flowchart and the outcome is that secure access control has been attained.

Then, the software part was designed in the matrix of the Arduino Integrated Development Environment. This involved the introduction of new codes on the microcontroller of the system as well as optimizing the system functions. There are numerous resources and tools included in the Arduino Integrated designing environment that helped in coding and debugging a lot.

Finally, a performance study and an evaluation of the system were made. This included evaluating the credibility of the system as well as seeking to determine the extent of effectiveness without yielding a percent accuracy. The obtained outcome meets the project's requirements as their ability to predict will be high along with stringent security measures.

Summing up, the project has successfully designed a facial recognition smart door lock system and has more scope and goals for further improvement and marketing.



5.2 Future Works

Several methods can be used to improve the face recognition door lock system in the future by using the ESP32-CAM process outcome:

Installation of an Alarm Notification System: It is essential to put in place a notification system that will instantly notify security staff or designated individuals of any attempts to gain unauthorized entry. You can use push notifications, email, or SMS to accomplish this. Integration with current security surveillance systems should be taken into consideration to provide a coordinated reaction. [16] [17]

Improvement of User-Friendliness: User-friendliness and intuitive operation should be given top priority in the design of the keypad and RFID interface. Enhancing usability can be accomplished by using aural signals, straightforward navigation, and clear labeling. Support for many languages, voice prompts, and illuminated keys are some features that should be incorporated to accommodate users with varying degrees of technical expertise.[18]

Implementing Emergency Power Backup: Having a backup power source is crucial to ensuring that the door lock system operates continuously and unhindered, particularly in the event of an emergency. By serving as a buffer against future power disturbances or outages, this backup power source keeps the system operating around-the-clock.

Adoption of Energy Efficiency Measures: Increasing power efficiency is essential to the system's reliable and long-lasting operation. Energy usage may be greatly decreased by using low power modes, such as sleeping devices like the ESP32-CAM while not in use. Overall power consumption may be reduced by using motion or infrared sensors to

only activate the system when necessary and by choosing energy-efficient hardware, such as ESP32 microcontrollers. Because of these improvements, the system's battery life will be extended, making it appropriate for areas with spotty electrical supply.

5.3 Project Commercialization

Throughout the process of commercializing the ESP32-CAM facial recognition door lock system, it is imperative to undertake a series of carefully planned actions to ensure successful market penetration and sustainable viability. This section outlines the key stages involved in transforming this innovative technology from a mere prototype into a commercially feasible product.

5.3.1 Examination of the Market

A comprehensive market study is the initial step in the commercialization process to identify potential customers, market size, and competition. The demand for enhanced security solutions in institutional, commercial, and residential settings is increasing, indicating a significant market opportunity. Key market segments include smart home enthusiasts, security-conscious individuals, and companies or organizations looking to upgrade their access control systems. Analyzing the strength, weakness, and market position of existing face recognition and IoT-based security solution should be the primary focus of competitor analysis.

5.3.2 Product Development

In order to ensure durability, ease of use, and reliability, it is imperative to enhance the prototype through further refinement. Key development endeavors encompass the following:

- **Enhanced User Interface:** Developing a user-friendly interface with functions like voice guidance, multilingual support, and illuminated keys for the keypad and RFID system.
- **Energy Efficiency:** Utilizing power-saving methods, such as backup power options and low-power modes, to extend battery life and ensure uninterrupted operation.
- **Alert System:** Implementing an alarm notification system that delivers push notifications, emails, or SMS alerts to users or security personnel in case of unauthorized access attempts. [19] [20]

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

REFERENCES

- [1] D. A. Wangean, S. Setyawan, F. I. Maulana, G. Pangestu, and C. Huda, "Development of Real-Time Face Recognition for Smart Door Lock Security System using Haar Cascade and OpenCV LBPH Face Recognizer," in *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, 2023, pp. 506–510.
- [2] T. G. Reddy, S. C. Sai, B. P. Kumar, R. T. Venkatesh, K. Sathwik, and K. Singh, "Face Recognition Door Lock System Using Raspberry Pi," in *2023 Third International Conference on Secure Cyber Computing and Communication (ICSCCC)*, 2023, pp. 218–221.
- [3] A. Saroha, A. Gupta, A. Bhargava, A. K. Mandpura, and H. Singh, "Biometric Authentication Based Automated, Secure, and Smart IOT Door Lock System," in *2022 IEEE India Council International Subsections Conference (INDISCON)*, 2022, pp. 1–5.
- [4] O. B. Doshi, H. N. Bendale, A. M. Chavan, and S. S. More, "A Smart Door Lock Security System using Internet of Things," in *2022 International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 2022, pp. 1457–1463.
- [5] R. Sivaprasad, C. Yazhini, G. S. Harini, J. Jayashree, and K. Prathibanandhi, "Automatic Door Locking System in Households Using IoT," in *2023 Intelligent Computing and Control for Engineering and Business Systems (ICCEBS)*, 2023, pp. 1–4.

- [6] L. Nabila, W. Priharti, and Istiqomah, "Design of Home Security System Using Face Recognition with Convolutional Neural Network Method," in *2022 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, 2022, pp. 78–83.
- [7] G. Orna, D. S. Benítez, and N. Pérez, "A Low-Cost Embedded Facial Recognition System for Door Access Control using Deep Learning," in *2020 IEEE ANDESCON*, 2020, pp. 1–6.
- [8] M. B. H, J. S, K. D. M, L. C. Govindapillai, and J. C. J. R, "Face Recognition Door Lock System Using Raspberry Pi, year=2022," in *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 1645–1648.
- [9] G. Surla, S. Manepalli, N. A. Shaik, and N. Saritha Gurram, "IoT and Face Recognition based Automated Door Lock System, year=2023," in *2023 Second International Conference on Electronics and Renewable Systems (ICEARS)*, pp. 648–651.
- [10] N. R. S, R. Venkatasamy, J. A. Dhanraj, S. Aravinth, K. Balachandar, and D. N, "Design and Development of IOT based Smart Door Lock System," in *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICI-CICT)*, 2022, pp. 1525–1528.
- [11] M. Y. S. Krishna, A. Arya, S. Ansari, S. Awasya, J. Sushakar, and N. Uikey, "Real Time Door Unlocking System using Facial Biometrics based on IoT and Python," in *2023 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, 2023, pp. 1–5.

- [12] M. Yashashwini, K. S. Kumar, R. Pitchai, K. S. Sai Sankeerth, G. Arun Prasath, and D. Trinath, "Face Recognition Based Smart Door Lock System using Convolution Neural Network," in *2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMK-MATE)*, 2023, pp. 1–6.
- [13] S. Sharma, M. Sharma, G. Sharma, and A. Bhasney, "IoT-Enabled Smart Door Lock System Using Temperature Sensor," in *2024 2nd International Conference on Disruptive Technologies (ICDT)*, 2024, pp. 360–365.
- [14] M. Hemalatha, J. S. Priya, J. R. P. S, T. Porselvi, and S. S, "An Intelligent Authentication System for Improved Security," in *2022 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, 2022, pp. 1–5.
- [15] S. Bhatlawande, S. Shilaskar, T. Gadad, S. Ghulaxe, and R. Gaikwad, "Smart Home Security Monitoring System based on Face Recognition and Android Application," in *2023 International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, 2023, pp. 222–227.
- [16] S. Jahnavi and C. Nandini, "Smart Anti-Theft Door locking System," in *2019 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE)*, 2019, pp. 205–208.
- [17] V. Pandit, P. Majgaonkar, P. Meher, S. Sapaliga, and S. Bojewar, "Intelligent security lock," in *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, 2017, pp. 713–716.

- [18] S. K. Tilli, N. A. Ibrahim, S. Thomas, S. Isah, U. Yahaya, and A. N. Obadiah, "Design and Construction of Secure Door Lock System Using RFID Authentication," in *2023 2nd International Conference on Multidisciplinary Engineering and Applied Science (ICMEAS)*, 2023, pp. 1–5.
- [19] S. Senthilkumar, E. Saranya, M. Kavitha, A. Selvakumar, S. Rajasri, and R. Ramiya, "A Novel and Smart Administrative Door Lock and Open System using Face Recognition," in *2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, 2023, pp. 916–922.
- [20] R. Priyakanth, N. S. Krishna, G. Karanam, M. L. Prassna, S. Baby Poojitha, and J. Mounika, "IoT Based Smart Door Unlock and Intruder Alert System," in *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, 2021, pp. 6–11.