Innovation and Design of Enhanced Home Security Using Face Recognition and Real-Time Intruder Alerts

by

Kanij Rejwana Tamim
ID: CSE2102023058

Suma Akter
ID: CSE1903018050

Mymona Akter Shoa
ID: CSE2102023097

Supervised by
Salma Tabashum

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

September 2023

Innovation and Design of Enhanced Home Security Using Face Recognition and Real-Time Intruder Alerts

by

Kanij Rejwana Tamim
ID: CSE2102023058

Suma Akter
ID: CSE1903018050

Mymona Akter Shoa
ID: CSE2102023097

Supervised by
Salma Tabashum

Submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

September 2023

APPROVAL

The thesis titled **"Innovation and Design of Enhanced Home Security Using Face Recognition and Real-Time Intruder Alerts"** submitted by **Kanij Rejwana Tamim** (CSE2102023058), **Suma Akter** (CSE1903018050), and **Mymona Akter Shoa** (CSE2102023097) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

Salma Tabashum — **Supervisor**
Lecturer & Asst. Coordinator
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 1
(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 2
(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

Examiner 3
(Examiner Name and Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Salma Tabashum,** Lecturer & Asst. Coordinator**,** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this thesis has been or is being submitted elsewhere for the award of any degree.

Countersigned

(Salma Tabashum)
Supervisor

Signature

Kanij Rejwana Tamim
ID: CSE2102023058

Suma Akter
ID: CSE1903018050

Mymona Akter Shoa
ID: CSE2102023097

ABSTRACT

Safety is a fundamental concern in today's world, especially when it comes to protecting our homes from unauthorized access and potential threats. With the increasing need for advanced security systems, traditional locks and keys no longer offer the level of protection required to safeguard personal property. To address these challenges, this project focuses on enhancing home security using face recognition technology combined with real-time intruder alerts, creating an automated, efficient, and secure solution. The system is designed to automatically identify authorized individuals and alert homeowners of potential intruders, offering both convenience and safety. The project utilizes a combination of cutting-edge hardware and software components to achieve this goal. The hardware includes an ESP32 Cam, which captures and processes facial data, and an SG90 servo motor that controls the door's locking mechanism based on facial recognition. To provide instant feedback and notifications, a buzzer is used to alert the homeowner in case of unauthorized access. The system's connectivity is managed via an FTDI Adapter, and there are reset switches for Wi-Fi and system management. The technical approach includes a backend powered by FastAPI for server-side logic and SQLite managed through Prisma ORM to store and manage facial recognition data. The face recognition Python library, with an accuracy of 97%, is used for identifying faces and granting access. Additionally, the system employs machine learning (ML) and deep learning (DL) techniques to improve recognition accuracy under various conditions. When someone approaches the door, the system attempts to recognize the individual's face. If the homeowner's face is detected, the door unlocks automatically within 3-10 seconds. In the event of an unrecognized face, the system triggers an alarm to alert the homeowner of a potential security threat. This innovative combination of technologies ensures a high level of security and convenience, revolutionizing home protection.

ACKNOWLEDGMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the schedule time. We are auspicious that we had the kind association as well as supervision of **Brig. Gen. (Retd) Prof. Habibur Rahman Kamal, ndc, psc,** Dean**,** Faculty of Science & Engineering, **Prof. Bulbul Ahamed**, Head, Department of Computer Science & Engineering and **Salma Tabashum,** Lecturer & Asst. Coordinator, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

Finally, our deepest gratitude and love to my parents for their support, encouragement, and endless love.

LIST OF ABBREVIATIONS

AC	Access Control
ASS	Automated Security System
BAS	Buzzer Alert System
DL	Deep Learning
ESP	ESP32 Cam
FB	FastAPI Backend
FRT	Face Recognition Technology
FSL	FastAPI Server-side Logic
HS	Home Security
HP	Home Protection
IA	Intruder Alerts
ID	Intruder Detection
ML	Machine Learning
PFR	Python Face Recognition
PORM	Prisma ORM
RTN	Real-time Notifications
SG90	SG90 Servo Motor
SLM	Servo-controlled Locking Mechanism
SQLD	SQLite Database
UA	Unauthorized Access
WFC	Wi-Fi Connectivity

TABLE OF CONTENTS

Title			Page No.
DECLARATION ……..…….			iii
ABSTRACT ……...…….			iv
ACKNOWLEDGEMENT ……..……			v
LIST OF ABBREVIATION …….……			vi
CHAPTER 1			1 – 6
INTRODUCTION			
1.1	Introduction.....		1
1.2	Problem Statement.....		2
1.3	Objectives of Project Work.....		3
	1.3.1	Development of an Automated Security System.....	3
	1.3.2	Integration of Facial Recognition Technology.....	3
	1.3.4	Real-Time Access Control and Feedback.....	3
	1.3.4	Alert System for Unauthorized Access.....	3
	1.3.5	Improvement of Recognition with Machine Learning.....	3
	1.3.6	Utilization of IoT Components for Connectivity.....	4
	1.3.7	Backend Development with Efficient Data Management.....	4
	1.3.8	Enhancement of User Experience.....	4
	1.3.9	Testing and Validation of System Performance.....	4
	1.3.10	Exploration of Future Enhancements.....	4
Ss			
CHAPTER 2			5 – 6
LITERATURE REVIEW			
2.1	Evolution of Home Security Systems.....		5
2.2	Facial Recognition Technology in Security.....		5
2.3	Machine Learning and Deep Learning in Face Recognition.....		6
2.4	Real-Time Alerts and Notification Systems.....		6
2.5	Challenges and Future Trends.....		6
CHAPTER 3			7 – 17
SYSTEM DESIGN AND ARCHITECTURE			
3.1	Overview of System Architecture.....		7

3.1.1 Hardware Components..... 7
3.1.2 Software Components..... 7
3.2.3 System Workflow..... 8
3.1.4 User Interface (UI) 8
3.2 Backend Architecture..... 8
3.2.1 Overview of Backend Components..... 8
3.2.2 Backend Workflow..... 9
3.2.3 Security and Authentication..... 11
3.3 Database Design..... 11
3.3.1 Database Tables and Structure..... 12
3.3.2 Relationships Between Tables..... 14
3.3.3 Example Data Entries..... 14
3.3.4 Data Security Considerations..... 15
3.3.5 Schema Design..... 16
3.3.6 Data Flow and Access Patterns..... 16
3.3.7 Database Security Considerations..... 17

CHAPTER 4 18 - 27

IMPLEMENTATION DETAILS

4.1 Development Environment Setup..... 18
4.2 Hardware Setup..... 19
4.3 Software Setup..... 21
4.4 Development Tools..... 23
4.5 Connectivity and Integration..... 23
4.6 System Configuration..... 24
4.7 Testing and debugging..... 25
4.8 Security Considerations..... 27

CHAPTER 5 29-36

KEY FEATURES OF HOUSE SECURITY

5.1 Smart Access Control..... 29
5.2 Overview of Database Design..... 31
5.3 Video Surveillance Integration..... 32
5.4 Smart Alarms and Sirens..... 33

5.5 Smart Home Automation Integration..... 34
5.6 Access Logs and Activity Monitoring..... 36

CHAPTER 6 38-41
TESTING AND VALIDATION
6.1 Hardware Testing..... 38
6.2 Integration Testing..... 39
6.3 Testing Scenarios..... 39
6.4 Response Time Testing..... 40
6.5 Performance Benchmarking..... 41

CHAPTER 7 43 - 44
CHALLENGES AND LIMITATIONS
7.1 Technical Challenges Faced..... 43
7.2 Face Recognition Accuracy and Reliability..... 43
7.3 Hardware Integration and Communication..... 43
7.4 Real-Time Processing and Speed..... 44
7.5 Environmental Factors Affecting Recognition..... 44
7.6 Security and Privacy Concerns..... 44
7.7 Power Consumption..... 44

CHAPTER 8 45-47
CONCLUSION AND FUTURE WORK
9.1 Conclusion..... 45
9.2 Contribution to the Rental Market 45
9.3 Future Enhancements and Scalability..... 46

REFERENCES ……..... 47

LIST OF FIGURES

Figure No.	Title	Page No.
Fig.1.1	Reset Switches	8
Fig.3.2.1	Backend Architecture.	10
Fig.3.3.1	Simplified Entity-Relationship Diagram.	15
Fig.4.1	Development Environment Process.	18
Fig.4.1.1	ESP32 Cam.	19
Fig.4.1.2	SG90 Servo Motor	20
Fig.4.1.3	buzzer is an essential component in the system's alert mechanism.	20
Fig.4.1.4	FTDI Adapter.	21
Fig.5.1	Hardware Components and Software Components system.	31

LIST OF TABLES

Table No.	Title	Page No.
Table.1	User Table for personal details, and contact information.	12
Table.2	Access User Login Table.	13
Table.3	System Configuration Table.	13
Table.4	Notification History Table.	13
Table.5	Farcical Recognition Data Table.	13
Table.6	For example, user data table.	14
Table.7	Access Logs Data Information Table.	15
Table.8	System Configuration Table.	15
Table.9	Notification History Table.	15
Table.10	Finical Data Information Table.	15

CHAPTER 1

INTRODUCTION

1.1 Introduction

In today's rapidly evolving world, the safety and security of our homes and personal spaces have become a growing concern. Traditional security measures, such as physical locks, keys, and even simple alarm systems, often fail to provide the level of protection necessary in modern society. With an increasing number of sophisticated threats, unauthorized access to homes is an ongoing issue that traditional methods can no longer address effectively. This has led to the development of more advanced security systems, incorporating modern technologies such as biometrics, artificial intelligence, and machine learning [1].

One such technology that has gained considerable attention in recent years is facial recognition. Facial recognition is a highly accurate and non-intrusive method for identifying individuals, making it ideal for applications in home security. Unlike traditional systems that rely on keys, access cards, or PIN codes methods that can be lost, stolen, or forgotten facial recognition offers a contactless and reliable solution. With the potential to not only identify authorized individuals but also detect and alert homeowners to the presence of potential intruders, this technology is poised to revolutionize home security systems.

This project explores the design and development of a home security system that integrates facial recognition technology to automate the process of identifying and granting access to authorized individuals. The system is designed to enhance convenience, increase security, and provide real-time alerts in the event of unauthorized access. By leveraging cutting-edge hardware and software, the system aims to provide a seamless user experience while ensuring maximum safety for homeowners. At the heart of this system is an ESP32 Cam, a compact and powerful device capable of capturing and processing facial data. The SG90 servo motor acts as the actuator that controls the door’s locking mechanism, ensuring that access is granted only to recognized individuals. To further enhance the security, a buzzer is incorporated to alert homeowners if an unrecognized face is detected, prompting immediate action. The system is connected via Wi-Fi, enabling real-time communication and notifications to the homeowner [1].

The backend of the system is powered by FastAPI, a modern web framework that allows for efficient server-side logic and data handling. SQLite is utilized for managing the facial recognition data, stored and accessed via Prisma ORM, which simplifies database interactions and ensures data integrity. The system utilizes an advanced face recognition Python library, which boasts an impressive 97% accuracy rate for identifying faces under various conditions. The system’s ability to recognize and authenticate faces in real time is further enhanced by the use of machine learning (ML) and deep learning (DL) techniques, which continuously improve the accuracy of the recognition process over time, adapting to various lighting, angles, and environmental conditions. The user experience is central to the design of this system. When a person approaches the door, the system immediately detects

and attempts to recognize their face. If the face matches that of an authorized user, the system grants access by unlocking the door within a few seconds. In the event that an unauthorized individual is detected, the system triggers an alarm and notifies the homeowner, ensuring that potential security threats are quickly addressed. The real-time nature of the alerts enables homeowners to take immediate action, whether that involves remotely locking the door, contacting authorities, or activating additional security measures.

This project not only addresses the need for a more secure and automated home entry system but also demonstrates the power of integrating advanced technologies such as facial recognition, machine learning, and IoT (Internet of Things) devices. It represents a significant step toward modernizing home security systems, providing a high level of protection while enhancing the user experience. Through this innovative combination of hardware and software, the system offers a future-proof solution to the growing security challenges faced by homeowners in an increasingly digital world [1, 2].

By providing a solution that can accurately and reliably recognize faces and offer immediate feedback, this project introduces a new standard for home security systems, one that is more efficient, intelligent, and user-friendly than traditional alternatives. The integration of real-time alerts and automated door control ensures both convenience and peace of mind, making this system a valuable addition to any home.

1.2 Problem Statement

As the world becomes more connected through the Internet of Things (IoT), ensuring the safety and security of personal spaces is becoming increasingly complex. Traditional home security systems, including manual locks, keys, and simple alarm systems, are often inadequate in the face of modern security threats. These outdated solutions are prone to being bypassed or compromised, whether through the loss or theft of keys, the vulnerability of lock-picking methods, or the limited effectiveness of basic alarm systems. Furthermore, these systems often lack the intelligence to distinguish between authorized users and intruders, making it difficult for homeowners to prevent unauthorized access in real time.

A major limitation of existing security systems is the reliance on physical authentication methods, such as keys, passwords, or PIN codes. These methods are inherently flawed; keys can be lost or stolen, passwords can be forgotten or hacked, and PIN codes can be shared or observed. Additionally, many home security systems rely on cameras for surveillance, but these cameras only provide passive monitoring without actively preventing access or notifying homeowners of potential threats. In many cases, even when an intrusion is detected, response times are delayed, leading to potential security breaches [1].

The absence of a real-time, automated system that can identify individuals based on their facial features presents a significant gap in current security solutions. With the growing demand for more sophisticated and efficient security measures, the need for a system that can not only authenticate users but also provide instant alerts when unauthorized access is attempted is clear.

The primary challenge addressed by this project is the development of an intelligent, automated home security system that combines face recognition technology with real-time alerts to effectively manage access control and ensure prompt security responses. By integrating facial recognition, machine learning, and IoT hardware, the system aims to provide a robust solution that enhances both security and convenience for homeowners.

This project addresses the critical need for a security system that:

- Can automatically identify authorized individuals through facial recognition.
- Alerts homeowners immediately in the event of an intruder or unauthorized access attempt.
- Eliminates the reliance on traditional methods such as physical keys and passwords.
- Provides a real-time, automated, and highly accurate system that can learn and adapt to different conditions.

By solving these problems, the project seeks to revolutionize home security by offering a more intelligent, efficient, and reliable solution that not only improves security but also enhances the overall user experience.

1.3 Objectives of Project Work

The primary objective of this project is to design and implement an advanced, automated home security system that leverages facial recognition technology to enhance both the security and convenience of homeowners. The system aims to address the shortcomings of traditional security solutions by integrating real-time identification, access control, and immediate alerts in case of unauthorized access. The key objectives of this project include:

1.3.1 Development of an Automated Security System

- To create a fully automated system capable of identifying authorized individuals through facial recognition, granting or denying access accordingly, and triggering immediate alerts for unauthorized access attempts.

1.3.2 Integration of Facial Recognition Technology

- To implement an efficient and reliable facial recognition system with an accuracy of 97% or higher, capable of detecting and recognizing faces in real time under various conditions, such as different lighting or angles.

1.3.3. Real-Time Access Control and Feedback

- To design a system that enables instantaneous access control through facial recognition, where the door unlocks automatically for authorized individuals within 3-10 seconds, and a buzzer triggers an alert for unauthorized individuals.

1.3.4 Alert System for Unauthorized Access

- To incorporate a real-time alert system that notifies homeowners immediately when an unauthorized person attempts access, ensuring prompt security response and peace of mind.

1.3.5 Improvement of Recognition with Machine Learning

- To enhance the face recognition process using machine learning and deep learning techniques, allowing the system to adapt and improve over time based on real-world usage, environmental conditions, and different facial appearances.

1.3.6 Utilization of IoT Components for Connectivity

- To ensure reliable communication and management of the security system by integrating IoT devices such as the ESP32 Cam, SG90 servo motor, and Wi-Fi connectivity, ensuring seamless operation across different devices.

1.3.7 Backend Development with Efficient Data Management

- To build a robust backend system using FastAPI for server-side logic and SQLite for database management, ensuring smooth storage, retrieval, and updating of facial recognition data with the help of Prisma ORM.

1.3.8 Enhancement of User Experience

- To design a user-friendly interface that allows homeowners to easily manage their security system, configure settings, and view notifications, offering convenience without sacrificing security.

1.3.9 Testing and Validation of System Performance

- To conduct comprehensive testing and validation of the system to ensure its accuracy, speed, and reliability under real-world conditions, including various environmental factors and face recognition challenges.

1.3.10 Exploration of Future Enhancements

- To identify potential areas for future upgrades, such as the integration of additional biometric features, improved recognition algorithms, or the ability to integrate with other smart home devices for a fully integrated security solution.

Through these objectives, this project aims to create a state-of-the-art home security system that not only addresses current security challenges but also offers scalability and flexibility for future innovations in the field of automated home protection.

CHAPTER 2
LITERATURE REVIEW

In recent years, the demand for advanced security systems has grown substantially, driven by increasing concerns about safety, the advancement of technology, and the desire for more automated and efficient solutions. Traditional security systems, such as mechanical locks, alarm systems, and manual access control, are being gradually replaced by more intelligent, automated systems that leverage modern technologies like biometrics, machine learning, and the Internet of Things (IoT). This chapter explores existing research, technologies, and systems relevant to the development of home security systems based on facial recognition, providing a foundation for the project.

2.1 Evolution of Home Security Systems

Traditional home security systems have primarily relied on physical locks, keys, and alarm systems. These solutions are reactive rather than proactive, offering limited protection against unauthorized access. As technology has advanced, the demand for smarter and more secure systems has increased. The integration of smart technologies, such as smart locks, automated alarms, and surveillance cameras, has led to the development of smart home security systems. However, many of these solutions are still vulnerable to security breaches, particularly when access methods, such as keys or codes, are compromised. Thus, there has been a clear need for systems that can identify individuals and offer higher levels of security and automation [1].

2.2 Facial Recognition Technology in Security

Facial recognition technology has emerged as one of the most effective methods for personal identification and access control. According to a study by Jain et al. (2008), facial recognition systems work by analyzing distinct facial features, such as the distance between eyes, nose shape, and chin contour, to create a unique biometric template for each individual. This biometric method has become a key component in various security applications due to its non-intrusiveness, accuracy, and ease of implementation compared to other biometric systems like fingerprint recognition or iris scanning.

Kusuma et al. (2020) demonstrated that facial recognition technology can significantly improve the security of access control systems by providing a more reliable and secure method of identification. Unlike traditional methods, facial recognition does not require physical contact or cooperation from the user, making it a contactless solution that is both user-friendly and hygienic. Additionally, facial recognition systems are capable of detecting multiple faces simultaneously and can perform identification in real-time, offering increased efficiency and security. In a home security context, facial recognition technology allows homeowners to automatically identify individuals at their door without requiring them to manually unlock doors or use physical keys. Moreover, face detection algorithms, such as those used in OpenCV and Haar Cascades, have made it easier to implement real-time face recognition systems that can accurately detect and recognize faces even in varying lighting conditions and angles. Research by Zhang et al. (2019) shows that deep learning techniques, such as convolutional neural networks (CNNs), can further enhance the accuracy and reliability of facial recognition systems, making them less prone to errors [1, 2].

2.3 Machine Learning and Deep Learning in Face Recognition

The integration of machine learning (ML) and deep learning (DL) into face recognition systems has revolutionized the field. Early facial recognition systems relied on simple algorithms and image processing techniques, but with the advent of deep learning, these systems have become significantly more accurate. Deep learning algorithms, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), enable systems to learn from large datasets and improve recognition over time [7, 8].

Sun et al. (2014) introduced the concept of deep neural networks (DNNs) for face recognition, which allowed systems to learn from images in a hierarchical manner, improving their accuracy in detecting and identifying faces. A study by Cheng et al. (2020) explored the use of deep learning models for face recognition in real-time security systems, showcasing how these models can be optimized for deployment in embedded systems, such as cameras and door locks. This advancement in machine learning and deep learning allows for better generalization, enabling facial recognition systems to work under various real-world conditions, including changes in facial expression, aging, and environmental variations. Incorporating machine learning into home security systems, as seen in this project, offers several benefits, including increased adaptability and reduced error rates. The ability of these models to improve over time by learning from new data makes them highly efficient and capable of recognizing faces in various contexts, including day/night cycles, seasonal changes, and different [2, 3].

2.4 Real-Time Alerts and Notification Systems

An essential feature of modern security systems is the ability to provide real-time alerts. According to Singh et al. (2017), real-time notifications are critical for empowering homeowners to act quickly when a potential security threat arises. Real-time alert systems can be implemented through buzzer alarms, mobile app notifications, or even remote-control systems that allow homeowners to lock/unlock doors or trigger additional security measures.

In this project, an alarm system is integrated with the facial recognition system to provide immediate alerts in case an unauthorized individual attempts access. By integrating this with mobile notifications or web-based alerts, the system ensures that homeowners are promptly informed and can take action as necessary [3].

2.5 Challenges and Future Trends

Despite the advancements in facial recognition and smart security systems, there are several challenges that still need to be addressed. Privacy concerns related to the collection and storage of facial data, as well as potential biases in recognition algorithms, remain significant issues in the field of biometric security. Studies by Noyes et al. (2020) highlight that facial recognition systems can have higher error rates for certain demographics, such as people with darker skin tones or women, which raises concerns about fairness and accuracy. Future developments in the field are likely to focus on enhancing the accuracy of facial recognition systems, improving security protocols, and addressing privacy concerns. The integration of multi-modal biometric systems (combining face recognition with other forms of authentication, such as voice or fingerprint recognition) may also emerge as a way to increase security while minimizing the risks of false identification [6, 7].

CHAPTER 3

SYSTEM DESIGN AND ARCHITECTURE

3.1 Overview of System Architecture

The system architecture of the home security solution presented in this project is designed to provide an integrated, efficient, and reliable automated security system using facial recognition technology, real-time alerts, and access control mechanisms. The architecture combines hardware and software components, each playing a crucial role in ensuring seamless operation and a high level of security. The system architecture follows a modular approach, where different components work together to achieve the core objective of identifying authorized individuals, controlling access, and alerting the homeowner of any unauthorized attempts. The following is an overview of the major components of the system architecture:

3.1.1 Hardware Components

1. ***ESP32 Cam:*** The ESP32 Cam is a key hardware component of the system. This small, cost-effective microcontroller is equipped with both a camera and Wi-Fi capability. The camera captures images of individuals approaching the door, and the onboard processing unit processes the images for face recognition. The Wi-Fi connectivity allows the ESP32 to communicate with the server for real-time processing and to send notifications to the homeowner in case of an unauthorized access attempt [2].
2. ***SG90 Servo Motor***: The SG90 servo motor is used to automate the locking and unlocking of the door based on the results from the facial recognition process. When the system detects a recognized face, the servo motor triggers the door lock to open. If an unrecognized face is detected, the system does not unlock the door and instead triggers an alarm [3].
3. ***Buzzer:*** The buzzer serves as an immediate feedback mechanism. When the system detects an unrecognized face, the buzzer sounds, alerting the homeowner of a potential security breach. This real-time alert ensures that unauthorized access attempts are detected quickly [4].
4. ***FTDI Adapter***: The FTDI (Future Technology Devices International) Adapter provides a communication link between the ESP32 Cam and other components. It ensures the proper communication and management of the system's hardware setup, especially for tasks related to system resets, Wi-Fi configuration, and overall control .
5. ***Reset Switches:*** The system is designed with reset switches to manage both the Wi-Fi connection and the overall system. These switches allow the user to reboot or reconfigure the system as needed, ensuring that the system can be restored to operational status if any issues arise [4].



Fig.1: Reset Switches

3.1.2 Software Components

1. ***Face Recognition Library:*** The core software component for the identification process is the face recognition library written in Python. This library uses advanced image processing algorithms to compare the captured face with a database of authorized faces. Using deep learning models, the system processes images captured by the ESP32 Cam and checks them against the stored face data to either grant or deny access. The system is designed to have an accuracy rate of 97%, providing reliable identification under varying environmental conditions [4].
2. ***FastAPI Backend:*** The FastAPI framework serves as the backend server that handles the system's logic and communication. FastAPI processes requests from the ESP32 Cam and performs the necessary actions based on the outcomes of the facial recognition process. It manages the overall logic of the system, such as granting or denying access, activating the servo motor to unlock the door, and triggering the buzzer for unauthorized access [5].
3. ***SQLite Database:*** The SQLite database is used for storing and managing the facial recognition data. Each authorized user's facial features are stored as biometric templates in the database. The backend uses Prisma ORM to interact with the SQLite database, ensuring efficient storage and retrieval of facial data. The system continuously checks the captured faces against the database to determine whether access should be granted [5].
4. ***Machine Learning and Deep Learning Models:*** The system employs machine learning (ML) and deep learning (DL) techniques to improve the accuracy of the face recognition system. These models allow the system to adapt to changes in users' appearances, such as aging or variations in lighting and angles. The convolutional neural network (CNN) architecture is employed for processing images and improving recognition performance [4, 5].
5. ***Wi-Fi Communication:*** Communication between the ESP32 Cam and the backend server is handled via Wi-Fi. This enables real-time data transfer, allowing the system to promptly analyze faces and provide feedback to the homeowner. The Wi-Fi connection ensures that all components of the system remain in sync and responsive.

3.1.3 System Workflow

The following describes the key steps in the system's workflow:

1. ***Face Detection:*** The system detects an individual approaching the door through the ESP32 Cam. The camera captures an image and sends it to the backend server via Wi-Fi.
2. ***Face Recognition:*** Once the image is received by the backend, it is processed by the face recognition library, which compares the captured face with the stored faces in the SQLite database. The system then determines whether the individual is authorized or not.
3. ***Access Control:*** If the individual is recognized as an authorized user, the system sends a signal to the SG90 servo motor to unlock the door, allowing the person to enter.

If the individual is not recognized, the buzzer is triggered to alert the homeowner of the unauthorized access attempt. The system also sends a real-time notification to the homeowner, informing them of the potential threat.

Notification and Alerts

The system can send push notifications to the homeowner's mobile device or app, alerting them to the unauthorized access attempt. These notifications can include information such as the time of the attempt, the location, and any other relevant data [5].

3.1.4 User Interface (UI)

While the focus of the system is on automated access control, the system also includes a user interface for configuration and management. The UI, accessible through a web portal or mobile app, allows homeowners to:

- View the list of authorized users.
- Add or remove users by enrolling or deleting their facial data.
- Manage system settings, such as notification preferences and alarm settings.
- Monitor system status and view logs of access attempts.

3.2 Backend Architecture

The backend architecture of the home security system is a crucial part of the system, handling logic, data management, face recognition processing, and communication with hardware components. The backend is responsible for ensuring that the system functions efficiently, securely, and in real-time, by providing a structured flow of data between the hardware components (such as the ESP32 Cam) and the software components (including the facial recognition algorithms, database, and user interface). This section outlines the design of the backend architecture, the technologies used, and the flow of operations [5, 6].

3.2.1 Overview of Backend Components

The backend architecture consists of several interconnected components, including the server-side logic, database management, face recognition model, and communication protocols with hardware. The backend is responsible for handling requests, processing facial images, managing facial data, and sending appropriate instructions to hardware components (such as unlocking the door or triggering alerts). The key backend components are as follows:

FastAPI Framework

The FastAPI framework is the main server-side component. It is responsible for handling HTTP requests, processing face recognition data, managing user authentication, and serving the appropriate responses back to the system. FastAPI's asynchronous capabilities allow it to handle multiple requests simultaneously with high performance, making it an ideal choice for real-time applications like this home security system [7].

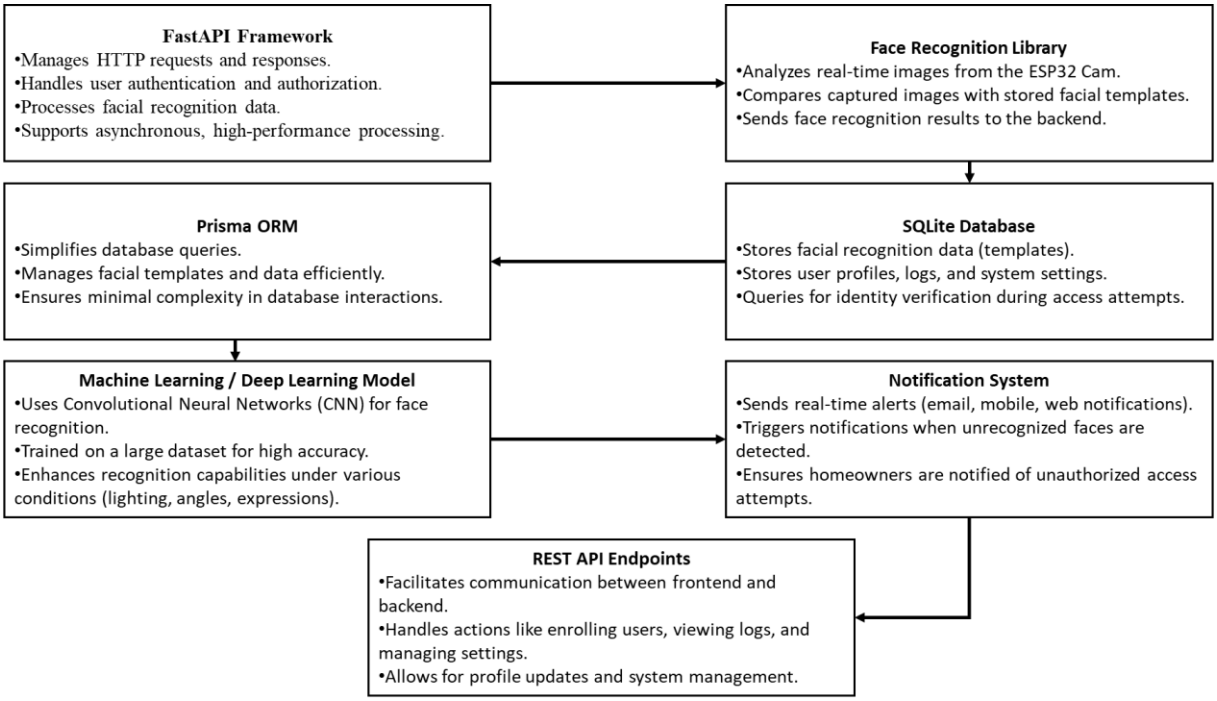


Fig.3.2.1: Backend Architecture.

Face Recognition Library

The core functionality of the system revolves around face recognition, and the Face Recognition Library is responsible for analyzing images, comparing them with stored data, and determining whether access should be granted. The system uses pre-trained deep learning models to detect and recognize faces with high accuracy. The library interacts directly with the ESP32 Cam to process images and return results.

SQLite Database

The SQLite database stores all the facial recognition data. It contains the facial templates for authorized individuals, and the backend queries the database to verify the identity of anyone attempting to gain access. The database is also used to store system settings, logs, and user management data (such as adding/removing faces or configuring notification preferences) [7].

Prisma ORM

Prisma ORM is used as an Object-Relational Mapping (ORM) layer to interact with the SQLite database. It simplifies querying, inserting, and updating data within the database, making the database operations more efficient and easier to manage.

Machine Learning/Deep Learning Model

The system employs a machine learning or deep learning model for face recognition. This model is trained on a large dataset of faces and is used to compare the real-time image captured by the ESP32 Cam with the stored templates in the database. Deep learning models, such as Convolutional Neural Networks (CNNs), are used to increase accuracy, especially under varying conditions such as different lighting, angles, and facial expressions.

Notification System

The notification system sends real-time alerts to homeowners when unauthorized access is detected. These notifications are triggered by events in the backend, such as the detection of an unrecognized face. Alerts can be sent via email, mobile apps, or web-based notifications, ensuring that the homeowner is instantly notified.

REST API Endpoints

The backend exposes various REST API endpoints that allow for communication between the system's frontend (user interface) and the backend. These endpoints facilitate actions such as enrolling new users, viewing access logs, updating user profiles, and managing system settings.

3.2.2 Backend Workflow

The backend workflow involves several stages, from receiving the image data from the camera, processing it, comparing it with stored data, and taking actions based on the recognition results. The following is an overview of the backend processing flow:

Image Capture

When someone approaches the door, the ESP32 Cam captures an image of the individual and sends it to the backend for analysis via a Wi-Fi connection. The image is transmitted as a data packet that the backend system can process.

Face Detection and Recognition

Once the image reaches the backend, it is processed by the Face Recognition Library, which first performs face detection to locate and isolate the face from the image. The detected face is then compared with the templates stored in the SQLite database using deep learning models to determine whether the individual is authorized.

Decision Making

Based on the comparison:

- If the face is recognized and matches an authorized user, the backend sends a signal to the hardware components (via the ESP32 Cam) to activate the SG90 servo motor to unlock the door.
- If the face is unrecognized, the backend triggers the buzzer to sound an alarm, notifying the homeowner of the unauthorized access attempt. It also sends a real-time notification to the homeowner via the chosen communication channels (e.g., mobile app, email).

Database Interaction

The backend interacts with the SQLite database at various points in the workflow:

- Face Enrollment: When a new user is added, their facial template is extracted from a previously taken photo and stored in the database.
- Access Log: Every face recognition attempt, whether successful or not, is logged in the database to allow the homeowner to review access history.
- System Configuration: The backend stores system settings, including notification preferences, user configurations, and alarm thresholds in the database.

User Interface Interaction

The backend also provides a connection to the user interface. This interface allows homeowners to manage the system, such as adding/removing users, viewing access logs, and configuring system preferences. The REST API handles these requests, ensuring the UI and backend stay synchronized.

3.2.3 Security and Authentication

Given the critical nature of the system, ensuring data security and user privacy is a priority. The backend architecture implements various security measures:

Secure Data Storage

Sensitive information, such as facial data and system configurations, is stored securely in the SQLite database. The facial templates are stored in a hashed format, ensuring that even if the database is compromised, the actual facial data cannot be accessed.

API Security

API authentication is enforced using techniques such as API keys or OAuth to ensure that only authorized devices or users can interact with the backend. This prevents unauthorized access to the backend APIs.

Encrypted Communication

Communication between the ESP32 Cam and the backend is encrypted using HTTPS to protect the data during transmission. This ensures that images and personal data cannot be intercepted or tampered with during transmission.

3.4 Database Design

The Database Design of the home security system is crucial for organizing and storing essential data related to user profiles, facial recognition templates, system configurations, access logs, and notifications. The use of SQLite provides a lightweight and efficient way to manage this data [7].

3.4.1 Database Tables and Structure

The system utilizes several tables in the database to store different kinds of data. The key tables in the database are as follows:

1. **Users Table**
    - Stores information about users, including their facial recognition templates, personal details, and contact information.

Column Name	Data Type	Description
user_id (PK)	INTEGER	Unique identifier for each user
name	TEXT	Full name of the user

Column Name	Data Type	Description
face_template	BLOB	Encoded facial recognition template (binary data)
email	TEXT	Email address of the user for notifications
phone_number	TEXT	Phone number of the user
is_active	BOOLEAN	Indicates if the user is active (authorized for access)

Table.1: User Table for personal details, and contact information.

2. **Access Logs Table**
   - Stores logs of each access attempt, including the timestamp, result, and user involved.

Column Name	Data Type	Description
log_id (PK)	INTEGER	Unique identifier for each log entry
user_id (FK)	INTEGER	References user_id in the **Users** table
timestamp	DATETIME	Timestamp of the access attempt
status	TEXT	Status of access (e.g., "granted", "denied")
attempt_type	TEXT	Type of attempt (e.g., "facial recognition")
location	TEXT	IP address or device from which the attempt was made

Table.2: Access User Login Table.

3. **System Configuration Table**
   - Stores various system configuration settings, such as notification preferences and alarm thresholds.

Column Name	Data Type	Description
config_id (PK)	INTEGER	Unique identifier for each configuration entry
setting_name	TEXT	Name of the configuration setting (e.g., "alarm_threshold")
setting_value	TEXT	Value for the setting (e.g., threshold value, "enabled")
updated_at	DATETIME	Timestamp of the last update to the setting

Table.3: System Configuration Table.

4. **Notification History Table**
   - Stores records of notifications sent to users when an unauthorized access attempt occurs.

Column Name	Data Type	Description

Column Name	Data Type	Description
notification_id (PK)	INTEGER	Unique identifier for each notification record
user_id (FK)	INTEGER	References user_id in the **Users** table
notification_type	TEXT	Type of notification sent (e.g., "email", "SMS")
status	TEXT	Status of the notification (e.g., "sent", "failed")
sent_at	DATETIME	Timestamp when the notification was sent

Table.4: Notification History Table.

5. **Facial Data Table (Optional)**
    - Stores detailed facial data, such as facial landmarks or additional biometric data if required.

Column Name	Data Type	Description
facial_data_id (PK)	INTEGER	Unique identifier for each facial data entry
user_id (FK)	INTEGER	References user_id in the **Users** table
facial_landmarks	BLOB	Encoded facial landmarks data (binary format)
image_data	BLOB	Image data used for facial recognition (optional)

Table.5: Farcical Recognition Data Table.

3.4.2 **Relationships Between Tables**

The relationships between tables ensure that data is well-organized, accessible, and consistent across the system. Here's an outline of how the tables relate to one another:

- Users - Access Logs: The Users table is linked to the Access Logs table via the user_id field. This means that each access attempt is associated with a user and can be tracked in the Access Logs table.
- Users - Notification History: The Users table is also linked to the Notification History table via the user_id field. This relationship tracks notifications that have been sent to specific users regarding unauthorized access attempts.
- Users - Facial Data (Optional): If additional facial data (such as landmarks) is stored, the Users table is linked to the Facial Data table via the user_id field. This table could be optional depending on the requirements of the system and the level of detail needed for biometric data storage.
- Users - System Configuration: Although not directly related to individual users, the System Configuration table affects the entire system. For example, the system's alarm thresholds or notification settings might apply globally to all users, and the settings can be adjusted from the System Configuration table.

3.4.3 **Example Data Entries**

Users Table:

user_id	name	face_template	email	phone_number	is_active
1	John Doe	[Binary Data]	john@example.com	123-456-7890	TRUE
2	Jane Smith	[Binary Data]	jane@example.com	987-654-3210	TRUE

Table.6: For example, user data table.

This table showing for example, user data some information, user name face template, email, phone number for massage, also active TRUE and False close number.

Access Logs Table Example:

log_id	user_id	timestamp	status	attempt_type	location
1	1	2025-02-23 10:45:00	granted	facial recognition	192.168.0.1
2	2	2025-02-23 11:00:00	denied	facial recognition	192.168.0.2

Table.7: Access Logs Data Information Table.

System Configuration Table Example:

config_id	setting_name	setting_value	updated_at
1	alarm_threshold	5	2025-02-23 09:30:00
2	notification_email	enabled	2025-02-23 09:30:00

Table.8: System Configuration Table.

Notification History Table Example:

notification_id	user_id	notification_type	status	sent_at
1	1	email	sent	2025-02-23 10:46:00
2	2	SMS	failed	2025-02-23 11:01:00

Table.9: Notification History Table.

Facial Data Table (Optional) Example:

facial_data_id	user_id	facial_landmarks	image_data
1	1	[Binary Data]	[Binary Data]

Table.10: Finical Data Information Table.

3.4.4 **Data Security Considerations**

Given that the database will store sensitive information, particularly facial recognition data and personal details, certain security measures should be implemented:

1. Data Encryption:
    - Sensitive data such as face template, facial landmarks, and image data should be encrypted before storage to ensure privacy.
2. Access Control:
    - Proper access control mechanisms should be in place to ensure that only authorized users can interact with the database. For example, only system administrators or trusted personnel should be allowed to modify or delete user records.
3. Backup and Recovery:
    - Regular backups of the database should be performed to avoid data loss in case of system failure. A recovery plan should also be in place to restore critical data in the event of an emergency.
4. Audit Trails:
    - All sensitive changes, such as adding or removing users, changing configurations, or modifying facial data, should be logged and stored for auditing purposes.

3.4.3 **Schema Design**

Below is a simplified Entity-Relationship Diagram (ERD) illustrating the structure of the database:

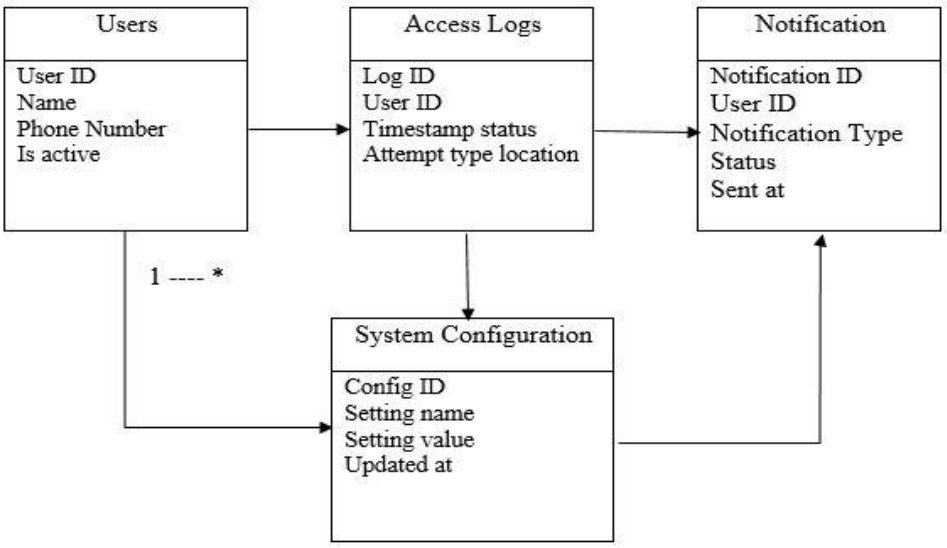


Fig.1: Simplified Entity-Relationship Diagram.

3.4.4 **Data Flow and Access Patterns**

The system interacts with the database in the following ways:

1. User Enrollment: When a new user is added to the system, their facial data is processed, and a record is created in the Users table. The face template is stored and associated with the user's profile.
2. Access Attempt Logging: Each access attempt, successful or unsuccessful, is recorded in the Access Logs table with relevant details such as timestamp, user identification, and status.
3. System Configuration Updates: The system's settings can be modified, such as changing the threshold for triggering an alarm or updating the user notification preferences. These updates are stored in the System Configuration table.
4. Notification Tracking: When unauthorized access is detected, the system triggers notifications, which are logged in the Notification History table for tracking and auditing purposes.

3.4.5 **Database Security Considerations**

Given the sensitive nature of the stored data (such as face templates and personal information), the database design implements several security measures:

- Encryption: Facial templates and other sensitive data can be encrypted before being stored in the database to protect user privacy.
- Access Control: The system uses role-based access control to restrict who can modify database records (e.g., only system administrators can add new users).
- Regular Backups: Periodic backups of the database ensure that the data is recoverable in case of hardware failure or system issues.

CHAPTER 4

IMPLEMENTATION DETAILS

4.1 Development Environment Setup

The successful implementation of the face recognition-based home security system depends heavily on the establishment of a well-structured and compatible development environment. This environment encompasses a combination of both hardware and software components, each playing a crucial role in ensuring the system's functionality, reliability, and efficiency. The integration of these components is not merely about connecting various devices and writing code; it requires careful planning and configuration to ensure smooth communication and seamless operation. The hardware elements, such as the ESP32 Cam, SG90 Servo Motor, and buzzer, serve as the physical interface between the user and the system, making it vital that these components are appropriately connected and configured. On the software side, tools like FastAPI, SQLite, and machine learning algorithms are used to process facial recognition data, manage the backend server, and ensure real-time performance of the system. Setting up the development environment is crucial to making sure that all the components from the microcontroller that captures facial images to the backend system that processes them and triggers real-time actions like unlocking the door work harmoniously together. Each step in the setup, from hardware wiring to API endpoint configuration, contributes to the overall success of the project. A small issue in the development environment setup could lead to significant errors in performance, which is why meticulous attention is given to every detail [8, 9].

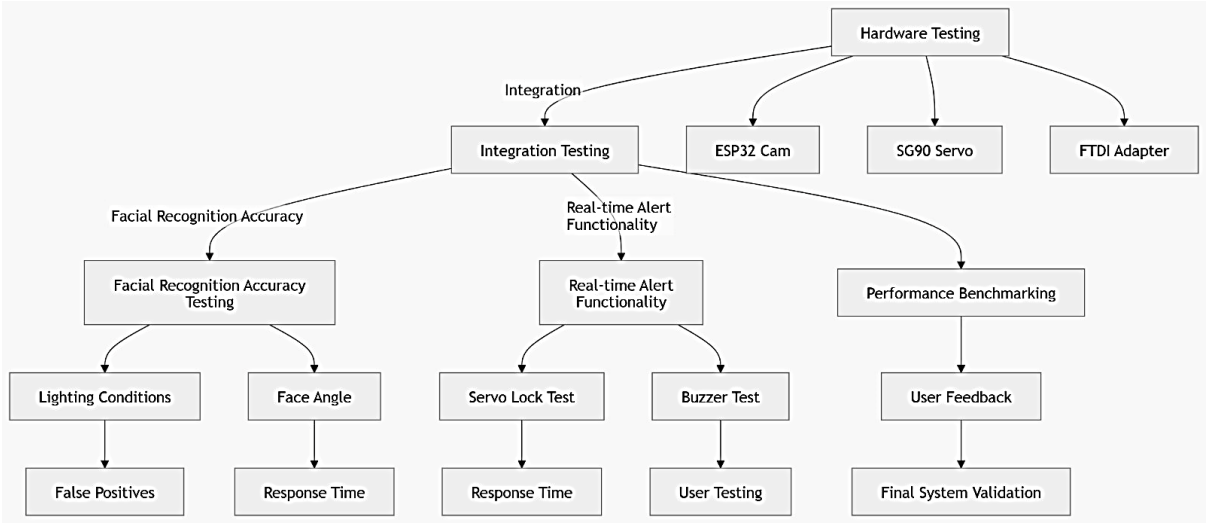


Fig.4.1: Development Environment Process.

In addition to ensuring that the hardware and software components communicate effectively, the development environment must be configured to handle specific challenges that may arise. For instance, the ESP32 Cam needs to capture clear facial data in varying lighting conditions, while the backend must process this data with high accuracy and minimal delay. Furthermore, security measures need to be in place to protect sensitive data, such as the homeowner's facial images and access logs, and ensure that unauthorized access is prevented.

The development environment setup also involves selecting appropriate tools and frameworks to facilitate development and testing. Tools like Visual Studio Code for writing code, Git for version control, and machine learning libraries for improving the facial recognition model are key to ensuring the smooth progression of the project. Moreover, with the system being Wi-Fi dependent, a reliable network setup is integral to allow seamless communication between the hardware and software components. We will dive deeper into the specifics of the hardware and software setup, the configuration process, and the necessary tools for implementing this innovative and secure face recognition-based home security system. The goal is to ensure that the development environment is optimized to provide both security and convenience for homeowners while maintaining a high level of reliability in all components of the system [8].

4.1.1 Hardware Setup

The hardware required for this project includes:

ESP32 Cam: The ESP32 Cam is a powerful yet compact microcontroller, specifically chosen for its ability to capture real-time facial data and process it efficiently. Equipped with an onboard camera, the ESP32 Cam allows for the seamless integration of facial recognition technology, providing an ideal platform for this home security system. The ESP32 Cam's affordability makes it a cost-effective choice for building a face recognition system, while its versatility and high processing power allow it to handle the demanding tasks of facial image processing and recognition. The camera module enables the system to capture clear facial images, which are then analyzed using the face recognition library to verify authorized users. The ESP32 Cam communicates with other components, such as the server, to ensure that the system works without interruption. Additionally, it supports Wi-Fi connectivity, making it easy to integrate the system with online services for remote notifications, updates, and control. Its compact size and all-in-one functionality make the ESP32 Cam an optimal device for this security solution [9, 10].



Fig.4.1.1: ESP32 Cam.

SG90 Servo Motor: The SG90 Servo Motor is a small yet reliable motor that plays a crucial role in the security system's locking mechanism. This servo motor is used to physically control the door's lock, based on the outcome of the facial recognition process. Once the ESP32 Cam processes the captured image and identifies the individual, the servo motor is activated to engage or disengage the locking mechanism. If an authorized face is detected, the door will unlock within seconds, allowing the individual to enter. Conversely, if the system

detects an unrecognized face, the door will remain locked, preventing unauthorized access. The SG90 motor is particularly suitable for this task due to its precision and compact size, allowing it to operate the lock mechanism smoothly and quickly. Its ability to be precisely controlled by the ESP32 Cam ensures a reliable and automated security feature that directly contributes to the safety and convenience of the homeowner [9].



Fig.4.1.2: SG90 Servo Motor.

Buzzer: The buzzer is an essential component in the system's alert mechanism, providing a vital audible notification when an unrecognized face is detected at the door. In the event that the system fails to identify an authorized individual, the buzzer emits a loud sound to alert the homeowner about a potential security breach. This feature is especially useful in preventing unauthorized access, as it serves as an immediate warning signal, enabling homeowners to take action swiftly. The buzzer is also helpful in noisy environments, where visual alerts may be missed, ensuring that the homeowner is always informed. In addition to its role in security, the buzzer enhances the system's usability by offering instant feedback to the homeowner, reassuring them that the system is functioning properly and actively monitoring the door [10].



Fig.4.1.3: buzzer is an essential component in the system's alert mechanism.

FTDI Adapter: The FTDI Adapter is a vital tool used during the initial programming and configuration of the ESP32 Cam. It serves as a bridge between the microcontroller and the development computer, enabling the transfer of code and necessary firmware to the ESP32 Cam. The FTDI Adapter ensures smooth communication between the microcontroller and the

computer, allowing the programmer to upload the facial recognition software and set up other system parameters. It also allows the programmer to troubleshoot the system and monitor its operation during the development phase. Without the FTDI Adapter, it would be difficult to interact with the ESP32 Cam effectively, which is why this device is essential for setting up and testing the home security system. Its role in ensuring that the ESP32 Cam is properly programmed and configured makes it a key component of the development environment [11].



Fig.4.1.4: FTDI Adapter.

Wi-Fi Network: A stable and secure Wi-Fi network is a fundamental part of this face recognition-based home security system, ensuring that the system remains connected to the internet and can function effectively in real-time. The ESP32 Cam relies on Wi-Fi to send and receive data, including facial recognition results, status updates, and remote notifications. This connectivity allows the system to notify the homeowner instantly when an unrecognized face is detected or when the door is unlocked. Additionally, the Wi-Fi network facilitates remote access to the system, enabling the homeowner to monitor and control the security system from anywhere in the world using a smartphone or computer. It also enables communication with the backend server, where facial recognition data is processed and stored. A reliable Wi-Fi connection is crucial for maintaining the performance and responsiveness of the system, ensuring that security alerts and access control actions are triggered in real time. In case of network instability, the system may encounter delays or failures, which is why having a high-quality, secure Wi-Fi connection is essential for optimal performance [9. 10, 11].

4.1.2 Software Setup

The software stack for this project includes several essential components, each playing a critical role in the development and operation of the system:

Python: Python serves as the primary programming language for developing the face recognition-based home security system. It is chosen for its readability, versatility, and vast ecosystem of libraries, making it an ideal choice for projects that involve complex data processing, machine learning, and hardware integration. Python's simplicity allows for rapid prototyping and easy debugging, both essential for a project that requires constant iteration and testing. With Python, developers can leverage powerful libraries such as OpenCV for

image processing, NumPy for numerical computations, and TensorFlow or Keras for implementing machine learning models, all of which play a role in enhancing the system's performance. The ability to seamlessly interface with hardware, including the ESP32 Cam, via Python's GPIO libraries or communication protocols, ensures smooth integration of the software stack with the physical components of the system [11, 12].

Face Recognition Library: At the heart of the system's facial recognition functionality is the face_recognition library, which is built on top of dlib. This library is specifically designed for processing and identifying facial features in images, making it an ideal choice for the home security system. face_recognition utilizes one of the most efficient and accurate algorithms for facial recognition, offering high accuracy and fast processing, even with real-time video or image streams. The library works by extracting key facial landmarks (such as the eyes, nose, and mouth) from captured images and comparing them against pre-stored facial data in the SQLite database. The result is then sent back to the ESP32 Cam, allowing the system to take action based on recognition results (e.g., granting or denying access). The library offers an impressive recognition accuracy of approximately 97%, which makes it reliable for identifying authorized individuals and ensuring that only those registered in the system are granted access to the home. The system's reliance on dlib for feature extraction allows it to recognize faces in various conditions, such as changes in lighting, facial expressions, or different angles. To further enhance its accuracy and reliability, torch and torchvision are incorporated to fine-tune deep learning models for facial recognition. These models are trained to handle difficult recognition tasks, such as matching faces in challenging environments, ensuring that the system performs consistently under a wide range of real-world conditions. ONNXRuntime can be used to optimize these models, ensuring that inference times are fast and computational resources are efficiently utilized.[13].

FastAPI: FastAPI is used for building the backend server of the system, responsible for managing and processing requests made by the ESP32 Cam and other system components. As a modern web framework, FastAPI is known for its high performance, ease of use, and automatic generation of OpenAPI documentation. It allows for the rapid development of RESTful APIs, making it an ideal choice for building the backend of the system. FastAPI handles user requests for facial recognition, sends results to the ESP32 Cam, and updates the SQLite database with access logs. Additionally, it manages requests for system configuration, status updates, and alerts. FastAPI's asynchronous capabilities allow the backend to handle multiple tasks simultaneously, ensuring a responsive and efficient system that can process requests in real time. Its seamless integration with Python-based libraries and tools ensures smooth development and system performance [13].

SQLite Database: The SQLite database is a lightweight and efficient database solution used to store critical data, such as facial recognition models, access logs, and user credentials. SQLite is chosen for its simplicity, speed, and ease of integration with Python. Since the system does not require complex relational database operations, SQLite provides a perfect solution that operates in a single file and is easy to set up and maintain. The Prisma ORM is used to manage interactions between the backend application and the database. Prisma provides an abstraction layer that allows developers to interact with the database using high-level queries, simplifying database management and ensuring efficient data storage and

retrieval. The facial recognition data stored in SQLite is critical for the operation of the system, as it enables fast and accurate matching of faces with previously stored images, while also recording access attempts and logs to track system activity [14].

Machine Learning and Deep Learning Models: The system leverages Machine Learning (ML) and Deep Learning (DL) algorithms to enhance the accuracy and reliability of facial recognition. These models are trained to recognize faces under various conditions such as different lighting, angles, and expressions, ensuring that the system performs well in real-world environments. ML algorithms allow the system to learn and adapt to new faces over time, while deep learning models specifically Convolutional Neural Networks (CNNs) can improve facial feature extraction and matching. These models ensure that even slight variations in facial appearance do not result in false rejections or errors in access control. By continuously improving the recognition accuracy, the system can operate reliably in dynamic conditions, offering both high performance and robustness. Additionally, these models help optimize the system for speed, reducing latency when granting or denying access. Machine learning techniques can also be used for system improvements, such as automatic adjustment to changing environmental conditions or adding new users to the facial recognition database.

4.1.3 Development Tools

To support the development process, the following tools are used:

Visual Studio Code (VS Code): Visual Studio Code (VS Code) is a lightweight yet powerful integrated development environment (IDE) that plays a central role in the development of this face recognition-based home security system. Known for its clean and intuitive interface, VS Code is equipped with a range of features that simplify the coding process, such as syntax highlighting, auto-completion, and debugging tools. These features make it easier to write, test, and debug Python code, which is essential for the implementation of complex algorithms like facial recognition. VS Code also supports a wide array of extensions, allowing developers to enhance its functionality according to the needs of the project. For instance, extensions such as Python support for linting, IntelliSense for code completion, and Jupyter Notebooks for running Python code interactively help streamline the development process. Furthermore, VS Code's Git integration simplifies version control and collaboration, making it easy to commit changes, track revisions, and collaborate with other team members. This flexibility, combined with an easy-to-navigate interface, makes Visual Studio Code an indispensable tool for the efficient development of the security system [11, 12, 13, 14].

Git: Git is a distributed version control system that plays a pivotal role in managing and tracking changes in the codebase throughout the development lifecycle. By using Git, developers can easily record changes, revert to previous versions, and ensure that the codebase remains stable as the project evolves. In a project of this scale, where multiple components are being developed and integrated simultaneously, Git allows for proper versioning and tracking of every modification made to the code. This ensures that any bugs or issues introduced during development can be quickly identified and corrected. Moreover, Git enables seamless collaboration among multiple developers, as it allows them to work on different features or components of the system in parallel without overwriting each other's work. GitHub, a cloud-based Git hosting platform, is often used to host repositories for easy

sharing, collaboration, and issue tracking. With GitHub, the project team can monitor progress, track bugs, and collaborate efficiently through pull requests, comments, and project management tools. This powerful version control system ensures the smooth development, maintenance, and scalability of the home security system [12, 14, 16].

4.1.4 Connectivity and Integration

One of the most crucial aspects of the face recognition-based home security system is ensuring seamless connectivity and effective integration between its hardware and software components. This system involves several devices and technologies that must communicate with each other reliably and in real-time to ensure the proper functioning of the security mechanism. At the heart of this communication is the ESP32 Cam, which serves as the primary device for capturing facial images. The ESP32 Cam is equipped with Wi-Fi capability, allowing it to send captured facial data over the internet to the backend server for processing. The FastAPI backend handles incoming data requests from the ESP32 Cam, where the facial images are analyzed using the Face Recognition Library. After processing, the backend sends the authentication results—whether the individual is authorized or not back to the ESP32 Cam [15].

This back-and-forth communication between the ESP32 Cam and the FastAPI backend is made possible through a stable and secure Wi-Fi network. The ESP32 Cam relies on this network to transfer data to the backend and receive responses in real time. The Wi-Fi connection plays a pivotal role in maintaining the system's responsiveness, allowing the door to be locked or unlocked based on facial recognition results within seconds.

Once the authentication result is received, the ESP32 Cam performs the necessary action. If the individual is recognized and deemed authorized, the SG90 servo motor is triggered to unlock the door, allowing the homeowner or authorized individual to enter. The servo motor receives precise signals from the ESP32 Cam, which is responsible for controlling the door's locking mechanism. This step relies on the smooth integration of the ESP32 Cam's processing capabilities with the servo motor's mechanical functions, ensuring that the door operates efficiently. In the case where the system detects an unrecognized face, it triggers an immediate response from the buzzer, an integral part of the security system. The buzzer emits a loud audible alert, notifying the homeowner of a potential security threat. This ensures that even if the homeowner is not physically present at the door, they will still be aware of unauthorized attempts to gain access. The buzzer's integration is critical in providing real-time feedback to the homeowner, making it a valuable component in enhancing the overall security of the system. The entire process, from facial data capture to authentication and access control, depends on the flawless integration of hardware components and software systems. Connectivity and real-time communication are essential to maintaining a high level of security, accuracy, and efficiency. Any interruptions in data transmission or system processing could result in delays or failures in the security system. Thus, the effective integration and communication between the ESP32 Cam, FastAPI backend, servo motor, and buzzer are fundamental to the success of the project [16].

4.1.5 System Configuration

Before the face recognition-based home security system can function effectively, several critical configurations need to be set up. These configurations ensure that the hardware components communicate properly with each other and that the software functions as intended. The following steps outline the essential setup processes:

Facial Data Enrollment: The Facial Data Enrollment process is the first step in setting up the system. This step involves capturing and storing facial data from authorized individuals to build a reliable facial recognition model. During enrollment, the system captures multiple images of the homeowner's face from different angles, lighting conditions, and expressions to ensure high accuracy during the identification process. The ESP32 Cam captures these images, which are then processed using the Face Recognition Library to extract unique facial features. These features are stored in the SQLite database along with a unique identifier for each authorized person. This data will later be used to compare against images captured during access attempts. Proper facial data enrollment is essential for minimizing false positives and ensuring that the system only grants access to authorized individuals [17].

Wi-Fi Configuration: The Wi-Fi Configuration is a crucial step in ensuring that the ESP32 Cam is able to communicate with the backend server over the internet. The ESP32 Cam connects to the home's Wi-Fi network, which enables it to send facial data to the FastAPI backend for processing and receive authentication results in real time. The ESP32 Cam is configured to connect automatically to the Wi-Fi network on startup. The configuration process includes entering the network credentials (SSID and password) into the system's settings, which allows the ESP32 Cam to establish a stable connection. Once connected, the ESP32 Cam can send and receive HTTP requests to interact with the FastAPI server, ensuring continuous communication between the frontend and backend of the system. This Wi-Fi connection is fundamental for the system to function in real-time, providing notifications, processing facial data, and controlling the locking mechanism efficiently [16].

API Endpoints: The API Endpoints are the backbone of communication between the frontend (hardware) and the backend (server). These endpoints are configured in FastAPI to handle various tasks related to the facial recognition process, system status updates, and user management. The backend exposes several API routes, such as:

- /enroll_face: To upload and store facial data from authorized individuals during enrollment.
- /recognize_face: To process facial images captured by the ESP32 Cam, compare them with stored data, and return authentication results (authorized or unauthorized).
- /status: To provide the current status of the system, such as whether it is armed or disarmed, and any active alerts.
- /user_management: To manage the addition or removal of authorized users and update their facial data if necessary.

FastAPI is configured to handle incoming HTTP requests from the ESP32 Cam, ensuring smooth data flow and real-time responses. For example, when the ESP32 Cam captures a face, it sends an HTTP request to the /recognize_face endpoint, where the server processes the image, compares it with stored data, and sends a response to the ESP32 Cam. If the face is recognized, the ESP32 Cam receives the authentication response and triggers the SG90

servo motor to unlock the door. Similarly, if the face is not recognized, the system triggers the buzzer to alert the homeowner. In addition to managing the core functionality of facial recognition, these APIs allow for future expansion and updates to the system, such as adding new users or modifying system settings remotely [17].

4.1.6 Testing and Debugging

Once the system is set up and configured, thorough testing is essential to ensure that the facial recognition technology and all associated components function as expected in real-world conditions. This stage is crucial to identify any potential issues, validate the system's performance, and fine-tune its accuracy and responsiveness. The testing process covers multiple aspects, including accuracy, environmental factors, user variations, and system performance. The following outlines the key testing procedures and debugging strategies employed to ensure the system's reliability.

Accuracy Testing

The accuracy of the facial recognition system is a critical metric for its success. To ensure high accuracy, the system undergoes a series of tests under various conditions, such as:

Different Lighting Conditions: The system is tested under bright, dim, and low-light environments to assess its ability to recognize faces effectively in various lighting scenarios. This helps ensure that the recognition algorithm is robust and can handle fluctuations in lighting, which are common in real-world environments.

Varying Angles and Distances: The system is tested with faces presented at different angles (e.g., side, profile) and distances (near or far) from the camera. This helps determine whether the ESP32 Cam can accurately capture and process facial data from diverse viewpoints and distances, which are essential for real-world usage.

Multiple Users: The system is tested with multiple authorized users to ensure that it correctly identifies each person and grants access accordingly. It is important to verify that the system can differentiate between users with similar facial features and ensure accurate recognition even when multiple users are enrolled in the system [16, 17, 18].

Response Time Testing

Another key aspect of testing is ensuring that the system responds quickly and accurately when granting or denying access. The response time of the system is evaluated by measuring the time it takes from capturing the image to sending a response (either "grant access" or "deny access") back to the ESP32 Cam. Ideally, this should happen within a few seconds to provide a smooth user experience. The testing evaluates:

Facial Recognition Speed: The time taken for the system to process and identify the face after it has been captured.

Locking Mechanism Response: The speed at which the SG90 servo motor engages or disengages to lock or unlock the door based on the recognition result.

Alert Time: If an unrecognized face is detected, the buzzer should be activated immediately to alert the homeowner. Testing ensures that the buzzer response is swift enough to notify the user of potential security breaches in real-time.

Debugging and Real-Time Monitoring

To identify and resolve any issues in the system's operation, various debugging tools are employed. These tools help monitor the real-time performance of the system and diagnose potential bottlenecks or failures. Common debugging practices include:

Logging: System logs are carefully reviewed to track facial recognition results, API requests, and responses between the hardware and backend server. Any discrepancies or errors in processing are logged for later analysis and resolution.

Real-Time Monitoring: Tools such as serial monitors and network analyzers are used to observe real-time data transmission between the ESP32 Cam and the backend server. This helps identify any delays, communication errors, or connectivity issues that could affect the system's performance.

Step-by-Step Debugging: In case of failures or inconsistencies, step-by-step debugging is performed to isolate the root cause of the problem. This may involve testing individual components (e.g., facial recognition accuracy, Wi-Fi connection) or running specific tests to evaluate the system's performance under various scenarios.

Iterative Improvements

As issues are identified during testing and debugging, iterative improvements are made to the system. This may involve adjusting the recognition threshold, fine-tuning the camera settings, updating the database, or optimizing the code to improve response time. Each round of testing allows the developers to progressively enhance the system's functionality, ensuring that it operates reliably and efficiently under all expected conditions.

4.1.7 Security Considerations

Since the system handles sensitive data, such as facial images, user credentials, and access logs, security is an essential aspect of the project. The system is designed with robust security measures to safeguard user privacy and prevent unauthorized access. These security measures are implemented at various levels ranging from secure authentication protocols to encrypted data storage and transmission. Below are the key security features integrated into the system to ensure the protection of sensitive information [16].

JWT (JSON Web Tokens) for Secure Authentication

JWT is used as the primary mechanism for secure authentication in the system. When an authorized user enrolls in the system or accesses the interface, a JWT is generated. This token contains encrypted user information and serves as a proof of identity for subsequent interactions with the system. By using JWTs, the system ensures that only authorized users

with a valid token can access certain functionalities, such as enrolling new faces or managing system settings.

JWTs have several advantages, including:

Stateless Authentication: The server does not need to store session data, as the JWT itself contains all necessary information. This reduces the risk of server-side vulnerabilities and improves performance.

Tamper-Proof: JWTs are cryptographically signed, ensuring that the data cannot be tampered with during transmission between the client (ESP32 Cam) and the backend server.

Expiration and Revocation: JWTs have an expiration date, and the system can revoke tokens if suspicious activity is detected. This ensures that a token is only valid for a limited time, further securing the authentication process.

Encrypted Communication over the Network

To ensure the confidentiality and integrity of data during transmission, the system employs encrypted communication over the network. The connection between the ESP32 Cam and the FastAPI backend is encrypted using HTTPS (Hypertext Transfer Protocol Secure), which is the secure version of HTTP. HTTPS encrypts the data exchanged between the devices using SSL/TLS protocols, preventing eavesdropping or man-in-the-middle attacks. This encryption ensures that sensitive data, such as facial images and authentication results, is transmitted securely, making it extremely difficult for malicious actors to intercept or alter the data during transmission.

In addition to securing the communication channel, the system ensures that sensitive data, such as facial images and access logs, is handled securely throughout the entire process. The Wi-Fi network must also be properly configured to prevent unauthorized access, with appropriate encryption methods (e.g., WPA2) and strong passwords in place [17].

Secure Data Storage in SQLite Database

All sensitive user data, such as facial recognition features and access logs, are stored in the SQLite database. To protect this data from unauthorized access, the database is secured using encryption techniques. Additionally, user data is stored in a structured manner to minimize the risk of data breaches. For example:

Facial Data: Only essential facial features (such as key facial landmarks) are stored in the database, rather than full images of faces. This reduces the risk of sensitive biometric data being exposed in case of a database breach.

Access Logs: Access logs that track who enter the system and when are also stored securely. These logs are encrypted to prevent tampering or unauthorized viewing, ensuring that a comprehensive audit trail is maintained without compromising user privacy.

Furthermore, access control measures are implemented to restrict who can interact with the database. Only authorized personnel, such as the system administrator, have access to modify

or manage the database. This principle of least privilege ensures that sensitive data is only accessible to individuals who absolutely need it.

Data Privacy and User Consent

Given that the system deals with biometric data, it is important to consider user privacy and obtain explicit consent from individuals whose faces are enrolled in the system. The system provides users with an option to opt-in for facial data collection during enrollment. Clear privacy policies are established to inform users about how their data will be used, stored, and protected, ensuring that they are fully aware of the system's data handling practices [18].

Regular Security Audits and Updates

To further enhance the security of the system, regular security audits are conducted to identify vulnerabilities and potential threats. Any weaknesses discovered during the audit process are addressed promptly through patches, updates, or system redesigns. This proactive approach ensures that the system remains resilient against new threats and potential exploits. Additionally, the software components, such as FastAPI, Python libraries, and the Face Recognition Library, are kept up to date with the latest security fixes to mitigate any known vulnerabilities.

CHAPTER 5

KEY FEATURES OF HOUSE SECURITY

House security is a critical component of protecting homes from unauthorized access and potential threats. The modern approach to home security has evolved significantly with the integration of smart technologies, providing homeowners with more control, convenience, and peace of mind. This chapter outlines the key features of an advanced House Security System, focusing on technologies that ensure a high level of protection while offering ease of use and remote management.

5.1 Smart Access Control

An essential feature of any modern home security system is the ability to effectively control access to the property. Traditional methods, such as locks and physical keys, have become inadequate in addressing today's security challenges. Smart access control systems have revolutionized the way homeowners can manage who enters and exits their homes, offering enhanced security, convenience, and flexibility. These systems eliminate the need for conventional keys and offer a range of advanced functionalities that can be controlled remotely, increasing both the safety and ease of home management [16].

Face Recognition Access

One of the most advanced technologies integrated into modern house security systems is facial recognition. This technology uses unique biometric features, such as the distance between the eyes, nose shape, and jawline, to identify and authenticate individuals seeking access. This system goes beyond traditional keyless entry systems by offering a highly secure and reliable method of identifying individuals based on their facial characteristics.

Automatic Identification: The system automatically compares the captured facial data with a pre-enrolled database of authorized users, including family members and trusted visitors. If the system recognizes the person as authorized, access is granted. For security reasons, the system can be set to deny access if an unrecognized or unauthorized face is detected.

Profiles for Family Members and Trusted Guests: Homeowners can easily set up profiles for each member of their household. Additionally, they can pre-enroll trusted guests or service providers, such as cleaners or maintenance workers, to grant them temporary access during specific time slots, all without needing physical keys or meeting in person.

Enhanced Security: Since facial recognition is based on unique biological features, it is much harder to replicate or forge compared to traditional keys or cards, making this method one of the most secure forms of access control [17].

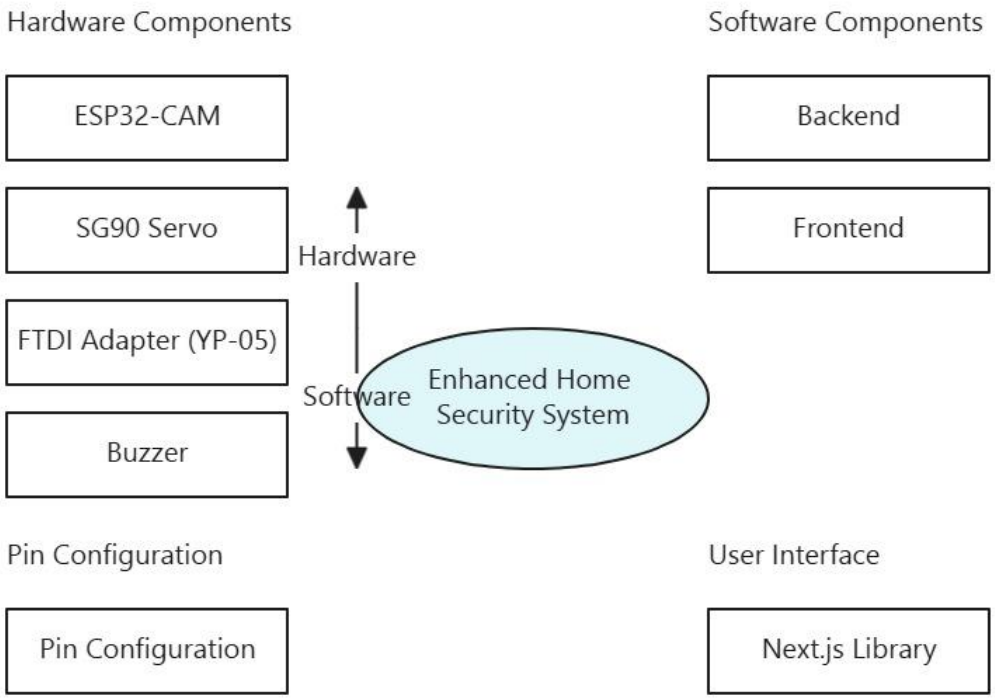


Fig.5.1: Hardware Components and Software Components system.

The convenience of face recognition access means homeowners no longer needs to worry about lost or stolen keys or forgettable PIN codes, providing peace of mind in knowing that only authorized individuals can enter the home.

Keyless Entry

Smart access control systems also introduce the concept of keyless entry, eliminating the need for traditional locks and physical keys altogether. This feature is especially valuable for homeowners who want a more secure and convenient way to enter and exit their property without the need to fumble with keys or risk losing them.

1. Smart Locks: Modern smart locks use wireless technology, such as Bluetooth or Wi-Fi, to control door access. These locks can be opened via a smartphone app, which allows the homeowner to unlock the door from anywhere. With no physical keys required, smart locks are more difficult to pick or tamper with compared to traditional locks.
2. Biometric Recognition: In addition to facial recognition, other forms of biometric recognition, such as fingerprint or iris scanning, can also be employed as keyless entry methods. These biometric systems ensure that only the person who matches the stored data can gain access to the home.
3. RFID Tags and Cards: Radio Frequency Identification (RFID) tags or cards offer another keyless entry option. Homeowners or authorized guests can use an RFID tag to simply wave in front of a sensor to unlock the door. These tags can be given to family members, friends, or even service providers to grant them secure access without the need for a physical key.

4. Keyless entry systems offer unparalleled convenience, as homeowners no longer have to worry about forgetting, losing, or misplacing physical keys. Additionally, these systems significantly reduce the risk of unauthorized access, as the data used for authentication is much harder to steal or replicate.

5.2 Real-Time Alerts and Notifications

A critical feature in modern house security systems is the ability to receive real-time alerts and notifications. These instantaneous updates enable homeowners to swiftly respond to potential security threats, providing a layer of protection that goes beyond traditional systems. With real-time alerts, homeowners can be immediately aware of suspicious activity, giving them the opportunity to take prompt action, whether it's investigating the situation or notifying authorities.

Smart home security systems are designed to provide immediate notifications through various channels, such as smartphone apps, emails, or even direct alerts to emergency services. By ensuring that homeowners are constantly updated, these systems help reduce the risk of threats escalating without notice, offering both convenience and heightened security [17].

Intruder Detection

One of the most important functions of a smart security system is the ability to detect intruders. Integrated motion sensors and security cameras continuously monitor the exterior and interior of the home for unusual activity, making it difficult for intruders to go unnoticed. These technologies work together to create a comprehensive surveillance network that keeps watch over the property 24/7.

Motion Sensors: These sensors are placed near vulnerable entry points such as doors, windows, and hallways. When motion is detected, the system immediately triggers an alert and can activate other devices such as cameras, alarms, or lighting to deter the intruder. The motion sensors are smart enough to differentiate between harmless movements, such as a pet, and potential security threats, minimizing false alarms.

Security Cameras: High-definition cameras integrated with the system can stream live footage directly to the homeowner's device. When a motion sensor detects movement, the camera is activated to capture video and can send an alert with a live feed to the homeowner, providing real-time insight into what is happening. Homeowners can remotely monitor the footage to assess whether the activity is indeed suspicious or harmless.

By receiving real-time notifications from these sensors and cameras, homeowners can respond swiftly and effectively to any signs of unauthorized access or suspicious behavior, enhancing overall security and reducing the chances of a break-in [17].

Door/Window Sensors

Door and window sensors are another essential component of a comprehensive home security system. These sensors are designed to detect any unauthorized attempt to open or tamper with entry points, such as doors and windows. Whether the homeowner is at home or away, the system can instantly notify them of any potential breaches.

Tamper Detection: These sensors are highly sensitive and can alert the homeowner to any unauthorized physical manipulation, such as attempts to pry open a window or door. If the system detects that a door or window has been opened without authorization, an immediate notification is sent to the homeowner's smartphone, allowing them to take action before an intruder gains full access to the property.

Silent Alerts: In some cases, it may be necessary for the homeowner to receive a silent alert when a sensor is triggered, especially during a break-in attempt. This alert ensures that the homeowner can discreetly monitor the situation without alarming the intruder.

Instant Notifications: In case of a breach, the system sends real-time alerts, including the specific sensor that was triggered, so homeowners can know exactly where the breach occurred and assess whether immediate action is needed. This feature provides an extra layer of security by keeping homeowners informed of every aspect of the system's operation.

These door and window sensors provide an added level of protection, especially for vulnerable entry points, such as ground-level windows or rear doors, ensuring that homeowners can monitor and secure their property at all times.

Emergency Alerts

In addition to intruder detection, modern home security systems are equipped to handle various emergency situations, such as fire, carbon monoxide poisoning, and flooding. These emergencies often require an immediate response, and having the system alert the homeowner and appropriate authorities can help prevent serious damage or loss.

Fire Detection: Smoke detectors integrated with the system can detect smoke or heat levels rising above safe thresholds. In the event of a fire, the system immediately sends an alert to the homeowner and can also notify emergency services. This enables a faster response time and helps minimize the risk of injury or property damage.

Carbon Monoxide Detection: Carbon monoxide (CO) is an odorless, colorless gas that can be deadly when present in high concentrations. Smart security systems equipped with carbon monoxide sensors can detect the presence of this dangerous gas and instantly alert homeowners to evacuate the property. In cases of extreme CO exposure, the system can also contact emergency services for immediate intervention.

Flood Detection: Water leak sensors or flood detectors can be placed in areas prone to water damage, such as basements, bathrooms, or kitchens. If the system detects a leak or rising water levels, it sends a notification to the homeowner to prevent damage to property and possessions. These sensors are particularly important in preventing costly repairs caused by undetected leaks or water-related damage.

Emergency alerts for fire, carbon monoxide, and flooding, a home security system becomes a comprehensive safety net, offering more than just protection from intruders. It helps ensure that homeowners are immediately informed about any potential life-threatening situations, allowing them to take quick action and possibly save lives or prevent further damage [18].

5.3 Video Surveillance Integration

Incorporating video surveillance systems into modern home security setups has become a key feature that enhances safety and peace of mind for homeowners. These systems provide continuous monitoring and allow for constant visibility of both the interior and exterior of the home. With the advancement of smart technology, video surveillance has evolved to offer more than just passive observation; it integrates seamlessly with other security features to provide real-time alerts, remote monitoring, and recorded evidence of any suspicious activity.

Smart security cameras now offer more advanced functionalities, such as high-definition video, motion detection, and remote access to footage. These features work together to ensure that homeowners have the most accurate and actionable insights into what is happening around their property at all times.

High-Definition Video Feed

One of the primary advantages of modern smart security cameras is their ability to provide high-definition (HD) video feeds. Whether monitoring the front door, backyard, or other vulnerable entry points, high-definition cameras capture clear, sharp images, even in low-light conditions.

Clear Visual Evidence: HD cameras provide clear and detailed video footage, which is crucial in identifying potential intruders, monitoring deliveries, or checking on service personnel. With resolutions of 1080p or even 4K in some models, homeowners can zoom in on specific areas without losing image quality, making identification of individuals and activities more accurate.

Wide-Angle Coverage: Many modern cameras are equipped with wide-angle lenses or 360-degree capabilities, offering expanded coverage of large areas around the property. This ensures that all critical zones are monitored without the need for multiple camera installations, allowing for more comprehensive surveillance.

Night Vision: Many security cameras also come with built-in night vision capabilities, enabling them to continue monitoring in low or no light. This ensures that the property is protected around the clock, regardless of the time of day or lighting conditions.

The high-definition video feed ensures that homeowners have access to clear, detailed visual information, which can be vital for security purposes and for verifying specific events.

5.4 Smart Alarms and Sirens

Incorporating a smart alarm system is a key feature of modern home security setups, providing an immediate and effective response to security breaches or emergencies. These alarm systems are designed not only to alert homeowners of suspicious activity but also to deter potential intruders and notify nearby individuals of a potential threat. Whether it's a loud siren to scare off intruders or a silent alert to discreetly notify the authorities, the smart alarm system plays a critical role in enhancing the overall safety and security of the home.

Loud Sirens

One of the primary functions of a smart alarm system is to deter intruders with loud sirens. When unauthorized access is detected, the system immediately triggers an alarm, emitting a powerful sound that can be heard throughout the neighborhood.

Intruder Deterrent: The loud siren is designed to startle the intruder and create an immediate sense of urgency, often causing them to flee the scene in fear of being caught. The piercing sound is intended to attract attention, making it more likely that the intruder will be detected before they can proceed with their plans.

Public Awareness: In addition to scaring off the intruder, the loud siren also alerts nearby residents or passersby, prompting them to be aware of the situation. Neighbors may notice the alarm and take action, whether it's by calling the authorities or keeping a watchful eye on the property until help arrives.

Security Reinforcement: Many homeowners find peace of mind knowing that the loud sirens will be triggered immediately if there is a breach. This enhances the security of the home and acts as an additional deterrent against criminals who may avoid homes with visible or audible security measures.

The loud sirens not only serve as a deterrent but also ensure that the alarm is heard by anyone in the vicinity, increasing the chances of quick intervention or discovery of the intruder.

Silent Alerts

While loud sirens are effective for deterring intruders, there are situations where a silent alert may be necessary. For example, during a robbery or break-in, a loud siren might alert the intruder that they have been detected, which could escalate the situation and endanger the homeowner. In these instances, the system can send a silent alert to the homeowner or designated security service without alerting the intruder.

Discreet Notifications: Silent alerts provide a discreet way to notify the homeowner of an emergency situation. For instance, if an intruder gains access to the home and the homeowner is present, the system can send an immediate notification to their phone or another device without triggering a loud alarm. This allows the homeowner to remain aware of the situation while avoiding drawing attention to themselves or the intruder.

Security Services Notification: In some cases, the silent alert can be configured to notify professional security services or the local authorities directly. These services will then be able to respond without the intruder being aware that help is on the way. Silent alerts are particularly useful when there is a need for a covert response to avoid further escalation of the situation.

Real-Time Updates: Silent alerts are often accompanied by real-time updates, such as live video feeds or a summary of recent activity, so the homeowner or security services can monitor the situation and make informed decisions on how to handle the breach.

Silent alerts ensure that homeowners have a discreet option to notify authorities and take action in sensitive situations without compromising their safety or alerting the intruder.

5.5 Smart Home Automation Integration

As technology continues to evolve, house security systems are becoming more sophisticated by integrating with other smart home devices. This integration creates a fully connected and automated environment, allowing homeowners to control multiple systems from a single interface and enhance their home's security and functionality. By connecting security features to a broader smart home network, the security system can work in tandem with other devices to create an ecosystem that improves both safety and convenience.

Automated Lighting

One of the most common and effective integrations is with automated lighting. Lighting plays a crucial role in home security by simulating the presence of residents when they are away, deterring potential burglars who may target empty homes.

Simulated Occupancy: Lights can be programmed to turn on and off at specific times of day or night, mimicking the routine of the homeowner. For instance, lights can be set to turn on in the evening and off at night, giving the impression that someone is home, even when the house is vacant.

Randomized Lighting: For added security, the lights can be randomized, creating unpredictable patterns that make it more difficult for burglars to discern whether the house is actually empty. This randomness can be programmed through the security system to match the homeowner’s typical habits, such as lights coming on in certain rooms at different times, further enhancing the illusion of occupancy.

Motion-Activated Lighting: Automated lighting can also work in conjunction with motion sensors to turn lights on when someone approaches the house or enters a room. This feature adds an extra layer of security, as it makes it difficult for intruders to approach the home without being noticed. Motion-activated lights can also help homeowners navigate their property without needing to manually turn lights on or off. Integrating automated lighting with the house security system, homeowners can effectively deter burglaries and increase the overall safety of their property, whether they are at home or away.

Integration with Smart Thermostats

Another valuable integration in a smart home security system is with smart thermostats. This connection can help optimize comfort and energy efficiency, while also adding an extra layer of security when combined with the security system's other features.

Temperature Adjustment Upon Alarm Trigger: When the security system detects an intrusion or is armed/disarmed, the smart thermostat can automatically adjust the temperature. For example, the thermostat can be set to turn off the heating or cooling system when the security system is armed to save energy and reduce the risk of leaving systems running unnecessarily while the home is unoccupied.

Temperature Control When Homeowners Leave or Return: The security system can be programmed to detect when homeowners leave or return, and automatically adjust the thermostat to a comfortable temperature. Upon departure, the system can set the temperature

to energy-saving levels, and when the homeowner returns, the thermostat will ensure the home is at a comfortable temperature, providing both convenience and efficiency.

Remote Temperature Control: Homeowners can also use their smartphones or smart home apps to control the thermostat remotely. This integration allows them to adjust the temperature of their home while they are away, ensuring that the property is always comfortable and preventing energy waste when no one is home.

With the integration of smart thermostats, homeowners can automate their energy usage, improve efficiency, and enhance comfort, while also creating a more responsive and secure home environment.

5.7 Access Logs and Activity Monitoring

An advanced and highly valuable feature in modern house security systems is the ability to track and log access activity. These logs and reports provide homeowners with a detailed record of the movements within their home, offering critical insights into both authorized usage and potential security threats. Access and activity tracking not only enhances security but also gives homeowners peace of mind, knowing they can monitor every entry and exit from the property.

Access Logs

Access logs are one of the fundamental components of any comprehensive security system. These logs maintain detailed records of who enters and exits the property, along with the time and date of each event. By consistently tracking this data, homeowners can ensure that only authorized individuals are entering their home and at the expected times.

Tracking Authorized Access: Homeowners can monitor when family members, guests, or service personnel enter the house. This is especially important for larger households or families with children, as it ensures that everyone is accounted for at all times. It also provides a record of any deliveries or visits from trusted service providers.

Unauthorized Access: In case of any unauthorized access, the system logs every entry attempt. If someone attempts to enter without proper authorization, such as an intruder or an unrecognized individual, the system will log this event and can alert the homeowner immediately, allowing them to take action if necessary.

Time Stamping: Each log entry is timestamped, providing an exact record of when the access occurred. This feature is crucial for reviewing incidents, especially in case of a security breach. The precise timing can help identify patterns of suspicious behavior or unauthorized activity, allowing homeowners to act swiftly to address any concerns.

These access logs become a key part of maintaining a secure home, providing both real-time monitoring and a historical record that can be reviewed at any time.

Activity Reports

In addition to access logs, activity reports offer a high-level overview of significant events within the home. These reports summarize important actions, security breaches, or events that have triggered the system’s alarms or sensors. By providing homeowners with clear and

concise summaries, activity reports help ensure that they are kept up-to-date with the latest security activity.

Summary of Key Events: Activity reports can include details about when the system was armed or disarmed, when doors or windows were opened, when security cameras detected motion, or when alarms were triggered. This allows homeowners to review all critical actions that may have taken place during the day or week, ensuring they never miss important events.

Triggered Alarms: The report can also highlight instances where alarms were set off—whether due to unauthorized access, motion detection, or other security-related triggers. Homeowners can then analyze the circumstances surrounding the alarm, helping them determine if the alert was legitimate or a false alarm.

Reports for Authorities: Activity reports can be easily shared with law enforcement or other authorities if a security breach occurs. These reports provide authorities with a clear timeline of events, which can aid in investigations and help track down culprits. By documenting every significant action, the system ensures that accurate and reliable information is available if needed.

Customizable Report Settings: Many modern security systems allow homeowners to customize the types of reports they receive. They can set up alerts for specific events, or generate daily, weekly, or monthly summaries. This flexibility allows homeowners to choose the level of detail they wish to monitor and keep track of over time.

By incorporating activity reports, the system provides a comprehensive way for homeowners to stay informed about their property's security, offering not only real-time alerts but also detailed summaries of historical events.

CHAPTER 6

TESTING AND VALIDATION

This chapter delves into the extensive testing and validation process carried out to ensure the reliability, functionality, and accuracy of the home security system utilizing face recognition technology. The purpose of this phase is to evaluate how well the system performs under various real-world conditions and to confirm that it meets the desired security and convenience standards. The testing methodology is divided into several key components, each aimed at assessing a specific aspect of the system's performance.

Home Security System Testing and Validation

The system integrates hardware components (ESP32 Cam, SG90 Servo Motor, FTDI Adapter) with software components (face recognition algorithm, real-time alerts, server-side logic) to create an intelligent home security solution. This diagram illustrates the testing process at both the hardware and software levels, including how they interact.

6.1. Hardware Testing

The first stage of testing involves individual evaluation of the hardware components to ensure they work properly before integration.

ESP32 Cam Testing

- Purpose: This component captures facial images and is the first point of interaction with the user. It processes the images and sends them for recognition.

Test Focus:

- Image Capture: Verify the camera can capture clear, high-quality images under different conditions (e.g., low light, varying facial expressions).
- Data Transmission: Ensure that data (image) captured by the camera can be transmitted to the software (FastAPI backend) for processing.

SG90 Servo Motor Testing

- Purpose: The servo motor is responsible for physically locking and unlocking the door based on the recognized face.

Test Focus:

- Response Time: The servo motor must react quickly to an authenticated face, unlocking the door within a 3-10 second window.

Accuracy: Ensure the motor locks securely and does not malfunction.

FTDI Adapter Testing

- Purpose: Manages communication between the system’s hardware (ESP32 Cam, servo) and the internet through Wi-Fi for real-time updates and connectivity.

Test Focus:

- Wi-Fi Connectivity: Ensure the system connects to the home Wi-Fi network properly, ensuring smooth communication.
- Reset Functionality: Check that the system can be reset through the adapter without issues, ensuring recovery from errors.

6.2. Integration Testing

Once each hardware component is tested individually, they must work together as an integrated system. This step checks the interaction between components and validates how well they cooperate in a real-world setting.

Facial Recognition Accuracy Testing

- Purpose: The face recognition algorithm compares the face captured by the ESP32 Cam with stored authorized faces to determine whether access should be granted.

Test Focus:

- Environmental Factors: Test the system under different lighting conditions, such as bright light, dim light, or night-time conditions, to ensure that the camera can still recognize faces effectively.
- Facial Angles: Test recognition accuracy from different angles (e.g., frontal, side profile) to ensure robustness.
- False Positives/Negatives: Measure how often the system mistakenly grants access to unauthorized faces or fails to recognize authorized individuals.

Real-Time Alert Functionality

- Purpose: The system must alert the homeowner when unauthorized access is detected.

Test Focus:

- Buzzer Activation: Test that the buzzer triggers accurately and immediately upon the detection of an unauthorized face.
- Notification System: Ensure that the system can send notifications or alerts to the homeowner's mobile device (if integrated) or other communication systems, ensuring immediate action can be taken.

6.3. Testing Scenarios

Specific test cases are created to simulate real-life conditions to ensure the system's robustness and accuracy.

Lighting Conditions

- Test Purpose: Lighting can significantly affect the accuracy of facial recognition. Different environments (e.g., bright sunlight, low indoor lighting) will be tested to see how well the system adapts to various light levels.
- Expected Outcome: The camera should maintain a high recognition accuracy across different lighting conditions.

Face Angle

- Test Purpose: Faces viewed from different angles can present challenges for facial recognition.
- Expected Outcome: The system should reliably recognize faces even when they are presented at varying angles (e.g., frontal, side profile).

Servo Lock Response

- Test Purpose: Once a face is recognized as authorized, the servo motor must act promptly to unlock the door.
- Expected Outcome: The servo motor should engage the lock with precision, unlocking the door promptly without delays.

Buzzer Test

- Test Purpose: The buzzer should alert the homeowner if an unauthorized face is detected.
- Expected Outcome: The buzzer should sound clearly and immediately after detecting an unrecognized face.

6.4. Response Time Testing

One of the most important aspects of the home security system is the response time—how quickly the door unlocks once a recognized face is detected and how quickly alerts are triggered in case of unauthorized access.

Unlocking Response Time

- Test Purpose: Ensure that the door unlocks within 3-10 seconds of the authorized face being recognized.
- Expected Outcome: The system should consistently meet this time frame for all authorized users.

Unauthorized Access Response Time

- Test Purpose: Ensure that the system responds promptly to unauthorized access attempts by triggering the buzzer and notifying the homeowner.
- Expected Outcome: The system should immediately alert the homeowner upon detecting an unrecognized face.

6.5. User Testing and Feedback

The usability of the system is tested by real users who interact with the system, providing feedback on their experience. This phase is essential to ensure that the system is intuitive, user-friendly, and effective in providing security.

Test Focus:

- User Experience: Gather feedback from homeowners on how easy and convenient it is to use the system in daily life.
- Ease of Use: Evaluate whether the system is simple to set up, whether it requires any technical expertise, and how intuitive it is for everyday users.

6.6. Performance Benchmarking

At this stage, the system is compared against performance benchmarks or industry standards. This ensures the system not only functions correctly but also meet expectations in terms of security, accuracy, and usability.

Test Focus:

- Facial Recognition Accuracy Benchmarking: Compare the system’s recognition accuracy against known benchmarks for similar security systems.
- Real-Time Performance: Evaluate the system’s response time, especially in various real-world conditions.

6.7. Final System Validation

The final validation phase involves ensuring that the entire system meets the specifications and objectives laid out in the project. At this point, all the individual test results are reviewed, and any issues found during testing are addressed and refined. The system is then ready for deployment.

Test Focus:

- System Robustness: Verify that the system operates without major issues in real-world conditions and is stable over time.
- Security Validation: Confirm that the system provides adequate security by testing its ability to defend against unauthorized access attempts.
- User Satisfaction: Final user feedback ensures that the system meets the expectations and offers value in terms of both convenience and security.

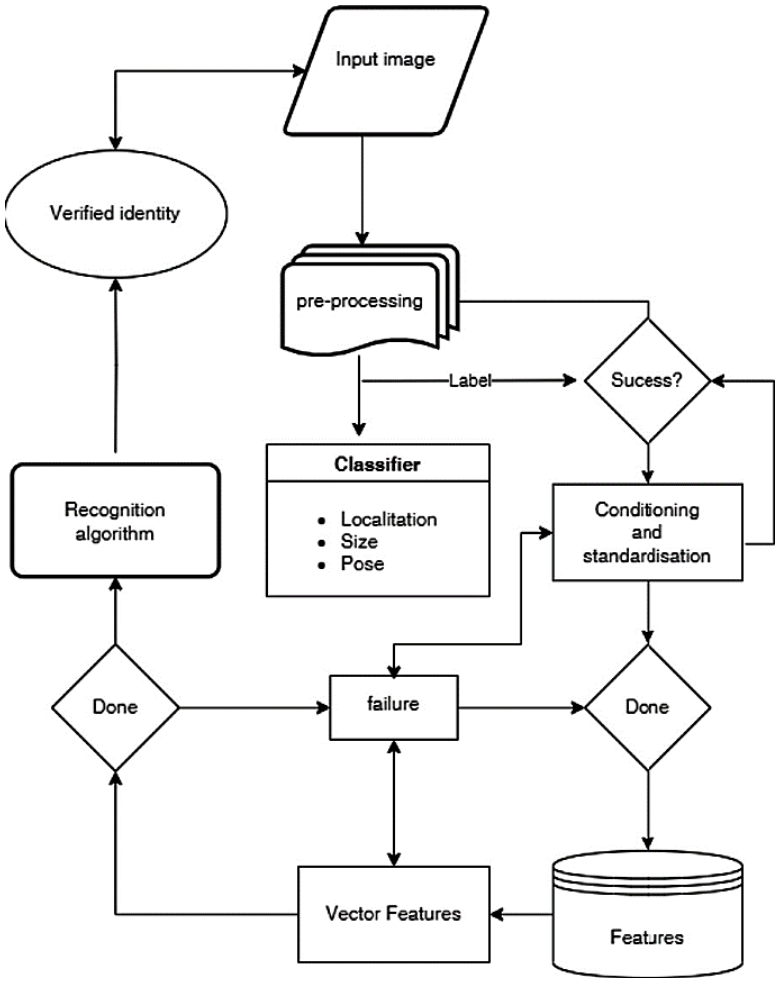


Fig.6.7.1: Application of Face Recognition testing and Dubbing.

Firstly, the facial recognition accuracy is thoroughly tested. This involves testing the system's ability to accurately identify authorized individuals under a variety of scenarios. Different lighting conditions, facial angles, and partial face obstructions are simulated to evaluate the robustness of the face recognition algorithm. Additionally, the system's performance is tested on a diverse set of individuals to determine how well it handles variations in facial features such as age, gender, and skin tone. The recognition accuracy is then quantified by comparing

the number of correct identifications to the total number of attempts, ensuring that the system consistently identifies authorized users with minimal false positives or false negatives.

The response time of the system is another critical factor under evaluation. The system is tested to ensure that the door unlocks within the target time frame of 3-10 seconds after an authorized face is recognized. This metric is essential for user convenience, as the system must provide quick and seamless access to homeowners while maintaining security. Testing is also conducted to evaluate the door locking mechanism's performance in the event of unauthorized access, ensuring that the servo motor functions properly and securely engage the lock.

The real-time alert functionality is also a key focus during testing. The buzzer and notification system are assessed to confirm that the homeowner is promptly alerted in the case of unauthorized access attempts. Various scenarios are created to simulate potential intruder attempts, and the system is evaluated on its ability to trigger alarms accurately and at the right time, ensuring that homeowners are aware of potential threats immediately.

Component testing is another important aspect of this chapter. Individual hardware components such as the ESP32 Cam, SG90 servo motor, and FTDI adapter are tested for their functionality in isolation before being integrated into the complete system. This ensures that each part works as expected and minimizes the possibility of hardware failures when components are combined.

Once individual components are verified, integration testing is carried out to assess the system's overall performance. This includes testing the communication between the hardware and software components, as well as the interaction between the facial recognition library, FastAPI server, and SQLite database. The integration testing phase ensures that data flows correctly between the system components and that the complete system functions as intended.

User feedback plays a crucial role in the validation process. User testing is conducted to assess the system's usability and practicality. Homeowners are asked to interact with the system and provide feedback on its ease of use, convenience, and overall effectiveness in real-life scenarios. This feedback is analyzed to identify any usability issues, user preferences, or potential improvements to the system’s design and functionality.

Finally, performance benchmarking is performed to compare the system’s results against expected outcomes and industry standards. The results of all tests are analyzed, and any issues identified during testing whether in recognition accuracy, response time, or alert functionality are addressed and refined to enhance the system's overall reliability and user experience. The chapter concludes by confirming that the system is capable of providing a robust, secure, and efficient solution for home security.

Through this comprehensive testing and validation process, the system is fine-tuned to ensure optimal performance in real-world deployment, offering homeowners a reliable and effective security solution.

CHAPTER 7

CHALLENGES AND LIMITATIONS

7.1 Technical Challenges Faced

During the development and implementation of the home security system using face recognition technology, several technical challenges were encountered. These challenges were primarily related to the integration of hardware and software, as well as ensuring the system's robustness under different conditions. Below are some of the key technical challenges faced during the project:

7.1.1. Face Recognition Accuracy and Reliability

One of the major challenges was ensuring the accuracy and reliability of the facial recognition algorithm. While face recognition technology has seen significant advancements, it is still sensitive to various factors such as:

- Lighting Conditions: Low light or overexposure can affect the quality of the captured image, leading to inaccuracies in recognition.
- Face Angle: The system struggles when faces are presented from extreme angles, leading to a decrease in recognition accuracy.
- Facial Changes: Variations in facial appearance due to factors like aging, changes in hairstyle, or facial accessories (glasses, masks) can cause the system to misidentify users. To overcome these challenges, extensive testing was conducted to fine-tune the recognition model and improve its robustness under diverse conditions. However, ensuring consistent performance in all environments was a continuous challenge.

7.1.2. Hardware Integration and Communication

Integrating the hardware components (ESP32 Cam, SG90 servo motor, and FTDI adapter) with the software posed another significant challenge. Each hardware component needed to work seamlessly with the others, and establishing reliable communication between them was critical.

- Wi-Fi Connectivity: The ESP32 Cam relies on a stable Wi-Fi connection to send data to the server. Any interruptions or delays in Wi-Fi connectivity could result in lag or failure in facial recognition, which affects the overall system's performance.
- Servo Motor Response: The SG90 servo motor needed to respond quickly to face recognition data to unlock the door in real-time. However, delays or inconsistencies in communication between the camera and the motor could result in a laggy or unresponsive system.
- Data Transfer and Processing: Ensuring fast data transfer and processing from the camera to the recognition algorithm and from the system to the motor was crucial. Any bottleneck in data transfer could delay the entire process, reducing the efficiency of the system.

7.1.3. Real-Time Processing and Speed

The need for real-time processing was another technical challenge. For a security system to be effective, it must process the captured images, match them against a database of authorized faces, and trigger the door's locking mechanism (or sound an alarm) in a very short time frame (3-10 seconds).

The processing speed of the facial recognition model had to be optimized to minimize delays. The need for quick facial detection and matching without compromising accuracy was a constant balancing act.

The system required rapid image capturing, face comparison, and servo motor activation. Ensuring that all these tasks occurred quickly while maintaining high accuracy was technically challenging, especially given the processing power limitations of the ESP32 Cam.

7.1.4. Environmental Factors Affecting Recognition

Another challenge was the impact of environmental factors on the system's performance. These factors include:

- Lighting Variability: As mentioned earlier, varying lighting conditions can affect how the camera captures the face, leading to recognition issues. Bright sunlight, shadows, or indoor low light all presented challenges in achieving consistent performance.
- Weather and Temperature Conditions: External factors such as extreme temperatures or humidity could affect the hardware components, especially the camera and servo motor, potentially leading to system malfunctions or inconsistent behavior.

7.1.5. Security and Privacy Concerns

Facial recognition systems, while effective for security, also raise significant privacy and data security concerns. Storing facial data securely in the database and ensuring that unauthorized access to this data is prevented was crucial.

Ensuring the integrity and confidentiality of the stored data became a challenge, especially considering the risks associated with hacking or unauthorized access to sensitive information.

The system needed to have proper encryption and security protocols in place to safeguard users' personal information.

7.1.6. Power Consumption

The power consumption of the system, especially with continuous operation of the ESP32 Cam and servo motor, posed another challenge. Efficient power management was necessary to ensure that the system remained functional over extended periods without excessive power drain.

The challenge was to maintain a balance between performance and power consumption to avoid the need for frequent recharging or power source interruptions.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

The Smart to Let platform has been successfully developed and implemented to address the existing gaps in the property rental market in Bangladesh. By leveraging modern web development technologies such as Django, React, and SQL, the platform aims to transform the traditionally broker-driven rental market into a streamlined, digital experience for both landlords and tenants. A key outcome of this project is the creation of a user-friendly interface that simplifies tasks such as user registration, property listing, property search, and payment processing. Tenants can find properties based on their preferences using advanced search and filtering options, while landlords can easily list their properties and manage tenant inquiries. The platform's design emphasizes ease of use, ensuring accessibility for a broad range of users, regardless of their technical expertise. The integration of a secure payment gateway ensures safe, reliable, and convenient transactions between tenants and landlords. The platform also includes real-time notifications and an efficient communication system, allowing tenants and landlords to manage rental agreements and resolve issues promptly. Transparency and trust are fundamental to the success of *Smart to Let*. By eliminating the need for brokers and middlemen, the platform fosters direct communication and dealings between landlords and tenants, contributing to a more open and transparent rental experience. Furthermore, the platform's scalability ensures that it can accommodate increasing numbers of users, properties, and transactions, positioning it as a viable solution for the growing digital rental market in Bangladesh. The successful implementation of *Smart to Let* lays a solid foundation for future developments, such as the addition of advanced features and the potential expansion to other regions or countries.

8.2 Contribution to the Rental Market

The Smart to Let platform makes a significant contribution to the rental market in Bangladesh by transforming how both landlords and tenants interact within the real estate ecosystem. Traditionally, the rental process in Bangladesh has been heavily reliant on brokers, leading to inefficiencies, higher costs, and a lack of transparency. The introduction of Smart to Let addresses these challenges by providing a digital solution that removes the need for intermediaries and facilitates direct communication between landlords and tenants. This innovation greatly enhances the overall efficiency of the rental market and lowers transaction costs. One of the most crucial contributions of Smart to Let is its focus on transparency. In traditional rental markets, there is often a lack of clarity regarding property details, rental terms, and payment methods. Smart To Let resolves these issues by offering clear, accurate, and comprehensive property listings. Tenants can view detailed property descriptions, photographs, and even virtual tours, which help them make informed decisions before contacting landlords. Similarly, landlords can manage their properties, set rental terms, and communicate directly with potential tenants, ensuring that there is no room for miscommunication or hidden fees. Additionally, the secure payment gateway integration is a significant contribution to the rental market in Bangladesh. Traditionally, rental transactions

involved a mix of cash payments and bank transfers, which could be prone to fraud or disputes. With Smart to Let's integrated payment system, transactions are secure and traceable, providing a safer environment for both tenants and landlords. The secure nature of online payments also builds greater trust between the parties, ensuring that payments are made promptly and without complications. Furthermore, Smart to Let contributes to localizing the rental experience. Unlike international platforms like Airbnb or Zillow, which cater to a global audience, Smart to Let focuses exclusively on the Bangladeshi market, addressing local needs, preferences, and challenges. This localization ensures that tenants and landlords in Bangladesh can benefit from a platform that understands and caters to the specific characteristics of the rental market in the country. Smart To Let makes a meaningful contribution to the rental market by enhancing efficiency, transparency, security, and user experience. Its innovative approach to property listing, management, and transaction processing paves the way for a more modern, digital, and accessible rental market in Bangladesh. Through the use of technology, Smart to Let has revolutionized property rental, making it easier, safer, and more efficient for everyone involved.

8.3 Future Enhancements and Scalability

The Smart to Let platform has the potential for continuous growth and improvement, with several future enhancements and scalability options to consider:

1. Mobile Application Development: Developing a mobile app for Android and iOS will expand accessibility, allowing users to manage listings, communicate with potential tenants, and make payments on-the-go.
2. AI-Powered Recommendations: Implementing AI algorithms to provide personalized property recommendations based on user preferences and behavior, making it easier for tenants to find their ideal accommodation.
3. Geolocation Features: Integrating geolocation to display nearby available properties, enhancing the user experience by helping tenants find homes closer to their desired locations.
4. Multilingual Support: Expanding the platform to include multiple languages, making it more accessible to users from different regions of Bangladesh and catering to a more diverse user base.

These enhancements will ensure that Smart to Let remains scalable, adaptable, and capable of meeting the growing demands of Bangladesh's rental market.

REFERENCES

[1] Jain, A. K., Ross, A., & Nandakumar, K. (2011). *Introduction to biometrics*. Springer Science & Business Media.

[2] Phillips, P. J., et al. (2005). Overview of face recognition algorithms and technology. *IEEE Transactions on Circuits and Systems for Video Technology, 14*(1), 5-25.

[3] Ge, R., & Han, S. (2020). Home security system based on Internet of Things (IoT). *International Journal of Advanced Computer Science and Applications (IJACSA), 11*(9), 344-350.

[4] Li, X., Wang, X., & Zhou, Z. (2017). Smart home security system based on cloud computing and IoT. *Procedia Computer Science, 107*, 175-179.

[5] Ibrahim, A., & Nordin, M. (2013). A study on the design of an automatic home security system with real-time alert mechanism. *International Journal of Computer Applications, 72*(17), 27-34.

[6] Dufresne, A. (2018). IoT-based intruder alarm system. *Proceedings of the 4th International Conference on Internet of Things, Big Data and Security (IoTBDS)*, 123-129.

[7] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778.

[9] Espressif Systems. (2017). *ESP32 Technical Reference Manual*. Espressif Systems.

[10] Yu, Z., & Jiang, X. (2020). ESP32-based real-time security system design and implementation. *2020 IEEE International Conference on Consumer Electronics (ICCE)*, 1-6.

[11] SQLite Development Team. (2020). *SQLite Documentation*. SQLite.org.

[12] Prisma. (2021). *Prisma ORM Documentation.* Prisma.io.

[13] D. S. (2020). *FastAPI Documentation*. FastAPI.readthedocs.io.

[14] Grinberg, M. (2018). *Flask Web Development: Developing Web Applications with Python*. O'Reilly Media.

[15] K. B. (2019). Servo motor control and applications. *International Journal of Scientific & Engineering Research, 10*(5), 700-705.

[16] Sounding Alerts with Buzzer: Design and Implementation of Alarm Systems. *IEEE Transactions on Industrial Electronics, 2021*.

[17] Bradski, G. (2000). *The OpenCV Library*. Dr. Dobb's Journal of Software Tools.

[18] L. Ross, & A. Jain (2003). Multibiometric Systems and Applications. *Springer*.