IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

EYES ON ACCESS (FACE RECOGNITION DOOR UNLOCK)

Submitted by

Akilesh Prabhu S(714022205007)

Kathiravan T(714022205051)

Manikandan M(714022205059)

SRI SHAKTHI

INSTITUTE OF ENGINEERING AND TECHNOLOGY

ABSTRACT

The Smart Door Notification System is an innovative solution aimed at enhancing security and convenience in residential and commercial spaces. This system utilizes advanced technologies to notify homeowners or users about the status of their doors remotely Companies are getting rid of traditional fingerprinting scanners and creating massive business opportunities by adopting AI-based facial recognition technology. Some of the applications where its usage has become crucial are security & surveillance, authentication/access control systems, digital healthcare, photo retrieval, etc. Our objective was to create a much-needed device for current scenario, a face detection enabled door lock mechanism which will prevent spread of infection through door knobs, door- bells and also provide stronger security system

Face Recognition has always been one of the most fascinating and intriguing technologies as it deals with human faces. Covid-19 outbreak has propelled the world to move towards touchless facial recognition technology. It is gaining huge traction worldwide owing to its contactless biometric features. Companies are getting rid of traditional fingerprinting scanners and creating massive business opportunities by adopting AI-based facial recognition technology. Some of the applications where its usage has become crucial are security & surveillance, authentication/access control systems, digital healthcare, photo retrieval, etc.

Our objective was to create a much-needed device for current scenario, a face detection enabled door lock mechanism which will prevent spread of infection through door knobs, door- bells and also provide stronger security system.

Our project is Multi User, that is, it will work for every person whose face was added into to the database. Our algorithm will also be Fool-proof, i.e., a real face is needed. It will not work on a digital photo or a printed photo.

CHAPTER1

INTRODUCTION

In an era marked by technological advancements and a growing emphasis on security and convenience, traditional methods of access control are being reimagined. The convergence of biometric recognition and smart access systems has given rise to innovative solutions, and one such groundbreaking technology is "Eyes on Access: Face Recognition Door Unlock."

Imagine a world where access to secure spaces is not reliant on keys, cards, or codes, but instead, a simple glance can grant you entry. This is the vision that "Eyes on Access" aims to fulfill through the integration of advanced face recognition technology and door unlocking mechanisms.

The Need for Evolution

Traditional access control methods, while effective to some extent, have their limitations. Keys can be lost or stolen, access codes can be forgotten or shared, and cards can be misplaced or cloned. Additionally, these methods often require physical interaction, which can be inconvenient and even unhygienic in certain contexts.

"Eyes on Access" seeks to address these challenges by harnessing the power of facial recognition – a biometric technology that is both inherently secure and remarkably convenient. The uniqueness of each individual's face, coupled with cutting-edge algorithms, enables accurate and swift identification. This revolutionary approach not only enhances security but also simplifies the user experience, transforming the way we interact with our surroundings.

How It Works

The core principle of "Eyes on Access" is the fusion of face recognition technology with door unlocking mechanisms. Enrolled users are securely registered within the system, with their facial features meticulously captured and stored as mathematical templates. When a user approaches a secured entry point, such as a door or gate, the system's camera captures their facial image in real-time. This image is then compared against the stored templates using advanced algorithms that analyze intricate facial details.

If a match is detected within a defined level of confidence, the door unlocking mechanism is

triggered, granting the user access. This entire process takes mere seconds, offering a seamless and efficient experience. Simultaneously, access events are logged, contributing to accountability and oversight.

Benefits at a Glance

"Eyes on Access" brings forth a host of benefits that redefine access control:

Enhanced Security: Face recognition is a robust biometric technology that is difficult to replicate or falsify.

Convenience Redefined: Users can access secure areas with a simple glance, eliminating the need for physical tokens.

Hygienic Solution: Contactless access aligns with modern hygiene standards and minimizes germ transmission.

Real-time Processing: Rapid recognition and response ensure minimal delays during access.

Audit Trail: A digital record of access events enhances monitoring and accountability.

As the world moves towards smarter, more connected environments, "Eyes on Access: Face Recognition Door Unlock" stands at the forefront of access control innovation. This documentation explores the intricate workings of this technology, from its technical underpinnings to practical implementation. Discover how "Eyes on Access" transforms the way we interact with security, efficiency, and the spaces around us.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the realm of modern security and access control, biometric authentication methods have gained significant prominence due to their inherent accuracy and convenience. Among these methods, facial recognition technology stands out as a powerful means of verifying an individual's identity. This technology holds immense potential for applications ranging from smartphone authentication to surveillance systems. One particularly intriguing application is its use for door unlocking, where the unique characteristics of an individual's face serve as the key to access. This literature review explores the evolution, advancements, challenges, and implications of employing facial recognition technology for door unlocking systems.Existing Solutions

2.2 Existing Solutions

The existing solution for "Eyes on Access: Face Recognition Door Unlock" might involve using a combination of hardware and software components. Here's a typical outline of the existing solution:

Hardware Setup:

A camera or a webcam is installed near the door to capture the image of the person standing in front of it.A microcontroller or a computer (like a Raspberry Pi) is used to process the images and control the unlocking mechanism.

Software Implementation:

Face detection and recognition software is installed on the microcontroller or computer.
The software analyzes the captured image to identify the person's face using facial features and patterns.
If the identified face matches a pre-registered face in the database, the software triggers the unlocking Mechanism

Database:

A database stores the facial features or patterns of authorized users. Each authorized user's face is captured and registered in the database.

Door Locking Mechanism:

An electric lock or mechanism controls the door's locking and unlocking.When the recognized face matches an authorized user, the software sends a signal to unlock the door.

2.2 Proposed Solution

The proposed solution could involve enhancements and improvements to the existing setup, addressing potential issues or limitations. Here are some ideas for a proposed solution:

Enhanced Face Recognition:

Use advanced facial recognition algorithms to improve accuracy and reduce false positives/negatives.

Implement anti-spoofing measures to prevent unauthorized access using photographs or videos.

Real-time Monitoring and Alerts:

Incorporate real-time monitoring capabilities to notify the system administrator or user when someone is at the door.

Send alerts or notifications via email, SMS, or a mobile app.

Integration with Access Control Systems:

Integrate the face recognition system with existing access control systems or smart home platforms for seamless management of access permissions.

User Management Dashboard:

Develop a user-friendly web-based dashboard to manage authorized users, add or remove access, and view access logs.

Privacy and Data Security:

Ensure compliance with data protection regulations by implementing secure data storage and encryption.

Provide users with control over their data and the ability to delete their information.

Redundancy and Failover:

Implement redundancy in case of hardware or software failures to maintain secure access.

User Experience:

Focus on a smooth and intuitive user experience during the face registration process and door unlocking

.

Remote Access:

Allow authorized users to grant access remotely to guests or visitors.

The proposed solution aims to enhance the accuracy, security, and usability of the face recognition door unlock system, making it a more reliable and user-friendly access control solution.CHAPTER 3

SYSTEM SPECIFICATION

3.1 HARDWARE SPECIFICATIONS

Processor : Intel® i-3 5th Gen (Minimum)

RAM : 4.00GB (3.76GB usable).

Hard disk Drive : 20GB

3.2 SOFTWARE SPECIFICATION

Operating System : Windows 8 (x64 bit) and above

Front-End : XML, Java

Back-End : SQLite

3.3 TECHNOLOGIES USED

3.3.1 Arduino IDE

The Arduino IDE (Integrated Development Environment) is a popular software platform used for programming and developing projects with a variety of microcontrollers, including the ESP32-CAM. The Arduino IDE provides a user-friendly interface and a set of libraries that simplify the process of writing code for embedded systems. Here's how the Arduino IDE is commonly used with the ESP32-CAM:

1. **Install the Arduino IDE:** To begin, you'll need to download and install the Arduino IDE on your computer. You can find the official Arduino IDE on the Arduino website. Make sure to select the appropriate version for your operating system.

2. **Install ESP32 Board Support:** The Arduino IDE doesn't natively support the ESP32 by default. However, you can add support for the ESP32 by installing the necessary board definitions. This enables the Arduino IDE to compile and upload code to the ESP32-CAM.

- Open the Arduino IDE.

- Go to "File" > "Preferences."
- In the "Additional Boards Manager URLs" field, add the following URL:

https://dl.espressif.com/dl/package_esp32_index.json

- Click "OK" to save the changes.
- Go to "Tools" > "Board" > "Boards Manager."
- Search for "esp32" and install the "esp32" platform by Espressif Systems.

3. **Select the ESP32-CAM Board:** Once the board support is installed, you can select the ESP32-CAM board in the Arduino IDE.

- Go to "Tools" > "Board" and select "ESP32 Wrover Module" as the board.
- Choose the appropriate options for "Flash Frequency" and "Flash Mode."

4. **Write and Upload Code:** With the ESP32-CAM board selected, you can now write your code in the Arduino IDE.

- Write your code in the Arduino IDE's editor.
- Use the "Verify" button to compile your code and check for errors.
- Use the "Upload" button to upload the compiled code to the ESP32-CAM. This will also initiate the process of flashing the firmware onto the board.

5. **Serial Monitor:** The Arduino IDE includes a serial monitor that allows you to communicate with the ESP32-CAM and monitor its output. This is particularly useful for debugging and viewing log messages.

6. **Library Usage:** The Arduino IDE supports libraries, which are pre-written code modules that simplify complex tasks. You can easily include libraries in your Arduino projects to add functionalities like Wi-Fi communication, camera control, and more.

7. **Examples and Documentation**: The Arduino community provides a wealth of examples, tutorials, and documentation for working with various components, including the ESP32-CAM. These resources can help you understand how to use the board's features effectively.

Remember that the specifics of using the Arduino IDE with the ESP32-CAM may vary based on updates, versions, and individual project requirements. Always refer to the most up-to-date documentation and resources for accurate guidance.

CHAPTER 4

SYSTEM DESIGN

4.1 BLOCK DIAGRAM

As Shown in fig 4.1 the video is divided into frames and later face is extracted from it afterpre-processing. The detected face is given as input for face recognition from which the Id of the detected face is obtained based on the obtained Id.

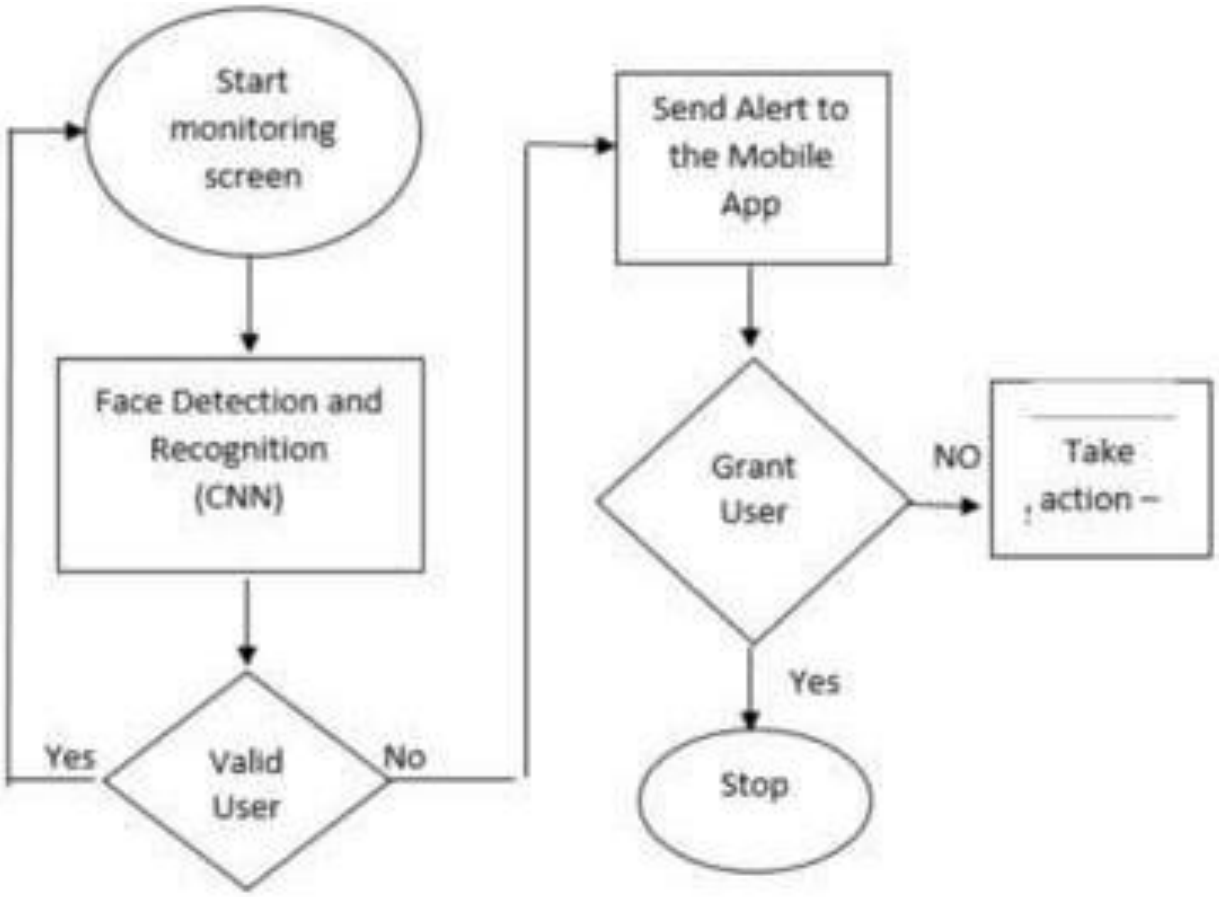


Figure 4.1: Block Diagram

4.2 FLOW CHART

The Algorithm flow of an admin is explained in Figure 4.2. It start from admin login,where this chart explain the flow of an admin workflow.

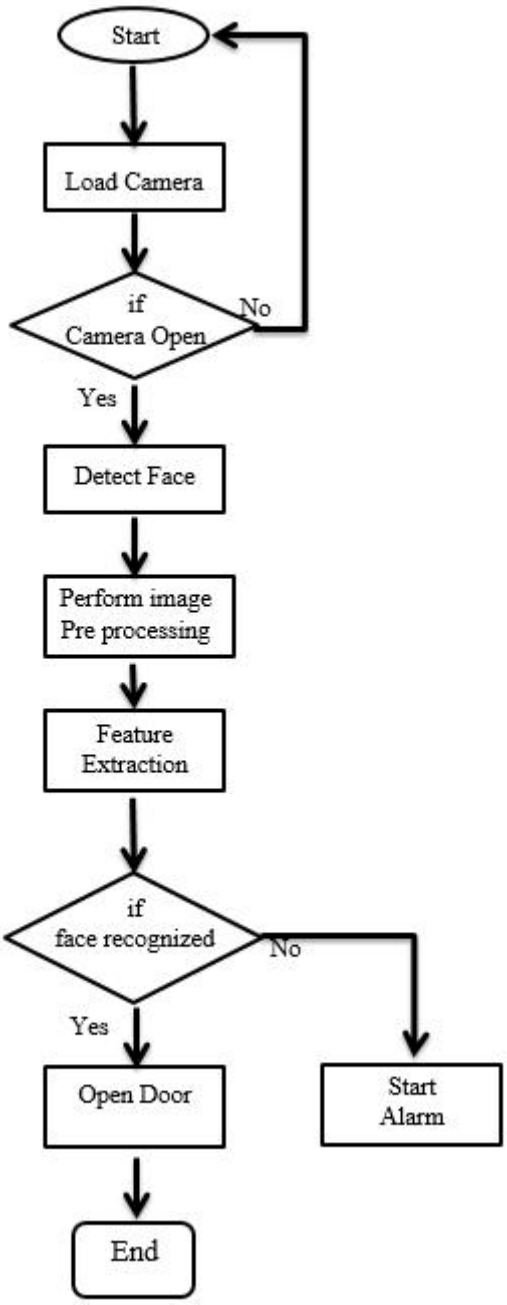


Figure 4.2: Flow Chart

4.3 PROCESS FLOW

The process flow of the project is explained in Figure 4.3. Initially the admin can able to add all the parking details and will be creating an parking slot and user can able to view that parkingfor the particular cities and able to book their parking and do the procedure to have payment

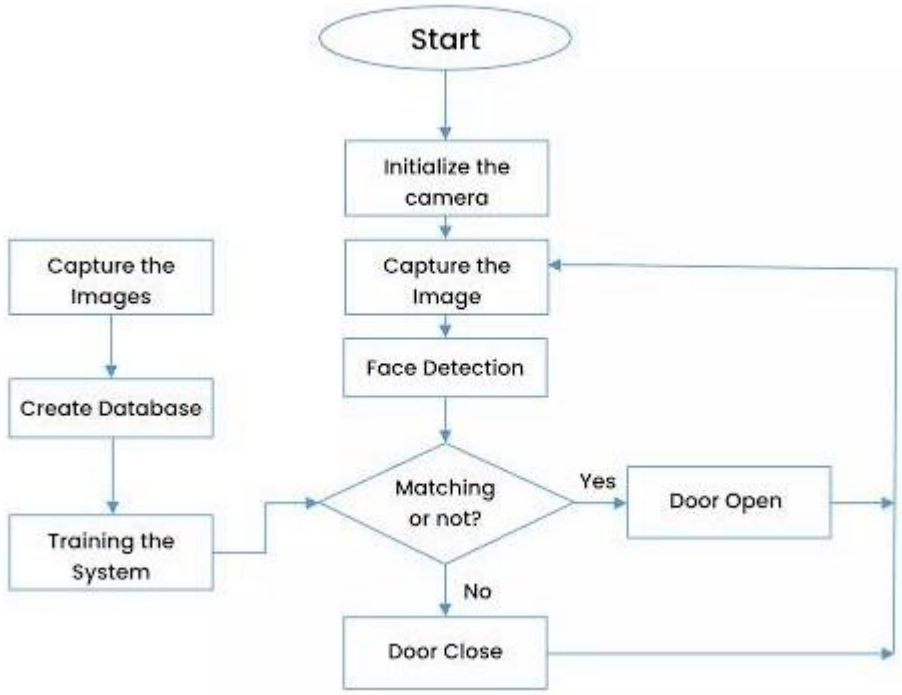


Figure 4.3: Process Flow

4.4 CLASS DIAGRAM

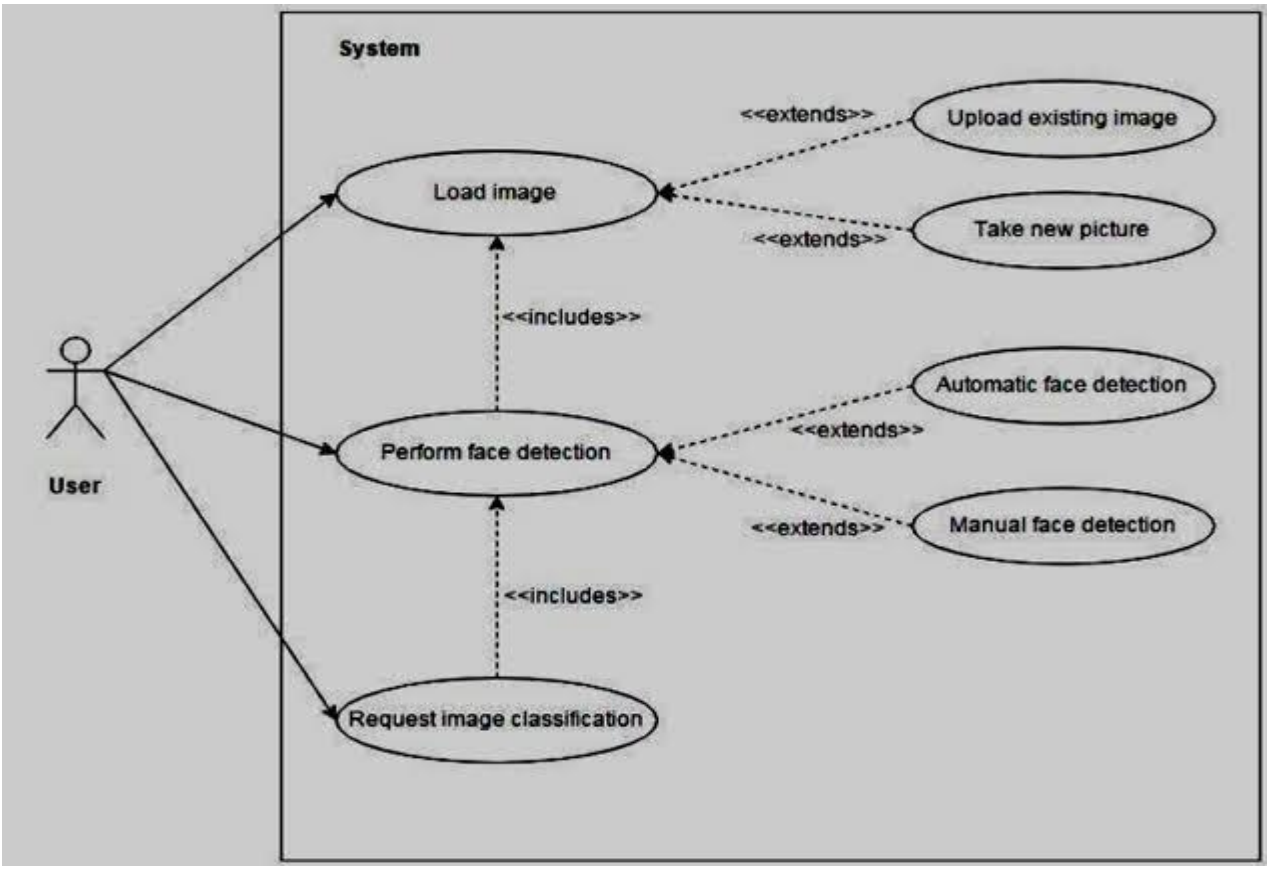


Figure 4.4: Class Diagram

4.5 Activity Diagram

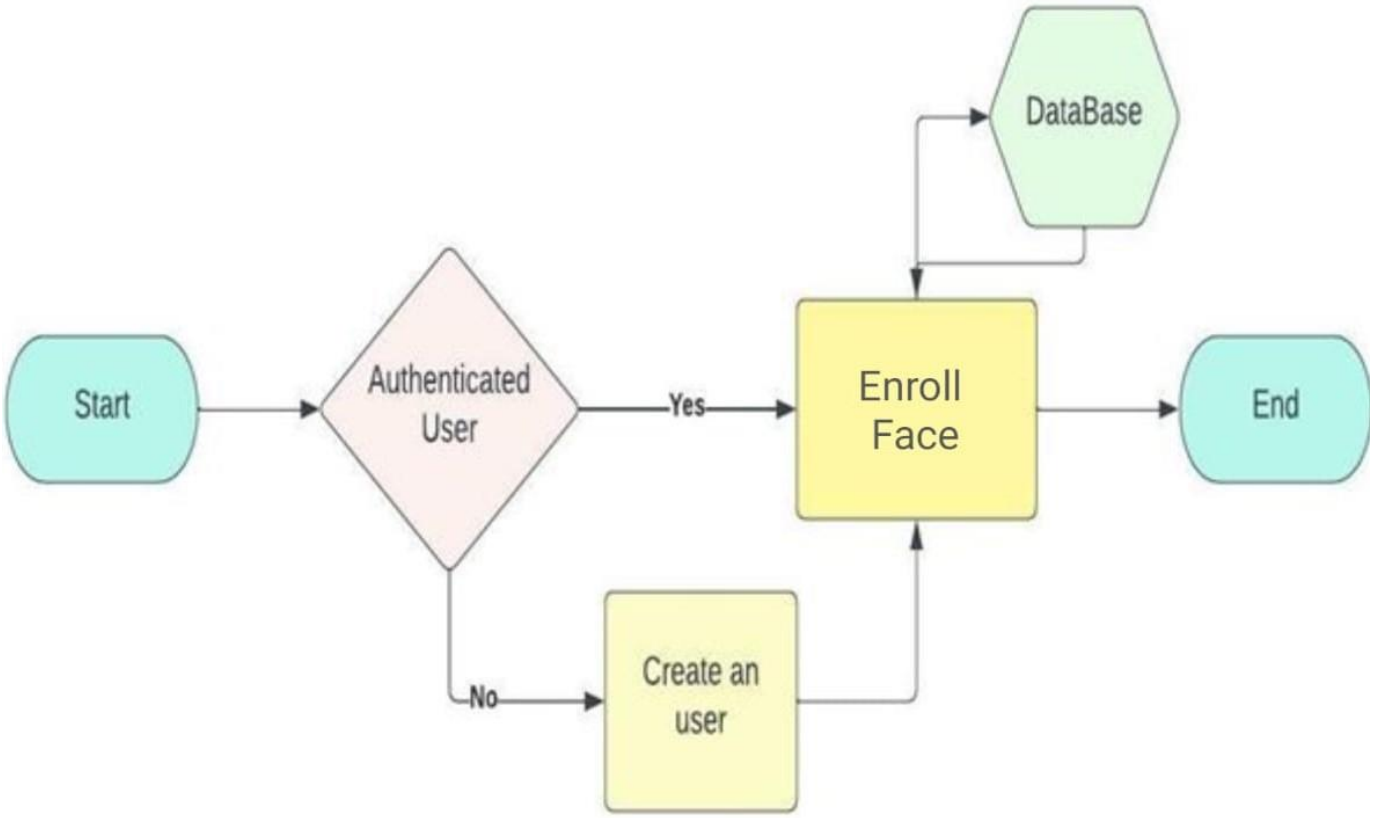


Figure 4.5: Activity Diagram

CHAPTER 5 IMPLEMENTATION

Security is at most concern for anyone nowadays, whether it's data security or security of their own home. With the advancement of technology and the increasing use of IoT, digital door locks have become very common these days. Digital lock doesn't require any physical key but it uses RFID, fingerprint, Face ID, pin, passwords, etc. to control the door lock. In past, we have developed many digital door locks applications using these various technologies. In this tutorial we build a **Face ID controlled Digital Door lock system using ESP32-CAM**.

The AI-Thinker ESP32-CAM module is a low-cost development board with a very small size OV2640 camera and a micro SD card slot. It has an ESP32 S chip with built-in Wi-Fi and Bluetooth connectivity, with 2 high-performance 32-bit LX6 CPUs, 7-stage pipeline architecture. We have previously explained Esp32-CAM in detail and used it to build a Wi-fi door Video doorbell . This time we will use the ESP32-CAM to build a **Face Recognition based Door Lock System** using a Relay module and Solenoid Lock.

COMPONENTS REQUIRED

- ESP32 CAM
- Relay Module
- Solenoid Lock
- LED
- Breadboard

- 12v power Supply / Battery
- 7805 Regulator
- 100uf 16V Capacitor
- wires

SOLENOIDAL LOCK

A solenoid lock works on the electronic-mechanical locking mechanism. This type of lock has a slug with a slanted cut and a good mounting bracket. When the power is applied, DC creates a magnetic field that moves the slug inside and keeps the door in the unlocked position. The slug will retain its position until the power is removed. When the power is disconnected, the slug moves outside and locks the door. It doesn't use any power in a locked state. To drive the solenoid lock, you would need a power source that can give 12V @ 500mA.

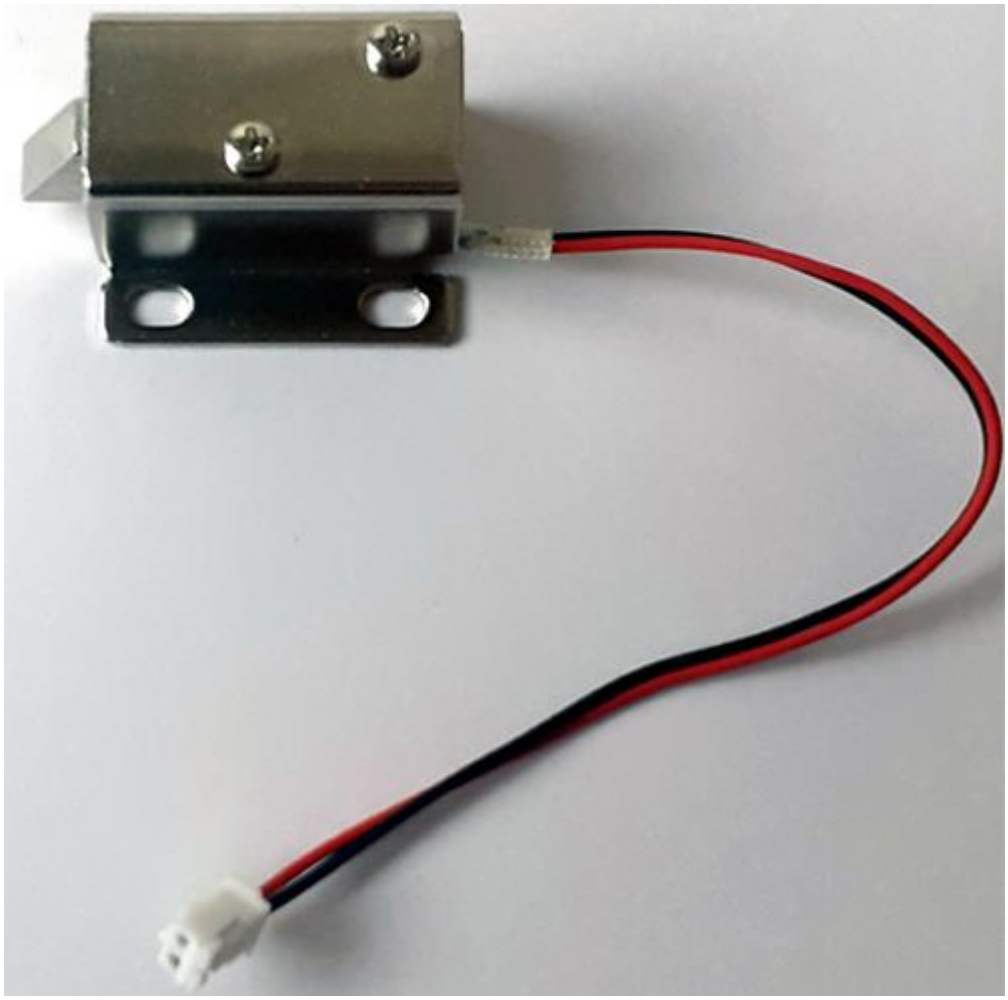


Figure 5.1 Solenoidal lock

CIRUCIT DIAGRAM

The Circuit Diagram for **ESP32-CAM Face Recognition Door Lock System** is given below:

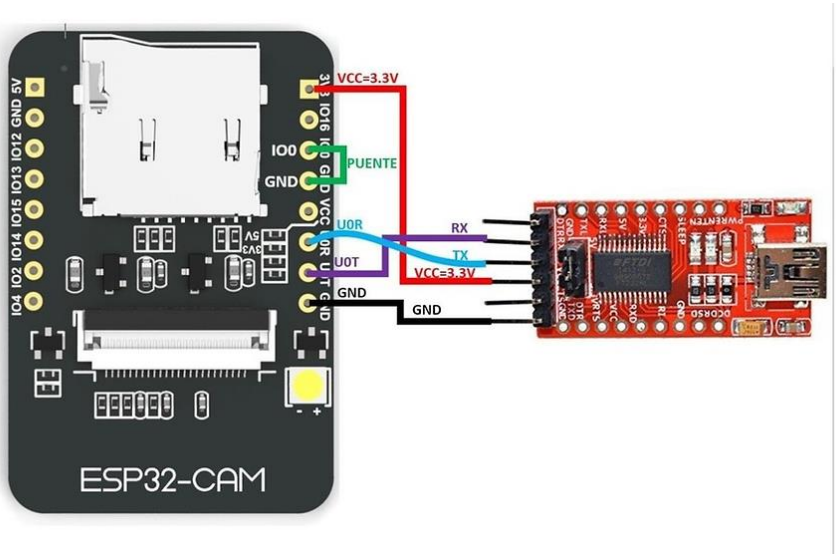


Figure 5.2 Cirucit Diagram 1

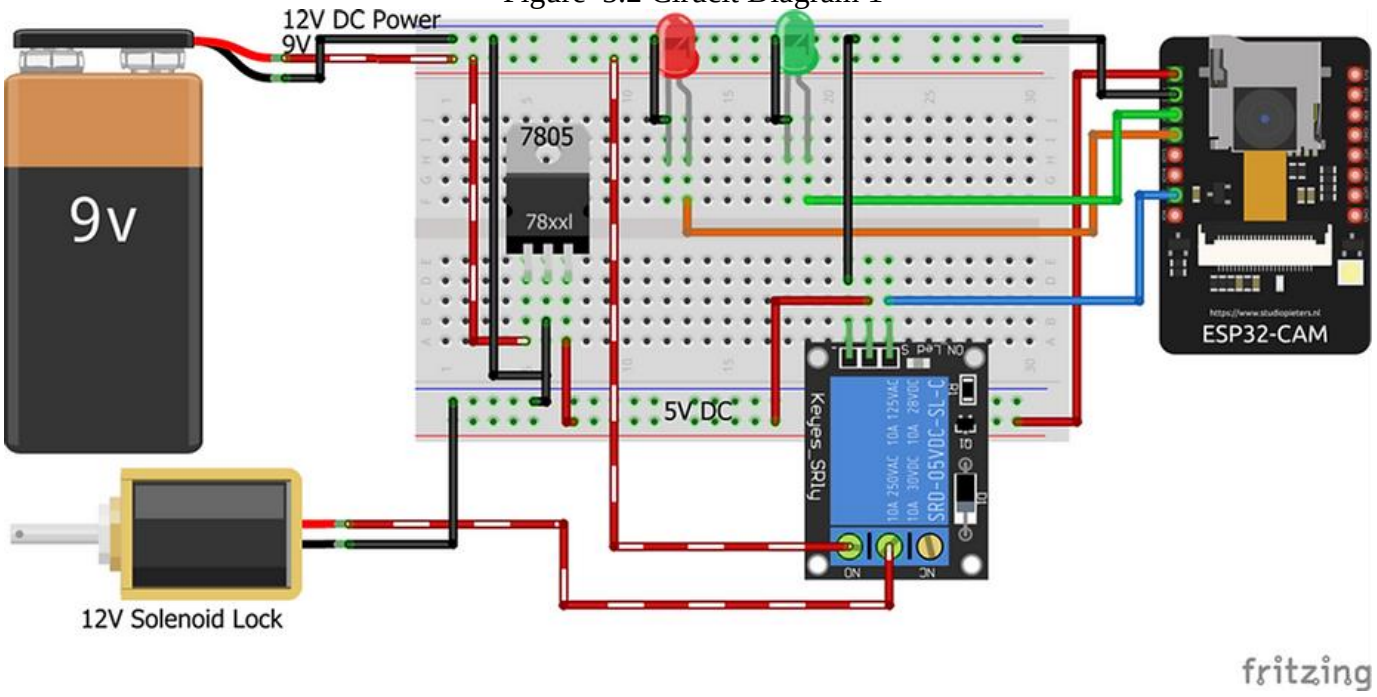


Figure 5.3 Circuit Diagram 2

The circuit above combined with an FTDI board, Relay Module, and Solenoid Lock. The FTDI board is used to flash the code into ESP32-CAM as it doesn't have a USB connector while the relay module is used to switch the Solenoid lock on or off. VCC and GND pins of the FTDI board and Relay module is connected to the Vcc and GND pin of ESP32-CAM. TX and RX of the FTDI board are connected to RX and TX of ESP32 and the IN pin of the relay module is connected to IO4 of ESP32-CAM.

CONNECTIONS

ESP32-CAM	FTDI Board
5V	VCC
GND	GND
UOR	TX
UOT	RX
ESP32-CAM	**Relay Module**
5V	VCC
GND	GND
IO4	IN

FIGURE 5.4 CONNECTION TABLE

INSTALL ESP32 BOARD ON ARDUINO IDE

Here Arduino IDE is used to program ESP32-CAM. For that, first, install the ESP32 add-on on Arduino IDE.

To install the ESP32 board in your Arduino IDE, go to ***File> Preferences.***

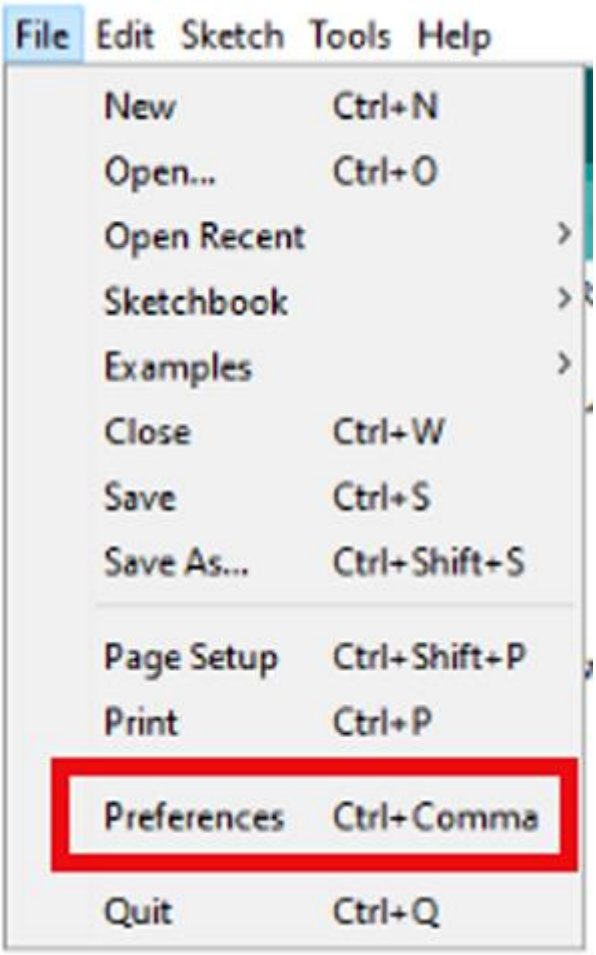


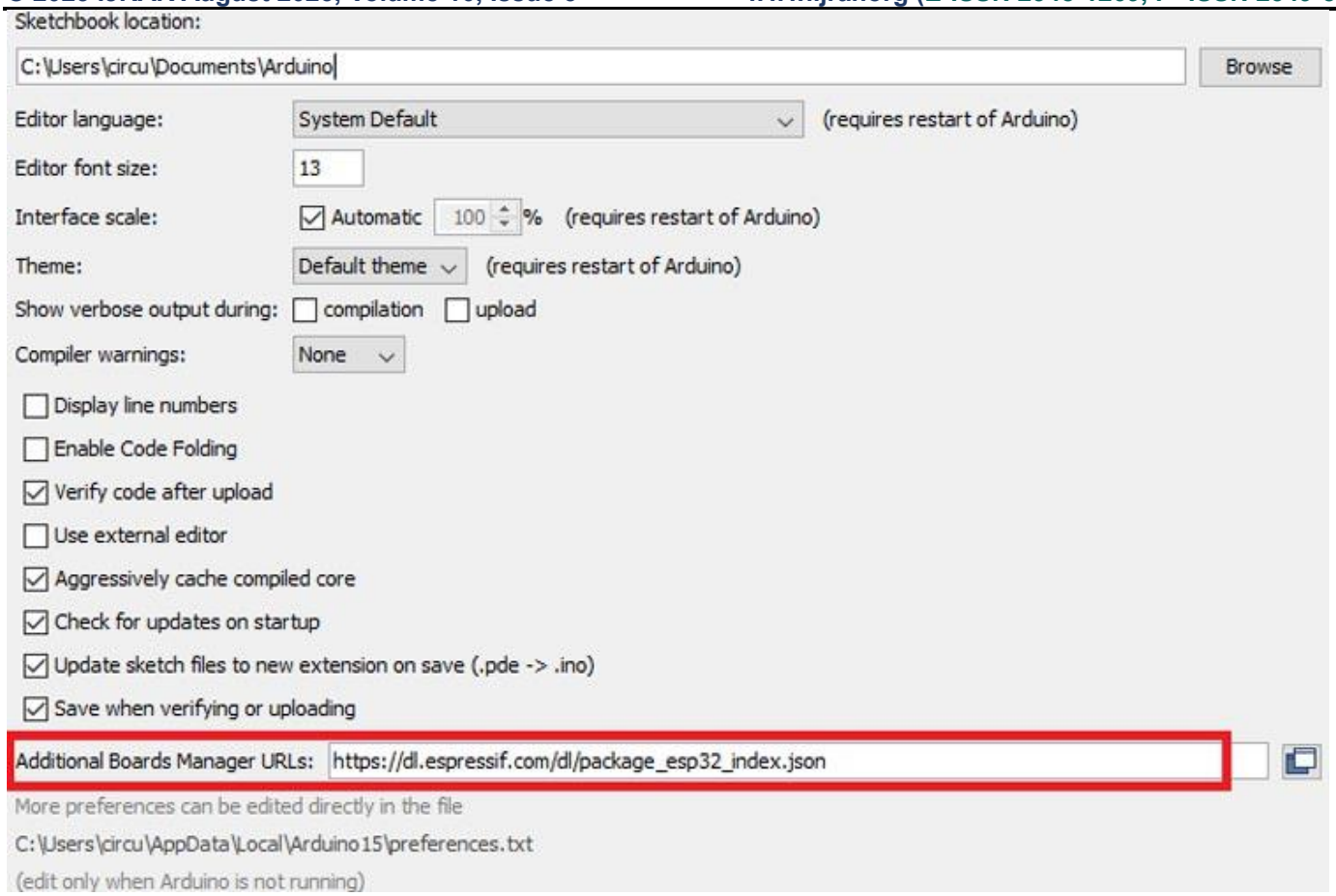
Figure 5.5 Prefrence

Now copy the below link and paste it into the “Additional Board Manager URLs” field as shown in the figure below. Then, click the “OK” button:

https://dl.espressif.com/dl/package_esp32_index.json

Figure 5.6 Linking Header Files

Now go to ***Tools > Board > Boards Manager***



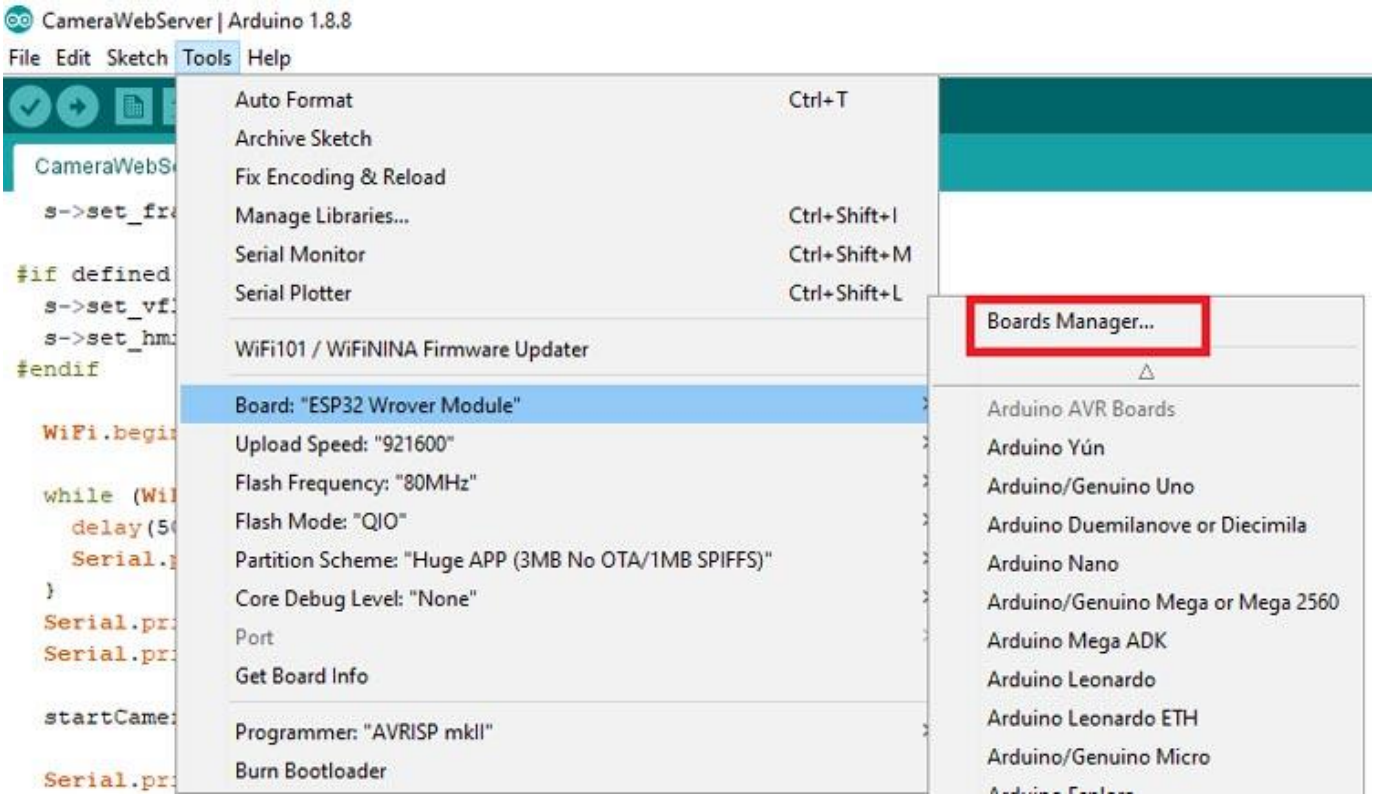


Figure 5.7 Board Manager

In Board Manager, search for ESP32 and install the "ESP32 by Espressif Systems".

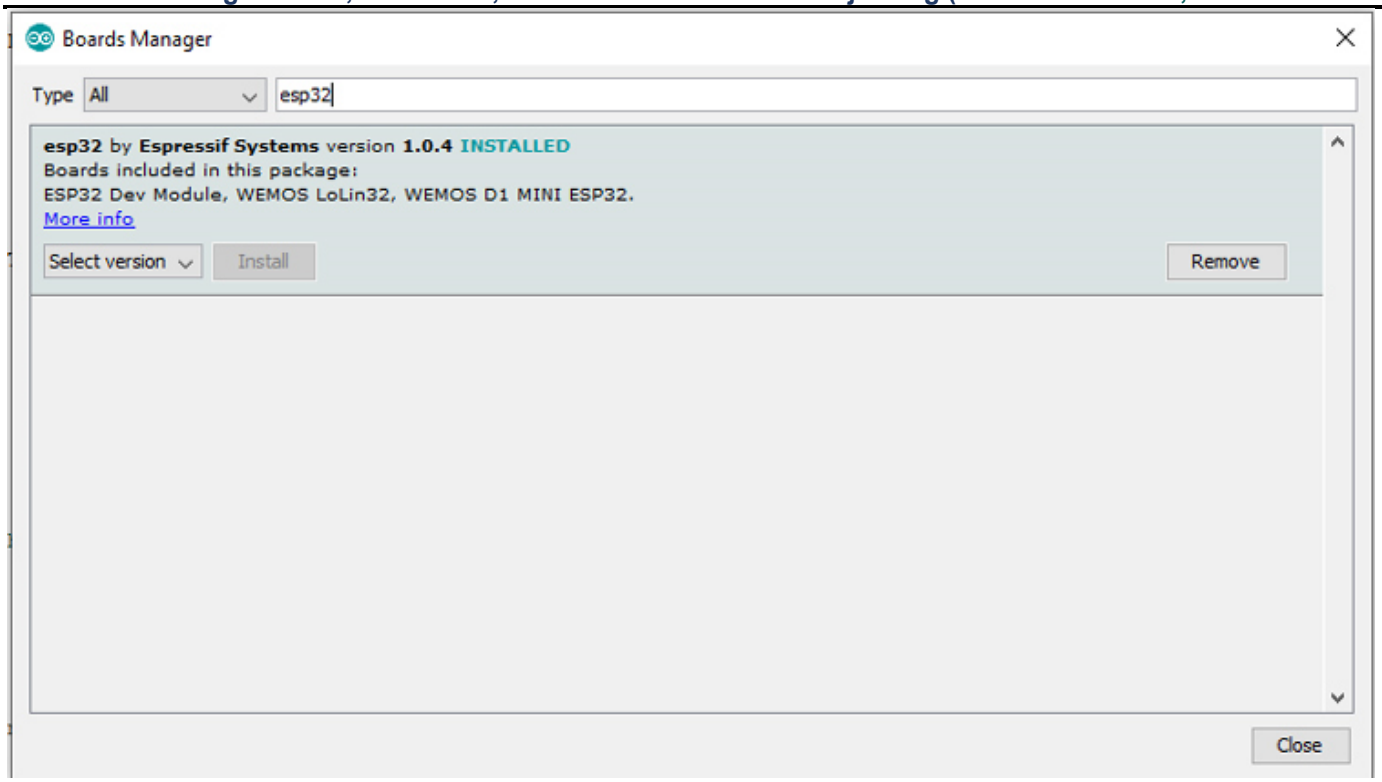


Figure 5.8 Installing Headers

After completing this verify and run the code

CHAPTER 6TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. There are various types oftests.Each testtype addresses a specific testing requirement.Testing is a series of different tests that whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all work should verify that all system element have been properly integrated andperformed allocated function. Testing is the process of checking whether the developed system works according to the actual requirement and objectives of the system.

The philosophy behind testing is to find the errors. A good test is one that has a high probability offinding an undiscovered error. A successful test is one that uncovers the undiscovered error. Test cases are devised with this purpose in mind. A test case is a set of data that the system will processas an input. However the data are created with the intent of determining whether the system will process them correctly without any errors to produce the required output.

6.1 UNIT TEST

Unit testing is an essential practice in software development that involves testing individual components or units of code in isolation to ensure they work as intended. When developing a complex system like "Eyes on Access: Face Recognition Door Unlock," unit testing helps identify and fix issues early in the development process. While unit testing might be more commonly associated with traditional software development, you

can still apply the concept to certain parts of your embedded system, particularly the software components**.**

Please note that unit testing on embedded systems can be more challenging due to hardware dependencies, limited resources, and the need for real-time interaction. The example provided is simplified and may need adjustments to fit your specific system and libraries.

For more comprehensive testing, you might consider using a testing framework compatible with your platform. While the Arduino environment may not offer the same level of testing tools as full-fledged software development environments, you can still adopt practices like code review, manual testing, and simulation to ensure the reliability of critical components such as your face recognition algorithm.

6.2 SYSTEM TEST

System testing is a crucial phase in the development lifecycle where the entire system is tested as a whole to ensure that it meets the specified requirements and functions correctly. In the context of "Eyes on Access: Face Recognition Door Unlock," system testing involves evaluating the complete integration of hardware, software, and other components to verify that the system behaves as expected and achieves its intended goals.

6.2.1 **Define Test Scenarios and Use Cases:**

i. Identify various scenarios that users might encounter when interacting with your system.
ii. Create test cases that simulate these scenarios, covering actions like enrollment, recognition, successful access, access denial, and error handling.

6.2.2 **Test Data Preparation:**

i. Gather a diverse set of test images that represent different lighting conditions, angles, facial expressions, and subjects.
ii. If possible, create a dataset with known faces for testing recognition accuracy.

6.2.3 **Functional Testing:**

i. Test the core functionalities of the system, such as face enrollment, recognition, door unlocking, and access denial.
ii. Verify that the system responds appropriately to different users and situations.

6.2.4 **Integration Testing:**

i. Test the integration between different components of the system, including the camera, facial recognition algorithm, access control mechanism, and user interface (if applicable).
ii. Verify that data is passed correctly between components.

6.2.5 **Performance Testing:**

Evaluate the system's performance under various conditions, such as different lighting levels, crowded environments, and rapid succession of access attempts.

CHAPTER 7 DEPLOYMENT AND MAINTENANCE

Program test

Finally to upload the code, connect the FDTI board to your laptop, and select the 'ESP32 Wrover Module' as your board. Also, change the other settings as shown in the Figure 6.1.

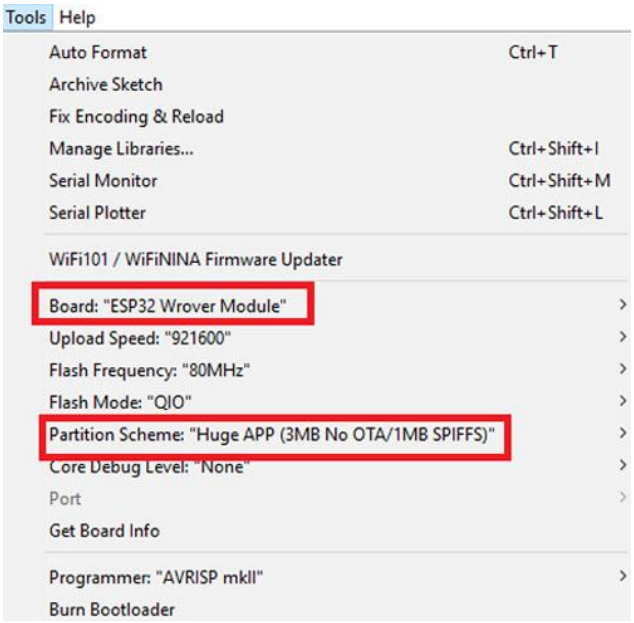


Figure 7.1 ESP32 Wrover Module

Don't forget to connect the IO0 pin to GND before uploading the code and also press the ESP32 reset button and then click on the upload button.

Note: If you get errors while uploading the code, check that IO0 is connected to GND, and you selected the right settings in the Tools menu.

After uploading the code, remove the IO0 and GND pin. Then open the serial monitor and change the baud rate to 115200. After that, press the ESP32 reset button, it will print the ESP IP address and port no. on the serial monitor.

```
ets Jun  8 2016 00:22:57

rst:0xc (SW_CPU_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1216
ho 0 tail 12 room 4
load:0x40078000,len:9720
ho 0 tail 12 room 4
load:0x40080400,len:6352
entry 0x400806b8

.
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.43.226' to connect
```

Figure 7.2 ESP-IP-Address

Now navigate to the browser and enter the ESP IP address that is copied from the Serial monitor to access the camera streaming. It will take you to the streaming page. To start the video streaming, click on the 'Start Stream' button at the bottom of the page.

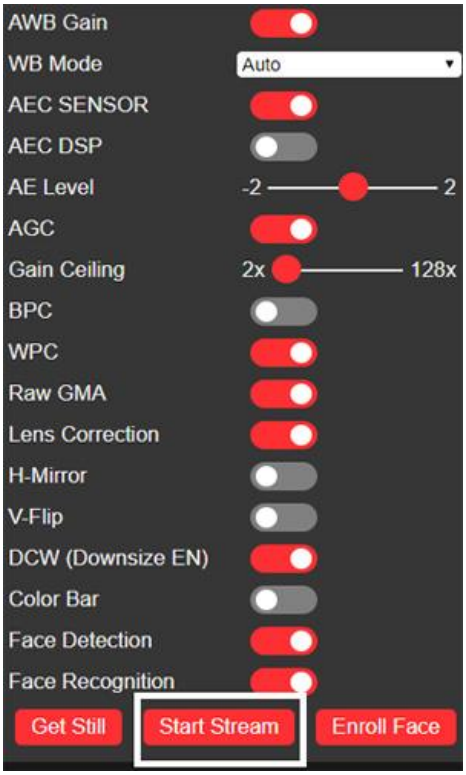


Figure 7.3 ESP-CAM-Testing

To recognize the faces with ESP32-CAM, first, we have to enroll the faces. For that, turn on the Face recognition and detection features from settings and then click on the Enroll Face button. It takes several attempts to save the face. After saving the face, it detects the face as **subject 0** where zero is the face number.

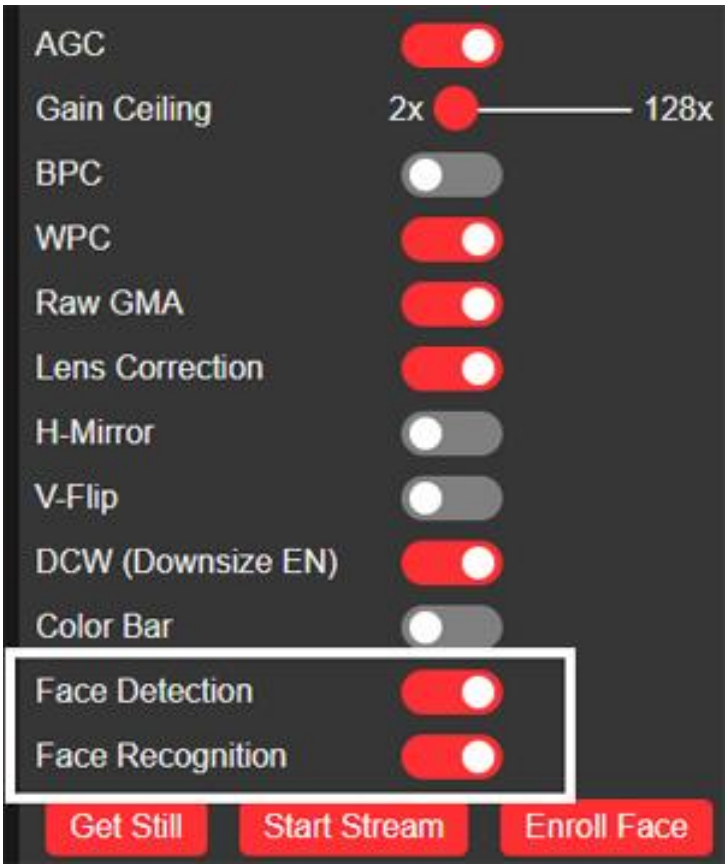


Figure 7.4 Face Recognition Using ESP32 CAM

Now enroll the face to detect and it takes 5 samples .After enrolling the faces, if a face is recognized in the video feed, ESP32 will make the relay module high to unlock the door.

CHAPTER 8 CONCLUSION AND FUTURE SCOPE

The "Eyes on Access: Face Recognition Door Unlock" system represents a significant advancement in access control technology, leveraging the power of facial recognition and smart door mechanisms to create a secure, convenient, and futuristic solution. By combining the unique biometric characteristics of individuals' faces with seamless access, this system addresses the limitations of traditional access methods and enhances both security and user experience.Through this documentation, we've explored the core principles of the system, delving into the technical workings of face recognition, its integration with the ESP32-CAM platform, and the implementation of a functional prototype. We've highlighted the benefits of contactless access, real-time identification, and audit trail logging, all contributing to a comprehensive access control solution.

Future Scope:

While the "Eyes on Access" system offers a groundbreaking solution, there are several avenues for further exploration and enhancement:

1.Multi-Factor Authentication: Introducing additional authentication factors such as voice recognition, fingerprint scanning, or RFID cards can bolster security.

2.Machine Learning: Implementing machine learning algorithms can continuously improve the accuracy of face recognition, adapting to different lighting conditions and facial expressions.

3.Cloud Integration: Integrating with cloud services allows remote monitoring, data storage, and remote access control management.

4.User Management: Developing a user-friendly interface for enrolling and managing users, along with user-level access permissions.

5.Privacy Considerations: Addressing privacy concerns through transparent data handling practices and uer consent mechanisms.

6.Customization: Allowing users to customize access rules and notifications based on their preferences and needs.

7.Security Enhancements: Implementing encryption and secure communication protocols to safeguard data during transmission.

The "Eyes on Access: Face Recognition Door Unlock" system has the potential to reshape access control practices, bringing us closer to a world where security and convenience seamlessly coexist. As technology evolves and user needs change, ongoing development and innovation will pave the way for safer, smarter, and more inclusive access solutions.

CHAPTER 9 REFERENCES

[1]Multi-faces recognition process using Haar cascades and eigenface method T Mantoro, MA Ayu - 2018

[2] A Survey of Face Recognition Techniques Rabia Jafri* and Hamid R. Arabnia* Journal of Information Processing Systems, Vol.5, No.2

[3]https://arsfutura.com/magazine/building-a-face-recognitionpowered-door-lock/

[4] Arduino UNO user manual https://www.arduino.cc/en/Guide/ArduinoUno

APPENDICES

A. SOURCE CODE

Fronnt End (Web Page):

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Your Website</title>
   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
   <a class="navbar-brand" href="#">Eye on access</a>
   <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-
controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
   </button>
   <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav ml-auto">
         <li class="nav-item active">
            <a class="nav-link" href="#home">Home</a>
         </li>
         <li class="nav-item">
            <a class="nav-link" href="#about">About</a>
         </li>
         <li class="nav-item">
            <a class="nav-link" href="#solution">Solution</a>
         </li>
         <li class="nav-item">
            <a class="nav-link" href="#architecture">Architecture</a>
         </li>
         <li class="nav-item">
            <a class="nav-link" href="#hardware">Hardware</a>
         </li>
         <li class="nav-item">
            <a class="nav-link" href="#software">Software</a>
         </li>

         <li class="nav-item">
            <a class="nav-link" href="#budget">Budget</a>
         </li>
         <li class="nav-item">
            <a class="nav-link" href="#contact">Contact</a>
         </li>
      </ul>
   </div>
</nav>
<div class="jumbotron text-center jumbotron-fluid" style="background-image: url('door.jpg');">
   <h1>Welcome to Our Website</h1>
```

```
    <br>
    <p>This is a simple jumbotron page created using Bootstrap.</p>
    <br><br><br><br><br><br><br><br>
    <a href="#" class="btn btn btn-outline-secondary ">All features</a>  
    <a href="#" class="btn btn btn-outline-secondary">View plans</a>
  </div>

<section id="about" class="py-5">
    <div class="container">
      <h2>About Us</h2>
      <p>Face recognition door unlock is a cutting-edge technology that leverages facial biometrics for secure and convenient access control. This system employs advanced algorithms to identify individuals based on unique facial features, such as the arrangement of eyes, nose, and mouth. When a person approaches the door, the built-in camera captures their face and analyzes key facial points. This data is then compared to a database of authorized individuals. If a match is found, the door is automatically unlocked, granting seamless entry.</p><br><p>

This technology offers several advantages over traditional methods. It eliminates the need for physical keys or access cards, reducing the risk of unauthorized duplication or loss. It also enhances security by making impersonation significantly more challenging. Additionally, face recognition door unlock systems are user-friendly, requiring only a natural glance at the camera, making them accessible to individuals of various abilities.</p><br><p>

However, concerns about privacy and potential inaccuracies in recognition algorithms have been raised. Developers are addressing these issues by implementing robust security measures and refining the technology's accuracy. As face recognition door unlock continues to evolve, it holds great promise for revolutionizing access control in various sectors, including residential, commercial, and industrial environments.</p>
    </div>
</section>
<section id="about" class="py-5">
    <div class="container">
      <h2>Solution</h2>
      <p>The solution of face recognition door unlock involves the integration of hardware and software components. Firstly, a high-resolution camera is installed near the entrance, positioned to capture the face of an individual approaching the door. The camera's feed is then connected to a processing unit, which can be a local device or a cloud-based server.</p><br><p>

On the software side, sophisticated facial recognition algorithms are employed. These algorithms analyze the captured facial image, identifying key facial landmarks and unique features. The data is then compared against a pre-existing database of authorized users. This database stores the facial templates of individuals granted access.</p><br><p>

If a match is found between the captured image and an entry in the database, the system triggers an unlocking mechanism. This mechanism could involve controlling an electronic door lock, releasing a latch, or activating an access control system.</p><br><p>

To enhance security and accuracy, the system incorporates anti-spoofing measures to prevent fraudulent attempts using photos or videos. Depth sensors, infrared cameras, and liveness detection techniques can be implemented to ensure that the face being presented is real and not a static image.</p><br><p>

Regular updates to the facial recognition model and continuous monitoring of system performance are crucial for maintaining effectiveness and adapting to changing conditions, such as variations in lighting or appearance due to factors like facial hair or accessories.</p><br><p>

Overall, the face recognition door unlock solution offers a blend of convenience and security, streamlining
```

*access control processes while safeguarding against unauthorized entry. *
* *
</section>
<section id="about" class="py-5">
* *
<h2>Architecture</h2>
* The architecture of a face recognition door unlock system typically involves multiple components working together to achieve accurate and secure access control. Here's a simplified overview of the architecture: *

*1. Camera: A high-resolution camera captures the facial images of individuals approaching the door. The camera should be positioned to ensure optimal image quality and capture various facial angles.
 *

*2. Image Preprocessing: The captured facial images undergo preprocessing, which includes tasks like image normalization, resizing, and noise reduction. This prepares the images for further analysis.
 *

*3. Feature Extraction: Advanced facial recognition algorithms analyze the preprocessed images to extract unique facial features and landmarks. These features are used to create a mathematical representation of the individual's face, often referred to as a facial template or embedding.
 *

*4. Database: The system maintains a database containing facial templates of authorized individuals. Each template is associated with user-specific information and access privileges.
 *

*5. Matching Algorithm: When a new facial template is extracted from a captured image, the system's matching algorithm compares it to the templates in the database. The algorithm calculates a similarity score to determine how closely the captured face matches the stored templates.
 *

*6. Threshold and Decision: A predefined similarity threshold is set to determine whether a match is successful. If the similarity score exceeds this threshold, the system considers the individual's face as authenticated.
 *

*7. Access Control Mechanism: Upon successful authentication, the system triggers an access control mechanism. This could involve unlocking an electronic door lock, granting access to a turnstile, or activating a gate.
 *

*8. Anti-Spoofing Measures: To prevent fraudulent attempts, the system may integrate anti-spoofing techniques. These measures detect indicators of spoofing attempts using photos, videos, or masks. Depth sensors, infrared cameras, or liveness detection algorithms can be used to ensure the presented face is live and not a reproduction.
 *

*9. Logging and Monitoring: The system maintains logs of access events, including timestamps and authenticated users. This information is crucial for audit trails and security monitoring.
 *

*10. Configuration and Management: Administrators can configure and manage the system, including adding or revoking access for individuals, adjusting settings, and monitoring system health.
 *

*11. Updates and Maintenance: Regular updates to the facial recognition algorithm, as well as maintenance of hardware components, ensure the system's accuracy and reliability over time.
 *

*12.Integration Options: The system can be integrated with other security measures, such as key card access or PIN codes, for multi-factor authentication.
 *

This architecture showcases the integration of hardware, software, and security measures to create a robust and effective face recognition door unlock system. It's important to note that actual implementations might involve more sophisticated components and considerations based on specific requirements and use cases.</p>

```
    </div>
</section>
<section id="about" class="py-5">
    <div class="container">
        <h2>Hardware</h2>
        <p>Your Hardware content goes here.</p>
    </div>
</section>
<section id="about" class="py-5">
    <div class="container">
        <h2>Budget</h2>
        <p>Your Budget content goes here.</p>
    </div>
</section>
<section id="about" class="py-5">
    <div class="container">
        <h2>Software</h2>
        <p>Your Software content goes here.</p>
    </div>
</section>

<section id="about" class="py-5">
    <div class="container">
        <h2>Contact</h2>
        <p>Your Contact content goes here.</p>
    </div>
</section>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>
</html>
```

Arduino IDE ESP32-CAM Connection code

```
#include "esp_camera.h"
#include <WiFi.h>

//
// WARNING!!! Make sure that you have either selected ESP32 Wrover Module,
//            or another board which has PSRAM enabled
//

// Select camera model
//#define CAMERA_MODEL_WROVER_KIT
//#define CAMERA_MODEL_ESP_EYE
//#define CAMERA_MODEL_M5STACK_PSRAM
//#define CAMERA_MODEL_M5STACK_WIDE
#define CAMERA_MODEL_AI_THINKER
#define Relay 2
#define Red 13
#define Green 12
#include "camera_pins.h"

const char* ssid = "-----------"; //Wifi Name SSID
const char* password = "----------"; //WIFI Password

void startCameraServer();

boolean matchFace = false;
boolean activateRelay = false;
long prevMillis=0;
int interval = 5000;

void setup() {
  pinMode(Relay,OUTPUT);
  pinMode(Red,OUTPUT);
  pinMode(Green,OUTPUT);
  digitalWrite(Relay,LOW);
  digitalWrite(Red,HIGH);
  digitalWrite(Green,LOW);

  Serial.begin(115200);
  Serial.setDebugOutput(true);
  Serial.println();

  camera_config_t config;
  config.ledc_channel = LEDC_CHANNEL_0;
  config.ledc_timer = LEDC_TIMER_0;
  config.pin_d0 = Y2_GPIO_NUM;
  config.pin_d1 = Y3_GPIO_NUM;
  config.pin_d2 = Y4_GPIO_NUM;
  config.pin_d3 = Y5_GPIO_NUM;
  config.pin_d4 = Y6_GPIO_NUM;
  config.pin_d5 = Y7_GPIO_NUM;
  config.pin_d6 = Y8_GPIO_NUM;
  config.pin_d7 = Y9_GPIO_NUM;
  config.pin_xclk = XCLK_GPIO_NUM;
```

```
  config.pin_pclk = PCLK_GPIO_NUM;
  config.pin_vsync = VSYNC_GPIO_NUM;
  config.pin_href = HREF_GPIO_NUM;
  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;
  config.pin_pwdn = PWDN_GPIO_NUM;
  config.pin_reset = RESET_GPIO_NUM;
  config.xclk_freq_hz = 20000000;
  config.pixel_format = PIXFORMAT_JPEG;
  //init with high specs to pre-allocate larger buffers
  if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
  } else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
  }

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

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

  sensor_t * s = esp_camera_sensor_get();
  //initial sensors are flipped vertically and colors are a bit saturated
  if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1);//flip it back
    s->set_brightness(s, 1);//up the blightness just a bit
    s->set_saturation(s, -2);//lower the saturation
  }
  //drop down frame size for higher initial frame rate
  s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE)
  s->set_vflip(s, 1);
  s->set_hmirror(s, 1);
#endif

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
```

```
  startCameraServer();

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

void loop() {
  if(matchFace==true && activateRelay==false)
  {
    activateRelay=true;
    digitalWrite(Relay,HIGH);
    digitalWrite(Green,HIGH);
    digitalWrite(Red,LOW);
    prevMillis=millis();
    }
    if (activateRelay == true && millis()-prevMillis > interval)
    {
      activateRelay=false;
      matchFace=false;
      digitalWrite(Relay,LOW);
      digitalWrite(Green,LOW);
      digitalWrite(Red,HIGH);
      }
}
```

Camera Pins (Header Files)

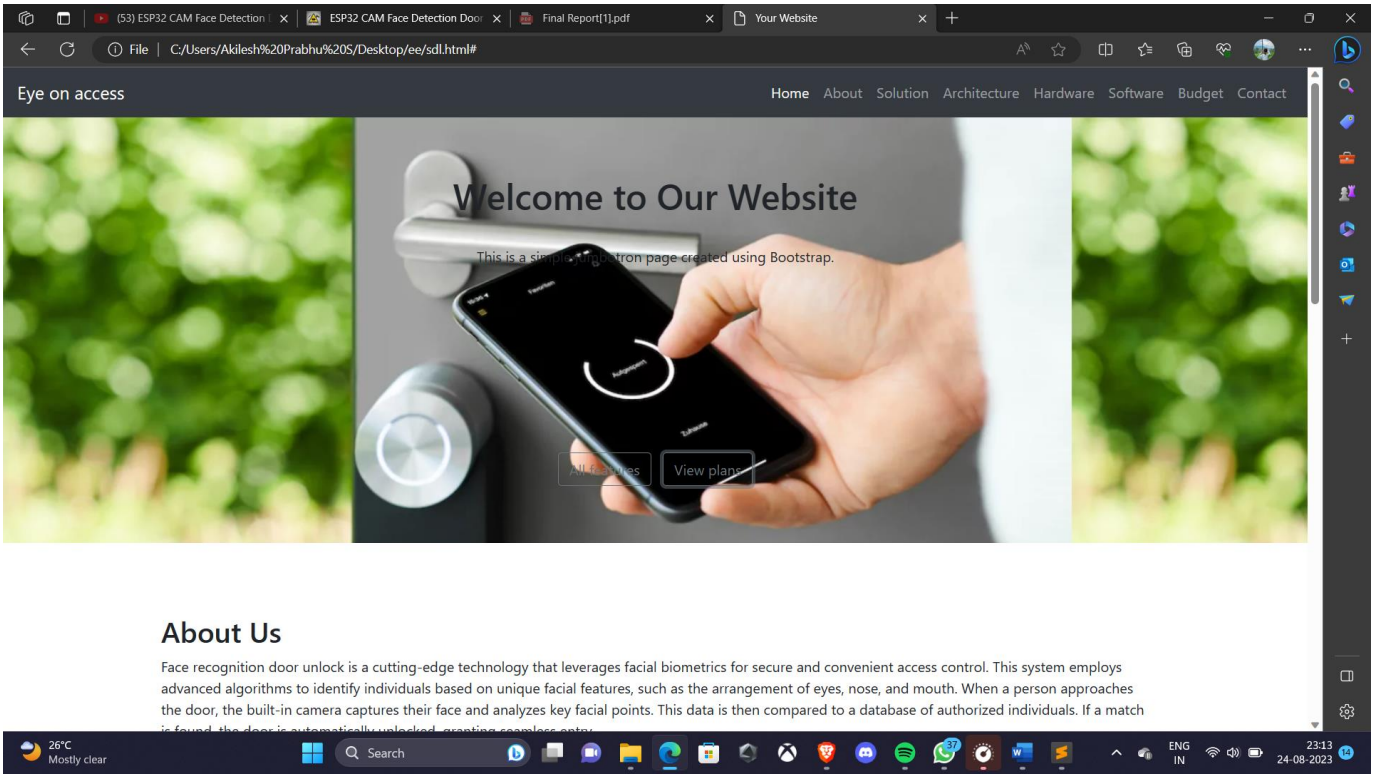
```
#if defined(CAMERA_MODEL_WROVER_KIT)
#define PWDN_GPIO_NUM    -1
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM    21
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27

#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
#define Y5_GPIO_NUM      19
#define Y4_GPIO_NUM      18
#define Y3_GPIO_NUM       5
#define Y2_GPIO_NUM       4
#define VSYNC_GPIO_NUM   25
#define HREF_GPIO_NUM    23
#define PCLK_GPIO_NUM    22

#elif defined(CAMERA_MODEL_ESP_EYE)
#define PWDN_GPIO_NUM    -1
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM    4
```

```
#define SIOD_GPIO_NUM    18
#define SIOC_GPIO_NUM    23

#define Y9_GPIO_NUM      36
#define Y8_GPIO_NUM      37
#define Y7_GPIO_NUM      38
#define Y6_GPIO_NUM      39
#define Y5_GPIO_NUM      35
#define Y4_GPIO_NUM      14
#define Y3_GPIO_NUM      13
#define Y2_GPIO_NUM      34
#define VSYNC_GPIO_NUM   5
#define HREF_GPIO_NUM    27
#define PCLK_GPIO_NUM    25

#elif defined(CAMERA_MODEL_M5STACK_PSRAM)
#define PWDN_GPIO_NUM     -1
#define RESET_GPIO_NUM    15
#define XCLK_GPIO_NUM     27
#define SIOD_GPIO_NUM     25
#define SIOC_GPIO_NUM     23

#define Y9_GPIO_NUM       19
#define Y8_GPIO_NUM       36
#define Y7_GPIO_NUM       18
#define Y6_GPIO_NUM       39
#define Y5_GPIO_NUM        5
#define Y4_GPIO_NUM       34
#define Y3_GPIO_NUM       35
#define Y2_GPIO_NUM       32
#define VSYNC_GPIO_NUM    22
#define HREF_GPIO_NUM     26
#define PCLK_GPIO_NUM     21

#elif defined(CAMERA_MODEL_M5STACK_WIDE)
#define PWDN_GPIO_NUM     -1
#define RESET_GPIO_NUM    15
#define XCLK_GPIO_NUM     27
#define SIOD_GPIO_NUM     22
#define SIOC_GPIO_NUM     23

#define Y9_GPIO_NUM       19
#define Y8_GPIO_NUM       36
#define Y7_GPIO_NUM       18
#define Y6_GPIO_NUM       39
#define Y5_GPIO_NUM        5
#define Y4_GPIO_NUM       34
#define Y3_GPIO_NUM       35
#define Y2_GPIO_NUM       32
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     26
#define PCLK_GPIO_NUM     21

#elif defined(CAMERA_MODEL_AI_THINKER)
#define PWDN_GPIO_NUM     32
```

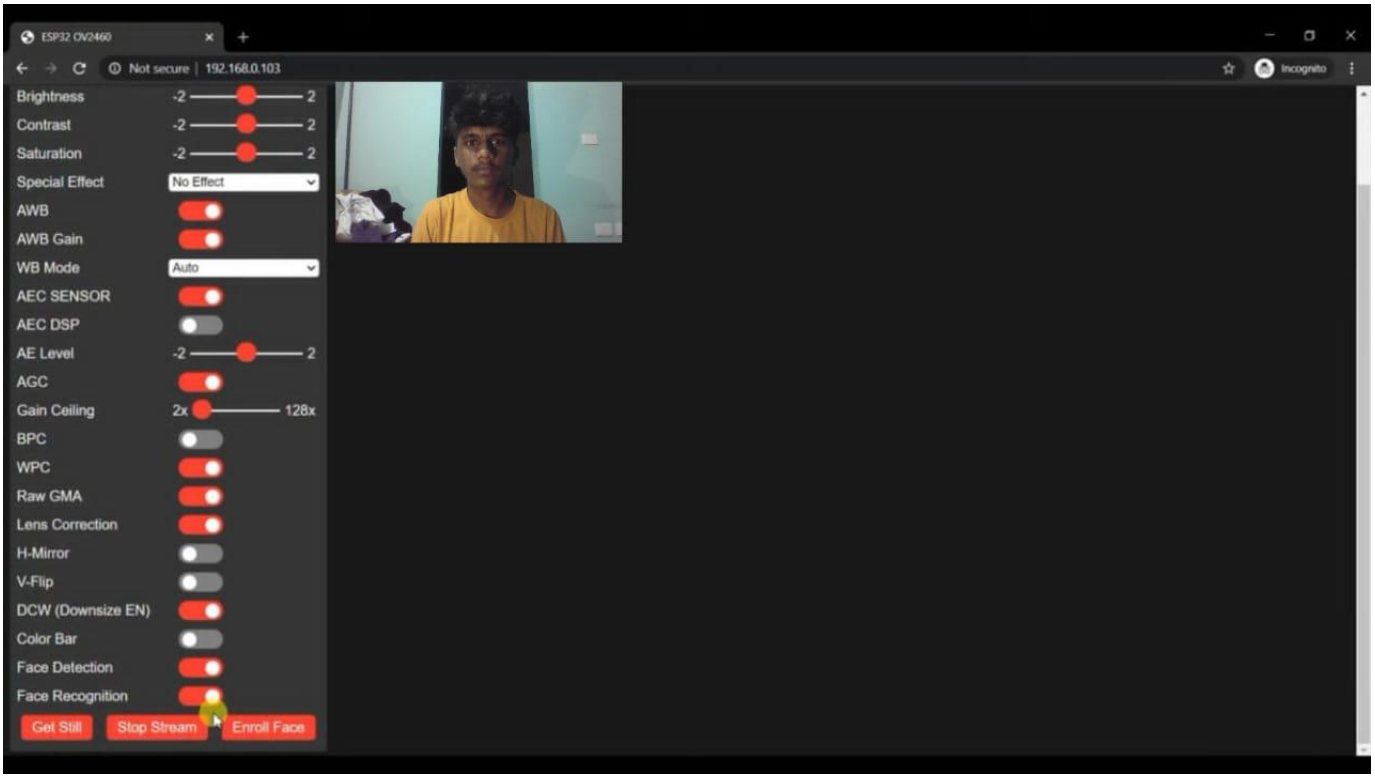
```
#define RESET_GPIO_NUM    -1
#define XCLK_GPIO_NUM      0
#define SIOD_GPIO_NUM     26
#define SIOC_GPIO_NUM     27

#define Y9_GPIO_NUM       35
#define Y8_GPIO_NUM       34
#define Y7_GPIO_NUM       39
#define Y6_GPIO_NUM       36
#define Y5_GPIO_NUM       21
#define Y4_GPIO_NUM       19
#define Y3_GPIO_NUM       18
#define Y2_GPIO_NUM        5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

#else
#error "Camera model not selected"
#endif
```

B. SCREEN SHOTS

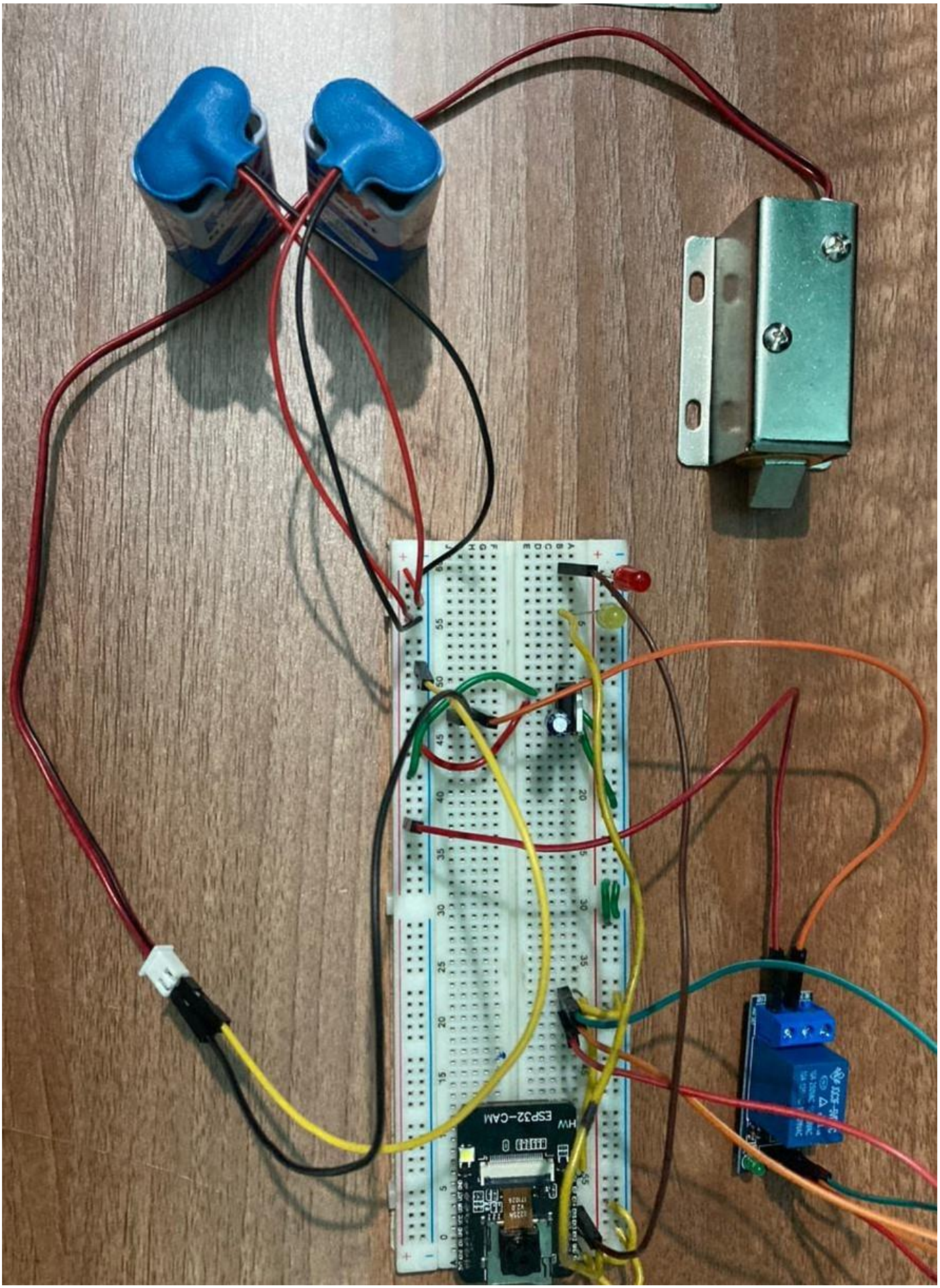
WEB PAGE



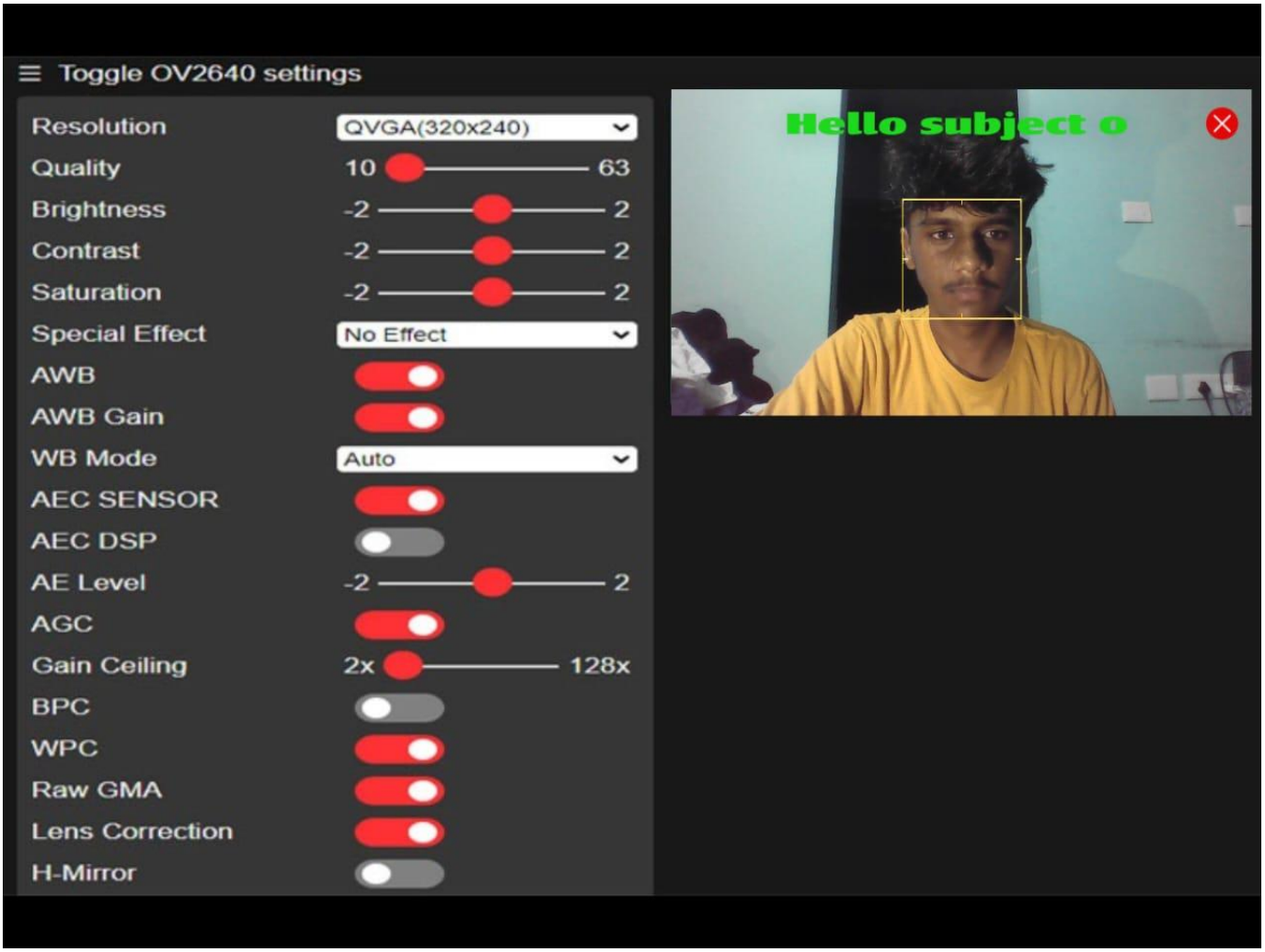
Face Enroll


ESP32 OV2460
Not secure | 192.168.0.103
Incognito
Brightness -2 2
Contrast -2 2
Saturation -2 2
Special Effect No Effect
AWB
AWB Gain
WB Mode Auto
AEC SENSOR
AEC DSP
AE Level -2 2
AGC
Gain Ceiling 2x 128x
BPC
WPC
Raw GMA
Lens Correction
H-Mirror
V-Flip
DCW (Downsize EN)
Color Bar
Face Detection
Face Recognition
Get Still
Stop Stream
Enroll Face

Lock Before Detecting Face



Face Detecting



Lock after detecting face