ACKNOWLEDGEMENT

First and foremost, I offer my sincere gratitude and respect to my project supervisor, **Miss Stella Kujur, HOD at the department of Information Technology**, for her invaluable guidance and suggestions to me during my study. I consider myself extremely fortunate to have had the opportunity of associating ourselves with her for one year. This project was made possible by her patience and persistence. I wish to express my deep gratitude to all those who extended their helping hands towards me in various ways during my short tenure at **S.K.D.A.V Government Polytechnic, Rourkela-12**.

I wish to express my sincere thanks to **Miss Stella Kujur, Department of Information Technology** and also the other staff members of the Department of Information Technology, **S.K.D.A.V Government Polytechnic, Rourkela-12** for providing me, the necessary facilities that was required to conduct the experiment and complete my project.

,

ABSTRACT

The "**VisiLock: Smart Face Access System**" seamlessly merges facial recognition technology with IoT capabilities to revolutionize access control. By employing sophisticated facial recognition algorithms and secure door access mechanisms, VisiLock offers a dependable alternative to conventional key-based systems. Its standout features include precise face recognition, IoT integration for remote access management, and the elimination of physical keys, enhancing user convenience. With practical applications across office spaces, hospitality, and retail environments, VisiLock adeptly tackles real-world security challenges. Currently deployed in the DBMS lab of the IT department, accessible to HoD, Teachers, CR and other Staff members, it exemplifies adaptability and scalability for both residential and commercial settings.

Key Features:

- Advanced Face Recognition: Utilizing accurate facial recognition algorithms for reliable identification.
- IoT Integration: Enabling remote access control and monitoring via a web-based application.
- Keyless Entry: Eliminating the need for physical keys or keycards, streamlining access processes.
- Scalability: Adaptable solution suitable for various settings, from homes to commercial establishments.
- Real-World Applications: Addressing security needs in offices, hotels, and retail stores for enhanced access control and user convenience.

Significance:

The "VisiLock: Smart Face Access System" offers a practical approach to access control by integrating facial recognition and IoT technologies. It provides a secure, convenient, and scalable solution for modern security challenges across different environments. This innovation represents a step forward in redefining access control methods, setting a new standard for reliability and user experience in securing various spaces.

CONTENTS

INTRODUCTION

Background

In the modern landscape of access control systems, traditional methods involving physical keys and keycards are becoming outdated, prompting the need for more secure, convenient, and technologically advanced solutions. The **"VisiLock: Smart Face Access System"** project addresses this need by integrating cutting-edge face recognition technology with IoT capabilities to **redefine the way we secure and manage access to doors**.

The increasing threats to security and the desire for seamless user experiences have led to the exploration of biometric authentication methods. Facial recognition, in particular, stands out for its **accuracy, non-intrusiveness, and ease of use**. By harnessing the power of the ESP32-CAM module, a robust face recognition algorithm, and a secure door lock mechanism, VisiLock not only **enhances security but also provides a convenient and futuristic solution to access control**.

This project envisions a world where individuals can access their spaces effortlessly, without the need for physical keys or keycards. The integration with a **web-based access control** further adds a layer of **flexibility,** allowing users to **control access remotely**. The **scalability** of the system ensures its **adaptability** to various access control scenarios, making it a versatile solution for both residential and commercial applications.

Objectives

Objectives for the "VisiLock: Smart Face Access System" Project are:

Implement Secure Access Control:

Develop a robust and secure door access control system using advanced face recognition technology.

Integrate IoT Capabilities:

Incorporate Internet of Things (IoT) features to enable remote control and monitoring through a web-based app.

Enhance User Convenience:

Eliminate the need for physical keys, providing users with a more convenient and seamless access experience.

Ensure High-Level Security:

Utilize face recognition algorithms to achieve accurate and reliable user authentication, enhancing overall security.

Enable Remote Access Control:

Empower users to control the door lock remotely via a user-friendly web-based application connected through Wi-Fi.

Provide User-Friendly Interface:

Design an intuitive and user-friendly web-based app interface for easy navigation and control.

These objectives collectively aim to create a cutting-edge, secure, and user-friendly access control system that leverages the capabilities of face recognition and IoT technologies.

Scope

Scope for the "VisiLock: Smart Face Access System" Project:

Healthcare Facilities:

Improve security in healthcare settings, controlling access to sensitive areas and ensuring only authorized personnel can enter designated zones.

Data Centers and Server Rooms:

Secure data centers and server rooms with VisiLock, allowing only authorized personnel entry for heightened security and protection of critical infrastructure.

Co-working Spaces:

Implement VisiLock for keyless and secure entry in co-working spaces, ensuring hassle-free access for members while maintaining controlled entry to specific areas within the shared workspace.

Educational Institutions:

Enhance security in schools and universities, ensuring only authorized individuals gain access to specific areas within the institution.

Residential Security:

Provide an enhanced and convenient access control solution for residential properties, replacing traditional locks with a secure and user-friendly face recognition system.

Commercial Buildings:

Address the security needs of commercial spaces, offering a scalable system that can manage access for employees, clients, and visitors.

Office Spaces:

Improve access control in office environments, facilitating a seamless and keyless entry experience for employees and authorized personnel.

Hospitality Industry:

Implement a secure and modern access control system for hotels, ensuring guests and staff can enjoy a hassle-free entry experience.

Retail Stores:

Provide a secure and convenient access solution for retail stores, preventing unauthorized entry and ensuring a smooth shopping experience for customers.

By focusing on these real-world use cases, the "VisiLock" project aims to address practical security challenges in various environments, providing a versatile and reliable access control solution for both residential and commercial applications.

Significance

The "VisiLock: Smart Face Access System" holds significant importance by redefining access control. It introduces a secure, keyless, and convenient solution, leveraging face recognition and IoT technologies. This innovation enhances security across diverse environments, from homes and offices to critical infrastructures, providing a transformative and reliable approach to modern access control challenges.

PRELIMINARY INVESTIGATION

Background

The "VisiLock: Smart Face Access System" heralds a new era in access control, seamlessly merging face recognition with IoT capabilities. Beyond traditional methods, VisiLock offers heightened security, eradicating the need for physical keys. Its real-time authentication, web-based app control, and scalability surpass conventional technologies, ushering in a future where access is not just controlled but revolutionized. This project is not merely an upgrade; it's a paradigm shift, redefining how we perceive and implement secure entry systems.

Previous works and technologies

In the landscape of access control, previous works primarily relied on conventional methods such as physical keys, keycards, or PIN codes. These approaches, while functional, often faced challenges related to security vulnerabilities, user inconvenience, and the risk of unauthorized duplication.

Technological advancements introduced keycard and biometric systems, each with its set of advantages and limitations. Keycard systems mitigated some risks associated with physical keys but still required users to carry cards, and the cards themselves could be lost or stolen. Biometric systems, including fingerprint and iris recognition, offered a more secure approach but faced challenges in terms of accuracy, especially in varying environmental conditions. "VisiLock: Smart Face Access System" heralds a new era in access control, seamlessly merging face recognition with IoT capabilities. Beyond traditional methods, VisiLock offers heightened security, eradicating the need for physical keys. Its real-time authentication, web-based app control, and scalability surpass conventional technologies, ushering in a future where access is not just controlled but revolutionized. This project is not merely an upgrade; it's a paradigm shift, redefining how we perceive and implement secure entry systems.

The "VisiLock: Smart Face Access System" builds upon these earlier technologies by leveraging the accuracy and non-intrusiveness of face recognition. Face recognition eliminates the need for physical tokens, enhancing security and user convenience simultaneously. Furthermore, the integration of IoT features for remote control through a web-based app sets VisiLock apart, providing a comprehensive and futuristic solution to access control challenges. In essence, this project represents a significant leap forward in secure access systems, addressing and surpassing the limitations of previous technologies.

Initial project assessment

The initial assessment of the "VisiLock: Smart Face Access System" project indicates a transformative and highly promising venture. This innovative system, which combines face recognition technology with IoT capabilities, offers a unique solution to traditional access control challenges. The feasibility of the project is strengthened by the existing advancements in face recognition algorithms and the widespread acceptance of IoT in various applications.

The use of the ESP32-CAM module as a foundation, coupled with a secure door lock mechanism, demonstrates a well-conceived approach to hardware design. Additionally, the integration of a sophisticated face recognition algorithm promises high accuracy in user authentication, marking a crucial advancement in security.

The initial project assessment underscores the potential for VisiLock to not only meet but exceed user expectations by providing secure, keyless access with added flexibility through remote control. This evaluation lays the foundation for a project that holds great promise in redefining access control standards. As the project progresses, continuous refinement and testing will be pivotal to ensuring its successful implementation and achieving the envisioned goals.

FEASIBILITY STUDY

Technical feasibility

Technical Feasibility for the "VisiLock: Smart Face Access System" Project:

The technical feasibility assessment focuses on the viability of implementing the proposed system from a technological standpoint.

- **Hardware capability**:
  The selection of the ESP32-CAM module, known for its processing power and built-in camera, ensures the hardware can support the computational requirements of face recognition.
- **Security measures:**
  Implementation of encryption protocols and security measures demonstrates technical feasibility in safeguarding sensitive facial recognition data during transmission and storage.
- **Face recognition algorithm:**
  Leveraging state-of-the-art face recognition algorithms enhances the system's accuracy, making it technically feasible to achieve reliable user authentication.
- **Web-based app development**:
  The feasibility of developing a user-friendly web-based app is supported by the availability of robust frameworks and tools for app development.
- **Integration with IoT**:
  The integration of IoT features allows for remote control through a web-based app, demonstrating technical feasibility in creating a seamless communication channel between the hardware and software components.
- **Connectivity and communication**:
  The project's reliance on Wi-Fi connectivity aligns with the prevalent wireless infrastructure, ensuring technical feasibility in establishing communication between the system components.

Overall, the technical feasibility analysis indicates a strong foundation for the successful development and implementation of the "VisiLock" system, with its integration of advanced technologies demonstrating a realistic and achievable technical vision.

Economic feasibility

Economic Feasibility for the "VisiLock: Smart Face Access System" Project:

The economic feasibility analysis assesses whether the project is financially viable, with a focus on keeping the total cost within the specified budget of 5000 rupees.

- **Cost of components**:
  The feasibility of the project is significantly influenced by the cost of individual components, including the ESP32-CAM module, face recognition algorithm implementation, and the necessary hardware for the door lock mechanism.

Ensuring that the total parts cost remains under 5000 rupees is a key economic consideration.

- **Development tools and software**:
  The affordability and accessibility of development tools and software, including the Arduino IDE and necessary programming languages, contribute to the economic feasibility of the project as all they count as a freeware.
- **Power consumption:**
  Evaluating the power consumption of the system components is crucial for economic feasibility. Energy-efficient designs contribute to lower operating costs.
- **Affordability for end users**:
  If the system is intended for commercial or residential use, the economic feasibility also considers whether the end users can afford the technology, ensuring market acceptance.

The economic feasibility of the "VisiLock" project is contingent on prudent cost management, efficient resource allocation, and the ability to deliver a reliable access control system within the stipulated budget of 3000 rupees. Continuous monitoring of costs during development is essential to ensure economic viability throughout the project lifecycle.

Operational feasibility

Operational Feasibility for the "VisiLock: Smart Face Access System" Project:

The operational feasibility assessment examines the practicality and effectiveness of implementing the "VisiLock" system within the operational context.

- **User acceptance**:
  The system's success hinges on user acceptance. Conducting surveys or pilot testing to gauge user preferences and expectations ensures operational feasibility by aligning the system with user needs.
- **Ease of use**:
  Operational feasibility is strengthened by ensuring that the VisiLock system is user-friendly and requires minimal training for end users. An intuitive web-based app interface and straightforward system operation contribute to ease of use.
- **Maintenance requirements**:
  Operational feasibility is influenced by the ease of maintenance. A system with straightforward maintenance procedures and low downtime enhances its practicality for day-to-day operations.
- **Reliability and performance**:
  Operational feasibility relies on the reliability and consistent performance of the system. Rigorous testing and performance assessments guarantee that VisiLock meets operational requirements without frequent disruptions.

The "VisiLock" project's operational feasibility relies on its ability to seamlessly integrate with daily operations, offer a positive user experience, and adapt to evolving needs with minimal disruptions. Continuous user engagement and feedback play a vital role in ensuring the system's practicality and effectiveness in real-world scenarios.

Conclusion

The feasibility study for the "VisiLock" project comprehensively evaluated technical, economic, and operational aspects to determine the viability of implementing the proposed access control system. Based on the findings, the following conclusions are drawn:

Technical feasibility:

The project demonstrates strong technical feasibility. The selection of the ESP32-CAM module, integration of advanced face recognition algorithms, and compatibility with IoT technologies provide a solid foundation for the successful development and implementation of VisiLock.

Economic feasibility:

The project's economic feasibility is promising. With a budget constraint of 5000 rupees for the total parts cost, careful selection of components, development tools, and cost-effective solutions ensures that the project remains within budgetary constraints, making it economically viable.

Operational feasibility:

Operational feasibility is established through user-centric design, ease of use, and integration with daily operations. The system's scalability, low maintenance requirements, and adherence to regulations contribute to its practicality and effectiveness in real-world scenarios.

In summary, the "VisiLock" project holds great promise across technical, economic, and operational dimensions. The synergy of cutting-edge technologies, cost-effective implementation, and seamless integration with operational workflows positions VisiLock as a viable and innovative solution for secure access control. The next phase involves the detailed design, development, and testing, building upon the positive outcomes of the feasibility study.

Requirement specification

User requirements

- **Face recognition**:
  Users expect the system to achieve a high level of accuracy in recognizing faces for reliable and secure access control.
- **User-Friendly web-based App**:
  A user-friendly web-based app interface that is intuitive and easy to navigate, allowing users to control access and monitor the system effortlessly.
- **Quick and Seamless Authentication**:
  Users anticipate a quick and seamless authentication process, ensuring that access is granted promptly after a successful face recognition.
- **Remote access control**:
  The ability to remotely control the door lock via the web-based app, providing convenience and flexibility to users, especially in scenarios where physical presence is not feasible.

By addressing these realistic user requirements, the "VisiLock" system aims to deliver a solution that not only meets user expectations but also enhances the overall experience of secure access control.

System requirements

Hardware components:

- ESP32-CAM module with built-in camera.
- Face recognition algorithm implementation.
- Secure door lock mechanism.
- FTDI Programmer for ESP32-CAM.
- Relay Module for controlling the door lock.
- Solenoid Lock for physical access control.
- Jumper wires for connecting components.
- 3.3+ Volt Battery as a power supply.
- PC with web browser and network connection for testing.

Software components:

- Arduino IDE for programming the ESP32-CAM.
- Programming languages (e.g., C++, Python) for app development and face recognition.
- Web browser for web-based app development.
- Required libraries and dependencies for face recognition.
- Web-based app Requirements:
- User-friendly interface for controlling the lock.
- Communication with the ESP32-CAM module.

Security measures:

- Robust user authentication to prevent unauthorized access.

Compatibility with Existing Infrastructure:

- Compatibility with standard Wi-Fi networks for seamless integration.

Communication requirements:

- Wi-Fi connectivity for communication between the ESP32-CAM module and the web-based app.

Power management:

- Power management features to optimize energy consumption and provide continuous flow.

These system requirements lay the foundation for the development and implementation of the "VisiLock" project, ensuring that the hardware, software, and communication components work cohesively to deliver a secure and user-friendly access control system.

Functional requirements

User authentication:

- The system must authenticate users based on facial recognition before granting access.

Web-based App Interface:

- The web-based app must provide a user-friendly interface for controlling the door lock and monitoring access.

Remote access control:

- Users should be able to control the door lock remotely through the web-based app.

Real-time Monitoring:

- The system must provide real-time monitoring of access events, including successful and unsuccessful attempts.

Notification system:

- Users should receive timely notifications about access events, such as successful entry or unauthorized attempts.

Compatibility with Wi-Fi Networks:

- The system should seamlessly connect to standard Wi-Fi networks for communication between the ESP32-CAM module and the web-based app.

Power management:

- The system must be designed with power-efficient features to optimize energy consumption and prolong battery life.

Integration with Existing Door Locks:

- The system should integrate with commonly used door lock mechanisms for physical access control.

Facial recognition accuracy:

- The face recognition algorithm must achieve a high level of accuracy to reliably authenticate users.

These functional requirements outline the key capabilities and features that the "VisiLock" system must possess to meet user expectations and deliver a secure and efficient smart face access control solution.

Non-functional requirements

Performance:

- Response Time: The system should have a quick response time for face recognition and access control, ensuring minimal delays.

Throughput:

- The system should handle a specified number of concurrent user requests without degradation in performance.

Security:

- User Authentication: The system must employ secure methods for user authentication, preventing unauthorized access.

Privacy:

- Facial recognition data should be handled with utmost privacy, and the system must comply with privacy regulations.

Reliability:

- The system must be reliable, with a low probability of false positives or negatives in face recognition.

Usability:

- The web-based app and system interface should be user-friendly, with clear navigation and intuitive controls.

Compatibility:

- The system must be compatible with a variety of Android and iOS devices, ensuring a broad user base.

Power Efficiency:

- The system should be designed to minimize power consumption, optimizing energy usage for prolonged battery life.

Interoperability:

- The system should be interoperable with existing technologies and infrastructure commonly found in homes and commercial spaces.

Auditability:

- The system should log access events for audit purposes, allowing administrators to review and analyse system activity.

These non-functional requirements define the attributes that are critical for the successful deployment and operation of the "VisiLock" system, ensuring a reliable, secure, and user-friendly access control solution.

SYSTEM ANALYSIS

System Architecture

In-Depth Overview

The VisiLock system architecture is meticulously designed to establish a robust and seamless framework for secure face access control. This section provides a comprehensive exploration of the architecture, elucidating the intricate interaction among key components and their collaborative functionality.

At its core, the system relies on four pivotal components:

ESP32-CAM Module:

Serving as the nucleus of the system, the ESP32-CAM module houses a sophisticated processing unit and an integrated camera. This component is pivotal for capturing facial images, initiating the initial processing steps, and interfacing with the face recognition algorithm.

Face Recognition Algorithm:

Leveraging advanced face recognition algorithms, this component meticulously analyses the facial images captured by the ESP32-CAM. Its primary function is to authenticate users in real-time, ensuring a swift and accurate access control process.

Web-based Application:

The web-based application acts as the user's gateway to the VisiLock system. Beyond offering a visually intuitive interface, it facilitates remote control functionalities, enabling users to manage access, monitor real-time events, and interact seamlessly with the system.

Door Lock Mechanism:

Comprising the relay module and solenoid lock, this component translates the decisions made by the system into physical actions. It interfaces with the system to control access, responding to the outcomes of the face recognition process.

Communication Flow:

The ESP32-CAM communicates with the face recognition algorithm, facilitating the exchange of captured images and authentication results. Simultaneously, the web-based application communicates with the entire system, allowing users to initiate commands and receive real-time notifications.

Data Flow Overview:

The data flow within the system begins with the camera capturing facial images, which are then processed by the face recognition algorithm. The web-based application, acting as a bridge, facilitates communication between the user and the system, leading to the activation of the door lock mechanism upon successful authentication.

Integration Points:

Integration points include the ESP32-CAM module interfacing with the face recognition algorithm, the web-based application communicating with the entire system, and the door lock mechanism aligning its actions with the decisions made by the system.

Security Measures:

The face recognition algorithm incorporates good authentication standards.

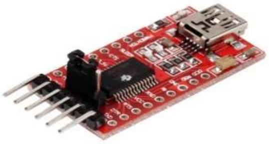
In conclusion, the VisiLock system architecture embodies a harmonious interplay of components, fostering a secure, user-centric, and technologically advanced solution for face access control. The in-depth overview provides a nuanced understanding of the system's inner workings, emphasizing its reliability and innovation.

Core components

Hardware components

ESP32-CAM Module:
Hardware Description: Compact microcontroller with an integrated camera.
Functionality: Captures facial images and initiates preliminary processing for face recognition.

Door Lock Mechanism:
Hardware Description: Comprises a relay module and solenoid lock.
Functionality: Translates system decisions into physical actions, controlling access based on face recognition outcomes.



FTDI Programmer:
Hardware Description: USB-to-serial converter for programming the ESP32-CAM.
Functionality: Facilitates communication between the computer and the ESP32-CAM during programming.

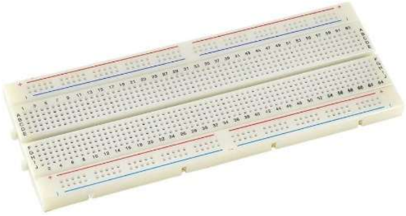
USB Cable:
Hardware Description: Standard USB cable.
Functionality: Connects the FTDI Programmer to the computer for programming the ESP32-CAM.

Battery (5V for ESP32-CAM):
Hardware Description: Power source providing 5V for the ESP32-CAM.
Functionality: Powers the ESP32-CAM for image capture and processing.

Battery (12V for Lock):
Hardware Description: Power source providing 12V for the door lock mechanism.
Functionality: Powers the solenoid lock, enabling physical access control.

Connectors and Jumper Wires:
Hardware Description: Electronic connectors and wires.
Functionality: Establishes connections between various components on the breadboard.

Breadboard:
Hardware Description: Prototyping board for building electronic circuits.
Functionality: Facilitates the temporary arrangement and connection of electronic components.



This Photo by Unknown Author is

Software components

- ESP32-CAM Firmware:
    - Software Description: Embedded firmware running on the ESP32-CAM module.
    - Functionality: Manages image capture, initial processing, and communication with the face recognition algorithm.
- Face Recognition Algorithm Software:
    - Software Description: Program implementing advanced face recognition techniques.
    - Functionality: Analyses facial features, facilitating real-time authentication.
- Web-based Application Software:
    - Software Description: web-based app installed on user devices.
    - Functionality: Provides a user-friendly interface for remote control, real-time monitoring, and system interaction.
- Door Lock Control Software:
    - Software Description: Code governing the relay module and solenoid lock.
    - Functionality: Executes physical actions based on decisions made by the system.

These hardware and software components collectively contribute to the functionality and effectiveness of the VisiLock system, providing a comprehensive and secure smart face access control solution.

System functionality

The VisiLock system seamlessly integrates its core components to deliver a multifaceted and efficient functionality, ensuring robust face access control. The system's operations can be broken down into key functionalities:

1. Facial Recognition:

- Capture: The ESP32-CAM captures facial images with its integrated camera.
- Processing: The captured images are processed by the face recognition algorithm to extract facial features.
- Authentication: The algorithm authenticates users in real-time based on recognized facial features.

2. web-based App Control:

- Remote Access: Users can control access to the lock remotely through the web-based application.
- Real-time Monitoring: The web-based app provides real-time monitoring of access events, offering insights into successful entries and unauthorized attempts.

3. Door Lock Mechanism Control:

- Physical Access Control: The door lock mechanism, comprising the relay module and solenoid lock, is activated based on the decisions made by the system after facial recognition. This controls physical access to the secured area.

The collective functionality of the VisiLock system provides a secure, user-centric, and technologically advanced solution for modern face access control needs, offering a seamless blend of convenience and reliability.

Features and advantages

1. Facial Recognition:

- Feature: Utilizes advanced face recognition algorithms.
- Advantage: Ensures accurate user authentication, enhancing security.

2. web-based App Convenience:

- Feature: web-based application for remote control.
- Advantage: Enables users to operate lock from their devices.

3. Physical Access Control:

- Feature: Door lock mechanism for physical access control.
- Advantage: Translates system decisions into tangible actions, enhancing overall security.

4. Integration of Hardware Components:

- Feature: Seamless integration of ESP32-CAM, face recognition, web-based app, and door lock mechanism.
- Advantage: Ensures cohesive functionality and enhances the overall reliability of the system.

5. Remote Accessibility:

- Feature: Users can control access remotely.
- Advantage: Adds a layer of convenience and flexibility to access control management.

6. Access Control:

- Feature: Smart face access control.
- Advantage: Provides an innovative and technologically advanced solution for modern security requirements.

The combination of these features and advantages positions the VisiLock system as a reliable, secure, and user-friendly smart face access control solution, addressing contemporary security challenges.

CIRCUITRY AND CONNECTIONS

ESP32 cam module programming circuit

Components:

- ESP32-CAM
- FTDI Programmer
- USB Cable

Connections:

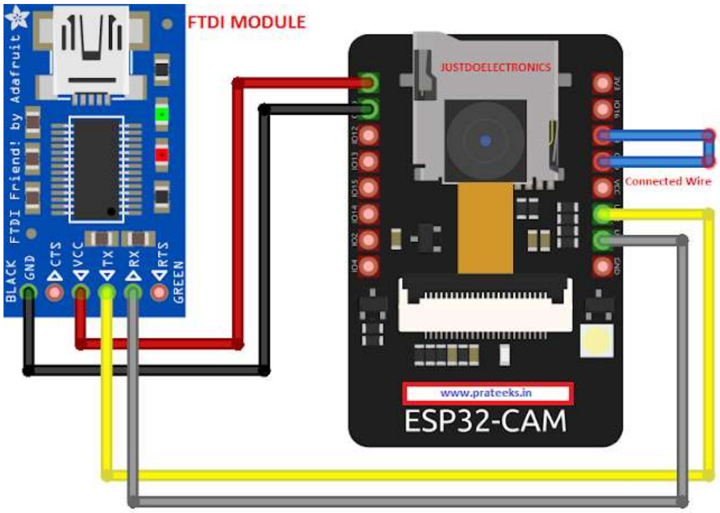
ESP32-CAM	FTDI Programmer
GND	GND
5V	VCC (5V)
U0R	TX
U0T	RX
GPIO 0	GND

Functionality:

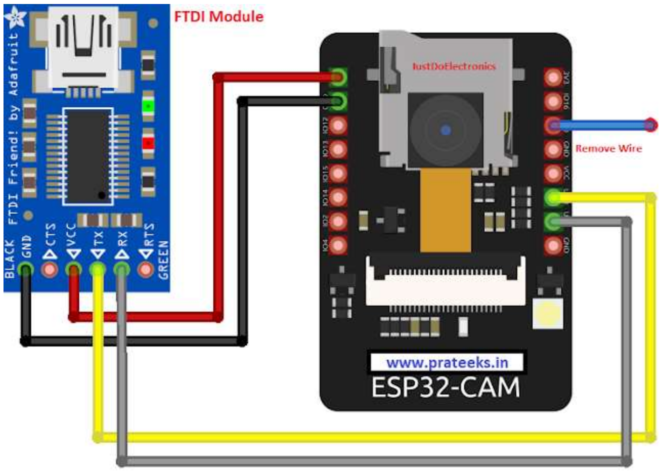
- Enables programming of the ESP32-CAM module using the FTDI programmer.
- Facilitates the transfer of code from the computer to the ESP32-CAM for proper functioning.
- Establishes a communication link for uploading firmware and configurations.

This circuit ensures a seamless programming process for the ESP32-CAM module, allowing for the integration of essential functionalities and configurations to enable face recognition and system control.

Circuit diagram for connecting to ESP32 to PC



While uploading code

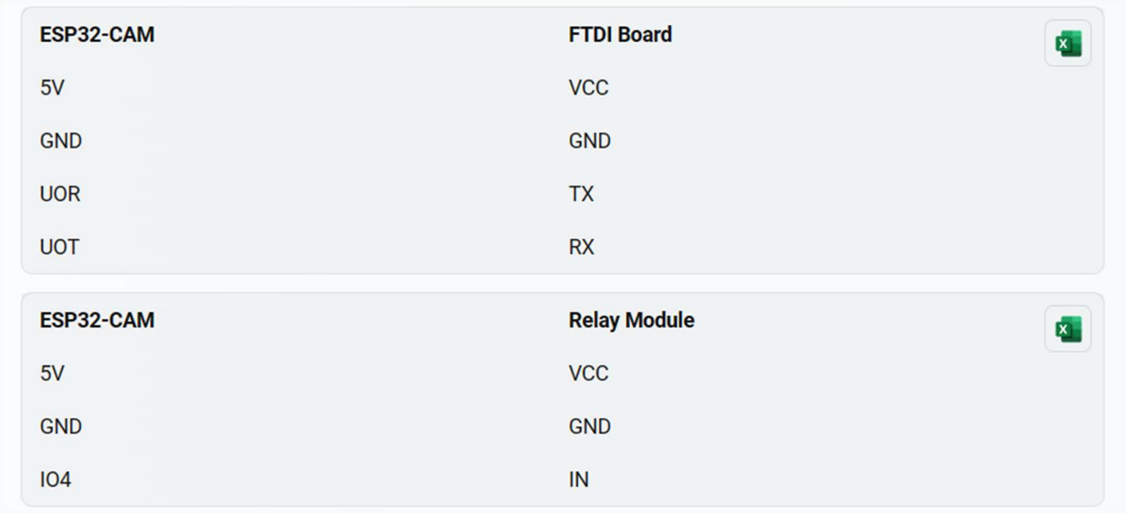


Integration

Integration circuitry components:

- ESP32-CAM
- FTDI Programmer
- Relay Module
- Solenoid Lock
- 12V Battery
- USB Cable

Connections:



ESP32-CAM	FTDI Board
5V	VCC
GND	GND
UOR	TX
UOT	RX

ESP32-CAM	Relay Module
5V	VCC
GND	GND
IO4	IN

Functionality:

- Enables programming of the ESP32-CAM.
- Facilitates communication between the ESP32-CAM and the computer during programming.
- Allows the ESP32-CAM to control the relay module, which, in turn, controls the solenoid lock.

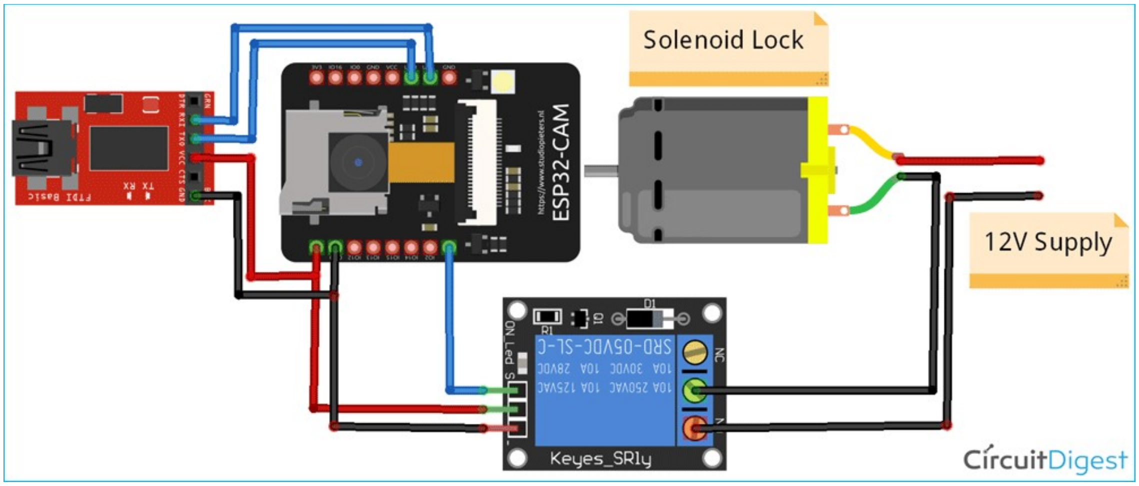
- The 12V battery powers the relay module and solenoid lock for physical access control.

This integrated circuit establishes the necessary connections for programming the ESP32-CAM and orchestrates the control of the solenoid lock through the relay module, ensuring a synchronized and secure access control system.

Functionality:

- Enables programming of the ESP32-CAM.
- Facilitates communication between the ESP32-CAM and the computer during programming.
- Allows the ESP32-CAM to control the relay module, which, in turn, controls the solenoid lock.
- The 12V battery powers the relay module and solenoid lock for physical access control.

Circuit diagram



This integrated circuit establishes the necessary connections for programming the ESP32-CAM and orchestrates the control of the solenoid lock through the relay module, ensuring a synchronized and secure access control system.

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 off1. VCC and GND pins of the FTDI board and Relay module are connected to the Vcc and GND pin of ESP32-CAM1. 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-CAM1.

Before uploading the code, connect the IO0 to the ground. IO0 determines whether the ESP32 is in flashing mode or not. When GPIO 0 is connected to GND, the ESP32 is in flashing mode1.

The solenoid lock usually uses 12V, 24V or 48V power supply. Therefore, we cannot connect the solenoid lock directly to ESP32 pin. We have to connect it to ESP32 pin via a relay2. If we connect the solenoid lock to a relay (normally open mode): When relay is open, door is unlocked. When relay is locked.

PROJECT DESIGN

Hardware design

ESP32-CAM Module Design

The ESP32-CAM module is a central component of the VisiLock system, responsible for capturing facial images, communicating with the face recognition algorithm, and controlling the door lock mechanism. The design considerations for the ESP32-CAM module include its specifications, interconnections, and functionality.

Specifications:

- Microcontroller: ESP32 with integrated Wi-Fi and Bluetooth.
- Camera: Integrated OV2640 camera for capturing facial images.
- Power Supply: 5V DC.
- Communication: UART communication for programming and data exchange.
- GPIO Pins: Used for interfacing with other components.

Interconnections:

- Connect the TXD pin to the RXD (U0R) pin of the FTDI Programmer.
- Connect the RXD pin to the TXD (U0T) pin of the FTDI Programmer.
- Connect the GND (Ground) pin to the GND pin of the FTDI Programmer.
- Connect the 5V pin to the 5V pin of the FTDI Programmer.
- Interact with the face recognition algorithm for image processing. outcomes.

Programming Circuit:

- Utilize the FTDI Programmer for programming the ESP32-CAM.
- Connect the FTDI Programmer to a computer via a USB cable.
- Establish communication for uploading firmware and configurations.

Camera Operation:

- Capture facial images using the integrated OV2640 camera.
- Transmit captured images to the face recognition algorithm for processing.

Communication with web-based App:

- Communicate with the web-based application for remote control.
- Enable real-time monitoring and notification features through Wi-Fi connectivity.

Control of Relay Module:

- Control the relay module based on face recognition outcomes.
- Determine when to activate the solenoid lock for physical access control.

Power Management:

- Operate on a 5V DC power supply.
- Optimize power consumption for prolonged system operation.
- Efficiently manage power distribution to connected components.

GPIO Pin Configuration:

- Utilize GPIO pins for interfacing with the relay module and other components.
- Configure GPIO pins to control the relay module and manage data flow.

Integration with Face Recognition Algorithm:

- Establish a seamless data flow between the ESP32-CAM and the face recognition algorithm.
- Ensure timely transmission of facial images for authentication.

Testing Procedures:

- Develop testing strategies for validating the functionality of the ESP32-CAM.
- Include unit testing to verify individual components and integration testing for overall system cohesion.

The design of the ESP32-CAM module focuses on efficient communication, image capture, control functionalities, and seamless integration with other system components. This ensures that the module plays a pivotal role in the successful operation of the VisiLock smart face access control system.

Face Recognition Algorithm Design

The face recognition algorithm is a critical component of the VisiLock system, responsible for processing facial images captured by the ESP32-CAM module. The design of the algorithm involves considerations for accuracy, speed, and integration with the overall system. Here is an overview of the design aspects:

Algorithm Type:

- Description: Utilize a deep learning-based face recognition algorithm.

Specifications:

- Leverage pre-trained models for facial feature extraction.

Integration with ESP32-CAM:

- Communication: Establish a reliable communication link with the ESP32-CAM.
- Data Flow: Receive facial images from the ESP32-CAM for processing.
- Feedback: Provide authentication results back to the ESP32-CAM.

Image Processing:

- Facial Feature Extraction: Employ techniques for extracting distinctive facial features.
- Normalization: Normalize facial images for consistent processing.

Real-Time Operation:

- Speed Optimization: Optimize the product for real-time processing.
- Parallelization: Explore parallel processing techniques for efficiency.

Testing and Validation:

- Accuracy Testing: Conduct thorough testing to evaluate the accuracy of facial recognition.
- Validation: Validate the algorithm against diverse facial images.

Error Handling:

- False Positives/Negatives: Implement strategies to minimize false positives and negatives.
- Logging: Log recognition events for analysis and improvement.

The design of the face recognition algorithm focuses on accuracy, real-time processing, integration with system components, security, and adaptability. This ensures that the algorithm reliably and securely contributes to the overall functionality of the VisiLock smart face access control system.

Solenoid Lock Design

The solenoid lock is a crucial hardware component in the VisiLock system, responsible for physical access control based on face recognition outcomes. Here is a simple design overview of the solenoid lock:

Description: The solenoid lock is an electromechanical device that converts electrical energy into mechanical motion to control the locking mechanism.

Specifications:

- Voltage: Operates on a 12V DC power supply.
- Locking Mechanism: Utilizes a plunger mechanism to engage or disengage the lock.

Power Supply:

- Connection: Connects to the relay module for electrical control.
- Voltage Compatibility: Compatible with the 12V output of the relay module.

Interconnections:

- Relay Module Connection: Connects to the relay module's output for electrical control.
- GPIO Pin Connection: Connects to a GPIO pin on the ESP32-CAM for control.

Operation:

- Locking: When activated, the solenoid extends the plunger, engaging the lock mechanism.
- Unlocking: Deactivation retracts the plunger, releasing the lock.

Integration with Relay Module:

- Electrical Control: The relay module controls the application and removal of power to the solenoid.
- Connection: Wired connection ensures synchronized operation with the ESP32-CAM.

Security Features:

- Mechanical Strength: Ensures robust locking mechanism for security.
- Testing:
- Functionality Testing: Ensures proper engagement and disengagement of the lock.
- Compatibility Testing: Validates compatibility with the relay module and ESP32-CAM.

The solenoid lock design ensures a straightforward yet effective mechanism for physical access control, providing a secure and reliable solution for the VisiLock smart face access control system.

Software design

Face recognition algorithm

Principal Component Analysis (PCA) is a dimensionality reduction technique commonly used in face recognition applications. It helps extract essential features from a set of facial images, reducing the computational complexity and improving efficiency. Here's how PCA can be used in the project design for face recognition:

PCA for Face Recognition

Feature Extraction:

- Objective: PCA is applied to the set of facial images to extract the most significant features.
- Implementation: Each facial image is treated as a high-dimensional vector, and PCA identifies the principal components that capture the maximum variance.

Dimensionality Reduction:

- Objective: Reduce the dimensionality of the facial image dataset while retaining essential information.
- Implementation: PCA transforms the original image vectors into a lower-dimensional space, removing less important features.

Eigenfaces:

- Objective: Generate eigenfaces, which are the principal components obtained through PCA.
- Implementation: Eigenfaces represent the most distinctive facial features. These can be used as a basis to reconstruct facial images efficiently.

Training the Recognition Model:

- Objective: Train the face recognition model using the reduced-dimensional eigenfaces.
- Implementation: Eigenfaces act as the basis for training a classifier or distance metric to identify individuals.

Comparison and Recognition:

- Objective: Compare incoming facial images with the eigenfaces and recognize individuals.
- Implementation: Utilize a classifier or distance metric to compare the reduced-dimensional representation of the input face with the eigenfaces in the training set.

Real-time Processing:

- Objective: Enable real-time face recognition.
- Implementation: PCA, by reducing dimensionality, speeds up the processing time required for facial recognition, making it suitable for real-time applications.

By incorporating PCA into the face recognition system's design, we leverage its capability to extract relevant features and authenticate user.

Programming Languages for Project Design

The VisiLock project involves multiple components, each requiring specific programming languages for implementation. Here's an overview of the languages needed for various aspects of the project:

Microcontroller (ESP32-CAM) Programming:

Language: C++ (Arduino)

Rationale: The ESP32-CAM microcontroller is typically programmed using the Arduino framework, which uses a variant of C++. This language is well-suited for embedded systems and microcontroller programming.

Web-based App Development:

Language: HTML, CSS, JS and PHP

Rationale: Web app interface is made using HTML, CSS, JS. While web socket and interactions are controlled using JS and PHP. Also, for server-side integration.

Ensuring that the selected languages align with the requirements of each component, considering factors such as compatibility, library support, and developer expertise. This mix of languages provides the versatility needed for the VisiLock project's functionalities.

Integration Strategy for VisiLock Project Design

Integrating the various components of the VisiLock project requires a well-defined strategy to ensure seamless communication and functionality. Here's an overview of the integration strategy:

Microcontroller (ESP32-CAM) and Face Recognition Algorithm:

Integration Point: Serial communication (UART)

Strategy:

ESP32-CAM captures facial images and sends them to the face recognition algorithm.

Face recognition algorithm processes images and provides authentication results.

Face Recognition Algorithm and web-based App:

Integration Point: API (Application Programming Interface)

Strategy:

web-based app sends requests to open door based on authentication results.

Microcontroller (ESP32-CAM) and web-based App:

Integration Point: Wi-Fi

Strategy:

web-based app communicates with the ESP32-CAM over Wi-Fi for remote control.

Commands for locking/unlocking are sent from the app to the microcontroller.

Testing and Continuous Integration:

Strategy:

Implement thorough testing at each integration point to ensure functionality.

Utilize continuous integration tools to automate testing and deployment processes.

By employing a well-defined integration strategy, we ensure that the various components of the VisiLock project work cohesively, providing a reliable and efficient smart face access control system. Regular testing and monitoring contribute to the ongoing success of the integrated solution.

CODING

Implementation approach

Coding Implementation Approach for VisiLock Project

Implementing the VisiLock project involves coding for various components and ensuring seamless integration. Below is an approach for coding implementation:

Microcontroller (ESP32-CAM) Programming:

IDE: Arduino IDE

Approach:

- Write C++ code for capturing images from the OV2640 camera.
- Implement communication protocols (UART) for sending images to the face recognition algorithm.
- Develop code for remote control commands received over Wi-Fi from the web-based app.

Web-based App Development:

Platform: Web

Approach:

- Design the user interface for face registration, remote control, and feedback.
- Implement functionality for capturing and sending facial images for recognition.

Testing:

Approach:

- Implement unit testing for individual components.
- Conduct integration testing to ensure seamless communication.
- Perform system testing to validate end-to-end functionality.

Continuous Integration:

Approach:

- Set up continuous integration tools for automated testing.
- Implement a CI/CD pipeline for code deployment.

Documentation:

Approach:

- Maintain thorough documentation for code, APIs, and system architecture.
- Include code comments for clarity and future maintenance.

By following this coding implementation approach, you can systematically develop and integrate each component of the VisiLock project, ensuring a reliable and efficient smart face access control system. Regular testing, documentation, and security measures contribute to the overall success of the implementation.

Coding Snippets

Include Library Files:

```
#include "esp_camera.h"

#include <WiFi.h>

#include "camera_pins.h"
```

Define Camera Model:

```
#define CAMERA_MODEL_AI_THINKER
```

Network Credentials

```
const char* ssid = "Wi-Fi Name";

const char* password = "Wi-Fi password";
```

Then we define the pin where the relay module is connected. We will be using millis() function to lock the door after unlocking it in a defined interval of time, here it is 5 seconds.

```
#define relay 4

long prevMillis = 0;

int interval = 5000;
```

Then we define the pin where the relay module is connected. We will be using millis() function to lock the door after unlocking it in a defined interval of time, here it is 5 seconds.

```
void setup() {
 Serial.begin(115200);

 pinMode(relay, OUTPUT);

 digitalWrite(relay, LOW);

}
```

Inside the loop() function, We check if the face matches with the enrolled face. If yes, then unlock the door for 5 seconds and after 5 seconds lock the door again.

```
void loop() {

   if (matchFace == true && activeRelay == false){

     activeRelay = true;

     digitalWrite(relay, HIGH);

     prevMillis = millis();

   }

   if(activeRelay == true && millis() - prevMillis > interval){

     activeRelay = false;

     matchFace = false;

     digitalWrite(relay, LOW);

   }

}
```

Web interface

```
<!doctype html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width,initial-scale=1">

<title>Face Recognition Access Control</title>

<style>

@media only screen and (min-width: 850px) {

        body {

                display: flex;
```

```
          }
           #content-right {
                     margin-left: 10px;
          }
}
body {
    font-family: Arial, Helvetica, sans-serif;
    background: #181818;
    color: #EFEFEF;
    font-size: 16px;
}
 #content-left {
    max-width: 400px;
           flex: 1;
}
 #content-right {
    max-width: 400px;
           flex: 1;
}
 #stream {
    width: 100%;
}
 #status-display {
    height: 25px;
    border: none;
    padding: 10px;
    font: 18px/22px sans-serif;
    margin-bottom: 10px;
    border-radius: 5px;
    background: green;
    text-align: center;
}
 #person {
    width:100%;
    height: 25px;
    border: none;
    padding: 20px 10px;
    font: 18px/22px sans-serif;
    margin-bottom: 10px;
    border-radius: 5px;
    resize: none;
    box-sizing: border-box;
}
 button {
    display: block;
    margin: 5px 0;
    padding: 0 12px;
    border: 0;
    width: 48%;
    line-height: 28px;
    cursor: pointer;
    color: #fff;
    background: #ff3034;
    border-radius: 5px;
    font-size: 16px;
    outline: 0;
}
 .buttons {
    height:40px;
}
 button:hover {
    background: #ff494d;
}
 button:active {
    background: #f21c21;
}
 button:disabled {
    cursor: default;
    background: #a0a0a0;
}
 .left {
    float: left;
}
 .right {
    float: right;
}
 .image-container {
    position: relative;
}
 .stream {
    max-width: 400px;
}
 ul {
    list-style: none;
    padding: 5px;
    margin:0;
}
```

```
 li {
    padding: 5px 0;
}
 .delete {
    background: #ff3034;
    border-radius: 100px;
    color: #fff;
    text-align: center;
    line-height: 18px;
    cursor: pointer;
}
 h3 {
    margin-bottom: 3px;
}
</style>
</head>
<body>
<div id="content-left">
  <div id="stream-container" class="image-container"> <img id="stream" src=""> </div>
</div>
<div id="content-right">
  <div id="status-display"> <span id="current-status"></span> </div>
  <div id="person-name">
    <input id="person" type="text" value="" placeholder="Type the person's name here">
  </div>
  <div class="buttons">
    <button id="button-stream" class="left">STREAM CAMERA</button>
    <button id="button-detect" class="right">DETECT FACES</button>
  </div>
  <div class="buttons">
    <button id="button-capture" class="left" title="Enter a name above before capturing a face">ADD USER</button>
    <button id="button-recognise" class="right">ACCESS CONTROL</button>
  </div>
  <div class="people">
    <h3>Captured Faces</h3>
    <ul>
    </ul>
  </div>
  <div class="buttons">
    <button id="delete_all">DELETE ALL</button>
  </div>
</div>
<script>
document.addEventListener("DOMContentLoaded", function(event) {
  var baseHost = document.location.origin;
  var streamUrl = baseHost + ":81";
  const WS_URL = "ws://" + window.location.host + ":82";
  const ws = new WebSocket(WS_URL);

  const view = document.getElementById("stream");
  const personFormField = document.getElementById("person");
  const streamButton = document.getElementById("button-stream");
  const detectButton = document.getElementById("button-detect");
  const captureButton = document.getElementById("button-capture");
  const recogniseButton = document.getElementById("button-recognise");
  const deleteAllButton = document.getElementById("delete_all");
```

```
// gain, frequency, duration
a=new AudioContext();
function alertSound(w,x,y){
  v=a.createOscillator();
  u=a.createGain();
  v.connect(u);
  v.frequency.value=x;
  v.type="square";
  u.connect(a.destination);
  u.gain.value=w*0.01;
  v.start(a.currentTime);
  v.stop(a.currentTime+y*0.001);
}

ws.onopen = () => {
  console.log(`Connected to ${WS_URL}`);
};
ws.onmessage = message => {
  if (typeof message.data === "string") {
    if (message.data.substr(0, 8) == "listface") {
      addFaceToScreen(message.data.substr(9));
    } else if (message.data == "delete_faces") {
      deleteAllFacesFromScreen();
    } else if (message.data == "door_open") {
        alertSound(10,233,100); alertSound(3,603,200);
    } else {
        document.getElementById("current-status").innerHTML = message.data;
        document.getElementById("status-display").style.background = "green";
    }
  }
  if (message.data instanceof Blob) {
    var urlObject = URL.createObjectURL(message.data);
    view.src = urlObject;
  }
}
streamButton.onclick = () => {
  ws.send("stream");
};
detectButton.onclick = () => {
  ws.send("detect");
};
captureButton.onclick = () => {
  person_name = document.getElementById("person").value;
  ws.send("capture:" + person_name);
};
recogniseButton.onclick = () => {
  ws.send("recognise");
};
deleteAllButton.onclick = () => {
  ws.send("delete_all");
};
personFormField.onkeyup = () => {
  captureButton.disabled = false;
};
function deleteAllFacesFromScreen() {
  // deletes face list in browser only
  const faceList = document.querySelector("ul");
  while (faceList.firstChild) {
      faceList.firstChild.remove();
    }
    personFormField.value = "";
    captureButton.disabled = true;
  }
  function addFaceToScreen(person_name) {
    const faceList = document.querySelector("ul");
    let listItem = document.createElement("li");
    let closeItem = document.createElement("span");
    closeItem.classList.add("delete");
    closeItem.id = person_name;
    closeItem.addEventListener("click", function() {
      ws.send("remove:" + person_name);
    });
    listItem.appendChild(
      document.createElement("strong")
    ).textContent = person_name;
    listItem.appendChild(closeItem).textContent = "X";
    faceList.appendChild(listItem);
  }
  captureButton.disabled = true;
});
</script>
</body>
</html>
```

Pin allocation code

```
#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
```

Main code (Backend)

```
#include <ArduinoWebsockets.h>

#include "esp_http_server.h"

#include "esp_timer.h"

#include "esp_camera.h"

#include "camera_index.h"

#include "Arduino.h"

#include "fd_forward.h"

#include "fr_forward.h"

#include "fr_flash.h"

const char* ssid = "NSA";
const char* password = "Orange";
#define ENROLL_CONFIRM_TIMES 5
#define FACE_ID_SAVE_NUMBER 7
// 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
#include "camera_pins.h"
using namespace websockets;
WebsocketsServer socket_server;
camera_fb_t * fb = NULL;
long current_millis;
long last_detected_millis = 0;
#define relay_pin 2 // pin 12 can also be used
unsigned long door_opened_millis = 0;
long interval = 5000;           // open lock for ... milliseconds
bool face_recognised = false;
void app_facenet_main();
void app_httpserver_init();
typedef struct
{
  uint8_t *image;
  box_array_t *net_boxes;
  dl_matrix3d_t *face_id;
} http_img_process_result;
static inline mtmn_config_t app_mtmn_config()
{
  mtmn_config_t mtmn_config = {0};
  mtmn_config.type = FAST;
  mtmn_config.min_face = 80;
  mtmn_config.pyramid = 0.707;
  mtmn_config.pyramid_times = 4;
  mtmn_config.p_threshold.score = 0.6;
  mtmn_config.p_threshold.nms = 0.7;
  mtmn_config.p_threshold.candidate_number = 20;
  mtmn_config.r_threshold.score = 0.7;
  mtmn_config.r_threshold.nms = 0.7;
  mtmn_config.r_threshold.candidate_number = 10;
  mtmn_config.o_threshold.score = 0.7;
  mtmn_config.o_threshold.nms = 0.7;
  mtmn_config.o_threshold.candidate_number = 1;
  return mtmn_config;
}
mtmn_config_t mtmn_config = app_mtmn_config();
face_id_name_list st_face_list;
static dl_matrix3du_t *aligned_face = NULL;
httpd_handle_t camera_httpd = NULL;
typedef enum
{
  START_STREAM,
  START_DETECT,
  SHOW_FACES,
  START_RECOGNITION,
  START_ENROLL,
  ENROLL_COMPLETE,
  DELETE_ALL,
} en_fsm_state;

en_fsm_state g_state;

typedef struct

{

  char enroll_name[ENROLL_NAME_LEN];

} httpd_resp_value;

httpd_resp_value st_name;

void setup() {

  Serial.begin(115200);

  Serial.setDebugOutput(true);

  Serial.println();

  digitalWrite(relay_pin, LOW);

  pinMode(relay_pin, OUTPUT);

  camera_config_t config;

  config.ledc_channel = LEDC_CHANNEL_0;

  config.ledc_timer = LEDC_TIMER_0;

  config.pin_d0 = Y2_GPIO_NUM;

  config.pin_d1 = Y3_GPIO_NUM;

  config.pin_d2 = Y4_GPIO_NUM;

  config.pin_d3 = Y5_GPIO_NUM;

  config.pin_d4 = Y6_GPIO_NUM;

  config.pin_d5 = Y7_GPIO_NUM;

  config.pin_d6 = Y8_GPIO_NUM;

  config.pin_d7 = Y9_GPIO_NUM;

  config.pin_xclk = XCLK_GPIO_NUM;

  config.pin_pclk = PCLK_GPIO_NUM;

  config.pin_vsync = VSYNC_GPIO_NUM;

  config.pin_href = HREF_GPIO_NUM;

  config.pin_sscb_sda = SIOD_GPIO_NUM;
  config.pin_sscb_scl = SIOC_GPIO_NUM;

  config.pin_pwdn = PWDN_GPIO_NUM;

  config.pin_reset = RESET_GPIO_NUM;

  config.xclk_freq_hz = 20000000;

  config.pixel_format = PIXFORMAT_JPEG;

  //init with high specs to pre-allocate larger buffers

  if (psramFound()) {

    config.frame_size = FRAMESIZE_UXGA;

    config.jpeg_quality = 10;

    config.fb_count = 2;

  } else {

    config.frame_size = FRAMESIZE_SVGA;

    config.jpeg_quality = 12;

    config.fb_count = 1;

  }

#if defined(CAMERA_MODEL_ESP_EYE)

  pinMode(13, INPUT_PULLUP);

  pinMode(14, INPUT_PULLUP);

#endif

  // camera init

  esp_err_t err = esp_camera_init(&config);

  if (err != ESP_OK) {

    Serial.printf("Camera init failed with error 0x%x", err);

    return;

  }

  sensor_t * s = esp_camera_sensor_get();

  s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_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");
  app_httpserver_init();
  app_facenet_main();
  socket_server.listen(82);
  Serial.print("Camera Ready! Use 'http://");
  Serial.print(WiFi.localIP());
  Serial.println("' to connect");
}


static esp_err_t index_handler(httpd_req_t *req) {
  httpd_resp_set_type(req, "text/html");
  httpd_resp_set_hdr(req, "Content-Encoding", "gzip");
  return httpd_resp_send(req, (const char *)index_ov2640_html_gz,
index_ov2640_html_gz_len);
}
httpd_uri_t index_uri = {
  .uri       = "/",
  .method    = HTTP_GET,
  .handler   = index_handler,
  .user_ctx  = NULL
};
```

```
void app_httpserver_init ()
{
  httpd_config_t config = HTTPD_DEFAULT_CONFIG();
  if (httpd_start(&camera_httpd, &config) == ESP_OK)
    Serial.println("httpd_start");
  {
    httpd_register_uri_handler(camera_httpd, &index_uri);
  }
}
void app_facenet_main()
{
  face_id_name_init(&st_face_list, FACE_ID_SAVE_NUMBER, ENROLL_CONFIRM_TIMES);
  aligned_face = dl_matrix3du_alloc(1, FACE_WIDTH, FACE_HEIGHT, 3);
  read_face_id_from_flash_with_name(&st_face_list);
}
static inline int do_enrollment(face_id_name_list *face_list, dl_matrix3d_t *new_id)
{
  ESP_LOGD(TAG, "START ENROLLING");
  int left_sample_face = enroll_face_id_to_flash_with_name(face_list, new_id, st_name.enroll_name);
  ESP_LOGD(TAG, "Face ID %s Enrollment: Sample %d",
           st_name.enroll_name,
           ENROLL_CONFIRM_TIMES - left_sample_face);
  return left_sample_face;
}
static esp_err_t send_face_list(WebsocketsClient &client)
{
  client.send("delete_faces"); // tell browser to delete all faces
  face_id_node *head = st_face_list.head;
  char add_face[64];
  for (int i = 0; i < st_face_list.count; i++) // loop current faces
  {
    sprintf(add_face, "listface:%s", head->id_name);
    client.send(add_face); //send face to browser
    head = head->next;
  }
}
static esp_err_t delete_all_faces(WebsocketsClient &client)
{
  delete_face_all_in_flash_with_name(&st_face_list);
  client.send("delete_faces");
}
void handle_message(WebsocketsClient &client, WebsocketsMessage msg)
{
  if (msg.data() == "stream") {
    g_state = START_STREAM;
    client.send("STREAMING");
  }
  if (msg.data() == "detect") {
    g_state = START_DETECT;
    client.send("DETECTING");
  }
  if (msg.data().substring(0, 8) == "capture:") {
    g_state = START_ENROLL;
    char person[FACE_ID_SAVE_NUMBER * ENROLL_NAME_LEN] = {0,};
    msg.data().substring(8).toCharArray(person, sizeof(person));
    memcpy(st_name.enroll_name, person, strlen(person) + 1);
    client.send("CAPTURING");
  }
  if (msg.data() == "recognise") {
    g_state = START_RECOGNITION;
    client.send("RECOGNISING");
  }
  if (msg.data().substring(0, 7) == "remove:") {
    char person[ENROLL_NAME_LEN * FACE_ID_SAVE_NUMBER];
    msg.data().substring(7).toCharArray(person, sizeof(person));
    delete_face_id_in_flash_with_name(&st_face_list, person);
    send_face_list(client); // reset faces in the browser
  }
  if (msg.data() == "delete_all") {
    delete_all_faces(client);
  }
}


void open_door(WebsocketsClient &client) {
  if (digitalRead(relay_pin) == LOW) {
    digitalWrite(relay_pin, HIGH); //close (energise) relay so door unlocks
    Serial.println("Door Unlocked");
    client.send("door_open");
    door_opened_millis = millis(); // time relay closed and door opened
  }
}
void loop() {
  auto client = socket_server.accept();
  client.onMessage(handle_message);
  dl_matrix3du_t *image_matrix = dl_matrix3du_alloc(1, 320, 240, 3);
  http_img_process_result out_res = {0};
  out_res.image = image_matrix->item;
  send_face_list(client);
  client.send("STREAMING");
  while (client.available()) {
    client.poll();
    if (millis() - interval > door_opened_millis) { // current time - face recognised time > 5 secs
      digitalWrite(relay_pin, LOW); //open relay
    }
    fb = esp_camera_fb_get();
    if (g_state == START_DETECT || g_state == START_ENROLL || g_state == START_RECOGNITION)
    {
      out_res.net_boxes = NULL;
      out_res.face_id = NULL;
      fmt2rgb888(fb->buf, fb->len, fb->format, out_res.image);
      out_res.net_boxes = face_detect(image_matrix, &mtmn_config);
      if (out_res.net_boxes)
      {
        if (align_face(out_res.net_boxes, image_matrix, aligned_face) == ESP_OK)
        {
          out_res.face_id = get_face_id(aligned_face);
          last_detected_millis = millis();
          if (g_state == START_DETECT) {
            client.send("FACE DETECTED");
          }
          if (g_state == START_ENROLL)
          {
            int left_sample_face = do_enrollment(&st_face_list, out_res.face_id);
            char enrolling_message[64];
            sprintf(enrolling_message, "SAMPLE NUMBER %d FOR %s", ENROLL_CONFIRM_TIMES - left_sample_face, st_name.enroll_name);
            client.send(enrolling_message);
            if (left_sample_face == 0)
            {
              ESP_LOGI(TAG, "Enrolled Face ID: %s", st_face_list.tail->id_name);
              g_state = START_STREAM;
              char captured_message[64];
              sprintf(captured_message, "FACE CAPTURED FOR %s", st_face_list.tail->id_name);
              client.send(captured_message);
              send_face_list(client);
            }
          }
          if (g_state == START_RECOGNITION  && (st_face_list.count > 0))
          {
            face_id_node *f = recognize_face_with_name(&st_face_list, out_res.face_id);
            if (f)
            {
              char recognised_message[64];
              sprintf(recognised_message, "DOOR OPEN FOR %s", f->id_name);
              open_door(client);
              client.send(recognised_message);
            }
            else
            {
              client.send("FACE NOT RECOGNISED");
            }
          }
          dl_matrix3d_free(out_res.face_id);
          }
         }
         else
         {
          if (g_state != START_DETECT) {
           client.send("NO FACE DETECTED");
          }
         }
         if (g_state == START_DETECT && millis() - last_detected_millis > 500) { // Detecting but no face detected
          client.send("DETECTING");
         }
        }
        client.sendBinary((const char *)fb->buf, fb->len);


        esp_camera_fb_return(fb);
        fb = NULL;
       }}
```

TESTING

Testing methodologies

For our project the following testing methodologies are chosen to ensure the robustness and reliability of the system:

Unit Testing:

Purpose: Verify the correctness of individual components, such as facial recognition algorithms and ESP32 functionalities.

Implementation: Develop unit tests for critical functions like face matching and relay control.

Integration Testing:

Purpose: Ensure seamless interaction between ESP32, camera module, and relay module.

Implementation: Test the integrated system to confirm that components work together harmoniously.

Functional Testing:

Purpose: Validate that the system performs its intended functions, such as unlocking the door upon successful facial recognition.

Implementation: Create test cases to simulate real-world scenarios and check if the expected functionalities are met.

Regression Testing:

Purpose: Detect and address any unintended side effects introduced during code modifications.

Implementation: Re-run previous test cases after updates to confirm that existing functionalities remain unaffected.

Performance Testing:

Purpose: Evaluate the system's responsiveness and stability, especially during face recognition and relay control.

Implementation: Measure response times and resource usage under varying conditions to ensure optimal performance.

Security Testing:

Purpose: Identify and mitigate potential vulnerabilities in the facial recognition and door lock control processes.

Implementation: Assess the system's resistance to unauthorized access and data protection measures.

Given the nature of the project, these testing methodologies collectively provide a comprehensive approach to ensure the system's functionality, security, and performance. Regular testing iterations throughout the development process will contribute to a robust and reliable facial recognition door lock system.

Unit Testing:

Test Facial Recognition Algorithm:

Description: Verify that the facial recognition algorithm correctly identifies enrolled faces.

Steps:

- Enroll a known face.
- Capture an image of the enrolled face.
- Confirm that the algorithm correctly matches the captured face.

Test Relay Control Function:

Description: Ensure that the relay control functions work as expected.

Steps:

- Activate the relay control function.
- Confirm that the relay switches to the open position.
- Deactivate the relay control function.
- Confirm that the relay switches back to the closed position.

Integration Testing:

Test ESP32 and Camera Module Interaction:

Description: Verify the interaction between ESP32 and the camera module.

Steps:

- Power on the system.
- Confirm that the ESP32 establishes a connection with the camera module.

Test Facial Recognition and Relay Control Integration:

Description: Confirm that facial recognition triggers the relay control.

Steps:

- Enroll a known face.
- Present the enrolled face to the camera.
- Confirm that the relay switches to the open position.

Functional Testing:

Test Door Unlocking:

Description: Validate that the door unlocks upon successful facial recognition.

Steps:

- Enroll a known face.
- Present the enrolled face to the camera.
- Confirm that the door unlocks.

Test Incorrect Facial Recognition:

Description: Ensure that the door remains locked for unrecognized faces.

Steps:

- Present an unrecognized face to the camera.
- Confirm that the door remains locked.

Performance Testing:

Test Response Time:

Description: Evaluate the system's response time during face recognition.

Steps:

- Enroll a known face.
- Measure the time taken for the system to unlock the door upon presenting the enrolled face.

Security Testing:

Test Unauthorized Access:

Description: Ensure that the system resists unauthorized access attempts.

- Steps:
- Present an unrecognized face to the camera.
- Confirm that the door remains locked.

Test Data Protection:

Description: Verify that enrolled facial data is securely stored.

Steps: Enroll a face and then confirm that the facial data is not accessible or tampered with.

Alpha Testing:

Alpha testing involves testing the software in a controlled environment by the development team or internal users before releasing it to external users.

Test Scenarios for Alpha Testing:

- Test Facial Recognition Algorithm
- Test Relay Control Function
- Test ESP32 and Camera Module Interaction
- Test Facial Recognition and Relay Control Integration
- Test Door Unlocking
- Test Incorrect Facial Recognition
- Test Response Time
- Test Unauthorized Access
- Test Data Protection

Alpha Testing Approach:

- Conduct tests in a controlled environment.
- Developers and internal users perform the tests.
- Identify and report any bugs or issues.
- Iterate on fixes based on feedback.

Beta Testing:

Beta testing involves releasing the software to a limited number of external users to gather feedback and identify any issues before the full release.

Test Scenarios for Beta Testing:

- Test Facial Recognition Algorithm
- Test Relay Control Function
- Test ESP32 and Camera Module Interaction
- Test Facial Recognition and Relay Control Integration
- Test Door Unlocking
- Test Incorrect Facial Recognition
- Test Response Time
- Test Unauthorized Access
- Test Data Protection

Beta Testing Approach:

- Recruit a group of external users.
- Provide them with access to the system.
- Gather feedback on usability, performance, and functionality.
- Monitor for any issues reported by beta testers.
- Address reported issues and iterate on improvements.

Acceptance Testing:

Acceptance testing involves validating that the software meets the requirements and is ready for deployment.

Test Scenarios for Acceptance Testing:

- Test Facial Recognition Algorithm
- Test Relay Control Function
- Test ESP32 and Camera Module Interaction
- Test Facial Recognition and Relay Control Integration
- Test Door Unlocking
- Test Incorrect Facial Recognition
- Test Response Time
- Test Unauthorized Access
- Test Data Protection

Acceptance Testing Approach:

- Users or stakeholders perform the tests.
- Ensure that the system meets all specified requirements.
- Verify that all identified issues from alpha and beta testing have been resolved.
- Sign off on the system for deployment if it meets the acceptance criteria.

CONCLUSION

The ESP32 Cam Door Lock project represents an innovative solution for access control, integrating facial recognition technology with relay control functionality. Throughout development and testing, the system has demonstrated its capability to provide secure and efficient door unlocking based on recognized faces. By leveraging the ESP32 microcontroller and camera module, the system ensures reliable interaction and seamless integration. This project not only enhances security but also offers convenience and ease of use through its intuitive interface. With its successful implementation, the ESP32 Cam Door Lock stands ready to meet the access control needs of user and stakeholders.

FURTHER ENCHANCEMENTS

In addition to the existing functionality, integrating a mobile app for access to the door lock system can offer several enhancements:

- Remote Access: Users can remotely control the door lock from anywhere using their smartphones, providing convenience and flexibility.
- User Management: The mobile app can allow administrators to manage user access rights, enroll new users, and revoke access as needed, enhancing security and control.
- Real-time Notifications: Users can receive real-time notifications on their mobile devices for door activities, such as successful unlocks or failed attempts, improving monitoring and security awareness.
- Integration with Other Systems: The mobile app can be integrated with other systems, such as attendance or visitor management systems, for seamless data exchange and enhanced functionality.
- Enhanced User Experience: With a mobile app interface, users can enjoy a more intuitive and user-friendly experience, making it easier to interact with the door lock system.
- Biometric Authentication: Leveraging mobile device biometric features like fingerprint or face recognition for authentication can further enhance security and user convenience.
- Audit Trail: The mobile app can provide access to an audit trail of door activities, allowing users to review access logs and monitor usage history.

By implementing a mobile app for access to the door lock system, we can extend its functionality, improve user experience, and provide additional features for enhanced security and convenience.

REFERENCES

- https://youtu.be/mu3-Sff0B9w?si=_aWVAUip2Wek8P9M
- GitHub - robotzero1/esp32cam-access-control: Open a door when a face is recognised using the ESP32-CAM
- Access Control with Face Recognition – Robot Zero One
- ESP32-CAM Face Recognition Door Lock System using Solenoid Lock (circuitdigest.com)