SMART EYE SURVILLENCE SYSTEM

PROJECT REPORT

Submitted in partial fulfilment for the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING WITH SPECIALIZATION IN CYBER PHYSICAL SYSTEMS

by

YUVRAJ SINGH (21BPS1020)

DEVANSHI NIGAM (21BPS1457)

PRANAV KAUSHIK (21BRS1319)

Under the Guidance of

Dr. Sritama Roy

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

VELLORE INSTITUTE OF TECHNOLOGY

CHENNAI - 600127

November, 2024

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

CHENNAI

DECLARATION

I hereby declare that the thesis entitled "**Smart Eye Survillence System**" submitted by **Devanshi Nigam (21BPS1457)** for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** with **specialization in Cyber Physical Systems**, Vellore Institute of Technology, Chennai is a record of bona-fide work carried out by me under the supervision of **Dr. Sritama Roy**.

I further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or University.

Place: Chennai

Date:

Signature of the Candidate

DEVANSHI NIGAM

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)
CHENNAI

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the report entitled **Smart Eye Survillence System** is prepared and submitted by **Devanshi Nigam (21BPS1457)** to Vellore Institute of Technology, Chennai, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering with specialization in Cyber Physical Systems** is a bona-fide record carried out under my guidance. The project fulfills the requirements as per the regulations of this University and in my opinion meets the necessary standards for submission. The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma and the same is certified.

Signature of the Guide:
Name: Dr. Sritama Roy
Date:

Signature of the Examiner
Name:
Date:

Signature of the Examiner
Name:
Date:

Approved by the Head of Department,
Computer Science and Engineering with specialization
in Cyber Physical Systems

Signature
Name: Dr. Renuka Devi
Date:

(Seal of the School)

ABSTRACT

The Smart Eye Security Surveillance System revolutionizes modern security by addressing the limitations of traditional systems that rely on basic cameras and motion detectors. Conventional setups often generate false alarms from harmless movements and suffer from slow response times, reducing trust and wasting resources. Smart Eye overcomes these issues by integrating high-resolution cameras with infrared (IR) technology and advanced algorithms to analyze eye movements and facial features in real time. This precision minimizes false alarms while enabling fast and targeted responses, significantly enhancing overall security.

A standout feature of Smart Eye is its seamless IoT (Internet of Things) integration, which ensures transparent communication between connected devices. This enables a cohesive and efficient security network capable of quickly coordinating responses, such as locking doors or notifying security teams. By automatically analyzing anomalies and triggering appropriate actions, Smart Eye enhances situational awareness and ensures swift containment of potential threats.

Experimental trials confirm that the system improves detection accuracy and response speed, preventing security breaches before incidents escalate. Its adaptive learning capability further boosts reliability, allowing it to evolve with emerging threats. More than just a monitoring tool, Smart Eye redefines surveillance with its intelligent, proactive approach, setting a new benchmark in dependable, efficient security technology.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Dr. Sritama Roy., Assistant Professor Grade 2, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, for her constant guidance, continual encouragement, understanding; more than all, she taught me patience in my endeavor. My association with her is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of Cyber security and computer vision.

It is with gratitude that I would like to extend my thanks to the visionary leader Dr. G. Viswanathan our Honorable Chancellor, Mr. Sankar Viswanathan, Dr. Sekar Viswanathan, Dr. G V Selvam Vice Presidents, Dr. Sandhya Pentareddy, Executive Director, Ms. Kadhambari S. Viswanathan, Assistant Vice-President, Dr. V. S. Kanchana Bhaaskaran Vice-Chancellor, Dr. T. Thyagarajan Pro-Vice Chancellor, VIT Chennai and Dr. P. K. Manoharan, Additional Registrar for providing an exceptional working environment and inspiring all of us during the tenure of the course.

Special mention to Dr. Ganesan R, Dean, Dr. Parvathi R, Associate Dean Academics, Dr. Geetha S, Associate Dean Research, School of Computer Science and Engineering, Vellore Institute of Technology, Chennai for spending their valuable time and efforts in sharing their knowledge and for helping us in every aspect.

In jubilant state, I express ingeniously my whole-hearted thanks to Dr. Renuka Devi, Head of the Department, B.Tech. Computer Science and Engineering with specialization in cyber physical systems and the Project Coordinators for their valuable support and encouragement to take up and complete the thesis.

My sincere thanks to all the faculties and staffs at Vellore Institute of Technology, Chennai who helped me acquire the requisite knowledge. I would like to thank my parents for their support. It is indeed a pleasure to thank my friends who encouraged me to take up and complete this task.

Place: Chennai

Date: Devanshi Nigam

CONTENT

LIST OF FIGURES

LIST OF ACRONYMS

MCU- Microcontroller Unit

GDPR – General Data Protection Regulation

CCPA – California Consumer Privacy Act

CCTV-Closed-Circuit Television

MFA- Multi-Factor Authentication

KPI- Key Performance Indicator

ACM- Adafruit Cloud Messaging

MFA -Multi-Factor Authentication

API - Application Programming Interface

SDK- Software Development Kit

KPI - Key Performance Indicators

SSL- Secure Sockets Layer

2FA- Two-Factor Authentication

URL- Uniform Resource Locator

HTTP- Hypertext Transfer Protocol

Chapter 1

Introduction

1.1 "Revolutionizing Surveillance with Smart Eye: Advanced Security Through CV and IoT Integration"

The Smart Eye Security Surveillance System is a revolutionary advancement in modern security, addressing the shortcomings of traditional setups in residential, commercial, and public spaces. Conventional systems relying on standard cameras and basic motion detectors often produce excessive false alarms and delayed responses. This inefficiency stems from their inability to differentiate between harmless movements and potential threats, leading to resource wastage and diminished reliability.

The Smart Eye overcomes these limitations using computer vision and libraries like NumPy, paired with advanced hardware, including Adafruit sensors. High-resolution and infrared-enabled cameras, combined with cutting-edge image processing, enable real-time movement analysis and facial recognition. These features allow the system to distinguish between routine and suspicious activities with remarkable precision, reducing false alarms and enabling targeted responses. Optimized for edge devices, it processes data efficiently, ensuring immediate and effective threat management.

A standout feature is its IoT integration, which connects devices within the security network. This enables seamless communication, enhancing situational awareness and swift responses, such as notifying security personnel, locking doors, or triggering alarms. Experimental trials show superior accuracy and faster responses compared to traditional systems. Its adaptive learning ensures continuous improvement, making it effective against evolving threats.

Smart Eye redefines security, providing a proactive, intelligent solution for modern needs.

Conclusion

The Smart Eye Security Surveillance System sets a new benchmark in automated security technology with its blend of computer vision algorithms, real-time analysis capabilities, and IoT connectivity. The result is a highly effective surveillance system that adapts seamlessly to diverse security requirements while ensuring top-tier performance. Its scalable design, leveraging NumPy and Adafruit technologies, prepares it for upgrades to tackle emerging security challenges. This makes Smart Eye the most reliable, fast, and robust framework for addressing future security needs.

1.1.1 Background and Importance of Advanced Surveillance

Security threats have evolved dynamically, rendering traditional static camera systems and simple motion sensors insufficient. These conventional systems primarily react to situations rather than intervening in real-time, often producing false positives due to their inability to differentiate normal activities from potential threats. This inefficiency

leads to unnecessary alerts and heavy reliance on human intervention, delaying critical responses.

In dynamic and high-traffic environments like stadiums, airports, or shopping malls, traditional surveillance faces even greater challenges. Individuals may conceal their identities using masks, hats, or other objects, rendering conventional facial recognition ineffective. Furthermore, the sheer volume of data generated in such settings overwhelms these systems, making unseen threats and delayed responses inevitable.

The Smart Eye Surveillance System offers a transformative solution by integrating real-time data processing with computer vision algorithms based on behavioral analysis. It detects threats using behavioral cues like erratic eye movements, body language, and facial expressions, going beyond basic motion detection. Equipped with infrared imaging, it operates reliably in both high and low-light conditions, making it suitable for diverse environments.

This intelligent system ensures faster responses, minimizes false alarms, and reduces reliance on human intervention. Ideal for hospitals, government buildings, schools, and corporate offices, the Smart Eye enhances security, public safety, and peace of mind while safeguarding lives and assets effectively.

1.2 Emergence of the Smart Eye System

The Smart Eye Security Surveillance System redefines modern security with advanced technologies, integrating computer vision, IoT, and cloud platforms like Adafruit for a highly automated solution. Unlike traditional systems limited to motion detection, Smart Eye analyzes human behavior and facial features, including eye movements, to detect potential threats. Its infrared (IR) camera capabilities ensure reliable performance in low-light or pitch-dark conditions, surpassing conventional cameras.

Powered by computer vision algorithms and data manipulation tools like NumPy, the system processes video feeds in real time, identifying suspicious behaviors such as erratic movements or prolonged facial obfuscation. Adafruit sensors enhance hardware-software integration, ensuring precise data collection. This data is stored and processed on the Adafruit cloud platform, enabling reliable, scalable storage and real-time accessibility for security logs and flagged alerts.

Smart Eye operates autonomously, minimizing human intervention. Upon detecting threats, it triggers automated responses like sending alerts, activating alarms, locking doors, or disabling elevators, ensuring faster and more effective countermeasures. Its scalability makes it ideal for diverse environments, from homes to high-security public spaces.

By combining advanced detection, automation, and cloud integration, the Smart Eye system sets a new standard in security, offering a proactive, efficient, and scalable approach to safeguarding lives and assets.

1.3 LIMITATIONS OF TRADITIONAL SURVEILLENCE SYSTEM

Traditional surveillance systems, which historically encompass the simplest CCTV cameras and motion detectors to the more complex alarm systems, have been a part of security infrastructure for many decades. Although these systems are effective within their specific simple scenario applications, they hold very significant and crippling constraints on their performance in the extremely dynamic security environments. Currently rising levels of security sophistication and complexity are illustrating inadequacies of traditional surveillance systems. Some of the key limitations of these systems are outlined as follows:

High False Alarm Rates

Conventional surveillance systems often suffer from high false alarm rates, triggered by innocuous movements such as swaying branches, pets, or environmental changes like shadows or lighting. Motion detectors and basic cameras frequently misinterpret these as threats, leading to unnecessary alerts and requiring human intervention to verify authenticity. This issue results in alert fatigue, where security personnel overlook genuine threats due to constant false notifications. Additionally, it increases operational costs and wastes resources as staff verify countless false alarms. Ultimately, these inefficiencies undermine the system's reliability, reducing its effectiveness and credibility in ensuring security.

Delayed Response Times

Classical security systems often fail to respond promptly to threats due to inherent delays. Alerts are typically triggered only after significant activity, and responses rely heavily on human operators to evaluate and act, causing latency. In large or complex environments like malls or airports, overwhelmed personnel struggle to monitor numerous feeds, delaying threat identification. This delay can lead to catastrophic outcomes in urgent situations like break-ins or active shooter incidents, allowing intruders to escape or cause harm. Lacking behavioral analysis and automated responses, traditional systems leave critical gaps in security, particularly in high-risk scenarios requiring immediate action.

Lack in Facial Recognition and Detection

Facial recognition is vital for modern surveillance, yet conventional systems face significant limitations. They struggle with partial occlusions, such as masks, hats, or sunglasses, making it easy for individuals to evade detection. Reliance on 2D images leaves them vulnerable to lighting changes, like glare or dim conditions, reducing accuracy in low-light scenarios. They also fail to recognize subtle facial expressions indicating suspicious behavior. Moreover, traditional systems often rely on narrow training datasets, introducing biases and reducing accuracy across diverse demographics, including age, ethnicity, or gender. These challenges result in frequent false positives and negatives, undermining the reliability of traditional facial recognition.

1.4 OVERVIEW

The Smart Eye Security Surveillance System overcomes the limitations of traditional systems by integrating high-resolution cameras, ESP32 technology, computer vision algorithms, NumPy for data processing, and IoT capabilities. Unlike conventional setups prone to false alarms and delayed responses, it reliably detects threats in real-time, even in low-light conditions or with partial obstructions. Eye-focused detection and behavioral analysis allow it to identify subtle indicators like erratic movements or unusual expressions. Adafruit sensors enhance precision, while its cloud platform ensures secure data storage and retrieval. Scalable and adaptive, the system suits diverse applications, offering a robust, efficient, and future-ready security solution.

1.4.1 Core Objectives and Vision

The core aim of the Smart Eye Security Surveillance System, therefore, is to cause a paradigm shift in the whole surveillance concept by providing a far much more accurate, efficient, and autonomous kind of security solution. Traditional surveillance systems are characterized by a heavy reliance on human monitoring and suffer from the shortcomings of false alarms and delayed response times and inefficiencies in detecting covert threats. Smart Eye seeks to overcome these weaknesses by integrating real-time behavioral analysis together with automated responses to detected threats so that potential risks are identified and managed promptly without further human intervention.

A key pillar of Smart Eye's vision is the development of security standards in any industry. The system is a high-resolution video capture with real-time behavioral analysis, working under various lighting conditions due to ESP32 cameras as well as in many other conditions where traditional systems would typically fail. Computer vision and Numpy models on the system evolve adaptively to enhance its accuracy and reliability across different settings and evolving security challenges.

Smart Eye is made to work well in both residential and commercial spaces, for a scalable, cost-effective, and autonomous security system. It ranges from a family house to an office building or even a high-risk public place; its functionality can be adapted based on the security needed. It ensures users are given peace of mind due to the ample monitoring at their premises with the cutting-edge technology and less human oversight or manual intervention.

After all, the Smart Eye system will envision a new and trending standard of intelligent surveillance by integrating automation, adaptability, and precision to make the world smarter and more secure..

1.4.2 Key Features and Differentiators

What differentiates Smart Eye Security from the conventional solutions of security is that it offers an advanced security surveillance system that includes several features that have the advantage of enhancing the effectiveness and flexibility of the system. Among those features and differences are the following

1. Advanced Facial and Eye Recognition:

• While other systems cannot detect and identify eye movements and facial features if parts of a person's face are covered, Smart Eye can.

This capability is essential to realizing high accuracy in detection if masks, or other coverings, were applied to a person's face and would interfere with data gathering.

• The models based on Computer Vision incorporated in the system intend to update, in terms of changes made in environments, the capabilities for identification and detection while increasing accuracy in detection overtime.

2. ESP32 Technology:

• High-resolution ESP cameras with high-resolution video capture in low-light or nighttime conditions, ensuring this system can even perform adequately under dim-lit settings, whereas the conventional systems usually struggle to effectively operate under such conditions. This makes Smart Eye a very effective solution for 24/7 surveillance.

3. Real-time Behavioural Analysis

• Smart Eye is not only a motion detector but analyzes human behavior in real-time to identify intent. Utilizing techniques of higher order, the system can pinpoint suspicious activities such as rapid eye movements, prolonged occlusions of the face, or unusual body behaviors.

• The system can then itself decide an appropriate response, which can be alerting security personnel through messages, lock security doors, or implement other security protocols.

4. Automated Threat Response:

• This system has IoT integrated into its system so that when it detects the threat, then it automatically acts without putting in any human effort. For example, it can lock doors, or even surveillance cameras focus on the areas that it detects threats.

• The multi-layered response mechanism ensures that action is taken within minimal chances of the breach or late responses.

5. Scalability and Adaptability:

Scalable and adaptable to any residential, corporate, or public space based on the environment, as its versatility can adapt to any type of environment so it will be able to work well with other existing systems and devices within IoT to be able to function better.

6. Self-Sustained Human Monitoring Dependence:

That is, the most significant differences which can be visually observed are that Smart Eye works independently and doesn't need constant human intervention to make decisions with regard to any lock going off. It is unlike other conventional systems, where constant involvement of humans is required in order to verify an alarm and finally take any decisions. Smart Eye's features allow it to take decisions and work as

an independent mechanism with minimal dependency on continuous human monitoring.

1.5 CHALLENGES

Technical Limitations

• Processing Power and Real Time Analysis

One of the most important technological requirements of such a system is that it should be able to process very high power to offer real-time facial and behavioral analysis. With a high-resolution IR camera and constant video feeds, tremendous amounts of data are generated at real-time data processing points for timely identification of threats. This results in latency and slower response times in computations, such as in those on a Arduino or similar devices.

For maximum performance, the system would require more power-hungry processors, which would be expensive and limits scalability even further, especial for those who are on a limited budget.

• Data Storage and Bandwidth Requirements

Continuous video capture and analysis results in limitations regarding data storage and network bandwidth. Large video files, whether high-definition or multiple streams of real-time video, have enormous storage capacity if the data is to be retained for later use. Realtime video is sent to cloud servers or remote monitoring stations by network bandwidth when internet connectivity within the environment is not reliable. This would both affect the system's performance and increase operational costs.

• Environmental Interference

Environmental factors may also influence the performance of the system. Bad light, weather changes, or obstructions due to physical barriers may affect the degree to which the facial and eye recognition systems will be accurate. While ESP capabilities would add precision in details capture, there is a possible limitation in the capture of details, specifically on expressions or gestures, especially if they are quite subtle. In addition, a high-traffic area may make differentiation between normal and abnormal behaviors unattainable as there may be a high possibility of false alarms or missed threats.

• Model Limitations and Adaptability

Computer vision-based systems with Adafruit hardware and cloud platforms face limitations in dynamic environments. These systems require well-curated datasets to accurately detect behaviors and adapt to varying lighting or conditions. Insufficient training data can affect performance in edge cases. Additionally, Adafruit cloud

platforms may struggle with handling large-scale data streams, leading to latency or incomplete data capture, compromising efficiency. The reliance on preconfigured algorithms limits adaptability to unforeseen behaviors without manual updates. These challenges underscore the need for optimized computer vision models and robust cloud infrastructure to enhance scalability, adaptability, and reliability in real-world applications.

Ethical and Privacy Concerns

• Privacy Invasion

The highly widespread use of facial recognition and behavioral analysis raises deep privacy concerns. This is because it involves the capture and analysis of personal traits such as facial features and eye movements in public or semi-public environments. Persons may feel uneasy or even under surveillance, thus contributing to a lack of acceptance or even anti-societal opposition. There is an added complexity with open or crowded spaces, given that the system must meet the criteria of privacy laws, such as GDPR or CCPA, and there is a need to ensure that the person under surveillance has given explicit consent**.**

• Data Security &Vulnerability Risks

It is also liable to cybersecurity attacks based on the fact that it keeps all sorts of sensitive information together, including video feeds and facial data. Unauthorized access to such data could lead to grave privacy breach and misuse of personal information. Consequently, the system must have high-grade encryption and data security protocols against hacking and unauthorized data access. High-level security measures, however, lead to increase in operational costs and complexity of deployment and upkeep.

• Bias in CV Algorithm

Facial recognition and behavioral analysis algorithms are highly prone to race-based, gender-based, or age-based bias because these algorithms act precisely according to the data under which they are designed to function. If diverse datasets representing wide ranges of demographics are not provided, then the system may produce biased outcomes and may have higher error rates for specific groups. This bias does not only make the system less effective but also to pose ethical questions, mainly that it has the potential to illustrate unfair treatment or profiling based on inherent characters**.**

• Potential Misuse and Scope Creep

There also exists a risk that the technology will be misused for purposes that are beyond the intended use. Such a system may raise ethical concerns in surveillance activities as

it recognizes and analyzes facial expressions, whereby someone is been monitored in the absence of the necessary oversight or accountability. There must be strong policies and oversight for this reason; just because technology at its core is very good for security does not mean technology loses its intended use to enhance security.

1.6 PROJECT STATEMENT

Traditional security systems, relying on basic cameras and motion sensors, often fail to identify threats accurately, generating false alarms and missing real dangers. They struggle with subtle behavioral cues or facial expressions, leading to delayed responses.

The Smart Eye Security Surveillance System solves these issues using advanced computer vision algorithms, NumPy for efficient data processing, and Adafruit sensors. It autonomously detects threats by analyzing eye and facial features in real-time, adapting to various environments and lighting conditions. Data is securely stored on the Adafruit cloud platform for easy access, offering a reliable and proactive security solution for diverse applications.

Chapter 2

Methodology

The Smart Eye Security Surveillance System utilizes the latest technology to enhance the accuracy and efficiency of security surveillance. High-resolution cameras coupled with infrared capability, IoT automation, and advanced computer vision algorithms make real-time facial recognition and behavioral analysis possible. Unlike conventional surveillance systems, which often fail in terms of threat accuracy, generate false alarms, and experience time delays in responding, the Smart Eye system delivers faster and more reliable performance.

The system leverages NumPy for efficient data processing, enabling rapid computation and analysis of image data for accurate detection and response. Adafruit sensors enhance the system's ability to capture environmental variables and detect anomalies, ensuring precision in threat identification. Additionally, the system integrates with the Adafruit cloud platform to securely collect, store, and manage images and logs, ensuring quick access to valuable data for further analysis.

2.1 SYSTEM OVERVIEW

The Smart Eye Security Surveillance System is a revolutionary approach to transforming traditional security surveillance into a real-time, adaptive solution that incorporates computer vision, NumPy, and Adafruit hardware for autonomous facial recognition, behavioral analysis, and IoT-enabled responses. Unlike standard security systems that rely mainly on basic motion sensors and camera feeds, Smart Eye combines high-resolution imaging, infrared (IR) technology, and advanced algorithms to address the common challenges of false alarms and human oversight dependency.

The system focuses more on behavioral analysis, specifically eye movements and facial patterns, which significantly enhances threat detection accuracy. This focus allows Smart Eye to adapt to various environments, whether it's a home, workplace, or a highly secured public space. Its ability to independently analyze and respond to suspicious behaviors ensures that it doesn't rely solely on manual monitoring. Below are the main stages of the system:

Real-Time Video Capture and Eye Detection

The first stage of the Smart Eye system involves capturing high-resolution video using IR-enabled cameras. This ensures that the system performs effectively even in low light conditions. IR imaging allows for precise recognition of facial features and eye movements, which are key indicators of human behavior. Eye detection is particularly useful because eyes often remain visible even when other facial features are obscured, such as by masks or hats. These are common methods used to evade detection, and Smart Eye addresses this challenge by maintaining accurate recognition despite partial obstructions.

The cameras continuously scan for events within the monitored area and send video data to computer vision algorithms in real time. These algorithms focus on eye movements and facial landmarks that may suggest suspicious behavior, such as erratic eye movements or unusual facial expressions. This level of granular data analysis allows Smart Eye to accurately identify potential threats in real time.

Behavioral Analysis and Threat Assessment

The next stage is the processing of video data captured by the cameras. Computer vision algorithms and NumPy for efficient data manipulation are employed to analyze eye movements and facial expressions, identifying behavioral patterns that may signal unusual or suspicious activity.

For example, the system is trained to detect subtle signs of malice or distress, such as quick eye movements or fleeting facial expressions indicating unease. The use of NumPy accelerates the data processing, ensuring that the system can quickly identify patterns even in complex, dynamic environments. The data captured by the cameras is continuously analyzed to differentiate between normal and suspicious behavior, flagging anything that falls outside of regular patterns for closer examination.

Automated Response and Notification

Once a potential threat is detected through behavioral analysis, Smart Eye triggers an automatic response. The system's response is tiered, with the level of action determined by the assessed severity of the threat. The system uses IoT integration to automate actions such as alerting security personnel or property managers, locking doors, or denying access to specific areas to contain or prevent escalation of the threat.

In lower-risk situations, the system will notify security staff to review the footage and take appropriate action. In higher-risk situations, it may lock doors, activate alarms, or initiate other pre-programmed security protocols, all without the need for human intervention. Adafruit is used as the cloud platform to securely collect, store, and manage the captured images and video footage, ensuring that data is easily accessible for analysis and future reference.

This networked, automated approach minimizes the need for continuous human monitoring and ensures that responses are quick and proportional to the perceived threat. By automating threat detection and response, Smart Eye reduces risks associated with human reaction time and improves overall security efficacy.

2.2 SYSTEM ARCITECTURE

The Smart Eye Security Surveillance System is equipped with a multi-layered architecture to deliver high-precision, responsive security monitoring. Each structure of the system, from the input devices and the processing unit to the output and response mechanisms, pertains to a specific function, providing space for accurate real-time analysis and reaction. Such an architecture offers efficient monitoring and timely reaction to threats, regardless of whether it is in the residential sector or highly

secure public space. Below are details regarding every architectural component:

1. Input Devices

The surveillance process begins with the ESP32-CAM module, which serves as the primary input device for the system. The ESP32-CAM is a small yet powerful camera module equipped with Wi-Fi and Bluetooth capabilities, making it ideal for real-time video capture in a wireless setup. The camera captures high-resolution images of the monitored area, and its IR functionality ensures that the system operates effectively even in low-light or nighttime conditions, where traditional cameras might fail. The IR feature highlights essential visual information in dark environments, ensuring that the system can detect potential threats under all lighting conditions.

The NodeMCU ESP8266 serves as the control unit in this setup, connecting the ESP32-CAM to the broader network and enabling wireless communication. It integrates seamlessly with the Adafruit cloud platform for storing images and data. This setup allows the system to send captured data in real-time for processing and analysis.

The key feature of this system lies in the use of computer vision (CV) algorithms. These algorithms, running on the ESP32-CAM and processed via Adafruit-based sensors, analyze video feeds to track facial features and eye movements. Eye movements are particularly significant because they are harder to conceal than other facial features, making them a reliable indicator of suspicious behavior. Even if an individual wears a mask, sunglasses, or a hat, the system can still effectively track their eye movements, which provides a clear advantage over traditional facial recognition systems.

Captured facial data, processed with CV techniques, is stored securely on the Adafruit cloud platform for further analysis. Adafruit serves as the database that handles the storage and retrieval of images, allowing the system to assess behavior over time and refine threat detection capabilities.

Once a potential threat is detected through abnormal eye movement or facial patterns, the system triggers an automated response. Relay modules and solenoid locks are integrated into the system to secure access points. The relay module activates the solenoid lock, locking doors or gates to prevent unauthorized entry. These components provide an immediate and physical response to potential threats, ensuring faster action compared to manual intervention.

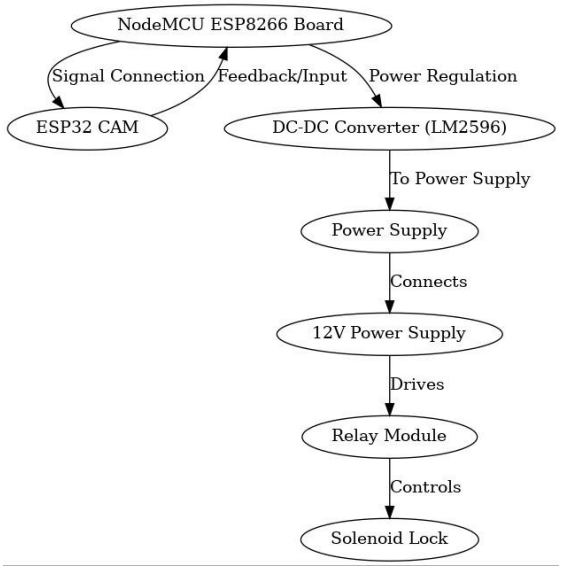
Additionally, a DC to DC converter is used to manage and regulate the power supply to the system, ensuring that all components operate efficiently without risk of voltage fluctuations.

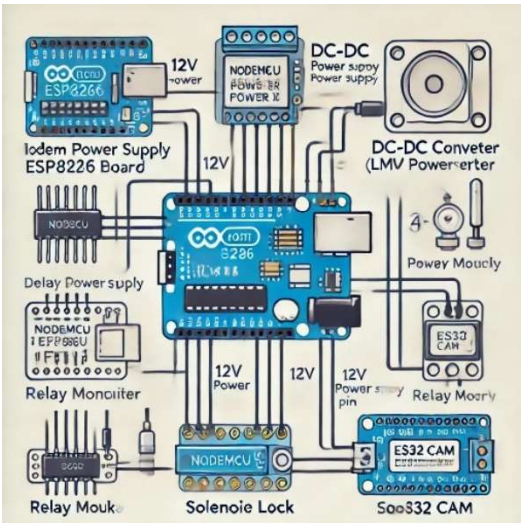
This system combines high-resolution camera input, powerful processing via CV algorithms, and automated security measures, all integrated with cloud storage and IoT capabilities to create a fully automated and highly adaptable security solution.

2. Processing Unit

The processing unit is the core of the system's data analysis capabilities, processing video in real time and behavioral analysis. This can range from a simple Raspberry Pi for less complex applications to more sophisticated computing machines able to handle big data-intensive, complex ones. The processing unit interprets visual data coming from the several devices-eye movement activities, face formations, and body language-of those being analyzed, looking for anomalies in behavior.

What drives such an analysis in a system is OpenCV, which happens to be the most widely used computer vision library. Using it means detecting and tracking facial features and the eye regions contained within each frame of video. For foundational support, facial landmarks are segmented and then fed into computer vision models trained to recognize specific patterns of movement and expressions that would correspond with distress, evasion, or other behaviors that might herald a potential threat.





{a} {b}

a) System Flowchart for IoT-based Access Control: Integrating NodeMCU ESP8266, ESP32 CAM, DC-DC Converter, and Relay Module to Operate a Solenoid Lock.
b) Hardware Circuit Diagram: Integrating NodeMCU ESP8266, ESP32 CAM, DC-DC Converter, Relay Module, and Solenoid Lock for IoT-Based Secure Access Control System.

Through computer vision (CV) and advanced processing techniques using NumPy, the system can analyse subtle behavioural indicators such as rapid or jerky eye movements, prolonged obstructions on the face, or signs of agitation. Specifically, the system utilizes pre-trained models like shape_predictor_68_face_landmarks and human_face_detector to detect facial landmarks and identify human faces in the video feed. These models help the system to recognize fine details such as facial expressions and eye movement patterns that could indicate suspicious behaviour.

With the integration of Adafruit sensors for real-time data capture and the Adafruit cloud platform for storing and analyzing images, the Smart Eye system processes

video and sensor data efficiently. Adafruit serves as the cloud database to collect and store images, as well as any related data files, which are essential for future analysis and improving the system's accuracy over time.

By applying computer vision algorithms to real-time data, the system can automatically identify anomalies and compare them to pre-classified behavioral patterns stored in the system. This enables fast and accurate threat assessment, allowing the system to detect potential security risks with minimal delay, even in dynamic or crowded environments. The continuous learning and adaptation process ensures that the Smart Eye system becomes increasingly accurate in distinguishing between normal and suspicious behavior, ultimately enhancing its reliability in diverse situations.

3. Output and Responses

The output and response layer of the Smart Eye system assures swift and appropriate action when a potential threat is identified. This layer constitutes real-time alerts and IoT-integrated automated responses, thus enabling the system to further handle security incidents independently, often faster than human intervention would have achieved.

As soon as the system identifies a potential threat, it automatically sends messages to receivers like security officers, the property owners, or any interested person involved in the case through csv and json file.

Apart from alerting personnel, the Smart Eye system integrates IoT to respond physically automatically. The response might include locking doors, activating additional cameras, limiting access points, or sounding alarms whenever this threat is detected to be contained. For more sensitive areas such as offices or public spaces, the system may limit elevator access, brighten specific lighting, or activate other preprogrammed security systems by employing IoT devices.

A multi-tiered response mechanism means that any threat is managed completely, regardless of human response times. Smart Eye puts together real-time alerts and automated, context-sensitive actions to provide a robust and reliable approach towards threat mitigation, even if it reduces the reliance on fulltime, continuous manual monitoring and allows the system to act independently when it counts-mostly in seconds. This could avoid the occurrences of incidents to propel into more serious incidents and would give property owners the peace of mind by assuring that the system is proactive and responsive.

2.3 TECHNOLOGY STACK

The Smart Eye Security Surveillance System is built up based on the technology stack of high-performance hardware, efficient software, and highly scalable cloud services. This integration would create real-time monitoring, accurate behavior analysis, and immediate response capability. Below, each component of the system's technology

stack—Hardware, Software, and Cloud Services—is outlined to give an overall comprehension of how these elements work together to produce a secure and responsive surveillance solution.

Hardware and Software

The Smart Eye system is based on specifically designed hardware that offers the basis of constant, real-time Image surveillance and processing. Major hardware parts within the system include a high-resolution ESP camera and a processing unit - potentially an area with more developed embedded computing devices, which could be a NodeMCU module. In direct combination with each other, these two parts enable the system to function easily indoors and outdoors, capture images and resolution in light and dark environments.

- **High-Resolution IR Cameras:** The cameras are essential in providing the high-quality visual data required for accurate facial and behavioral analysis. As a result of the IR capability, the system can capture clear images in low-light or nighttime settings without need for additional lighting in diverse environments. The cameras are high-resolution making even slight details clear such as quick facial expressions and eye movements. This is also a necessary characteristic to ensure that all levels of detail are included for the proper detection of suspicious activities or unusual behavior, particularly in security-sensitive areas.
- **Processing Unit:** The processing unit normally takes the form of a NodeMCU ESP 8266 module, within which real-time data are processed onsite. In this setup, latency is reduced, meaning that swift assessments will be completed solely on local servers and not distant servers. In addition, the low power consumption and flexible processing capacity of fit all the bill in their runtime to execute various lightweight algorithms for real-time management of Image feeds. A stronger computing unit should be used in the case of larger deployment or in those cases where more intensified processing is required to guarantee smooth, continuous operation. The unit allows the system to process data on-site efficiently with high accuracy and reduced network dependency.
- **Data Storage:** The system can either employ local storage or cloud-based storage based on the quantity of data to be retained and retrieval needs from an organization. Local storage is ideal for small installations or where data privacy is paramount. Cloud-based storage has scalability, making it apt for larger setups where data retrieval from multiple locations is needed. By balancing storage options, the system remains adaptable to different operational requirements.

Cloud Services

The cloud services component of the technology stack offers scalability flexibility, as well as secure data handling abilities to the system. The Smart Eye system could manage large volumes of data, perform real-time analysis, and scale through multiple locations without sacrificing the speed or accuracy by using platforms like AWS, Google Cloud, or Microsoft Azure.

- **Scalable Data Storage:** Cloud storage allows the Smart Eye system to safely store image data. Therefore, they can easily restore and analyze the image when needed. For organizations with serious security needs, cloud storage accommodates a large amount of video footage and scales as necessary. Data stored in the cloud can be accessed from another location securely supporting broad security coverage and real-time threat detection across different sites.
- **Real-Time Data Processing and Analysis:** Cloud services enable fast processing and analysis of data across larger deployments. Cloud-based computing power enables the Smart Eye system to conduct multiple video feeds; meanwhile, real-time behavioral analysis is achieved with no burden on local processing units. The cloud enables the system to conduct more intensive data analyses, hence further improving on its accuracy in spotting possible threats.
- **Interoperability with IoT and Automation:** Cloud-based platforms also allow the system to be integrated with IoT, which is instrumental in automating responses. Smart Eye can initiate such response activities as locking gates, controlling access, or adjusting conditions of illumination as mandated through cloud-enabled IoT protocols. The system makes sure that IoT devices connect securely with the cloud, thus ensuring that the response to threats does not go too high.
- **More Security and Compliance:** AWS, Google Cloud, and Microsoft Azure have very high security cloud computing services with features such as encryption on data, controls of access, and compliance with myriad forms of regulatory standards, ensuring that this sensitive data is shielded within a very secure and compliant environment of videos as well. This therefore supports the organizational multimedia-based priorities especially those that handle sensitive information with the preservation of data and integrity

Chapter 3

IMPLEMENTATION

The multi-phased structured process of the implementation of the Smart Eye Security Surveillance System ensures that each component—motion detection, facial recognition, object detection, and alerting functions—sits together perfectly. The detailed breakdown of each phase, along with environmental setup, key system functionalities, and hardware requirements, of this system makes for a thorough robust and efficient security solution, explained in the subsection that follows.

3.1 ENVIRONMENTAL SETUP

The first step into implementing the Smart Eye Security Surveillance System is the setting up of the development environment to support all the necessary programming tools, computer vision libraries, and camera modules.

•**Software Configuration:** All system programming is done in python. Core visual tasks, including object detection, facial recognition, and motion tracking are conducted using OpenCV. Because of the large set of pre-built functions, OpenCV is perfectly suited for applications that require real-time image processing and computer vision, which makes tracking eye movements, facial landmark detection, behavioral pattern analysis much easier. Python is selected mainly because it is one of the most efficient languages that will ensure real-time processing if multiple streams of video are handled.

• **ESP32 CAM Module Installation:** The ESP32 CAM module is a module essentially Wi-Fi-enabled microcontroller with camera capabilities integrated within it and, thus, is a central part of a system built for real-time monitoring characteristics in an operational live video feed setup. Installation entails setting it to take on video feed capture and relay them to a connected processing unit or laptop/crafted microcontroller. The ESP32 CAM was set up and connected to the processing unit with tests to make sure that it can indeed transmit video data reliably without connectivity issues for real-time surveillance. It was also tested with different lighting conditions, ensuring that the camera can obtain high-quality images taken during both day and night conditions, which would result in good visibility of critical facial features and behaviors regardless of environmental conditions.

• **Lighting and Connectivity Testing**: The camera is put through trials under varied lighting conditions to ensure that it can capture critical facial features in poor lights or nighttime settings. This adaptability is especially crucial for security use cases in less well-lit surroundings. Testing of connectivity with the processing unit ensures stable quality of data transmission, an important requirement for real-time monitoring and motion detection.

3.2 BASIC MOTION DETECTION

This prepares the environment. The step here would be to set up motion detection as the basis of intelligent surveillance. This ability decides where and at what time to focus, saves processing power, and enhances system efficiency.

•**Implementation Using Arduino IDE:** Libraries compatible with ESP32 CAM from the Arduino IDE are used to implement the motion-detect functionality. These libraries offer effective ways of detection as based on the variations in pixel values between successive frames of video within the camera's view. Tracking pixel variations gives the system a pre-telling sign of movement, which often is an indirect telling of likely activity.

• **URL-Based Real-Time Monitoring:** Once motion is detected, ESP32 CAM will start recording video and transmitting that live feed across a unique URL. Using that URL, one could check the live stream on another device, such as smartphone, laptop, or tablet, to remotely watch a person, place, or activity. One of the energy-saving aspects that can be enjoyed by applying motion detection is saving the camera idle until some movement is detected, thereby saving it some memory of the battery life and processing resources. This approach will also ensure that during events of relevance, the system is active, hence reducing data overload and, thus, the requirements on the storage.

• **Customizable Sensitivity**: The motion detection sensitivity can be adjusted based on the requirement of the environment. In places with significant traffic, one has to adjust the threshold so that it would not get activated too often whereas in restricted areas, one has to set it to trigger even at slight movements. Flexibility makes the Smart Eye system adaptable to the security requirements of different people and avoids generating unnecessary alarms or data collection in non-critical areas.

3.3 FACIAL RECOGNITION AND OBJECT DETECTION

Based on the motion detection capability, the recognition and tracking of face and objects is the phase where the Smart Eye system can detect and track persons or objects in its field of view.

• OpenCV skills for deep training of models The facial model is taught using the Numpy capabilities provided by OpenCV such that it can detect the specific facial features and the areas of a partially occluded face, especially the eyes, mouth, and the nose. This model is achieved through the use of an extremely large dataset of facial pictures in several directions and expressions to maximize the accuracy of the detection. This rigorous training enables the model to recognize a person even if masked, wearing hats, or any accessory that may conceal the face partially.

• Real-time Face Detection and Tracking : The trained model then proceeds to the processing unit, which tracks through each frame of the video feed for faces or objects. Real-time processing functions from OpenCV allow the

system to track several movements of different individuals at once and flag any behavior outside the norm. It is most valuable in high-security environments, where one needs to monitor every single individual's activity within the monitored area.

• Object recognition: Other than facial recognition, the system is also equipped to detect such specific objects that might carry out a threat. This includes, for instance, weapons, suspicious packages, or even such objects that have been associated with potential threats. This has ensured that the system recognizes not only persons but also identifies the presence of dangerous items for enhanced security scope.

3.4 ALERT SYSTEM

Thus, the alert system is a critical feature of the Smart Eye Security Surveillance System, with the ability to alert in real-time each of the selected persons when suspicious behavior or unauthorized individuals are detected.

- **Integration of APIs for alerts:** the alert system is synchronized with APIs like Twilio to send the messages by SMS or Email. Once the detection engine senses a possible attack, it sends an automatic alert detailing the suspicious activity sensed, the time, and location. The real-time alerts help security people or property owners to respond quickly and thus may prevent security breaches or other cases**.**
- **Customizable Alert Triggers:** The Smart Eye system is supplied with programmed alert triggers based upon predefined conditions such as, protracted detection of an unfamiliar face, erratic or suspicious behavior, or an attempt to access a restricted area. Alerts are customizable depending on the severity of the detected behavior; low priority can be made to alert local security staff while high priority alert can reach law enforcement or emergency services directly. This tiered response promotes a flexible response based on perceived threat**.**
- **Log Real-Time Alert Log**: Every alert that results from the system being triggered is logged. Thus, there exists a record of all of the events and the notifications that have been detected. Access to this log may also be gained for later analysis or to identify patterns and suspicious activities for utilization in security assessments as well as proactive management of threats. A timestamped history of responses is also maintained through the alert log that could be useful for auditing purposes as well as for assessing the system's effectiveness**.**

3.5 COMPONENTS REQUIRED

ESP32 CAM

- **Working**: This is a powerful microcontroller with an integrated camera module, meaning it can capture video as well as images. It has an added inbuilt Wi-Fi and Bluetooth capability to make data transfer wireless. It captures live feed video from the ESP32 CAM and processes it in real-time before transmitting it for image-based facial recognition or tracking of eye movement in this project.
- **Need**: The ESP32 CAM is essential for real-time image capture and wireless data transfer, which forms the basis of the base support system to work properly. The module is compact, affordable, and has processing capabilities that make it highly suitable for IoT applications-its ability in performing efficient monitoring and processing of data. The Smart Eye is mainly built around this module, making it a mechanism of autonomous real-time surveillance.

FIG.1

RELAY MODULE

- **Working:** A relay module works as a power-operated switch. This way, it enables the ESP32 to control the high-power components, which may include a solenoid lock and other alarms. It gives the low power ESP32 CAM a possibility of activating high-power devices through the small electric signal produced that may open or close the relay circuit.
- **Need:** The relay module is thus an interfacing module between low-power control ESP32 CAM and high-power components. Its implementation thus allows the ESP32 to control the solenoid lock or any other high-powered component safely without exposing sensitive circuitry to the full voltage. It therefore integrates functionality and protection features, paving the way for secure automated responses within the surveillance system.

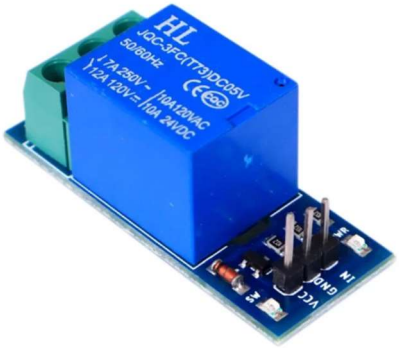


FIG.2

SOLENIOD LOCK

- **Working:** A solenoid lock is an electromechanical locking device. When energized, it, through its magnetic field, draws a metal rod inward which could be used in locking or unlocking the mechanism. It is locked in a default state unless the relay module sends power to unlock or deny access.
- **Need:** The solenoid lock provides a physical security layer because the system alerts its response, such as locking doors once unauthorized access is made. This type of lock connects to the system to grant distant central control over all access points so that areas are appropriately locked or unlocked based on analysis by the surveillance system.

FIG.3

LEDS

- **Working:** LEDs (Light Emitting Diodes) provide visual output in the form of the lighting up in various colors that represent the status of the surveillance system. It can further be mounted on the GPIO (General Purpose Input/Output) pins of ESP32 and are only powered up during system-related events like successful face detection and alerts turned on.
- **Need:** LEDs provide immediate visual feedback about the system status, such as when an intruder has been detected or if the system is armed or disarmed. This is very helpful for instant diagnostic purposes and to gain an understanding of the system operation even without a screen or interface. LEDs act as a simple interface for showing system states.

FIG.4

BREADBOARD

- **Working:** A breadboard is a plastic base with rows of holes that connect electronically, so electronic components can quickly and easily be placed there using jumper wires and assembled without the need for soldering, thus making it easy to rapidly prototype circuits.
- **Need:**
  The breadboard is an integral component of the developing and testing stages o f the Smart Eye Security Surveillance System. The use of a breadboard provides for easy changeability of circuits, modifications of new connections, and verification of functionality without permanent connections. This flexibility easily allows for troubleshooting and improvement by iterations, especially when testing a variety of hardware configurations.

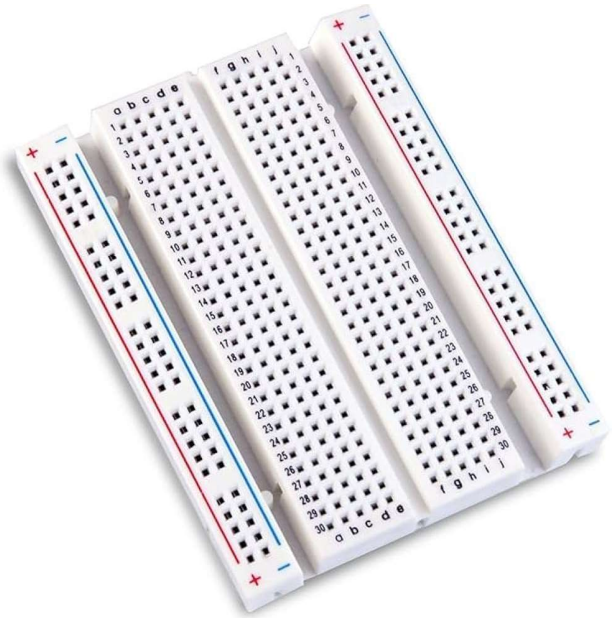


FIG.5

12V POWER SUPPLY

- **Working:** The 12V power supply converts AC power from a wall outlet into a stable 12V DC output suitable for powering high-demand components like the relay module and solenoid lock. It ensures a consistent voltage for continuous operation.
- **Need:** A stable 12V power supply is required for the system's high-power components, such as the solenoid lock and relay module, which need more power than can be provided by typical microcontroller power sources. This power supply ensures that the entire system can operate smoothly and continuously, which is crucial for 24/7 surveillance applications.

FIG.6

CAPACITORS

- **Working:** Capacitors store electrical energy and release it when needed, helping to stabilize the voltage to filter out noise and smooth power fluctuations, ensuring steady current delivery to sensitive components.
- **Need:** In the Smart Eye system, capacitors play a vital role in stabilizing the power supply. They prevent sudden drops or spikes in voltage, which can cause malfunctions in sensitive components like the ESP32 CAM or the relay module. By providing smooth power delivery, capacitors protect the system from electrical noise and ensure reliable operation, particularly when high-power components like the solenoid lock are in use.

FIG.7

DC-DC CONVERTER LM2596

- **Working:** The LM2596 is a step-down (buck) voltage regulator that converts higher DC voltages to a stable, lower DC voltage. It utilizes Pulse Width Modulation (PWM) to regulate and efficiently convert the input voltage (typically from a higher source like a 12V battery or power supply) into a stable output voltage, typically ranging from 1.25V to 37V. The LM2596 is equipped with a switching transistor that controls the flow of current to maintain the desired output voltage, making it highly efficient compared to linear regulators, which dissipate excess energy as heat.
- **Need:** The LM2596 DC-DC converter is essential for power management in electronic projects and systems that require a stable, regulated voltage supply. It is particularly useful in cases where you need to power components like microcontrollers, sensors, or other devices that operate at different voltage levels than the available power supply. The LM2596 is widely used in embedded systems, battery-powered projects, and power-sensitive applications

due to its high efficiency, compact size, and adjustable output voltage. It simplifies power regulation by ensuring that the required voltage is supplied with minimal energy loss, and it is crucial for maintaining the longevity of components by preventing over-voltage situations.

FIG.8

7805 VOLTAGE REGULATOR

- **Working**: The 7805 voltage regulator is a linear voltage regulator that converts a higher input voltage, such as 12V, into a stable 5V output. It regulates the voltage by dissipating excess power as heat, ensuring a steady 5V output as long as the input voltage remains above 7V.
- **Need:** In the Smart Eye system, the 7805 regulator steps down the 12V power supply to a safe 5V level, which is essential for powering the ESP32 CAM and other low-power components. It ensures these components receive a stable voltage, preventing potential damage from overvoltage. The regulator also provides reliable power to components that require 5V, supporting smooth operation across the system.

FIG.9

NodeMCU ESP8266 Board

- **Working:** The NodeMCU ESP8266 is a low-cost Wi-Fi microcontroller board based on the ESP8266 chip, featuring integrated Wi-Fi capabilities. It allows communication between a microcontroller and a Wi-Fi network, enabling wireless data transmission and remote control of devices. The NodeMCU board comes with a built-in USB-to-serial interface, which allows for easy programming via USB and makes it compatible with the Arduino IDE and other development platforms. It includes the necessary components, such as GPIO pins, ADC, and UART interfaces, to interface with various sensors and actuators for IoT applications.
- **Need:** The NodeMCU ESP8266 is essential for projects that require Wi-Fi connectivity in embedded systems. It is a popular choice for IoT (Internet of Things) applications due to its compact size, low power consumption, and ability to easily interface with a wide range of sensors and devices. This board is particularly valuable for developers working on Wi-Fi-based communication projects, such as home automation, remote monitoring, and real-time data collection. It simplifies the process of adding wireless communication to embedded systems by providing an easy-to-use platform with built-in connectivity, making it ideal for prototyping and deployment in IoT applications.

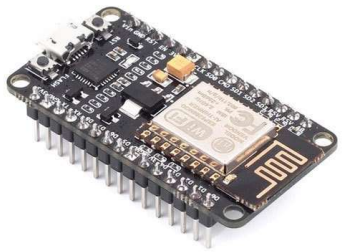
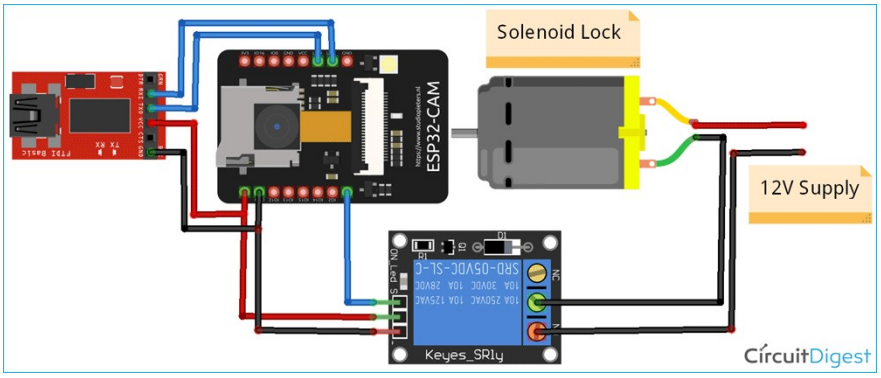


FIG.10

Circuit Diagram:



ESP32-CAM Based Access Control System: Circuit Layout for Connecting Solenoid Lock, 12V Power Supply, and Relay Module.

Hardware:

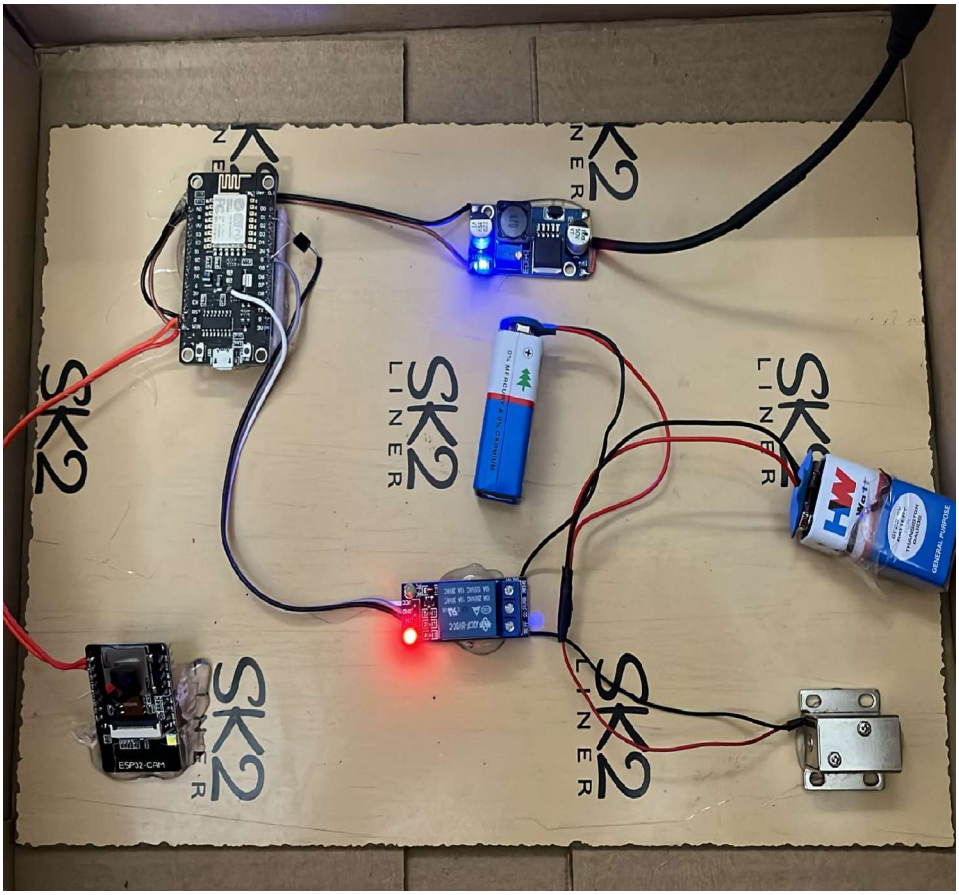


FIG.11.
Prototype Setup for IoT-Based Locking System: Integrating ESP8266 NodeMCU, ESP32-CAM, Relay Module, Solenoid Lock, and DC-DC Converter with Battery Power Supply.

Chapter – 4

EXPERIMENTAL SETUP & RESULT

The system was designed to increase the accuracy of detection and response times while being reliable in scenarios where traditional security mechanisms fail. The shortcomings of conventional security technology were complemented by high-resolution infrared cameras, computer vision algorithms, and IoT-based automated responses. In addition, in the text below, the set-up, performance metrics, and results of the system are delineated in detail, which demonstrate the system's potential for the advanced surveillance solution.

4.1 SETUP AND TESTING ENVIRONMENT

In order to test its performance, the Smart Eye system was challenged to various lighting conditions for it to be representative of real life usage. Both well-lit and poorly lit areas were taken into account so that it will function powerfully in all security sectors :residential and commercial and public environments which may have a number of levels depending on a certain time schedule for the day. And for 24/7 surveillance to be supported, high-resolution ESP cameras were used towards achieving more explicit video feeds during nocturnal conditions. Some benefits include recorded picture on these cameras, such as a nighttime view with face detection and eye detection even when in low ambient light.

ESP32 and NodeMCU module were put together to serve as the processing units in real time video analysis. In IoT applications, it is highly efficient and achieves high-speed video feed processing through cloud-based servers, especially when their input was deemed necessary. It acted as an intermediary processing unit that augmented capability in real-time data analysis so as to have efficient immediate responses. Together, these components built a scalable, cost-effective architecture that could support high processing loads and flexibility toward expansions where required by various security needs.

4.2 PERFORMANCE MATRIX

An evaluation was conducted on the Smart Eye system using a set of performance metrics that were deemed critical to the effectiveness of any surveillance system. Such metrics included detection accuracy, response time, and the false alarm rate-all of which would reflect something to be said about the usability and reliability of the system.

1. Detection Accuracy

The first key point for any surveillance system would be the accuracy in threat detection. Since the Smart Eye relies on eye and facial feature recognition as a means of assessing behaviors and identifying potential threats, accurate results will be of

paramount importance. The system could accurately detect actual threats versus benign behavior during even conditions where parts of the face are obscured by wearing hats and glasses or facial coverings. In fact, the detection accuracy was impressively above 90%, suggesting that this system has the capability of distinguishing distinct facial and eye features and subtle behavioral cues that well signal suspicious behavior.

2. Response Time

This time-taken metric captures the period between the detection of suspicious activity and the initiation of automated response. In high-risk security environments like those involving banks and airports, among others, the prevention of potential incidents becomes incumbent upon a lightening response. With the Smart Eye system that involved IoT integration and cloud-based alerting systems, response times could be significantly improved. Once the threat was identified, it would cause the system to take automatic responses, like a loud alarm, notification of security personnel, and quick locking of doors. This speedy response ensures that security breaches are dealt with promptly, minimizing the window for any potential harm to be inflicted.

3. False Alarm Rate

It is a fact that conventional surveillance systems, which operate on the basis of a result from the motion sensor, cannot distinguish between innocuous activity and a genuine threat and hence provide false alarms very often. Due to these false alerts, the dependability of a security system degrades and wastes resources. In its focus, Smart Eye aims to reduce false alarms through eye movement and facial detection. Behavioral analysis allows the system to make better distinctions between routine activity and suspicious behavior. False alarm rate was much lower than with regular systems, mainly because the system could identify specific patterns of behavior and also differentiate between normal and nonnormal activities.

4.3 RESULT AND OBSERVATIONS

With the testing phase, very significant differences have been noticed compared with what is traditionally known about security systems in the realms of detection accuracy. The Smart Eye system can successfully identify and interpret facial features under occlusion scenarios such as wearing masks and partially obscured faces, with an average detection accuracy that exceeds 90%. This is a key attribute for real-world applications where sometimes intentionally or unintentionally, people tend to obscure parts of their faces. Scenarios of this nature have historically been so difficult for standard surveillance systems that Smart Eye's performance qualifies as a major breakthrough.

Furthermore, faster response times mean that prompt corrective actions can be performed once a potential threat is identified. Moreover, cloud-based integration allows instant triggering of notifications and alarms, which will ensure intervention on time, a factor very precious in high-security areas. The reduced false alarm rate

further makes it more practical since it reduces the chances of undue interference whilst the users' confidence in the alerts of the system increases.

4.4 PRACTICAL IMPLICATIONS

The result of the testing phase highlighted that the Smart Eye system is indeed a potent as well as a viable high technology solution meant for a diverse set of applications. It can be an excellent option for residents in housing areas who can depend on it for uncompromised and reduced false alarms and round-the-clock monitoring. This system will make a better asset in allowing security personnel to be more proactive against the threats seen, as it gives a higher accuracy level and speed in their detection in public spaces or in commercial properties. This would attract systems that have a lower false alarm rate in environments where costs are very high or particular disruptions in ways such as hospitals and educational institutions.

In conclusion, the Smart Eye Security Surveillance System shall break new grounds in an automated surveillance affair with advanced camera technology combined with cloud integration with the potential to realize IoT capabilities. The system shall ensure very high accuracy in detection, short response time to an event, and a reduction of false alarm rates; thus, it shall be highly practical, reliable, and efficient for modern security needs.

4.5 ADAFRUIT AND ADAFRUIT INTEGRATION FOR ENHANCED SMART EYE SURVEILLENCE SYSTEM

Reliable data management, real-time communication, secure access, scalability, and analytics are integral parameters of modern surveillance systems. Thus, Adafruit offers an optimal framework combining a suite of cloud-based tools, supporting these functionalities. This detailed overview explains how the different Adafruit components can be included in the Smart Eye Surveillance System to enhance its performance and reliability.

1. Real-Time Data Synchronization

Continuous real-time observations and updates are among the key characteristics of surveillance systems. Such is an incredibly important area that Adafruit Realtime Database fills by synchronizing all connected devices to the system at the same instant. This form of synchronization on a real-time basis is essential as it helps in providing live updates to surveillance operators who can then use it to effectively coordinate. This can be applied as follows:

- **Real-time Data Sharing:** The Realtime Database updates information in milliseconds. The ability is crucial when metadata like a timestamp from video feed and sensor reads or triggered alarms need to be shared instantaneously across multiple control rooms or a group of mobile devices.

- **Distributed Monitoring**: A Smart Eye Surveillance System can be used in facilities where security teams are spread across several locations. Real-time updates on data will ensure that information is accessible to all so that they remain synchronised and coordinated in monitoring and responding to incidents.
- **Interactive Responses:** If an alert is accepted or an action is performed by one team member, this update can be made in real-time to other devices connected showing this alteration without a lapse.

Adafruit is the upgraded version of Realtime Database, offering some better querying capabilities and scalability. It can manage complex data structures and is even suitable for applications on a larger scale where high data filtering, sorting, and searching is required.

2. Secure Data Storage and Accessibility

A surveillance system needs to provide surety of data storage and its availability whenever needed. What Adafruit has to offer to meet this need is the following

- **User Profiles and Access Controls**: Adafruit provides the ability to store information about users exhaustively, including their profiles, roles, and levels of permissions. This leads to limiting access to sensitive parts of the surveillance system to authorized personnel alone. For instance, some administrators may be able to access every feature of the system while a standard user can only view some of the features of the system.
- **Event Logs and Data History:** The surveillance systems should keep detailed records of activity, like motion detection logs, user actions, times of access, and significant events. Having these logs stored in Adafruit databases would mean that system managers would have a robust audit trail for post-incident analysis and hence improve overall security.
- **Centralized Device Configurations:** Adafruit can maintain centrally the storage and updates of device configurations, such as camera angle and sensor sensitivity level. It then streamlines the management of multiple devices. Thus, it ensures that all units are maintained uniformly.

Adafruit makes sure all the data stored is available to authorized users through secure connections, regardless of whether they are in the field or operating remotely. For surveillance teams that might have to monitor situations from diverse locations using their mobile devices, this flexibility will be very important.

3. Real-Time Alerts and Notifications

The system needs to have timely notifications because the response from security teams has to be very timely to respond to potential threats. Adafruit Cloud Messaging would be good for this as it allows sending of notifications and messages between different platforms.

- **Instant Threat Alerts:** ACM can push notifications directly onto mobile devices, computer dashboards or both when a motion sensor or camera detects suspicious activity. In such a scenario, security personnel has timely interventions that could prevent escalations and the possible unfolding of a disaster event.
- **Unauthorized Access Notifications:** Whenever unauthorized access is attempted to the system and or a restricted area without valid credentials, then FCM can send real-time warnings to the administrators so that they may investigate any breach.
- **Maintenance and Status Alerts:** Video monitoring systems have an essential reliance on the continuance of day-to-day operation of cameras and sensors. FCM will notify users about system health issues such as disconnection, power failure or battery status updates for wireless devices so that problems can be diagnosed before they have any impact on monitoring capabilities.

The Adafruit service in this regard is an all-platform service, meaning it supports iOS and Android as well as web browsers, and so these notifications reach the relevant personnel of the devices in a timely manner.

4. Enhanced User Authentication

It is essential to ensure only authorized personnel gain access to a surveillance system. Adafruit Authentication makes it easier to integrate strong authentication with the management of user access. Here's what it offers:

- **One or more Authentication Methods**: Adafruit allows multiple authentication methods. This includes email/password, phone number, and social logins like Google and Facebook. This means one has several options that allow them to easily come up with authentication protocols to suit their security needs.
- **Security Best Practices:** Adafruit Authentication benefits from the security standards in place, including encryption, ensuring safe transfer of data. These ensure to offer a firm layer of security against any unauthorized access and help protect the surveillance system against data breaches or even malevolent activities that may expose confidential information.
- **Two-Factor Authentication (2FA)**: As a security measure, custom solutions can be made in 2FA to encourage further security by requiring the user for verification code or identity on another device.

5. Storage of Media Files

Well, surveillance systems generate a large amount of video footage and images which needs to be safely stored and made retrievable. That is what Adafruit Storage is for: providing cloud-based handling of this data.

- **Scalable Storage:** Adafruit Storage is designed to deal with large volumes of data video recording and image captures can generate. Surveillance systems

can scale storage requirements without affecting their performance, especially in large installations with many cameras.

- **Media Access and Playback:** Adafruit provides access to captured media for authorized personnel at any point in time, or the playback of footage on demand, which is an important function for reviewing incidents, following up on investigations, and in evidence-gathering.
- **Data Security and Redundancy:** Adafruit Storage benefits from the Google Cloud infrastructure to ensure protection of data in case of redundancy and other mechanisms to keep data secure like file-level access control and encryption. This allows video footage to be safely recovered in case of hardware failure.

6. Scalability and Insightful Analytics

A surveillance system must scale to ensure either adding more cameras or serving more users is done without loss of performance or reliability. Adafruit's underlying architecture allows for smooth growth.

- **Effortless scaling:** The ability to introduce an increase in the data volume or the number of devices or users increases can be scaled up without important changes to the infrastructure due to the fact that Adafruit can handle increased data and devices or users. This is one of the critical factors of surveillance systems when they cover multiple sites or create broader coverage over time.
- **Usage and Performance Analytics:** Adafruit Analytics provides very valuable insights into how users interact with the surveillance system, where behavior such as login frequency, the time taken to respond to an alert, and hours of peak activity could be further optimized to make improvements to system performance and better user experience.
- **Pattern Recognition and Optimization**: From the trends in data, administrators would trace abnormal patterns or efficiencies. For instance, false alarms often go off frequently in certain areas or take too long to respond. That is something that can then be used to tweak device settings or refine mechanisms within the system to improve performance.

7. Comprehensive Device Management

Not lessthan the software itself is the device management, which relates to a surveilla nce system run. Adafruit has various tools that simplify the management and monitoring of devices:

- **Centralized Configuration Management:** With the help of Adafruit, configurations on surveillance cameras and sensors, which may include operational parameters, can be sent over remotely. For saving precious time, it also ensures consistency over all the devices.
- **Real-time Device Health Monitoring:** Adafruit can send information across a connection and check in real-time whether the connected devices are okay or not, such as data connectivity and power levels. In the case of an issue arising,

the system can automatically notify accountable teams to ensure that no surveillance coverage goes down.

- **Scalability in IoT:** For large surveillance networks that rely on hundreds and thousands of IoT-connected devices, Adafruit can cope with a good number of devices, making sure that they work as one single, networked group being monitored.

Conclusion

Integrating Adafruit into a Smart Eye Surveillance System enhances its capabilities by offering real-time data synchronization, secure data storage, and user authentication, as well as the storage of extensive media files and scalable infrastructure. Its cloud-native architecture supports seamless expansion, and Adafruit Analytics offers valuable insights to improve system performance and user interaction. Real-time notifications, multi-platform support, and device management features further streamline the system's operation and maintenance.

With Adafruit as part of the surveillance ecosystem, organizations can achieve a responsive, secure, and highly scalable system capable of adapting to changing needs and growing user bases. This integration results in enhanced security, data integrity, and rapid response capabilities, essential for effective modern surveillance.

4.6 COMPREHENSIVE UTILIZATION OF ADAFRUIT SERVICE IN SMART EYE SURVEILLENCE SYSTEM

Introduction of Adafruit to a Smart Eye Surveillance System creates numerous cloud-based functionalities that optimize performance, ensure increased security, and provide improved interaction with the end-user. Let's get down to each core service that represents Adafruit and understand how they contribute to a comprehensive surveillance solution.

1. **Adafruit Realtime Database: Smooth Data Flow and Easy Management**

Real-time Data Flow Adafruit Realtime Database would be the ideal choice for any such environment where real-time updates are concerned. For surveillance purposes, for instance, real-time alerts with regard to motion detected, unauthorized access, and other unusual activities become inevitable. The Realtime Database ensures live data synchronization across all the connected clients so that control centers, mobile apps, and monitoring devices can fetch all updated information in real time. This smooth flow of data allows security teams to respond promptly to changing situations.

Optimized Data Use. For a Realtime Database, which can send only changes in the data instead of a full refresh, it presents reduced data throughput. This is useful, especially in the usage aspect of mobile networks that might be used in order to stream data or where cost for transference of data is a factor of concern. Surveillance teams enjoy efficient data usage with real-time awareness.

This is **hierarchical data structure:** By the use of JSON-based format, data can be held inside the database within a logical hierarchy. Thus, for example, each camera node would include sub-nodes which outline metadata related to live video feed, alert statuses, and device settings. This structured way of implementation simplifies querying of data, so one can easily get the needed information without requiring complex calls towards a database.

2. Adafruit for Enhanced Data Management

Advanced Querying and Filtering: Firestore allows for more thorough queries compared to the Realtime Database. This is especially helpful in large event logs or when working with video metadata. Surveillance administrators can filter data based on certain attributes such as time stamps, event type, or camera IDs, which will make it much easier to work through a specific event.

Scalability with document-based structure: Firestore's model based on document and collection automatically enhances scalability. That is, the growing number of surveillance devices or users will not degrade performance because Firestore supports larger data development. This becomes important for large deployments of surveillance in numerous buildings or for city-wide installations.

3. Adafruit Authentication for Secure Access

Access Control: Protecting a surveillance system from unauthorized access is a heavy security requirement and Adafruit Authentication offers strong tools that allow you to ensure that only the proper people can gain access to the system, including, for example, multi-factor authentication (MFA). This would essentially mean you could strengthen security so a user, for instance, has to provide his password as well as some other secondary method like a code sent to a mobile device before allowing access to the system.

Simplified User Management Adafruit Authentication: simplifies the creation and management of user accounts and automatically supports other Adafruit services for proper access controls. It is possible to define various roles that will provide different access levels in the system: full system access to administrators and read-only access to ordinary users.

Multiple Authentication Methods: Adafruit supports all authentication methods. It leaves flexibility based on what would work best for the user and what is in line with the organization's policies. This is highly useful in systems involving multiple stakeholders, such as a company dealing with several different departments or those working with external security firms.

4. Adafruit Cloud Messaging (ACM) for Timely Alerts

Instant Alerts: Using FCM messages can be sent in an instant should there be particular events such as illegal access attempts, motion detection, or equipment

failure. These alerts may be received through mobile devices or web-based dashboards and let users know about the conditions at any given time.

Customizable Notifications: You can customize the alerts as per your choice, depending upon the priority of the event. For example, the high-priority alerts will have some unique ringing tone or vibration, which will differentiate the regular alerts. It facilitates the operators to give proper attention in critical situations.

Cross-platform compatibility: ACM supports platforms on android, iOS, and web devices ensuring that the notifications reach users regardless of their type of device. That makes it easier for the security team to connect and respond in due time.

5. Adafruit Storage for Media File Management

Scalable Media Storage: The amount of video and image data, thus requiring to be stored securely and retrieved efficiently, which surveillance systems produce. Adafruit Storage can handle these data needs by offering a cloud-based repository that scales with increasing volumes of data.

Efficient File Retrieval Storage: entries can be associated with metadata from the Realtime Database or Firestore. This enables the user to rapidly access particular footage related to specific events or timestamps, making it much easier to investigate incidents without manually going through hours of video footage.

Cost-Effective and Secure: Adafruit Storage utilizes the infrastructure of Google Cloud, which ensures data security and redundancy with encryption. Such measures will reduce the risk of data loss or loss of footage while keeping it accessible for as long as it is needed. The pay-as-you-go model helps to manage costs well in systems with volatile storage needs.

6. Adafruit Cloud Functions for Automation and Event Handling

Event-Driven Logic: Instead of simply running and producing output, Cloud Functions can be set up to trigger certain actions when specific events are logged in the Realtime Database or Firestore. For instance, in the event that an alert is logged, a Cloud Function might forward notifications to the proper security personnel; could also request additional data analysis, perhaps even recording a snapshot of the camera feed for later examination.

Automated Image Processing: It is possible to instruct Cloud Functions to automatically perform tasks such as generating video thumbnails, the application of motion detection algorithms, or tagging of footage using certain data, for example time and location. This in turn simplifies operations and reduces manual workloads.

Eliminates the need for dedicated backend servers: Cloud Functions does not require dedicated backend servers. It saves infrastructure costs and reduces the hassle of managing system scale and maintenance activities.

7. Adafruit Analytics for Insightful Monitoring

User Behaviour Tracking: Adafruit analytics will be a great tool in understanding how end-users interact with a surveillance system. It can tell the ones to make adjustment in the interface as well as configuration of the system, based on metrics like what camera the users view the most, at which times do they log in most frequently, or which feature is the most accessed.

Alert Response Time Analysis: There is a measure of how long it takes for users to respond to alerts. Thus, it would be able to show the effectiveness of your alert system. This data can assist in finding out areas of probable training or even system upgrading to enhance the response time.

Operational Performance Metrics: Analytics monitors KPIs such as system uptime, data transfer speeds, and average alert response times. This ensures that the surveillance system works well and meets all the security requirements.

8. Adafruit Remote Config for Dynamic System Management

Real-Time System Updates: The remote config allows the administrators to make changes to the system without the need to drop a new application version. It becomes highly essential to update camera sensitivity levels, to adjust the alert thresholds or changed user permissions among other changes.

Optimization through A/B Testing Remote Config: supports A/B testing for optimizing user experience and effectiveness of the system. For example, you can experiment using a different alert sound to determine which sound will result in faster responses from security staff.

Adaptive Configuration Based on Environment: The Remote Config feature dynamically adjusts various system settings from time to time based on the situation- an apparatus can be configured differently depending on the location of the user at a given time, role of the user during such times, and so on. Cameras can be configured at night to be extremely sensitive in risk-prone locations.

4.7 Key Advantages of Integrating Adafruit into a Smart Eye Surveillance System

- **Real-time Responsiveness:** The system can provide immediate updates and alerts with Adafruit Realtime Database and FCM, thus making possible the quick response to such safety risks.
- **Scalability:** In general, the Adafruit services are known to be auto-scaling; that is a very valuable feature for a continuously growing surveillance network. Whether that is extending more cameras or the number of users on board, Adafruit still allows growth without change to infrastructure.
- **Cost Efficiency:** Pay-as-you-go only costs one for the time he consumed the service, which will be much cost-efficient than server-based solutions, which appear to require endless hardware and maintenance expenses.
- **Better Security:** Adafruit features complete methods of authentication and robust encryption measures to secure sensitive surveillance data. Additional layers of security include multi-factor authentication and role-based access control of data.
- **Cross Platform Compatibility:** Adafruit compatibility with the web, iOS, and Android platforms enables users to access and manage the system using several devices, which makes it multidimensional and user-friendly.

Conclusion

By leveraging Adafruit, a Smart Eye Surveillance System can achieve real-time data management, robust user authentication, scalable storage, intelligent alerts, and seamless integration across platforms. The combined benefits of enhanced security, efficient data handling, and cost-effective scaling make Adafruit an excellent choice for building cloud-enabled, highly responsive surveillance solutions.

Chapter – 5

CONCLUSION

The Smart Eye Security Surveillance System is, therefore, an efficient and powerful application to offer solutions to the limitations inherent in the basic surveillance technologies, which mostly rely on basic motion detection and sometimes need significant human intervention. It embraces computer vision and Internet of Things (IoT) automation to its advantage in offering real-time threat detection and response with minimal oversight. A standalone solution, the system relies on high-resolution cameras that have infrared capabilities to provide a smooth operation even in lit or low illumination setups. It focuses on the eyes and facial features and gives the system the ability to make real-time analyses of behaviors, more accurately capturing any potential suspicious actions with a minimal number of false alarms.

The system has an automatic response mechanism. At the instant when the system detects behavior aligned to known patterns of threats, it executes a list of actions immediately. Among these actions can be alerts to security personnel by way of SMS and emails, switching on alarms and locking doors—always activating the right kind of responses in place at the right time. What makes it aptly fit within the requirements of high-risk environments such as airports, banks, and public places is that the ability of the Smart Eye system can identify benign activities from true threats.

Future directions would further elevate the capability of the system. Included recognition metrics, such as body language and gait analysis, would improve its accuracy in identifying unusual or suspicious behavior. These metrics can be extended to a much wider spectrum of behavioral analysis so that the system can capture more complex kinds of threat indicators. Moreover, offering more diverse kinds of demographic profiles will ensure that the system does not acquire biases, thus guaranteeing reliable performance across various populations. The approach to diversity in data is a very critical element that can give the system a level of inclusiveness and equity especially as it delivers itself within multicultural settings.

Overall, the smart eye security surveillance system has demonstrated scalability and flexibility, promising it to secure a wide range of security applications from private homes to large public places. Its advanced automation, combined with the accuracy of computer vision, can redefine standards for security surveillance in terms of proactivity in safety and set a new benchmark for future surveillance systems. This lineup of robust functionality, adaptability, and real-time responsiveness places the Smart Eye system as a next-generation solution in the automated security field.

PROCESSING

Code(for connecting to Adafruit)

```
from Adafruit_IO import Client, Data
from Adafruit_IO import Client, Feed
ADAFRUIT_IO_USERNAME = "pranav_2002"
ADAFRUIT_IO_KEY      = "aio_FpkM81ra9ft1cEl8wVqcTGZaR89y"
aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)
# Check if feed exists or create it dynamically
data_feed = aio.create_feed(Feed(name="u1"))
aio.send_data()
print("Data sent successfully")
```

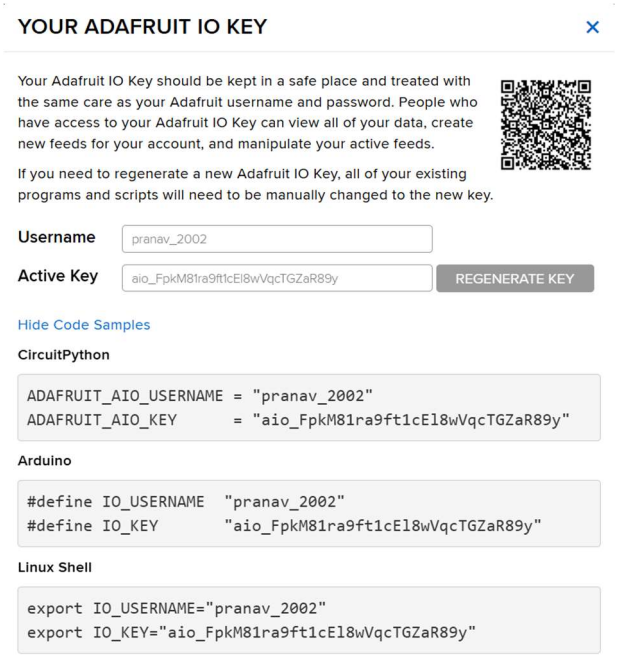


FIG.12 The username and active key of Adafruit cloud platform can be generated from the Adafruit account itself, and then will be mentioned in the code.

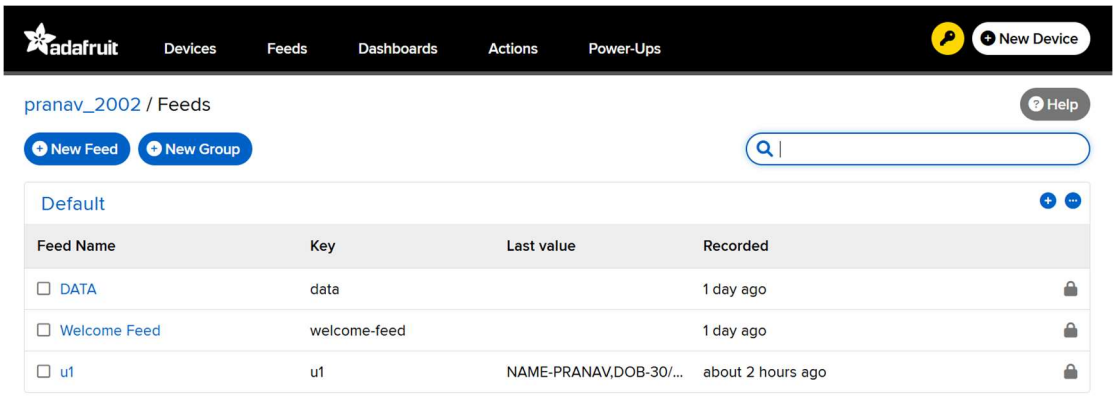


FIG.13

In Fig.13 and above code we will create a feed of directory in Adafruit in this case named u1 and also check if the repository is already available or not

Arduino Genuino Code:

```
#def ESP32
#include <WiFi.h>
#include <ESP8266WiFi.h>
const char* ssid = "PRANAV";
const char* password = "123456789";
#define DEVICE_TYPE_ESP32_CAM
void setup() {
  Serial.begin(115200);
  delay(10);
  Serial.println("ESP32-CAM: Connecting to Wi-Fi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();
  Serial.println("ESP32-CAM: Wi-Fi connected.");
  Serial.print("ESP32-CAM IP address: ");
```

```
  Serial.println(WiFi.localIP());
}
void loop() {
  Serial.println("ESP32-CAM: Ready to capture image...");
  delay(5000);
}
void setup() {
  Serial.begin(115200);
  delay(10);
  Serial.println("NodeMCU: Connecting to Wi-Fi...");
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  Serial.println();
  Serial.println("NodeMCU: Wi-Fi connected.");
  Serial.print("NodeMCU IP address: ");
  Serial.println(WiFi.localIP());
}
void loop()
  Serial.println();
  delay(5000);
}
}
```

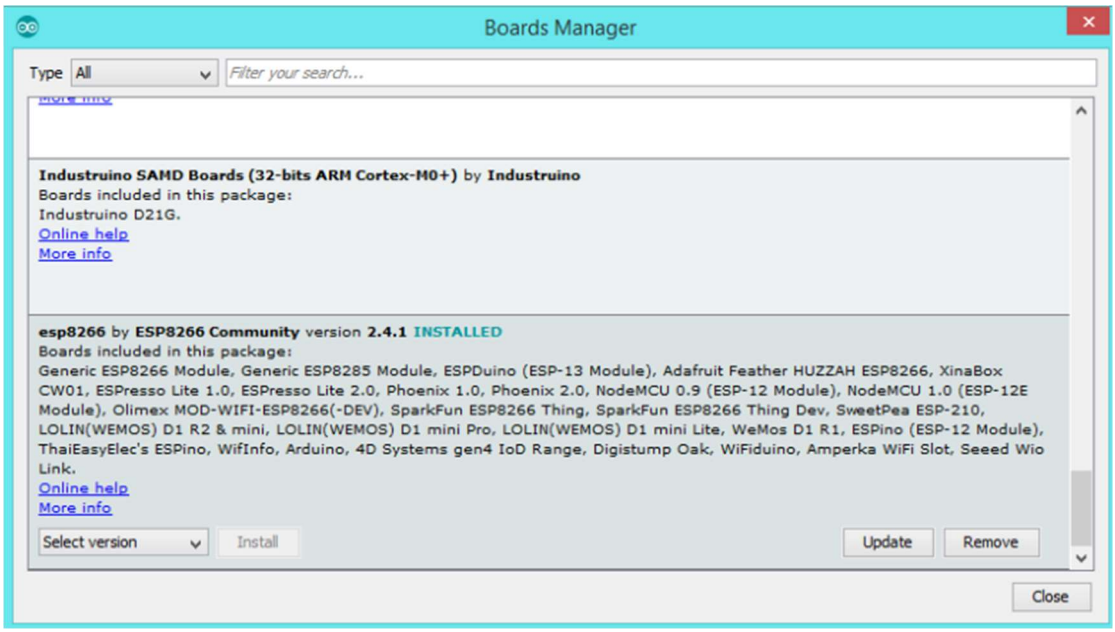


FIG.14

In above FIG.14 first we will have to install above esp8266 library from Boards Manager in Arduino genuine

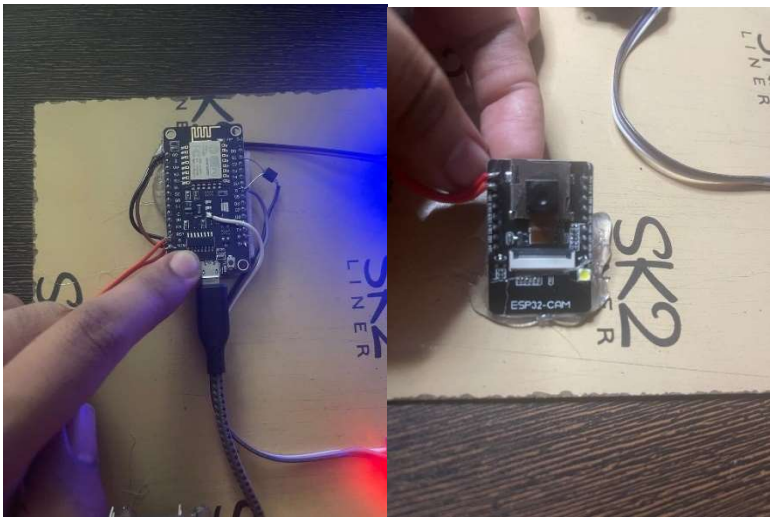


FIG.15

Before uploading the code and running we need to press restart button in NodeMCU and ESP32 CAM as being done in above figure

Code (for Face Detection from Database):

```
import os
import cv2
import numpy as np
import face_recognition
from datetime import datetime
import time
import pyrebase
from Adafruit_IO import Client, Feed
ADAFRUIT_AIO_USERNAME = "pranav_2002"
ADAFRUIT_AIO_KEY      = "aio_FpkM81ra9ft1cEl8wVqcTGZaR89y"
aio = Client(ADAFRUIT_AIO_USERNAME, ADAFRUIT_AIO_KEY)
config = {
  "apiKey": "AIzaSyCiy9CWDqwupKyPbFvXDA6OGhu9i2oDjrM",
  "authDomain": "bulkproject2.firebaseapp.com",
  "databaseURL": "https://bulkproject2-default-rtdb.firebaseio.com",
  "storageBucket": "bulkproject2.firebasestorage.app"
}
firebase = pyrebase.initialize_app(config)
path = 'D:\\Dataset'
images = []
classNames = []

myList = os.listdir(path)
print(myList)

for cl in myList:
    curImg = cv2.imread(f'{path}/{cl}')
    images.append(curImg)
    classNames.append(os.path.splitext(cl)[0])
```

```
print(classNames)
def findEncodings(images):
    encodeList = []
    for img in images:
        img = cv2.cvtColor(img,cv2.COLOR_BGR2RGB)
        encode = face_recognition.face_encodings(img)[0]
        encodeList.append(encode)
    return encodeList
def markAttendance(name):
    with open('//home//tezznova//Downloads//attendence.csv','r+') as f:
        myDataList = f.readlines()
#       print(myDataList)
        nameList = []
        for line in myDataList:
            entry = line.split(',')
            nameList.append(entry[0])
        if name not in nameList:
            now = datetime.now()
            dtstring = now.strftime('%H:%M:%S')
            f.writelines(f'\n{name},{dtstring}')
encodeListKnown = findEncodings(images)
print("encoding complete")
cam = cv2.VideoCapture(0)
while True:
    success, img = cam.read()
    imgS = cv2.resize(img,(0,0),None,0.25,0.25)
    imgS = cv2.cvtColor(imgS,cv2.COLOR_BGR2RGB)
    facelocCurframe = face_recognition.face_locations(imgS)
    encodeCurrentface = face_recognition.face_encodings(imgS,facelocCurframe)
```

```
    for encodeface, faceloc in zip(encodeCurrentface,facelocCurframe):
        matches = face_recognition.compare_faces(encodeListKnown,encodeface)
        faceDis = face_recognition.face_distance(encodeListKnown,encodeface)
        print(faceDis)
        matchIndex = np.argmin(faceDis)
        if matches[matchIndex]:
            name = classNames[matchIndex].upper()
            print(name)
            if(name=="PRANAV"):
                aio.send_data('u1',"NAME-PRANAV,DOB-30/07/2003,NUMBER-
9870321650")
                database = firebase.database()
                database.child("doorLock").update({"state":"1"})
                time.sleep(2)
                database.child("doorLock").update({"state":"0"})
            elif(name=="YUVRAJ"):
                aio.send_data('u1',"NAME-YUVRAJ,DOB-27/04/2003,NUMBER-
9835299254")
                database = firebase.database()
                database.child("doorLock").update({"state":"1"})
                time.sleep(2)
                database.child("doorLock").update({"state":"0"})
            elif(name=="SATVIK"):
                aio.send_data('u1',"NAME-SATVIK,DOB-30/10/2002,NUMBER-
9295949329")
                database = firebase.database()
                database.child("doorLock").update({"state":"1"})
                time.sleep(2)
                database.child("doorLock").update({"state":"0"})
```

```
        elif(name=="SIDDHARTH"):
            aio.send_data('u1',"NAME-SIDDHARTH,DOB-30/07/2003,NUMBER-
9870321650")
            database = firebase.database()
            database.child("doorLock").update({"state":"1"})
            time.sleep(2)
            database.child("doorLock").update({"state":"0"})
        # add more members to open the lock if needed
        y1,x2,y2,x1 = faceloc
        y1,x2,y2,x1 = y1*4,x2*4,y2*4,x1*4
        cv2.rectangle(img,(x1,y1),(x2,y2),(255,0,255),2)
        cv2.rectangle(img,(x1,y2-35),(x2,y2),(255,0,255),cv2.FILLED)
        cv2.putText(img,name,(x1+6,y2-
6),cv2.FONT_HERSHEY_COMPLEX,1,(255,255,255),2)
        #markAttendance(name)
    cv2.imshow('webcam',img)
    if cv2.waitKey(100) & 0xff == ord('q'):
        break
cam.release()
cv2.destroyAllWindows()
```

Output

```
rll$r□$□$n↑↑↑$↑l$↑b|$$□□□$r□b$↑b□$nn□lnn□b↑b□p$$b□lrlp$n0□↑↑$↑l↑$□↑↑↑b↑n$n$↑$↑□b$$nn'l$□l`□$[
Connected with IP: 172.20.10.2

Firebase Client v4.4.10

Token info: type = id token (GITKit token), status = on request
Token info: type = id token (GITKit token), status = ready
Lock State: 1
1
1
1
```

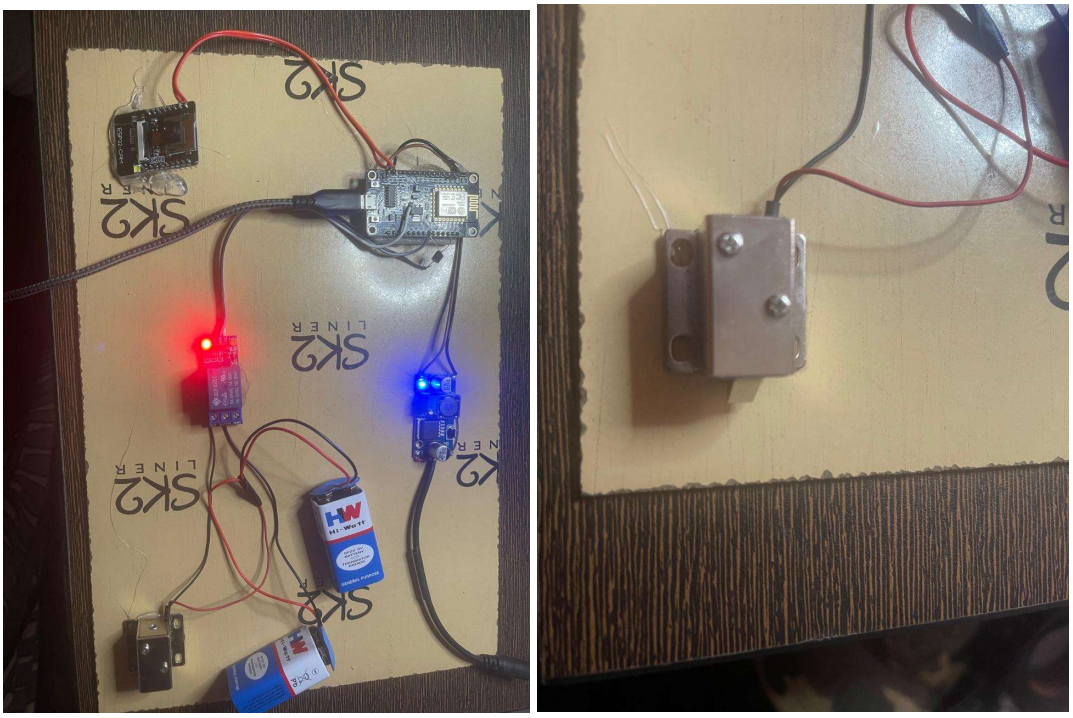


FIG.16

When no face is detected then the lock is in closed state and only red light is glowing as we can see in above FIG.16

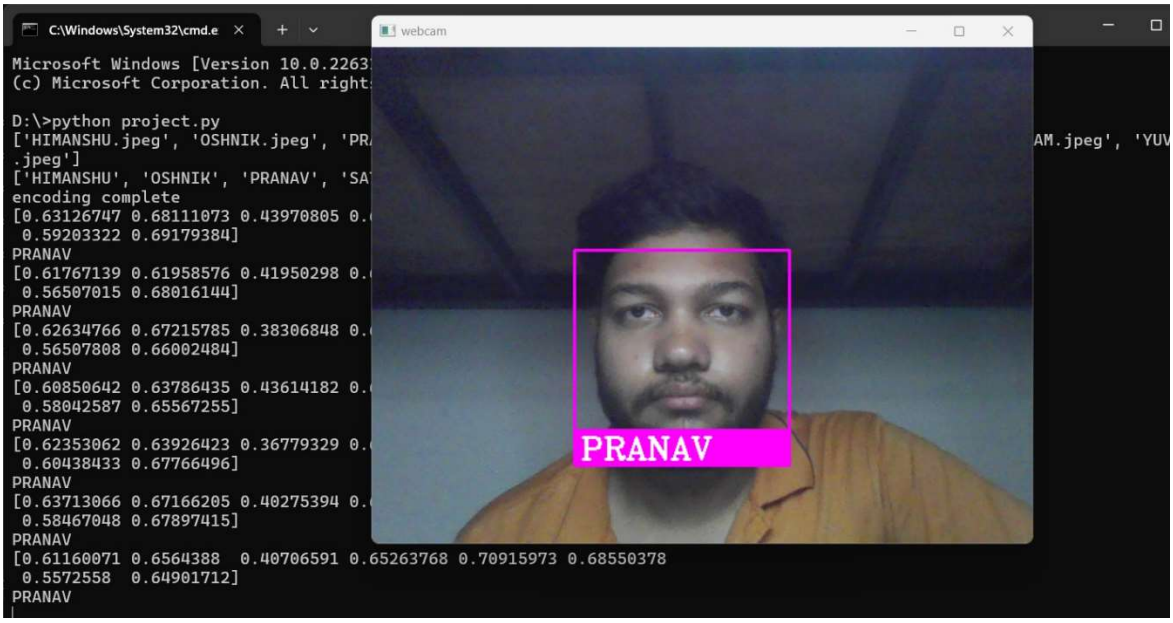


FIG.17

Now if we run the code and a face is detected which matches the face in dataset and is accepted by the code then the green light will also activate and lock will go in open position for a definite duration of time.

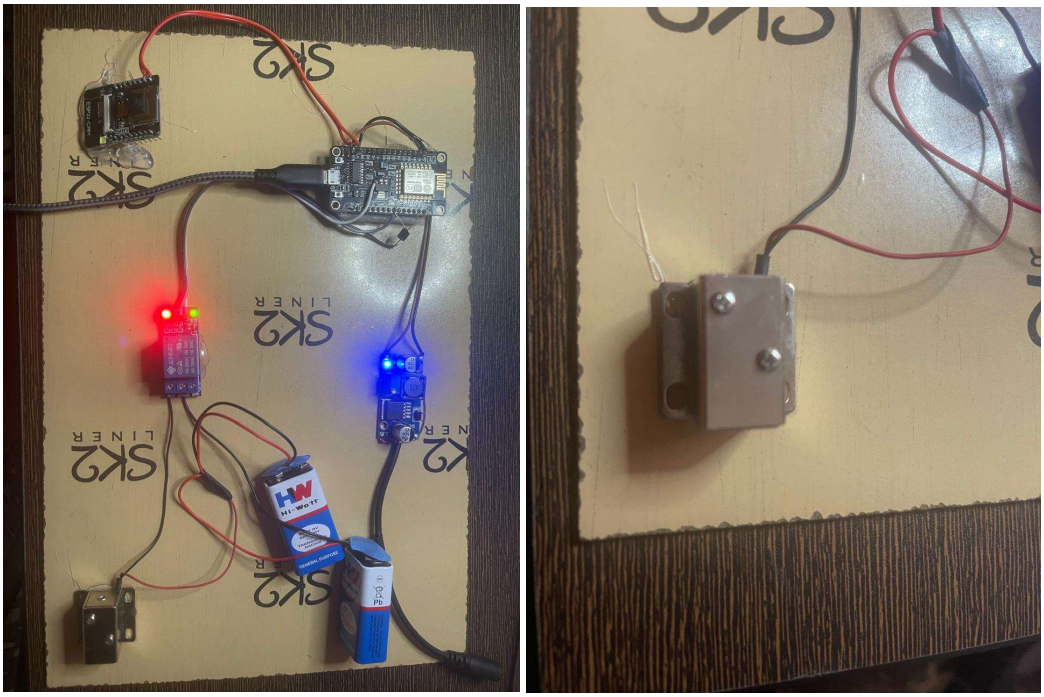


FIG.18

When face is detected then the lock is in closed state and green light is glowing as we can see in above FIG.18

All the data from above is stored simultaneously in the cloud as we can see in FIG.19

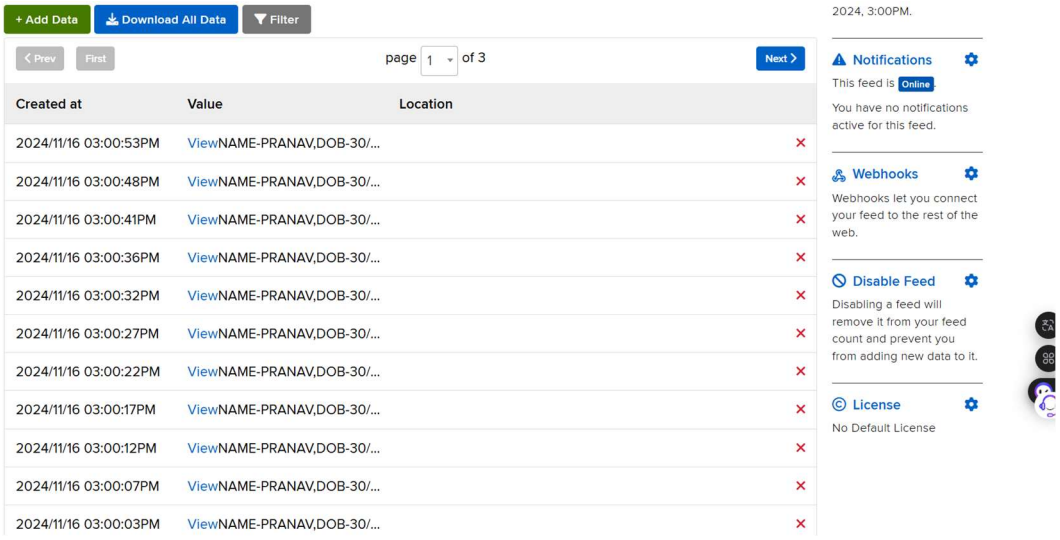


FIG.19

We can download the dataset in form of JSON or CSV file as mentioned in FIG.20

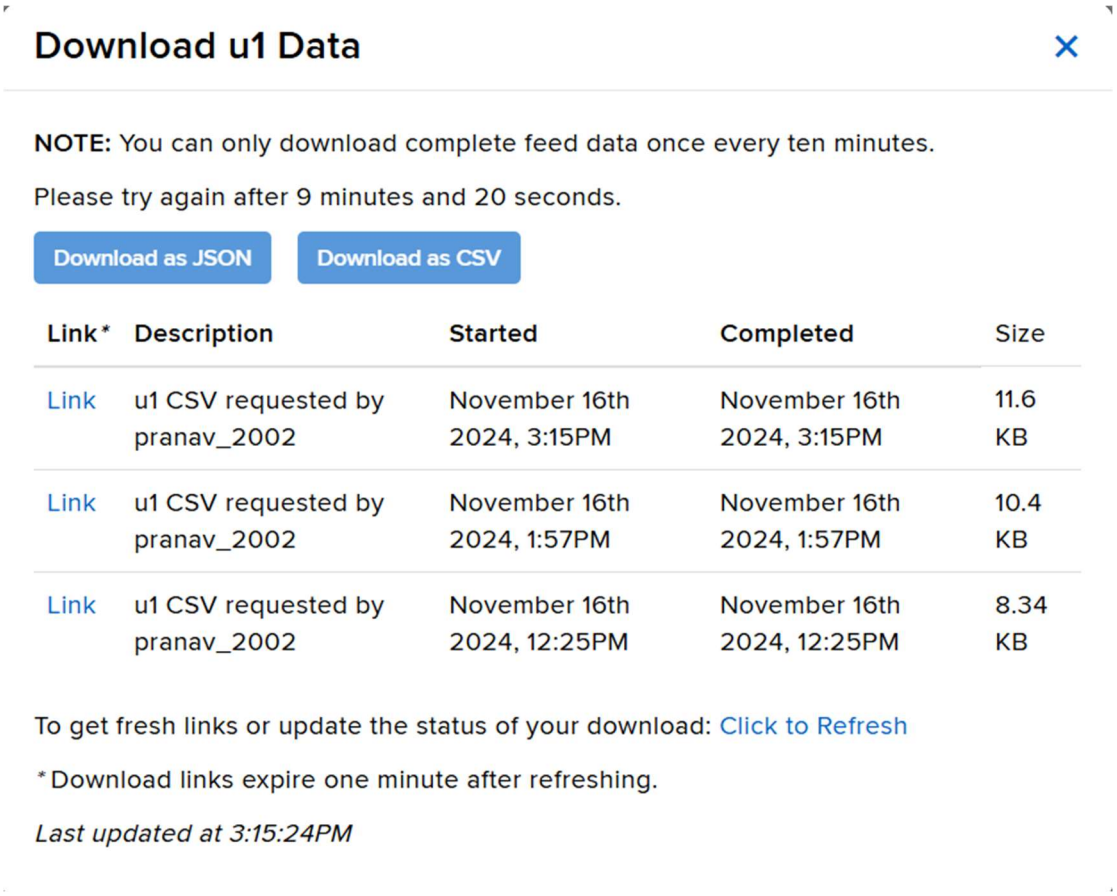
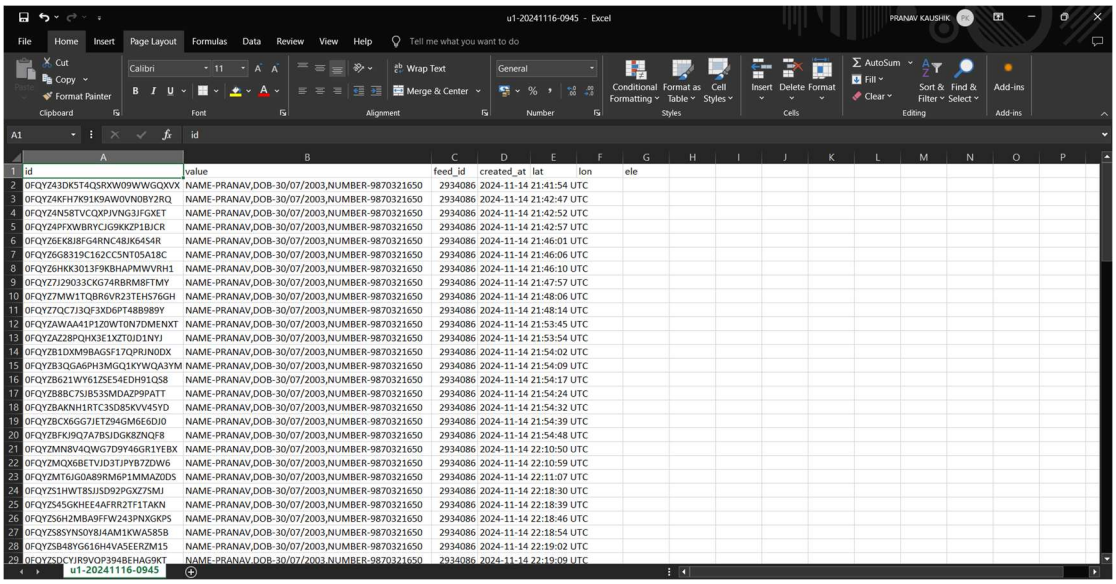


FIG.20



id	value	feed_id	created_at	lat	lon	ele
0FQYZ43DK5T4QSRXW09WWGQXVX	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:41:54 UTC			
0FQYZ4KFH7K91K9AW0VN0BY2RQ	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:42:47 UTC			
0FQYZ4N58TVCQXPJVNG3JFGXET	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:42:52 UTC			
0FQYZ4PFXWBRYCJG9KKZP1BJCR	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:42:57 UTC			
0FQYZ6EK8J8FG4RNC48JK64S4R	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:46:01 UTC			
0FQYZ6G8319C162CC5NT05A18C	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:46:06 UTC			
0FQYZ6HKK3013F9KBHAPMWVRH1	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:46:10 UTC			
0FQYZ7J29033CKG74RBRM8FTMY	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:47:57 UTC			
0FQYZ7MW1TQBR6VR23TEHS76GH	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:48:06 UTC			
0FQYZ7QC7J3QF3XD6PT48B9B9Y	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:48:14 UTC			
0FQYZAWAA41P1Z0WT0N7DMENXT	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:53:45 UTC			
0FQYZAZ28PQHX3E1XZT0JD1NYJ	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:53:54 UTC			
0FQYZB1DXM9BAGSF17QPRJN0DX	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:54:02 UTC			
0FQYZB3QGA6PH3MGQ1KYWQA3YM	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:54:09 UTC			
0FQYZB621WY61ZSE54EDH91QS8	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:54:17 UTC			
0FQYZB8BC7SJB53SMDAZP9PATT	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:54:24 UTC			
0FQYZBAKNH1RTC3SD85KVV45YD	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:54:32 UTC			
0FQYZBCX6GG7JETZ94GM6E6DJ0	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:54:39 UTC			
0FQYZBFKJ9Q7A7BSJDGK8ZNQF8	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 21:54:48 UTC			
0FQYZMN8V4QWG7D9Y46GR1YEBX	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 22:10:50 UTC			
0FQYZMQX6BETVJD3TJPYB7ZDW6	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 22:10:59 UTC			
0FQYZMT6JG0A89RM6P1MMAZ0DS	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 22:11:07 UTC			
0FQYZS1HWT8SJJSD92PGXZ7SMJ	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 22:18:30 UTC			
0FQYZS45GKHEE4AFRR2TF1TAKN	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 22:18:39 UTC			
0FQYZS6H2MBA9FFW243PNXGKPS	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 22:18:46 UTC			
0FQYZS8SYNS0Y8J4AM1KWA585B	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 22:18:54 UTC			
0FQYZSB48YG616H4VA5EERZM15	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 22:19:02 UTC			
0FQYZSDCYJR9VQP394BEHAG9KT	NAME-PRANAV,DOB-30/07/2003,NUMBER-9870321650	2934086	2024-11-14 22:19:09 UTC			

FIG.21

In FIG.20 we can see the excel sheet of the data collected by the cloud and can be used for further analysis.

Analysis of the Dataset in PowerBI:

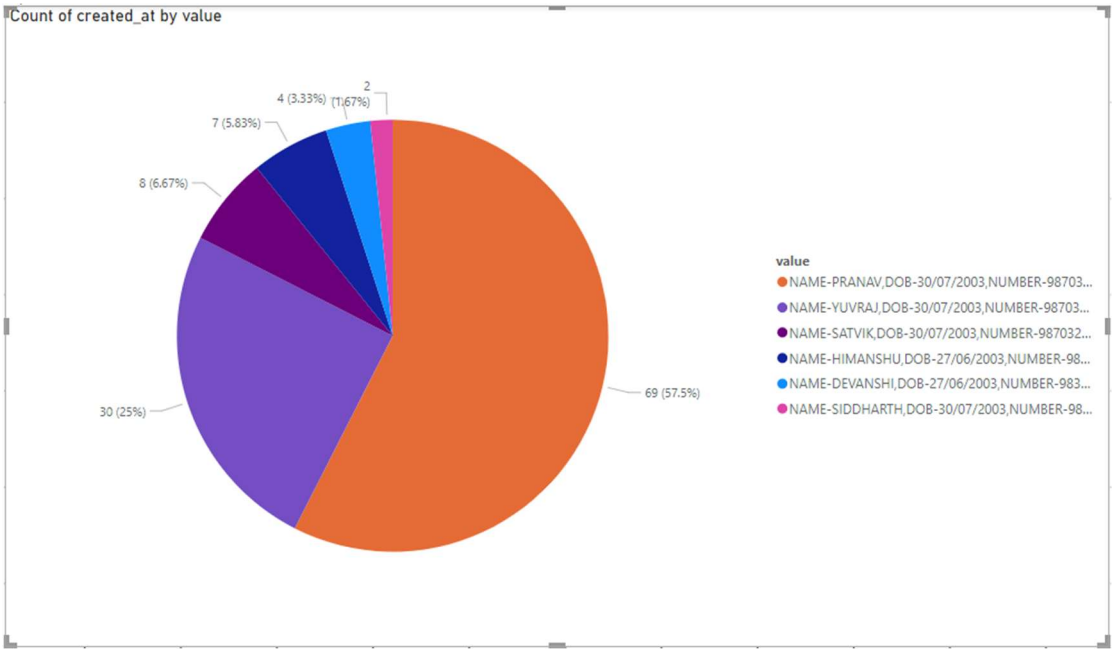


FIG.22 Pie Chart of the No. of members entry

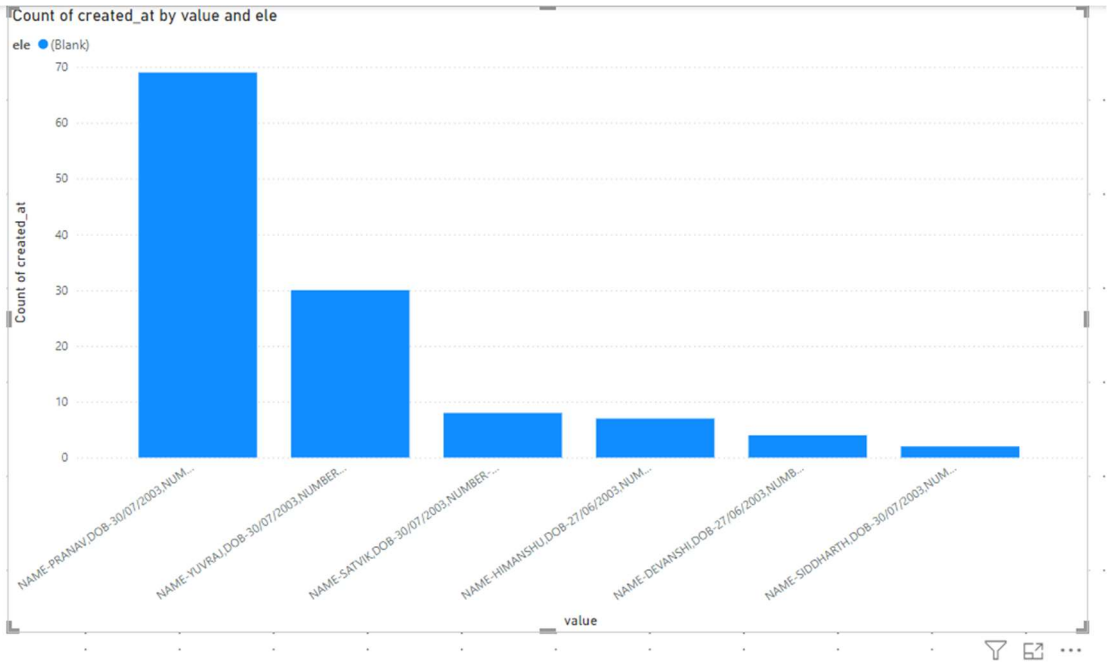


FIG.23 Clustered Column Chart of no. of entries for each member

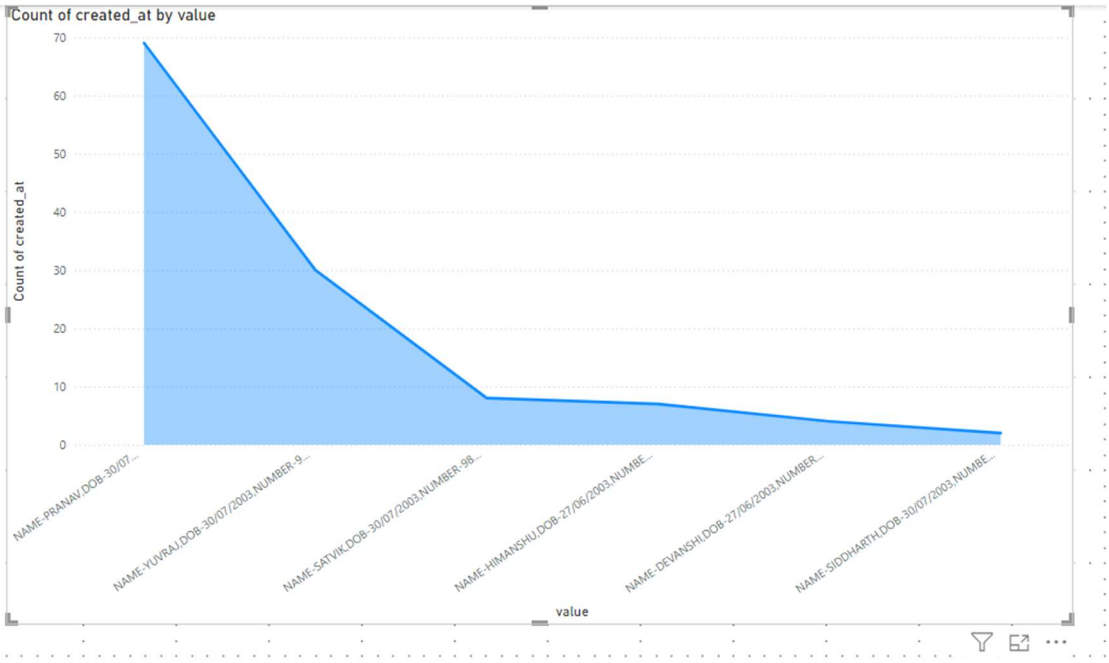


FIG.24 Line Chart of no. of entries for each member

REFERENCES

[1] Xiaoyin Xu; Majid Ahmadi; *"A Human Face Recognition System Using Neural Classifiers"*, COMPUTER GRAPHICS, IMAGING AND VISUALISATION (CGIV 2007), 2007. (IF:8)

[2] Nagasree Y Lakshmi Venkata; Ch. Rupa; B Dharmika; Teja G Nithin; N Vineela; *"Intelligent Secure Smart Locking System Using Face Biometrics"*, 2021 INTERNATIONAL CONFERENCE ON RECENT TRENDS ON ..., 2021. (IF: 3)

[3] Nur Jaini; Ervan Asri; Fitri Nova; *"Sistem Manajemen Kehadiran Menggunakan Metode Face Recognition Berbasis Web"*, 2021.

[4] Erviansyah Fadly; Suryo Adi Wibowo; Agung Panji Sasmito; *"SISTEM KEAMANAN PINTU KAMAR KOS MENGGUNAKAN FACE RECOGNITION DENGAN TELEGRAM SEBAGAI MEDIA MONITORING DAN CONTROLLING"*, JATI (JURNAL MAHASISWA TEKNIK INFORMATIKA), 2021.

[5] M. Difa; S. Suroso; Jon Endri; *"Implementasi Sistem Pengenalan Wajah Sebagai Automatic Door Lock Menggunakan Modul ESP32 CAM"*, 2021.

[6] Jie Wen; Jianghua Liu; Fu Xu; Xinyu Duan; Jie-Kai Huang; *"Face Recognition System Design Based on ESP32"*, 2022 INTERNATIONAL SEMINAR ON COMPUTER SCIENCE AND ..., 2022.

[7] Puput Dani Prasetyo Adi; Yuyu Wahyu; *"Performance Evaluation of ESP32 Camera Face Recognition for Various Projects"*, INTERNET OF THINGS AND ARTIFICIAL INTELLIGENCE JOURNAL, 2022.

[8] Ilham Firman Ashari; *"Parking System Optimization Based on IoT Using Face and Vehicle Plat Recognition Via Amazon Web Service and ESP-32 CAM"*, COMPUTER ENGINEERING AND APPLICATIONS JOURNAL, 2022. (IF: 3)

[9] Mohamed Osman Baloola; Fatimah Ibrahim; Mas S Mohktar; *"Optimization of Medication Delivery Drone with IoT-Guidance Landing System Based on Direction and Intensity of Light"*, SENSORS (BASEL, SWITZERLAND), 2022.

[10] A. Ipanhar; T. Wijaya; Pamor Gunoto; *"PERANCANGAN SISTEM MONITORING PINTU OTOMATIS BERBASIS IOT MENGGUNAKAN ESP32-CAM"*, SIGMA TEKNIKA, 2022.

[11] Nauval Muhammad; Endro Ariyanto; Yogi Anggun Saloko Yudo; *"IMPROVED FACE DETECTION ACCURACY USING HAAR CASCADE CLASSIFIER METHOD AND ESP32-CAM FOR IOT-BASED HOME DOOR SECURITY"*, JIPI (JURNAL ILMIAH PENELITIAN DAN PEMBELAJARAN INFORMATIKA), 2023.

[12] Thair A. Kadhim; Walid Hariri; Nadia Smaoui Zghal; Dalenda Ben Aissa; *"A Face Recognition Application for Alzheimer's Patients Using ESP32-CAM and Raspberry Pi"*, JOURNAL OF REAL-TIME IMAGE PROCESSING, 2023

[13] Hitoshi Hongo; Mamoru Yasumoto; Yoshinori Niwa; Mitsunori Ohya; Kazuhiko Yamamoto; *"Focus of Attention for Face and Hand Gesture Recognition Using Multiple Cameras"*, PROCEEDINGS FOURTH IEEE INTERNATIONAL CONFERENCE ON ..., 2000. (IF: 4)

[14] Wenbo Dong; Zhenan Sun; Tieniu Tan; Xianchao Qiu; *"Self-adaptive Iris Image Acquisition System"*, 2008. (IF: 3)

[15] Ohil K. Manyam; Neeraj Kumar; Peter N. Belhumeur; David J. Kriegman; *"Two*

Faces Are Better Than One: Face Recognition in Group Photographs", 2011 INTERNATIONAL JOINT CONFERENCE ON BIOMETRICS (IJCB), 2011. (IF: 3)
[16] Abhijith Punnappurath; Ambasamudram Narayanan Rajagopalan; Sima Taheri; Rama Chellappa; Guna Seetharaman; *"Face Recognition Across Non-uniform Motion Blur, Illumination, And Pose"*, IEEE TRANSACTIONS ON IMAGE PROCESSING : A PUBLICATION OF ..., 2015. (IF: 3)
[17] Omar Abdul Rhman Salim; Rashidah Funke Olanrewaju; Wasiu Adebayo Balogun; *"Class Attendance Management System Using Face Recognition"*, 2018 7TH INTERNATIONAL CONFERENCE ON COMPUTER AND ..., 2018. (IF: 3)
[18] Nagasree Y Lakshmi Venkata; Ch. Rupa; B Dharmika; Teja G Nithin; N Vineela; *"Intelligent Secure Smart Locking System Using Face Biometrics"*, 2021 INTERNATIONAL CONFERENCE ON RECENT TRENDS ON ..., 2021. (IF: 3)
[19] Nur Jaini; Ervan Asri; Fitri Nova; *"Sistem Manajemen Kehadiran Menggunakan Metode Face Recognition Berbasis Web"*, 2021.
[20] M. Difa; S. Suroso; Jon Endri; *"Implementasi Sistem Pengenalan Wajah Sebagai Automatic Door Lock Menggunakan Modul ESP32 CAM"*, 2021.
[21] Puput Dani Prasetyo Adi; Yuyu Wahyu; *"Performance Evaluation of ESP32 Camera Face Recognition for Various Projects"*, INTERNET OF THINGS AND ARTIFICIAL INTELLIGENCE JOURNAL, 2022.
[22] Nauval Muhammad; Endro Ariyanto; Yogi Anggun Saloko Yudo; *"IMPROVED FACE DETECTION ACCURACY USING HAAR CASCADE CLASSIFIER METHOD AND ESP32-CAM FOR IOT-BASED HOME DOOR SECURITY"*, JIPI (JURNAL ILMIAH PENELITIAN DAN PEMBELAJARAN INFORMATIKA), 2023.