**SMART SURVEILLANCE SYSTEM :**

**FACE RECOGNITION AND INTRUDER DETECTION USING MEDIAPIPE AND DLIB**

**A Mini project Report submitted**

**In the Partial of Fulfillment of the Requirements for**

**ECE IOT – III Semester PTI Lab**

*by*

**1. BT22ECI036 Mohit Talwar**

**2. BT22ECI039 Plawang Shishu**

**3. BT22ECI057 Parth Bhanudas Kadu**

*Under the Guidance of*

**Dr. Snehal B. Shinde**



भारतीय सूचना प्रौद्योगिकी संस्थान, नागपुर

Indian Institute of Information Technology, Nagpur

**CERTIFICATE**

This is to certify that the Project Based Learning-I work titled “**SMART SURVEILLANCE SYSTEM : FACE RECOGNITION AND INTRUDER DETECTION USING MEDIAPIPE AND DLIB**” that is being submitted by **MOHIT TALWAR (BT22ECI036), PLAWANG SHISHU (BT22ECI039), PARTH BHANUDAS KADU (BT22ECI057)** is in partial fulfillment of the requirements for **B. Tech. (ECI) – III Semester**, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Dr. Snehal Bankatrao Shinde

**Guide**

**The Report is satisfactory / unsatisfactory**

**Approved by**

|  |
| --- |
| **Mini-Project coordinator** |
| Dr. Snehal B. Shinde |

**ABSTRACT**

In today's security-conscious world, the need for robust and efficient security systems has become paramount. This project introduces a real-time security solution which leverages the power of **OpenCV**, **MediaPipe**, and **dlib** to enhance property security.The system integrates OpenCV for real-time video stream processing, MediaPipe for **accurate face detection**, and dlib for **facial recognition**. This combination allows the system to monitor live video feeds and trigger alerts when it detects an intruder, distinguishing between recognized and unrecognized individuals. Key features of the system include live video streaming, instantaneous face detection and recognition, as well as the capability to activate a buzzer and secure a door in response to potential security threats. It also sends email and Telegram notifications, making it a **comprehensive security solution**. This report provides insights into the project's technical implementation, the technologies employed, and its performance in terms of intruder detection and response times. It serves as a practical example of how computer vision, deep learning, and IoT technologies can be harnessed for security and surveillance, with applications in home security, office monitoring, and various security-sensitive environments. The project exemplifies the effective integration of open-source technologies for real-world security applications, offering a significant advancement in real-time security solutions.

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Chapter** | **Title** | **Page Number** |
| **1.** |  | **Abstract** | **3** |
| **2.** |  | **Table of Contents** | **4** |
| **3.** | **1**  **1.1**  **1.2**  **1.3** | **Introduction**  **Objectives**  **Literature Survey**  **Organization of the Report** | **6**  **6**  **7**  **9** |
| **4.** | **2**  **2.1**  **2.2**  **2.3**  **2.4**  **2.4.1**  **2.4.2** | **SMART SURVEILLANCE SYSTEM : FACE RECOGNITION & INTRUDER DETECTION**  **Existing System**  **Proposed System**  **Data Flow Diagram**  **System Details**  **Software**  **Hardware** | **10**  **10**  **11**  **12**  **12**  **12**  **13** |
| **5.** | **3**  **3.1**  **3.2**  **3.3**  **3.4**  **3.5** | **Implementation**  **Importing required libraries**  **Face detection with mediapipe**  **Face recognition with facerecognition**  **Real-Time Face Recognition**  **Hardware Setup** | **15**  **15**  **17**  **17**  **18**  **19** |
| **6.** | **4**  **4.1**  **4.2**  **4.3** | **Results and Discussion**  **Accuracy and Recognition Rate**  **Processing speed**  **Achievements and Limitations** | **20**  **20**  **20**  **21** |
| **7.** | **5** | **Conclusion and Future Works** | **22** |
| **8.** | **7** | **References** | **23** |

**CHAPTER 1**

**INTRODUCTION**

Machine Learning (ML), a subfield of Artificial Intelligence (AI), has witnessed a remarkable integration within the realm of Computer Vision. It constitutes the crux of how computer systems interpret and discern information, autonomously determining or categorizing tasks, all with or without human intervention. Within the domain of Computer Vision, ML algorithms have found increased utilization, particularly in the arenas of Face Recognition and Detection, where the amalgamation of data-driven approaches and human-like pattern recognition capabilities has revolutionized the way we address security and surveillance challenges.

The Significance of Computer Vision in Security Systems

In an era marked by incessant technological progress and digital innovation, the domains of surveillance and security have transcended their conventional boundaries, evolving into indispensable facets of our daily lives. Whether safeguarding private residences, commercial establishments, or public spaces, the necessity to vigilantly monitor and respond to security breaches in real-time has grown paramount.

The synergy of diverse technologies, including Computer Vision, Internet of Things (IoT), and multi-channel communication frameworks, has paved the way for the development of comprehensive surveillance and security systems. These systems no longer rely solely on human vigilance but harness the power of ML to augment their capabilities significantly.

* 1. **Objectives**

The below mentioned are the objectives of this project:

* Detect and recognize individuals entering a defined area, distinguishing between known and unknown individuals.
* Provide real-time alerts and notifications when unusual activity, such as unauthorized access, is detected via Email and Telegram.
* Enable remote monitoring through a live video stream accessible via the internet i.e . Responsive UI which can be accessed from anywhere.
* Controls and triggers Actuators like locks and alarms in Realtime.
  1. **Literature Survey**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| AUTHOR | TITLE | YEAR  And  PUBLISHER | METHODOLOGY | ACCURACY | OBSERVATIONS |
| Nico Surantha and Wingky R. Wicaksono | An IoT based House Intruder Detection and Alert System using Histogram of Oriented Gradients | 2019  and  Journal of Computer Science | Histogram of Oriented Gradients,  SVM classification, PIR Sensor, Internet of Things, Raspbery Pi | The simulation results show that the system can detect an intruder within seconds with accuracy of **90%** with processing time around 2 seconds. | 1. HoG-based Intruder Detection: Captures, processes, and analyzes images for intruder detection using HoG, including grayscale conversion, gradient calculation, and R-HOG feature extraction. 2. SVM Classification: Utilizes an SVM classifier to distinguish between people and non-people based on extracted features, triggering email notifications and buzzer when people are detected. |
| Nourman S. Irjanto , Nico Surantha | Home Security System with Face Recognition based on Convolutional Neural Network | 2020 and  International Journal of Advanced Computer Science and Applications | CNN Alexnet; facial recognition; Raspberry Pi;  Home door security | The training involves passing the collected facial data , adjusting model weights and parameters to optimize accuracy.  1048 facial data of the face of the homeowner where the results are quite accurate and the accuracy is the **97.5%** | The algorithm employs a CNN Alexnet model. CNNs consist of layers such as Convolutional, Pooling, and Fully Connected layers that are designed to automatically learn and extract features from input images.  Once the CNN model is trained, it is implemented on a Raspberry Pi, serving as a microcontroller to control access. The system captures and processes the face of the homeowner to grant or deny access. |
| G.Mallikharjuna Raoa , Haseena Palleb , Pragna Dasaric , Shivani Jannaikoded | Implementation of Low Cost IoT Based Intruder Detection System by Face Recognition using Machine Learning | 2021 and Turkish Journal of Computer and Mathematics Education | Machine Learning Algorithms, face detection, face recognition, OpenCV, Internet of Things | Haar Cascades have greater accuracy in real time detecting and recognizing the human face. The image processing rate attained by the system is noted to be around 28 images in 1 sec for the complete process. | Analyzing simple rectangular patterns of dark and light pixels, which are used as features to identify objects. Trained on datasets containing positive images (with the object of interest) and negative images Email and Buzzer Notifications are followed if intrusion is detected using SMTP. |
| Dr. M. Gayathri , M. Lakshmanan , D. Lalitha Venkata Krishna | Object Surveillance Detection | 2023 and  International Journal for Research in Applied Science & Engineering Technology (IJRASET) | Security Surveillance and Real Time Security Surveillance, Face Detection, Twillio API. | YOLO's efficiency and speed make it suitable for real-time applications, like autonomous vehicles and surveillance systems. | YOLO rapidly and precisely detects objects in images, leveraging anchor boxes for location prediction, handling multiple object classes, and integrating localization and classification for precise results. By gridding images and forecasting bounding boxes and class probabilities, the system provides instant notifications for any detected unwanted activity via email, SMS, or real-time monitoring of the surroundings. |

**1.3 Organization of the Report**

The remaining chapters of the project report are described as follows:

* Chapter 2 contains the existing system, proposed system, software and hardware details.
* Chapter 3 describes implementation of the project.
* Chapter 4 discusses the results obtained after the project was implemented.
* Chapter 5 concludes the report and gives idea of future scope.
* Chapter 6 consists of code of our project.
* Chapter 7 gives references.

**CHAPTER 2**

**AN EFFECTIVE CLASSIFICATION OF HEART DISEASE PREDICTION SYSTEM USING ML**

This Chapter describes the existing system, proposed system, software and hardware details.

* 1. **Existing Systems**

In the context of face detection, the existing systems explored in the literature survey make use of various machine learning algorithms, encompassing techniques such as Hologram of Gradients, Convolutional Neural Networks (CNNs), Haar Cascades, and the YOLO (You Only Look Once) algorithm. These systems have their own distinct advantages and limitations, which are assessed based on factors such as real-time performance, accuracy, and implementation.

**Positives**:

* **Efficiency and Real-time Performance**: Some of the existing systems, such as those utilizing YOLO or certain CNN architectures, demonstrate high efficiency and real-time object detection capabilities. This is particularly important for applications like surveillance, security, and robotics where immediate response is crucial.
* **Accuracy and Robustness**: Machine learning algorithms like CNNs are known for their accuracy in face detection. They can handle variations in lighting conditions, poses, and facial expressions, making them suitable for a wide range of applications.
* **Wide Applicability**: The variety of algorithms used in existing systems means that they can be tailored to suit different use cases. For instance, Haar Cascades are efficient for simple face detection, while more complex models like YOLO are versatile for object detection beyond faces.

**Negatives**:

* **Real-time Performance Variability**: While some algorithms excel in real-time performance, others may not be as fast. For instance, more complex models like YOLO might require powerful hardware for real-time operation, which can be a limitation in resource-constrained environments.
* **Resource Intensiveness**: Models like CNNs and YOLO tend to be resource-intensive, demanding significant computational power. This can be a drawback in applications with limited resources, such as embedded systems or IoT devices.
* **Similar Notification Mechanisms**: Many of the existing systems employ basic notification mechanisms, such as triggering a buzzer or sending an email. While these are reliable and simple, they may lack the flexibility to integrate with more modern communication platforms like Telegram or other social media.

The choice of an appropriate system depends on the specific requirements and constraints of the application. As technology advances, there is also room for improvement in terms of notification mechanisms and the integration of modern communication platforms to enhance the overall functionality of these systems.

**2.2 Proposed System**

The proposed system is a comprehensive solution for real-time face recognition and detection with live streaming capabilities, designed to be cost-effective and easily accessible through an ESP32-based PCB. This system leverages a variety of technologies and components to achieve its objectives. Here, we'll delve into the key aspects of the proposed system, highlighting its strengths and potential challenges.

**Positives:**

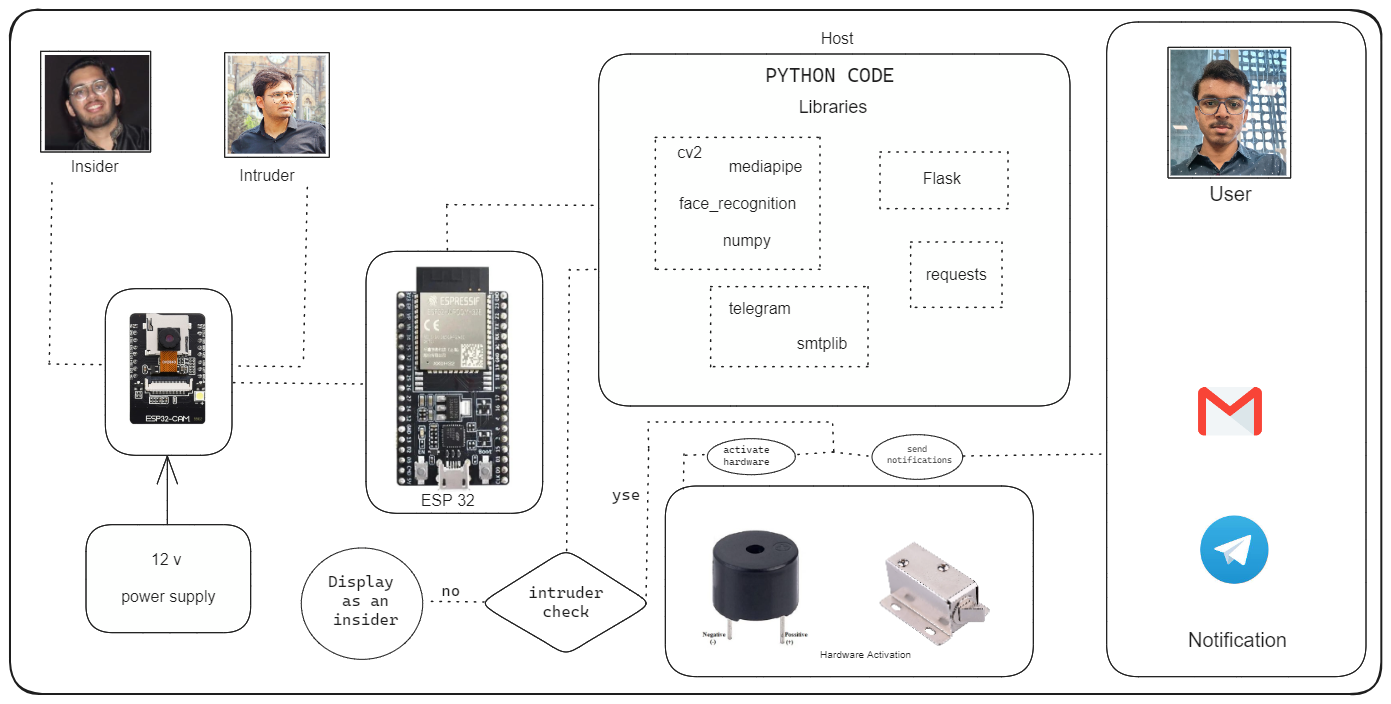
* **Live Streaming and Real-Time Face Detection**: The system offers live streaming capabilities by capturing images from an ESP32-CAM device. It uses the OpenCV library to process these images in real-time, enabling face detection through the MediaPipe Face Detection module. This provides immediate feedback and enhances security applications.
* **Face Recognition**: The system combines face detection with face recognition. It can recognize known individuals by comparing the detected faces to a pre-defined set of known face encodings. This feature is valuable for personalized access control and security.
* **Notification Mechanisms**: The system includes versatile notification mechanisms. When an unrecognized face is detected, it triggers a series of notifications, including sounding a buzzer, locking a door, sending an email, and even sending a Telegram message. This comprehensive approach enhances security and situational awareness.
* **Cost-Effective ESP32 Integration**: The system's integration with ESP32, a low-cost and versatile microcontroller, makes it accessible to a broad range of users. This is advantageous for DIY enthusiasts, small businesses, and projects with budget constraints.
* **Flexibility for Customization**: Users can easily customize the system by adding new faces to the recognition database and adjusting notification settings. This flexibility ensures that the system can adapt to various use cases and environments.

**Challenges:**

* **Resource Requirements**: While the system aims to be cost-effective with ESP32, resource-intensive tasks like face recognition can still strain the device's capabilities. Users may need to ensure they have sufficient hardware resources to run the system effectively.
* **Integration and Setup**: Implementing the system might require some technical expertise in setting up and configuring the ESP32, defining recognized faces, and establishing notification channels. A user-friendly interface or setup wizard could enhance accessibility.
* **Maintenance and Updates**: Like many complex systems, the proposed system may require periodic updates and maintenance, particularly when new faces need to be added or the system's notification behavior needs adjustments.

In summary, the proposed system offers a powerful and cost-effective solution for live streaming, face recognition, and face detection. It combines a variety of notification mechanisms to enhance security and situational awareness. However, users should be mindful of the resource requirements and setup complexities, along with the need for ongoing maintenance and security considerations.

**2.3 Data Flow Diagram**

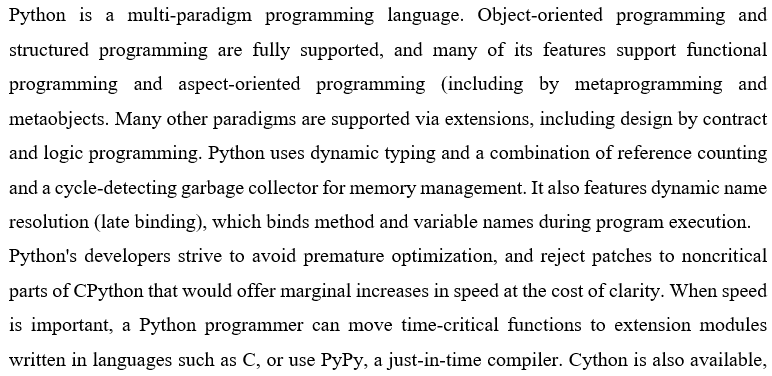
****

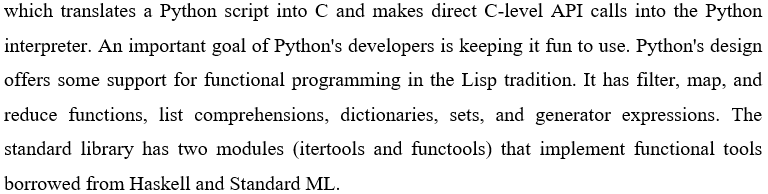
**2.4 System Details**

This section describes the software and hardware details of the system:

**2.4.1 Software Details**

1. **Python**





BENEFITS OF PYTHON

* + Presence of Third-Party Modules
  + Extensive Support Libraries
  + Open Source and Community Development
  + Learning Ease and Support Available
  + User-friendly Data Structures
  + Productivity and Speed
  + Highly Extensible and Easily Readable Language.

**ii) IDLE**

IDLE (Integrated Development and Learning Environment) is an integrated development environment (IDE) for the Python programming language. It comes bundled with the standard Python distribution, so you don't need to install it separately. IDLE provides a graphical user interface for writing, running, and debugging Python code.

IDLE can be used to execute a single statement and create, modify, and execute Python scripts. IDLE provides a fully-featured text editor to create Python scripts that include features like syntax highlighting, autocompletion, and smart indent.

**2.4.2 Hardware Details**

Hardware devices used in this project are ESP32, ESP32 CAM, solenoid lock, Relay, Power supply (12 V).

**i) ESP32**

The ESP32 is a highly integrated microcontroller module that offers a robust computing platform for various applications. Developed by Espressif Systems, it features a dual-core processor, Wi-Fi, and Bluetooth capabilities, making it an ideal choice for IoT (Internet of Things) projects.

**Key Features:**

* **Dual-Core Processor**: The ESP32 is equipped with two powerful Xtensa LX6 CPUs, allowing for efficient multitasking and high-performance computing.
* **Wireless Connectivity**: It supports both Wi-Fi (802.11 b/g/n) and Bluetooth (BLE) protocols, enabling seamless communication with other devices and networks.

**ii) ESP32 CAM**

The ESP32-CAM is a versatile microcontroller module that integrates an ESP32 system-on-

chip (SoC) with Wi-Fi and Bluetooth capabilities, along with a camera module. It allows you to

capture images or video and transmit them wirelessly, making it ideal for various IoT and

surveillance applications.

For this project, the ESP32-CAM can be configured to capture images or stream video,

which can then be transmitted over Wi-Fi to a central server or accessed remotely via a web

interface. Motion detection algorithms can be implemented to trigger image capture or video

streaming when movement is detected within the camera's field of view

**iii) Solenoid lock**

A solenoid lock is an electromechanical device used to control the access to a door, gate, or other

entry points. It operates using an electric current to generate a magnetic field, which in turn

controls the locking mechanism. When the solenoid is energized (current is applied), it produces

a magnetic force that either engages or disengages the lock.

**iv) Relay**

A relay is an electromechanical device that is used to control electrical circuits and devices by

using a small electrical signal to switch a larger load on or off. It acts as a kind of electrically

operated switch, allowing a low-power circuit to control a high-power circuit.

**v) Power Supply**

12V DC Power Supply

**CHAPTER 3**

**IMPLEMENTATION**

**3.1 Importing required libraries**

**i) cv2**

The cv2 library, also known as OpenCV (Open Source Computer Vision Library), is a

popular open-source computer vision and machine learning software library written in C++

and designed for Python. It provides a wide range of tools and algorithms for tasks related to

computer vision, image processing, and machine learning.

* **Image and Video Processing:** cv2 provides a set of functions for reading, writing, and

manipulating images and videos. This includes tasks like resizing, rotating, cropping, and

converting between different image formats

* **Face Detection and Recognition:** OpenCV provides tools for detecting faces in images and videos, as well as tools for facial landmark detection and face recognition.
* **Image Filtering and Enhancement:** You can apply various filters to images for tasks like

blurring, sharpening, edge detection, and noise reduction. This is useful for tasks like image preprocessing before further analysis.

**ii)** **mediapipe**

Mediapipe is an open-source library developed by Google that provides a comprehensive

framework for building real-time applications that involve various types of media processing

tasks, such as pose estimation, hand tracking, face detection, and more. It is particularly

useful for tasks related to computer vision and machine learning.

* **Face Detection and Tracking:** Mediapipe offers a face detection module that can identify and track faces in a video stream. This can be used to monitor the presence of individuals in a given area.

**iii) Numpy**

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array

objects and a collection of routines for processing those arrays. Using NumPy, mathematical

and logical operations on arrays can be performed.

* Mathematical and logical operations on arrays..
* Operations related to linear algebra. NumPy has in-built functions for linear algebra and

random number generation.

**iv) face\_recognition**

The face\_recognition library in Python is a popular open-source library that provides tools for

face detection, face recognition, and facial landmarks detection. It is built on top of the Dlib

library, which is a toolkit for machine learning and computer vision

* **Face Detection:** The library can detect faces in images or video streams. It uses a deep learning model to locate faces in images or frames of a video.
* **Face Recognition:** Once faces are detected, the library can perform face recognition to

determine if a detected face matches a known face. It does this by comparing facial features (like eyes, nose, mouth positions) with a pre-trained model.

**v) smtplib**

The smtplib library in Python is used for sending emails using the Simple Mail Transfer

Protocol (SMTP). It provides a convenient way to send emails from your Python programs.

**vi) requests**

The requests library in Python is a popular HTTP library that allows you to send HTTP

requests and interact with web resources. It provides a simple and elegant way to make

HTTP requests, handle responses, and manage sessions. This library is widely used for tasks

like web scraping, interacting with APIs, and more.

**vii) flask**

Flask is a lightweight web framework for Python. It's designed to make it easy to build web

applications quickly and with minimal code. Flask is known for its simplicity, flexibility, and

ease of learning, which makes it a popular choice for building web applications, APIs, and

even simple websites.

**Flask can be used in various ways:**

* **Web Interface:** Flask can be used to create a web-based interface for viewing surveillance

footage. You can create a dashboard where users can log in, view live streams, access

recorded videos, and interact with the surveillance system.

* **Integration with Cameras:** Flask can be used to handle communication with IP cameras or other types of surveillance devices. It can receive and process video streams, send commands to control cameras, and manage their configurations.

**vii) telegram**

The python-telegram-bot library is a popular Python wrapper for the Telegram Bot API. It

allows developers to easily interact with Telegram bots using Python. With this library, you

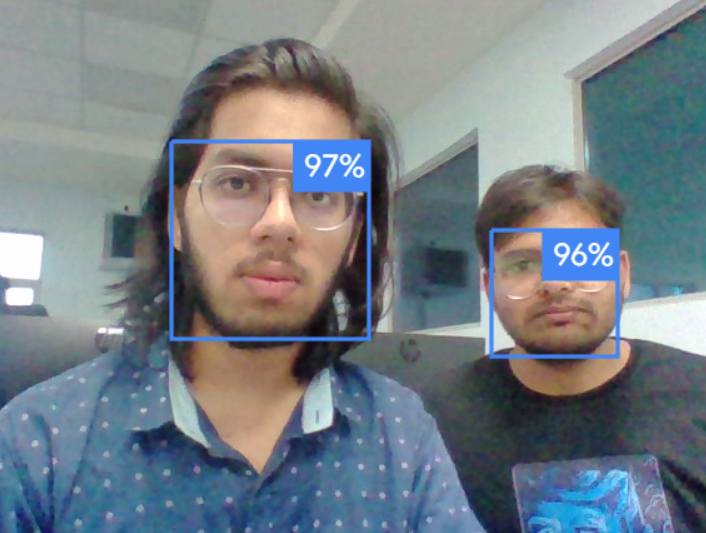
can create, configure, and manage Telegram bots, send messages, receive updates, and

perform various other tasks.

**3.2 Face Detection with MediaPipe**

The face detection component of the system relies on the MediaPipe Face Detection module. Here's a closer look at the ML aspects of this component:

* **Machine Learning Model**: MediaPipe utilizes a machine learning model specifically designed for face detection. This model has undergone extensive training to recognize faces in various poses, lighting conditions, and orientations.
* **Min Detection Confidence**: A crucial parameter in the implementation is the min\_detection\_confidence. This parameter sets the threshold for the model's confidence in detecting a face in a given frame. Faces with confidence scores below this threshold are not considered.
* **Bounding Box Localization**: The model provides a bounding box that specifies the face's location within the frame. The model's ability to accurately localize faces relies on training data that includes a wide range of facial variations.

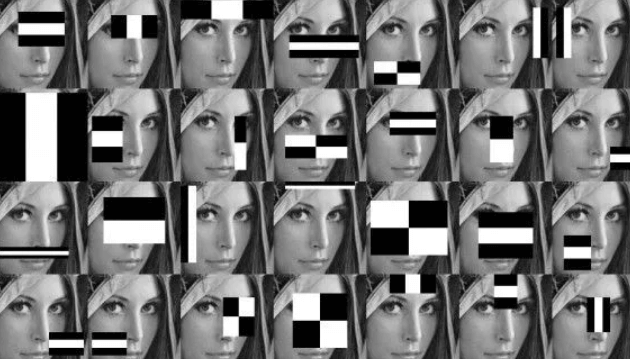


*Image: An example of face detection using the MediaPipe Face Detection module.*

**3.3 Face Recognition with face\_recognition Library**

The face recognition component is built upon the face\_recognition library. Here are the key ML aspects of this component:

* **Face Encoding**: Face recognition is performed by encoding known faces. The face\_recognition library extracts distinctive features from the face, creating a numerical representation called a face encoding. This encoding captures unique facial characteristics.
* **Comparing Face Encodings**: During real-time face recognition, the system compares the face encoding of the detected face with the encodings of known individuals. This involves computing the similarity or distance between the two encodings.



*Image: An example of Haar Cascades Algorithm used in encoding.*

* **Threshold for Recognition**: The system employs a recognition threshold to determine whether a detected face matches a known face. The threshold value can be adjusted to control the sensitivity of recognition.
* **Classification as Recognized or Unknown**: A classification decision is made based on the results of encoding comparisons. If the similarity score between the detected face and a known face encoding exceeds the recognition threshold, the system recognizes the individual. Otherwise, the face is classified as unknown.

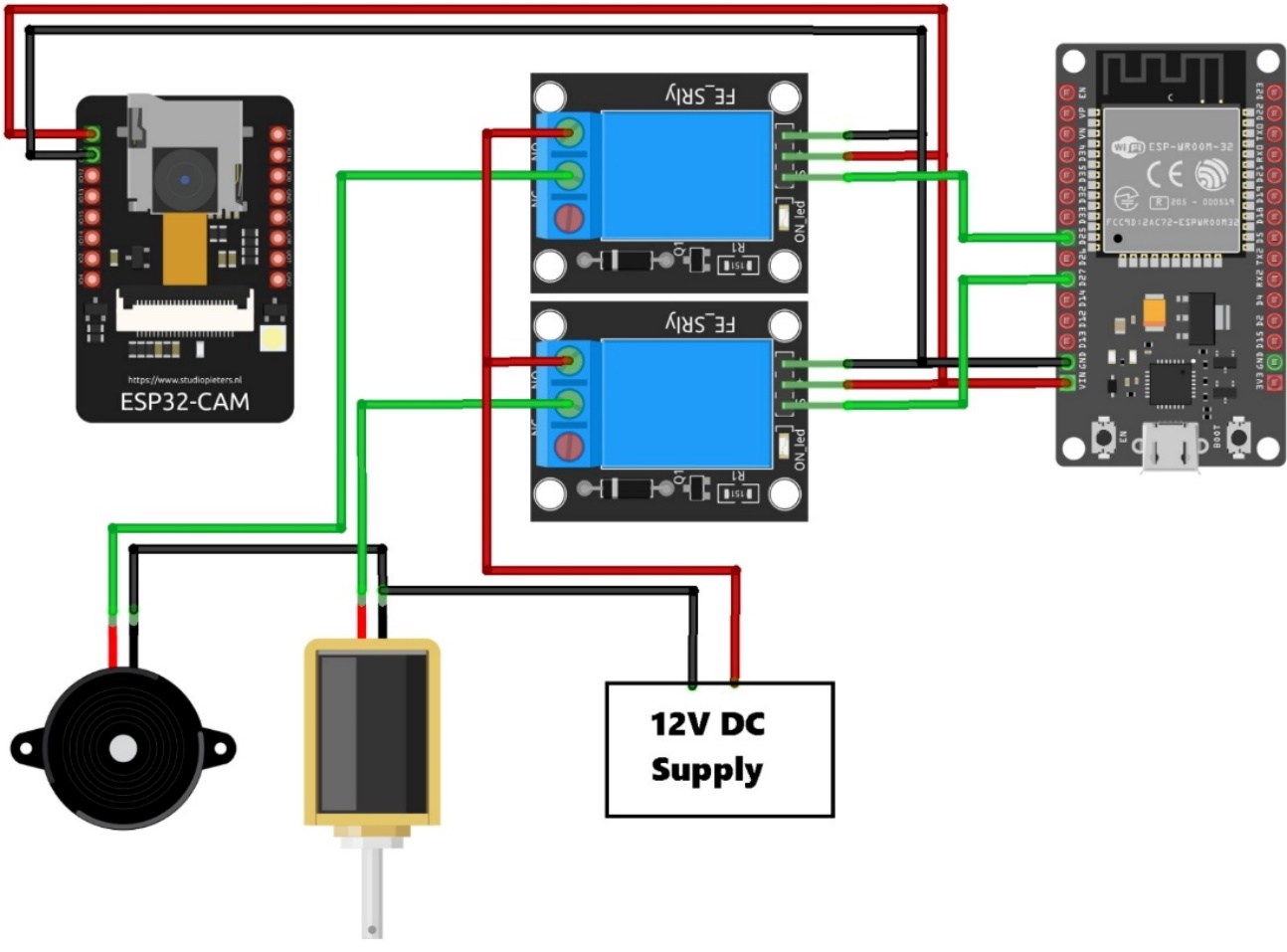
**3.4 Real-Time Face Recognition**

The real-time face recognition process combines face detection with face recognition. Here's how these components work together from an ML perspective:

* **Face Detection and Cropping**: The MediaPipe Face Detection model identifies faces in the video stream and provides bounding box coordinates. Once detected, the system crops the face region from the frame. The ability to accurately detect faces is a critical aspect of successful recognition.
* **Resizing and Encoding**: To perform face recognition, the cropped face image is resized to a consistent scale and converted to an RGB format. This standardized image is then encoded using the known faces' encodings.
* **Matching and Classification**: The system conducts a matching process by comparing the encoding of the detected face with the encodings of known individuals. This ML-based comparison determines whether the face is recognized or classified as unknown.
* **Annotation and Visual Feedback:** When a recognized individual is detected, their name is displayed on the video feed. This visual feedback is enabled by the ML-based recognition process.

**3.5 Hardware Setup**

The hardware setup is a critical component of the system. It involves the selection and configuration of hardware elements like the ESP32-based PCB, the ESP32 Camera Module, the buzzer, and the door lock mechanism. These components work in tandem with the ML-driven software to enable real-time face detection and recognition.



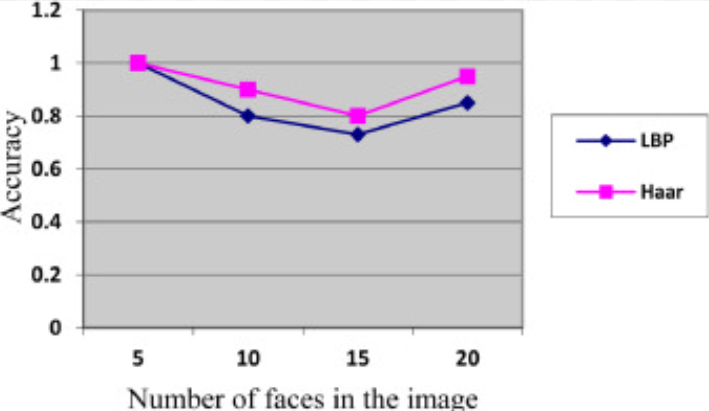
**CHAPTER 4**

**RESULTS AND DISCUSSIONS**

In this chapter, we present the results of the face recognition and detection system's implementation. We discuss the system's performance, its effectiveness in recognizing known individuals and detecting unknown faces, and the implications of its real-time capabilities. Furthermore, we delve into the technical and practical aspects of the system's operation.

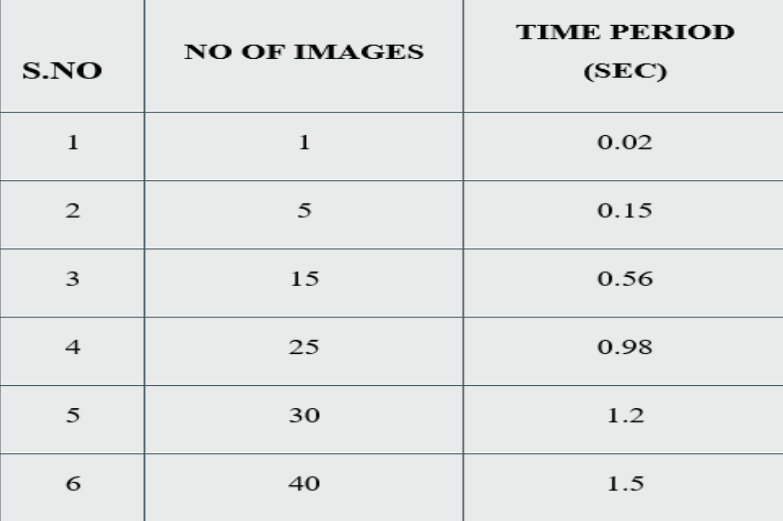
4.1 Accuracy and Recognition Rate

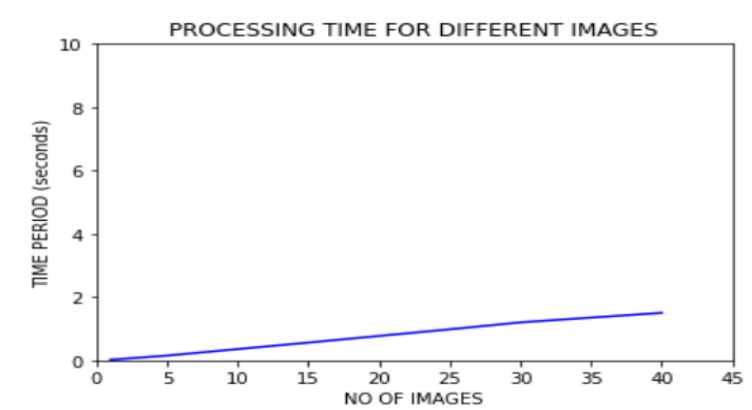
The performance of the face recognition component is assessed in terms of accuracy and recognition rate.



4.2 Processing Speed

The system's real-time capabilities are evaluated in terms of processing speed. We measure the time it takes for the system to detect and recognize faces in a live video feed. The system aims to achieve a processing speed that is sufficient for practical security and access control applications.





4.3 Achievements and Limitations

The results demonstrate the system's achievements in accurate face recognition and real-time detection. It excels in recognizing known individuals, and its performance in reducing false positives and false negatives is notable. Additionally, the system achieves a processing speed suitable for real-time applications.

However, certain limitations should be acknowledged. The system's resource-intensive nature might pose challenges on hardware with limited capabilities. It requires consistent updates and maintenance, and security measures should be in place to protect the system from unauthorized access.

**CHAPTER 5**

**CONCLUSION AND FUTURE WORK**

In conclusion, the face recognition and detection system successfully combines real-time face recognition and notification mechanisms, providing a versatile and cost-effective solution for enhancing security. The system's adaptability, user-friendly interface, and integration of machine learning models contribute to its effectiveness. However, future work should focus on security enhancements, resource optimization, scalability, and continued learning to further refine its capabilities and applications.

The future of face recognition and detection systems is promising, with potential advancements in security measures, integration with other devices and platforms, and continuous learning to improve accuracy. The evolution of these systems aligns with the growing importance of access control and surveillance in a rapidly changing technological landscape, making them a valuable asset in various industries. The future scope of face recognition and detection systems extends to various domains, including security, smart homes, and business applications. With ongoing advancements in artificial intelligence and machine learning, these systems will likely become more sophisticated, offering enhanced accuracy in recognizing faces under diverse conditions. Moreover, integrating them with IoT devices and edge computing technologies will enable real-time decision-making and provide a foundation for innovative applications, such as responsive environments that adapt to individuals' preferences and requirements. As privacy and ethical considerations gain prominence, future developments will also focus on responsible usage, data protection, and transparency, ensuring that these systems continue to evolve while respecting individual rights and societal values.

**REFERENCES**

|  |
| --- |
|  |
| 1. Sahoo, K. C., & Pati, U. C. (2017, May). IoT based intrusion detection system using PIR sensor. In *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)* (pp. 1641-1645). IEEE. 2. Boyko, N., Basystiuk, O., & Shakhovska, N. (2018, August). Performance evaluation and comparison of software for face recognition, based on dlib and opencv library. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)* (pp. 478-482). IEEE. 3. Zhou, H., Chen, P., & Shen, W. (2018, March). A multi-view face recognition system based on cascade face detector and improved Dlib. In *MIPPR 2017: Pattern Recognition and Computer Vision* (Vol. 10609, pp. 37-42). SPIE. 4. T. Mantoro, M. A. Ayu and Suhendi, "Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods," 2018 6th International Conference on Multimedia Computing and Systems (ICMCS), Rabat, Morocco, 2018, pp. 1-5, doi: 10.1109/ICMCS.2018.8525935. 5. Suwarno, S., & Kevin, K. (2020). Analysis of face recognition algorithm: Dlib and opencv. *Journal of Informatics and Telecommunication Engineering*, *4*(1), 173-184. 6. Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., ... & Grundmann, M. (2019). Mediapipe: A framework for building perception pipelines. *arXiv preprint arXiv:1906.08172*. 7. Surantha, N., & Wicaksono, W. R. (2019). An IoT based house intruder detection and alert system using histogram of oriented gradients. *Journal of Computer Science*, *15*(8), 1108-1122. 8. Irjanto, N. S., & Surantha, N. (2020). Home security system with face recognition based on convolutional neural network. *International Journal of Advanced Computer Science and Applications*, *11*(11). 9. Rao, G. M. (2021). Implementation of Low Cost IoT Based Intruder Detection System by Face Recognition using Machine Learning. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, *12*(13), 353-362. 10. Gayathri, M., Lakshmanan, M., & Krishna, D. L. V. Object Surveillance Detection. |
|  |  |