INTRUSION DETECTION SYSTEM USING REGULATED PATROLLING ROBOTS FOR APARTMENTS

A PROJECT REPORT

Submitted by

NANDHA GOPAL M 211420104175

MOHAMED ASHEIM A 211420104160

NIRMAL KUMAR M 211420104183

in partial fulfillment for the award of

the degreeof

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **" INTRUSION DETECTION SYSTEM USING REGULATED PATROLLING ROBOTS FOR APARTMENTS "** is the bonafide work of **NANDHA GOPAL M (211420104175), MOHAMED ASHEIM A (211420104160), NIRMAL KUMAR M (211420104183)** who carried out the project work under my supervision.

Signature of the HOD with date

Dr. L. JABASHEELA M.E., Ph.D.,

PROFESSOR AND HEAD,

Department of Computer Science and Engineering,
Panimalar Engineering College,
Chennai -600 123.

Signature of the Supervisor with date

Dr. G. SENTHIL KUMAR M.C.A., M.Phil., M.B.A., M.E., Ph.D.,

SUPERVISOR, PROFESSOR,

Department of Computer Science, and Engineering,
Panimalar Engineering College,
Chennai -600 123.

Certified that the above candidates were examined in the End Semester Project Viva- Voce Examination held on 26/03/2024

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **NANDHA GOPAL M (211420104175), MOHAMED ASHEIM A (211420104160), NIRMAL KUMAR M (211420104183)** hereby declare that this project report titled **"INTRUSION DETECTION SYSTEM USING REGULATED PATROLLING ROBOTS FOR APARTMENTS"** , under the guidance of **DR. G. SENTHIL KUMAR, M.C.A., M.Phil., M.B.A., M.E., Ph.D.,** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

NANDHA GOPAL M [211420104175]

MOHAMED ASHEIM A [211420104160]

NIRMAL KUMAR M [211420104183]

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We wish to express our sincere thanks to our beloved Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHIKUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHIKUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We thank the Head of the CSE Department, **Dr. L. JABASHEELA, M.E., Ph.D.,** for the support extended throughout the project.

We would like to express our sincere thanks to **Project Coordinator DR. G. SENTHIL KUMAR, M.C.A., M.Phil., M.B.A., M.E., Ph.D.,** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

Pantech e Learning
DIGITAL LEARNING SIMPLIFIED
www.pantechelearning.com

13/03/2024

COMPLETION CERTIFICATE

This is to acknowledge that students from **"PANIMALAR ENGINEERING COLLEGE"** has completed **Project** on the **title** of **"INTRUSION DETECTION SYSTEM USING REGULATED PATROLLING ROBOTS FOR APARTMENTS"** at our concern from **NOV 2023** to **APRIL 2024.**

1. **NIRMAL KUMAR M**
2. **MOHAMED ASHEIM A**
3. **NANDHA GOPAL M**

For Pantech e learning.,

Authorized Signatory

Pantech eLearning Pvt Ltd.,
II Floor, Kotta Srinivasiah Charities Building,
Thanjavur Street, Near Duraisamy Subway, T.Nagar,
Chennai – 600017Phone: 91 44 42606470 |
hr@pantechmail.com

ABSTRACT

With the increasing need for enhanced security in residential apartments, there is a growing demand for advanced intrusion detection systems. This project proposesan innovative approach to address this need by utilizing controlled patrolling robotsas a component of an apartment-specific Intrusion Detection System (IDS). An automated patrolling robot network makes up the suggested IDS equipped with sensors and surveillance capabilities. These robots are deployed strategically throughout the apartment complex to monitor and detect any unauthorized intrusions. The robots autonomously navigate predefined paths, ensuring comprehensive coverage of the premises. The system incorporates various sensors such as motion detectors and cameras to identify suspicious activities or movements. When an intrusion is detected, the patrolling robot captures images and streams live video to provide real-time evidence of the intrusion. The captured data is processedusing advanced image recognition algorithms to differentiate between authorized residents and unauthorized individuals. In the event of an unauthorized intrusion, the system triggers immediate responses to mitigate security threats. These responses include activating alarms, sending alert notifications to security personnel, and initiating appropriate emergency protocols. Additionally, the regulated patrolling robots are equipped with the ability to engage in real-time communication with security personnel, enabling efficient coordination and response.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

CNN	Convolutional Neural Networks
RNN	Recurrent Neural Networks
DNN	Deep Neural Networks
IDS	Intrusion Detection System
MC	Microcontroller
PLC	Programmable Logic Controller
ICS	Industrial Control System
LBP	Local Binary Pattern
CCTV	Closed Circuit Television
WSN	Wireless Sensor Networks
DFD	Data Flow Diagram
UV	Ultra Violet
IR	Infrared Rays

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

In the realm of building security, the integration of advanced technologies has opened new avenues for efficient and cost-effective protection. This project presents a comprehensive system comprising supervised autonomous platforms equipped with intruder detection, obstacle avoidance, video capturing and streaming capabilities, and message indication for theft prevention. These platforms are centrally commanded from an integrated control station, offering seamless control and monitoring of the entire security infrastructure.

Traditional intrusion detection systems in apartments often rely on static surveillance devices and passive alarms. These systems face limitations in immediate threat response and the ability to differentiate between known and unknown individuals. In response to these challenges, the proposed system leverages deep learning for person identification through cameras mounted on regulated patrolling robots. The microcontroller, buzzer, DC motor, and motor drive facilitate a swift and intelligent response to detected intrusions, marking a paradigm shift in residential security.

1.2 PROBLEM DEFINITION

Women's safety is becoming a major worry in many parts of the world, and there is still worry in isolated locations for both women and men. Women half-filled of the world's population. But their survival has always been a concern, whether it's on the roads, trains, cabs, schools, or elsewhere, when it comes to living with honor and dignity. Women's empowerment in the country can be

achieved once their safety and security are assured, whether at home, in isolated locations, or while traveling.

1.3 OBJECTIVE

The objectives of implementing an intrusion detection system using regulated patrolling robots for apartments are:

Enhance Security:

The primary objective is to improve the security of apartments by deploying patrolling robots equipped with advanced sensors. These robots will effectively detect and prevent intrusions, reducing the risk of break-ins, theft, and other security breaches.

Comprehensive Coverage:

Ensure comprehensive coverage of apartment premises by designing robots capable of navigating complex environments, including multiple floors, corridors, and rooms. The objective is to deploy robots that can patrol the entire area, minimizing blind spots and ensuring thorough surveillance.

Accurate Intrusion Detection:

Develop algorithms and integrate high-quality sensors to accurately detect intrusions. The objective is to minimize false alarms while ensuring that the system can reliably differentiate between normal activities and potential security threats.

Integration with Existing Infrastructure:

Integrate the intrusion detection system with existing security infrastructure in apartments, such as CCTV cameras and access control systems. The objective is to create a cohesive and synchronized security framework that optimizes the utilization of resources and enhances overall effectiveness.

Scalability and Cost-Effectiveness:

Design a system that is scalable and cost-effective, allowing for deployment in different apartment complexes. The objective is to develop a solution that can be implemented without incurring exorbitant costs while maintaining efficiency and effectiveness.

CHAPTER 2
LITERATURE REVIEW

2.1 EXISTING APPROACHES TO INTRUSION DETECTION SYSTEMS

Intrusion Detection Systems (IDS) have been extensively studied and developed to address security concerns in various domains. Traditional approaches to IDS in residential settings typically involve static surveillance systems, such as Closed-Circuit Television (CCTV) cameras and motion sensors. While these systems can provide some level of security, they often lack the ability to respond promptly to intrusions and may have blind spots in coverage. More advanced approaches have emerged, such as Wireless Sensor Networks (WSNs) and smart home security systems. WSNs utilize a network of distributed sensors to monitor the environment and detect intrusions. Smart home security systems integrate various sensors, cameras, and communication devices to enable remote monitoring and control. However, these systems may still have limitations in terms of coverage, scalability, and real-time response.

In October, 2017, Tahzib Mashrik, Hasib Zunair, Maofic Farhan Karin Designed and Implemented Security Patrol Robot using Android Application. Their project aimed at designing a low-cost autonomous mobile security robot based on a multi sensor system that is user friendly and is also affordable. This project did not have surveillance system to enable the robot operator remotely observe the set target area to take pictures and capture video clips. [8]

In 2016, singoee sylvestre sheshai designed a raspberry pi based security system whose main aim was to design and develop a security system that included features such as motion detection, image processing and emailing or SMS to

facility owner. The system was based on Raspberry Pi SBC. The drawback of this project is that it did not have remote control, so the system required t4o be remotely controlled. [1]

The first security surveillance robot was proposed by Everett, H. & Gage, D.W, 1999 in "Mobile Detection Assessment and Response System (MDARS)" [9]. Since then security robots have become a growing interest with increasing developments in research and application. Sneha Singh and his team described IP Camera Video Surveillance system using Raspberry Pi technology. The Researchers aimed at developing a system which captures real time images and displays them in the browser using TCP/IP. The algorithm for face detection is being implemented on Raspberry Pi, which enables live video streaming along with detection of human faces. The research did not include any of surveillance reactions.

In 2014, Sanjana Prasad and his colleagues worked on developing a mobile smart surveillance system based on SBC of Raspberry Pi and motion detector sensor PIR. Their development boosts the practice of portable technology to offer vital safety to our daily life and home security and even control uses. The objective of their research is to develop a mobile smart phone home security system based on information capturing module combined with transmitting module based on 3G technology fused with web applications. The SBC will control the PIR sensor events and operates the video cameras for video streaming and recording tasks. Their system has the capability to count number of objects in the scene [11].

A team of intelligent mobile security robots patrols different floors of a building. During the occurrence of an abnormal event, the mobile robot transmits the relation location (floor number) of the event to the supervised computer [12].

An automatic patrolling vehicle acts as a security patroller in the security system, which can monitor those dead zones of the traditional fixed surveillance system. The remote monitoring capabilities can also be enhanced by using the wireless network. And the face detection system is adapted to record and analyses the invaders [13].

Yoichi Shimosasa et al., combining security surveillance and service system together, developed an autonomous guard robot which can guide visitors in daytime and patrol at the night [14].

"Obstacle Detection and Avoidance by a Mobile Robot". In this paper, deals with detection and avoidance of the various obstacles found in an environment, it was found that given a number of obstacles, the robot is able to detect and avoid the obstacle with an average accuracy of 86.62%. [15]

While several research and implementations of security robots are available, the technology used in security robots reduces its affordability. To solve this problem, the paper proposes a low-cost spy robot system based on raspberry pi, Pi camera and obstacle avoidance system that is user friendly and is also affordable. The spy robot system will be implemented using mobile application

2.2 ROBOTS FOR SECURITY PURPOSES IN RESIDENTIAL AREAS

The use of robots for security purposes has gained traction in recent years. Robots offer the advantage of mobility and adaptability, allowing for efficient coverage of large areas. In the context of residential settings, robots can be employed to enhance security measures by autonomously patrolling the premises,

detecting intrusions, and providing real-time alerts. Several studies have explored the use of robots for security purposes in residential settings. For example, mobile robots equipped with cameras and sensors have been deployed for surveillance and patrolling in residential areas. These robots can navigate autonomously, collect data from various sensors, and transmit the information to a central control system. However, these studies often focus on general security applications and do not specifically address the unique challenges of apartment complexes.

2.3 PRIOR WORK ON REGULATED PATROLLING ROBOTS FOR INTRUSION DETECTION

While the use of robots for security purposes has been explored, there is limited prior work specifically focused on regulated patrolling robots for intrusion detection in apartment complexes. Regulated patrolling robots offer the advantage of controlled and consistent monitoring of the complex, ensuring comprehensive coverage and minimizing blind spots. Some research studies have investigated the use of robots for security patrolling in specific environments, such as warehouses or industrial settings. These studies often employ algorithms for navigation, obstacle avoidance, and path planning to enable effective patrolling. However, the application of regulated patrolling robots specifically tailored for apartment complexes is an area that requires further exploration. The proposed intrusion detection system using regulated patrolling robots for apartments bridges the gap between existing approaches and the specific security needs of apartment complexes. By integrating sensors, communication systems, and navigation capabilities, this system aims to provide comprehensive coverage, real-time monitoring, and prompt response to potential security threats. The literature review highlights the need for a tailored approach to intrusion detection in apartment complexes and sets the foundation for the proposed research.

CHAPTER 3
SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Current intrusion detection systems predominantly utilize stationary cameras and alarms for identifying potential threats. While these systems can detect intruders, they lack the capability to actively respond to security breaches in real-time. The proposed system addresses these limitations by incorporating dynamic patrolling robots with integrated cameras, enabling immediate identification and response to unauthorized access.

DISADVANTAGES:

- Limited Coverage
- Power Consumption
- Static Monitoring

3.2 PROPOSED SYSTEM

The proposed system integrates deep learning algorithms for person identification with regulated patrolling robots equipped with cameras, microcontrollers, buzzers, DC motors, and motor drives. When an unknown person is detected, a signal is sent to the microcontroller, activating the buzzer to alert residents. Simultaneously, the regulated patrolling robots are deployed, leveraging their mobility and surveillance capabilities to investigate and address the potential security threat. This holistic approach ensures a proactive and comprehensive intrusion detection system for apartments.

ADVANTAGES:

- Real-Time Response
- Autonomous Operation
- Improved Identification Accuracy
- Reduced Human Risk
- Interactive Alerting
- Women and Children Safety
- Immediate Alert Notification to the Residents

3.3 SCOPE OF THE PROJECT

"Intrusion Detection System Using Regulated Patrolling Robots for Apartments" encompasses the development and implementation of a comprehensive security solution tailored specifically for residential apartments. The project aims to address the growing need for enhanced surveillance and prompt response to potential security breaches in apartment complexes.

The primary focus is on the utilization of regulated patrolling robots equipped with sensors and cameras to monitor the surroundings of the apartments. These robots will follow predefined paths and continuously scan the area, capturing images and detecting any signs of intrusion or unauthorized access.

The scope of the project includes the design and integration of the hardware components, such as the patrolling robots, sensors, and cameras, along with the necessary software systems. The system will employ computer vision techniques

for image processing and analysis to identify any anomalies or suspicious activities.

Furthermore, the project will involve the development of an intelligent algorithm that can accurately detect and classify intrusion events based on the processed images. The system will generate real-time alerts, triggering immediate notifications to the security personnel and providing live video streaming to aid in swift and effective response.

The scope of the project is to create a robust and efficient intrusion detection system using regulated patrolling robots that can significantly enhance the security measures in residential apartments, mitigating the risks of unauthorized access and improving the overall safety of residents.

CHAPTER 4
SYSTEM DESIGN

4.1 UML DIAGRAMS

Unified Modeling Language (UML) is a general-purpose modelling language. The main aim of UML is to define a standard way to visualize the way a system has been designed. It is quite similar to blueprints used in other fields of engineering.

4.1.1 USE CASE DIAGRAM

A use case describes how a user uses a system to accomplish a particular goal. A use case diagram consists of the system, the related use cases and actors and relates these to each other to visualize the system, actors and use case. Use cases help ensure that the correct system is developed by capturing the requirements from the user's point of view.

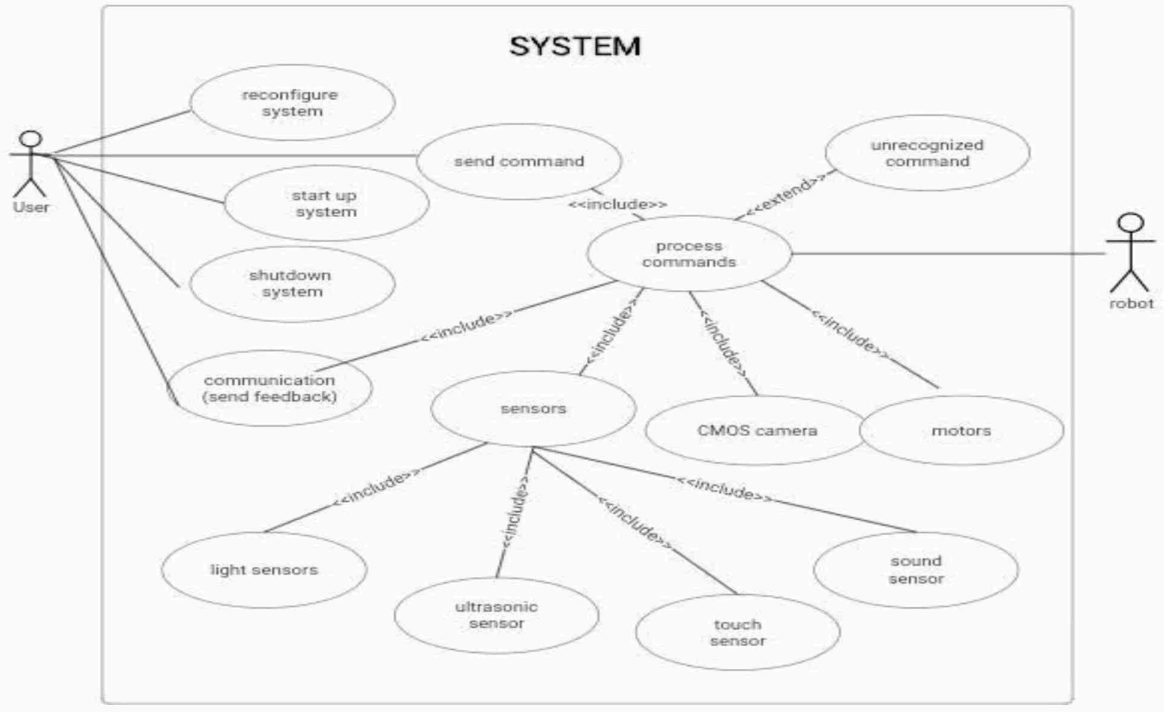

Figure 4.1.1 Use Case Diagram Of IDS

4.1.2 CLASS DIAGRAM

A class diagram describes the structure of an object-oriented system by showing the classes in that system and the relationships between the classes. A class diagram also shows constraints, and attributes of classes. Class is represented as rectangular box showing class name, attributes, and operations. An attribute is a logical data value of an object. Attributes of a classifier also called structural properties in the UML.

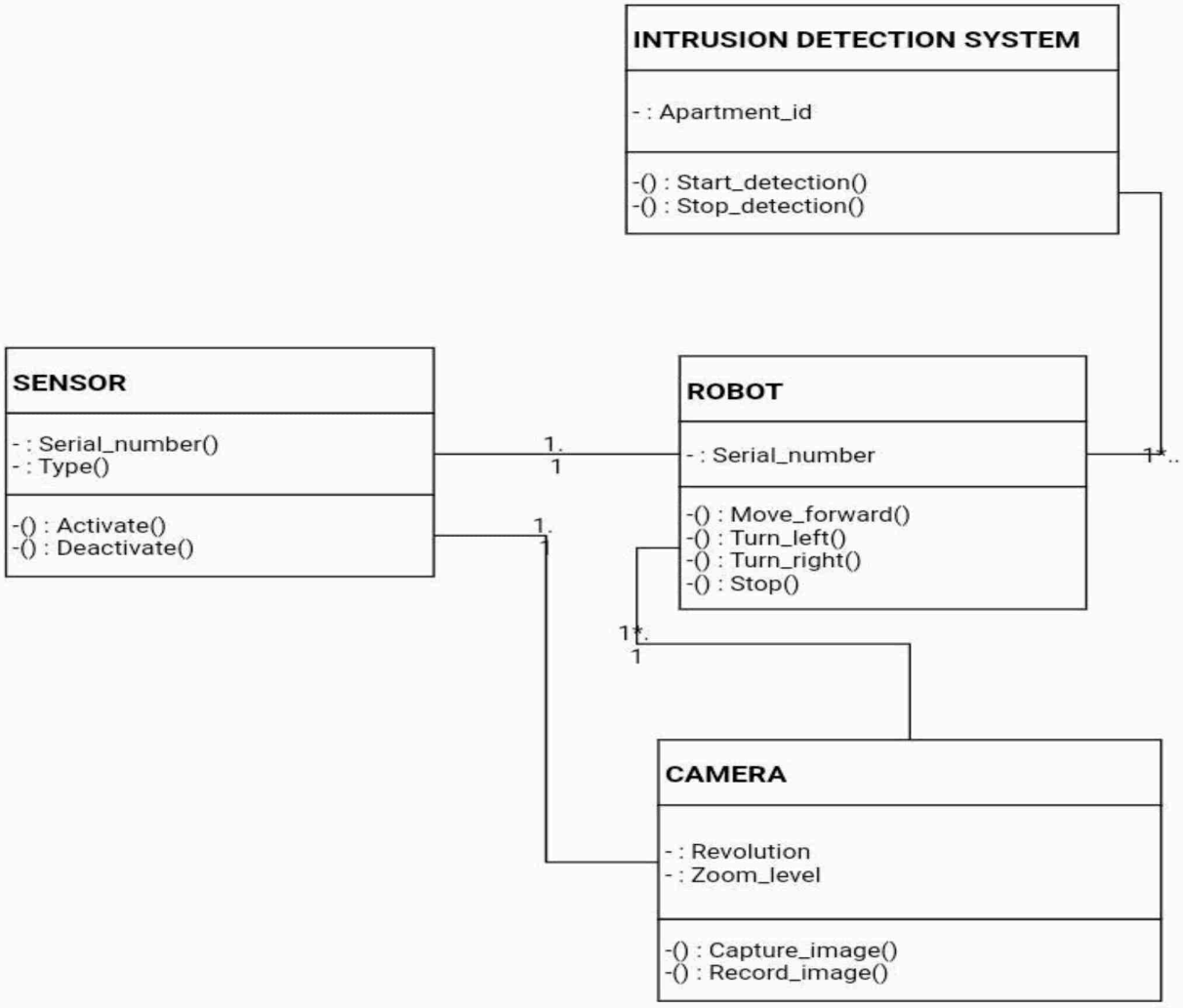

Figure 4.1.2. Class Diagram Of IDS

4.1.3 SEQUENCE DIAGRAM

A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during the interaction. A sequence diagram shows the sequence of messages passed between objects. Sequence diagrams can also show the control structures between objects.

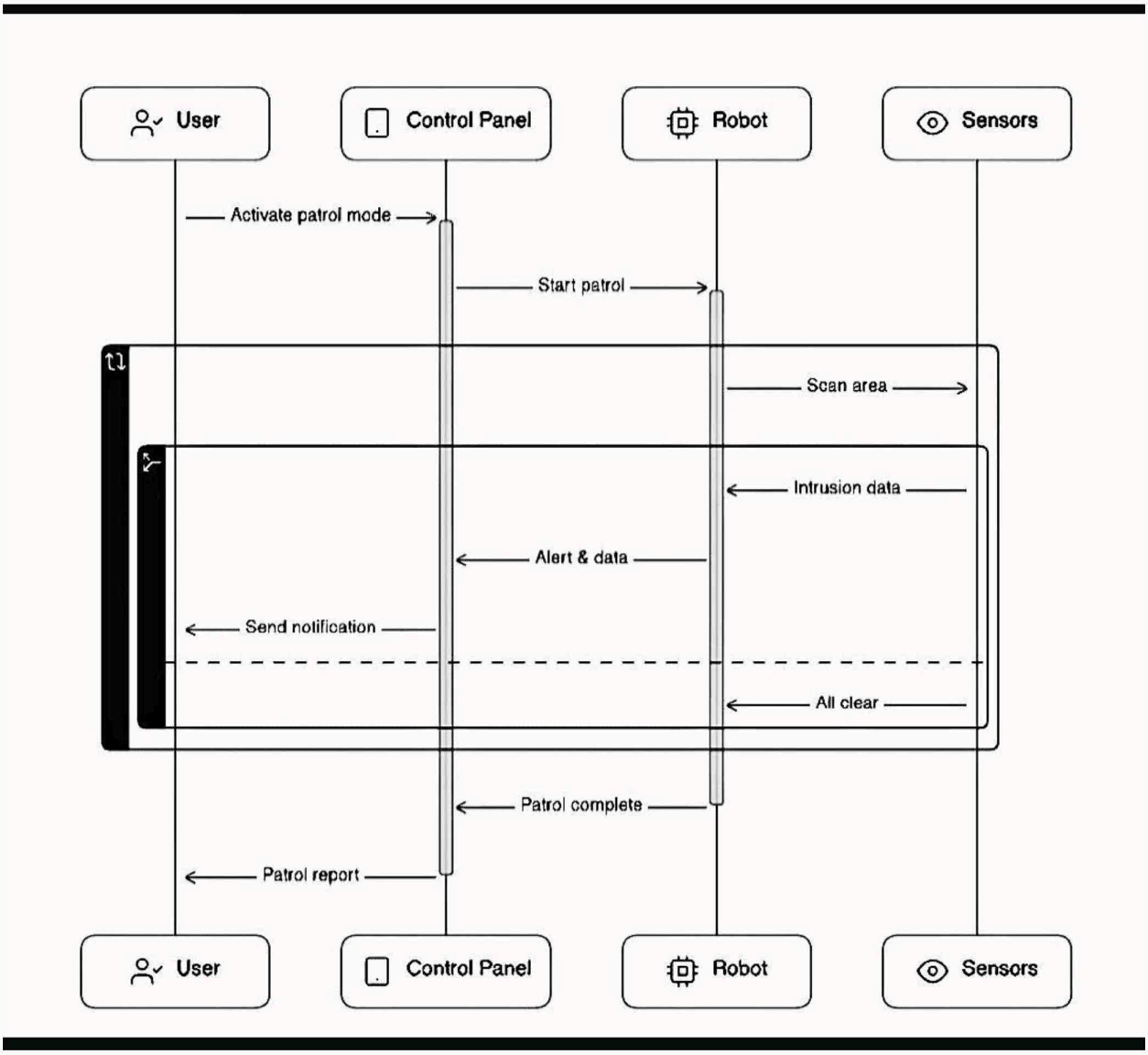

Figure 4.1.3. Sequence Diagram Of IDS

4.2 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. It can be used for the visualization of data processing (structured design). Data flow diagrams are also known as bubble charts. DFD is a designing tool used in the top-down approach to Systems Design. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond. DFD Level 0 is also called a Context Diagram.

Level 0:

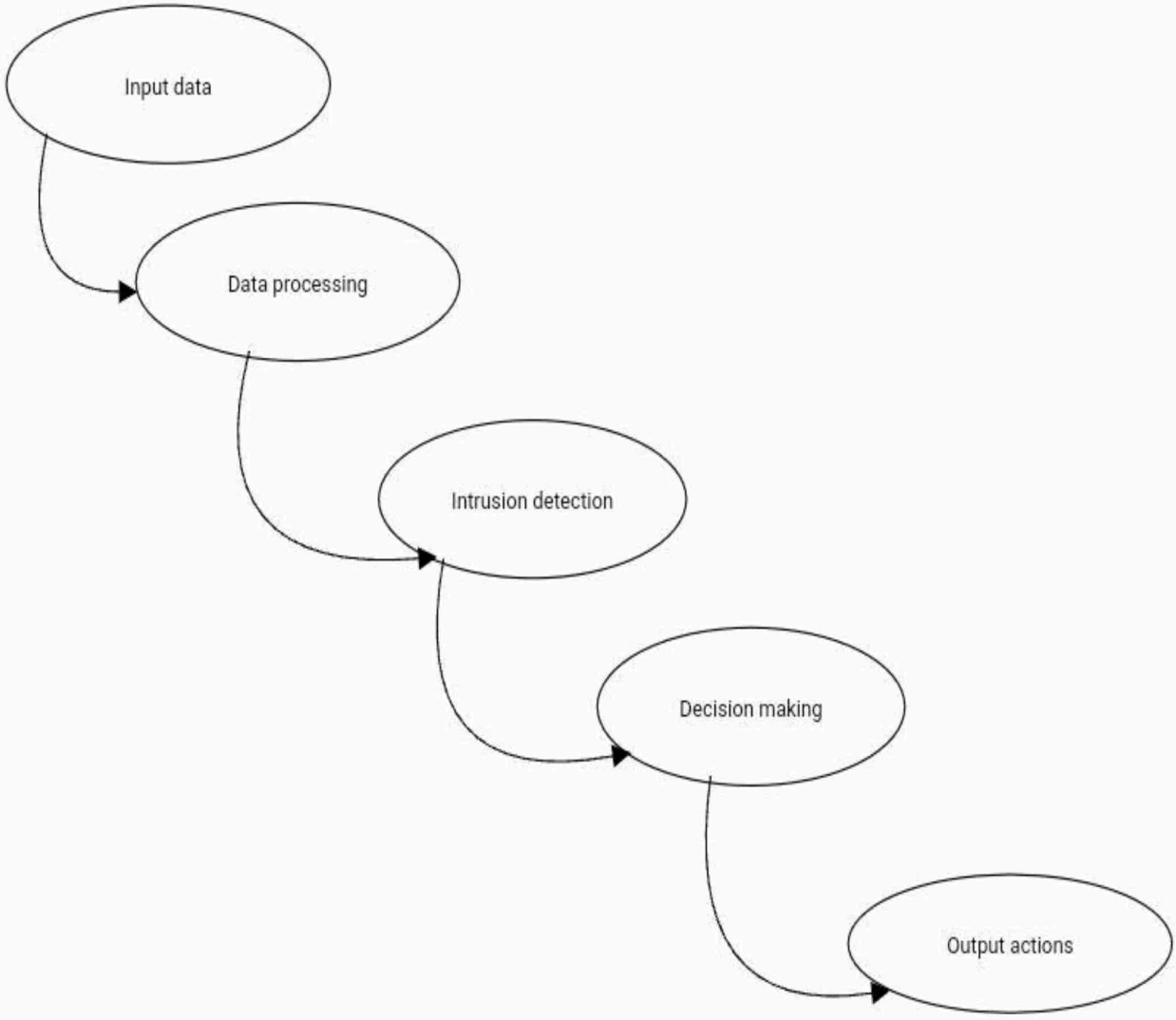

Figure 4.2.1. Level 0 DFD Diagram Of IDS

Level 1:

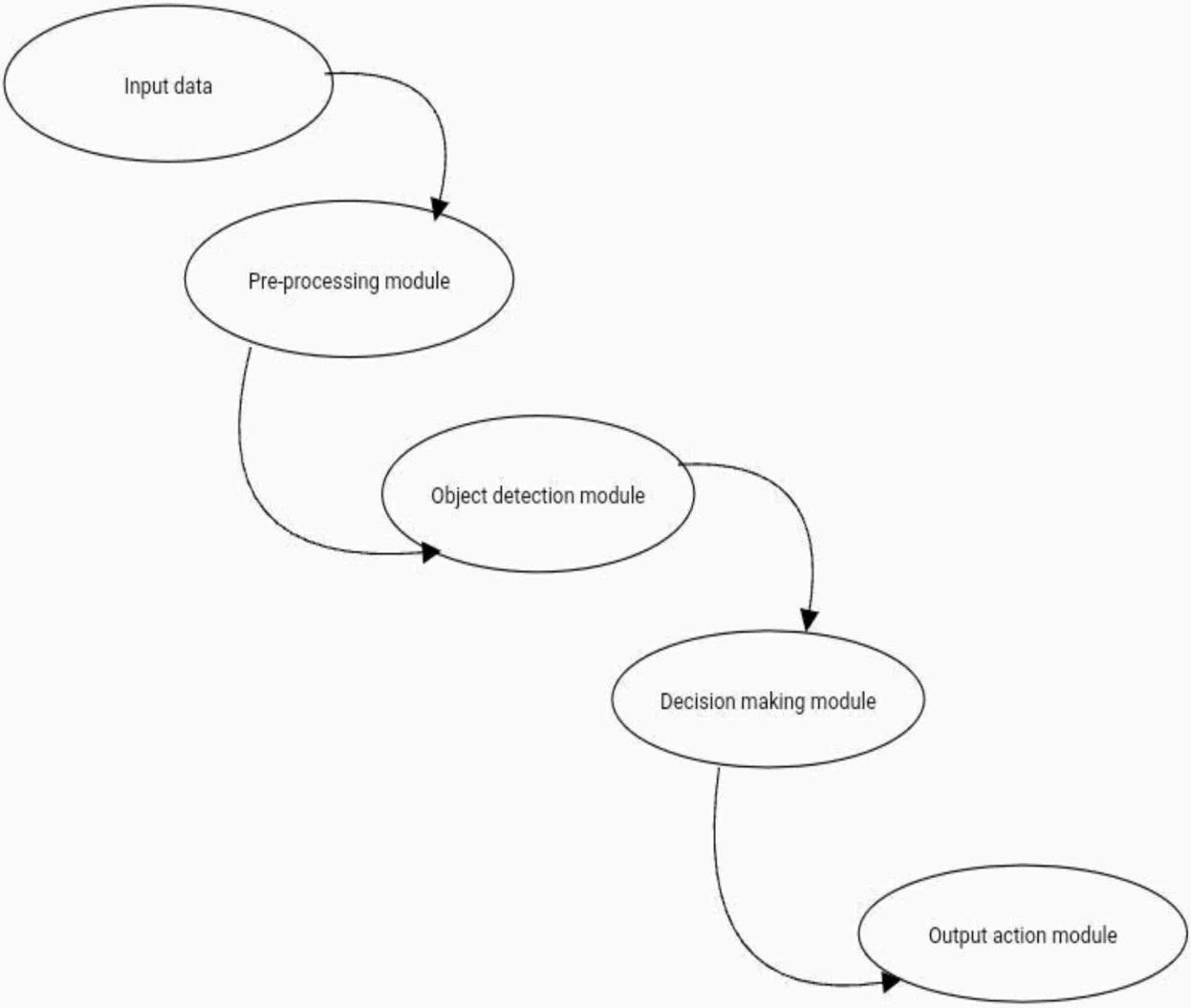

Figure 4.2.2. Level 1 DFD Diagram Of IDS

CHAPTER 5
REQUIREMENT SPECIFICATIONS

5.1 FUNCTIONAL REQUIREMENTS

Functional requirements for an Intrusion Detection System (IDS) using regulated patrolling robots for apartments can include the following.

ROBOT PATROLLING:

The system should support autonomous patrolling of regulated robots in designated areas of the apartment complex. The robots should follow predefined paths, monitor surroundings, and detect any potential security threats.

SENSOR INTEGRATION:

The regulated patrolling robots should be equipped with sensors such as cameras, motion detectors, smoke detectors and buzzers. The system should integrate and utilize data from these sensors to detect and analyse potential intrusions.

INTRUSION DETECTION:

The IDS should employ algorithms and techniques to analyse sensor data and detect intrusions or suspicious activities within the apartment complex. It should be able to identify unauthorized access attempts, detect abnormal behaviour, and raise alerts when security breaches occur.

ALERT GENERATION AND NOTIFICATION:

The system should generate timely and accurate alerts when an intrusion is detected. It should notify security personnel, property managers, or residents through various communication channels such as mobile apps, email, or SMS.

The alerts should include relevant information about the intrusion, such as the location and nature of the threat.

5.2 HARDWARE REQUIREMENTS

MICRO CONTROLLER

Figure 5.2.1 Micro Controller

The Micro Controller is powered by a 1.2 GHz quad-core ARM Cortex-A53 CPU, providing improved performance compared to its predecessors. It has 1GB of LPDDR2 RAM, allowing for smooth multitasking and efficient operation of applications. The board includes built-in Wi-Fi 802.11n and Bluetooth 4.2, eliminating the need for additional adapters and making it easy to connect to wireless networks and devices. It includes four USB 2.0 ports for connecting peripherals such as keyboards, mice, and external hard drives, CSI camera port for connecting a camera module

USB CAMERA FIGURE

Figure 5.2.2. USB Camera

The figure showcases an external camera to incorporate visual input into a project, the easiest approach is to combine a Raspberry Pi with a supported camera module.

FIRE SENSOR

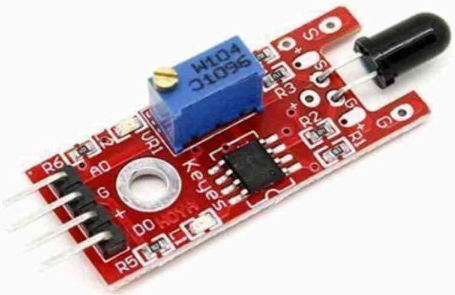

Figure 5.2.3 Fire Sensor

A fire sensor, also known as a fire detector or smoke sensor, is a crucial component in fire detection and alarm systems. Flame detectors are designed to detect the presence of flames by sensing the characteristic Infrared (IR) or Ultraviolet (UV) radiation emitted by the flames. When the sensor detects the characteristic radiation, it triggers the alarm.

MOTOR DRIVER

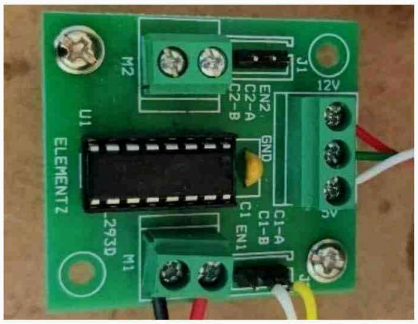

Figure 5.2.4 Motor Driver

A motor driver, also known as a motor controller, is an electronic device or circuit that controls the speed, direction, and operation of an electric motor. It acts as an interface between a microcontroller or other control signals and the motor, providing the necessary power and control signals to drive the motor efficiently and safely.

5.3 SOFTWARE REQUIREMENTS

PYTHON PROGRAMMING LANGUAGE

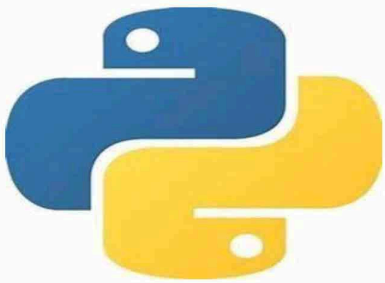

Figure 5.3.1 Python Programming Language

Python is a high-level programming language with an interpreted execution model and object-oriented paradigm. It boasts dynamic semantics,

making it an appealing choice for Rapid Application Development, scripting, and integration purposes.

OpenCV

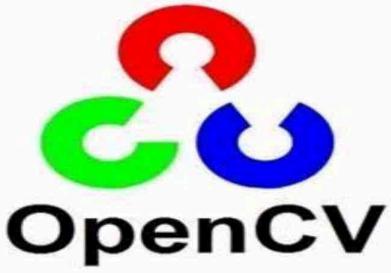

Figure 5.3.2. Open CV

Open CV is an open-source library of computer vision and image processing functions widely used in various applications, including robotics, augmented reality, object detection, and video analysis. It provides a comprehensive set of tools and algorithms for manipulating and analyzing images and videos. The library provides a wide range of functionalities, including image and video I/O, image processing operations (e.g., filtering, transformations, and morphological operations), feature detection and extraction, object tracking, camera calibration, and machine learning algorithms for computer vision tasks. One of the notable features of OpenCV is its ability to handle real-time computer vision applications, leveraging the power of GPUs for high- performance processing.

CHAPTER 6

SYSTEM ARCHITECTURE

6.1 SOFTWARE BLOCK DIAGRAM OF IDS

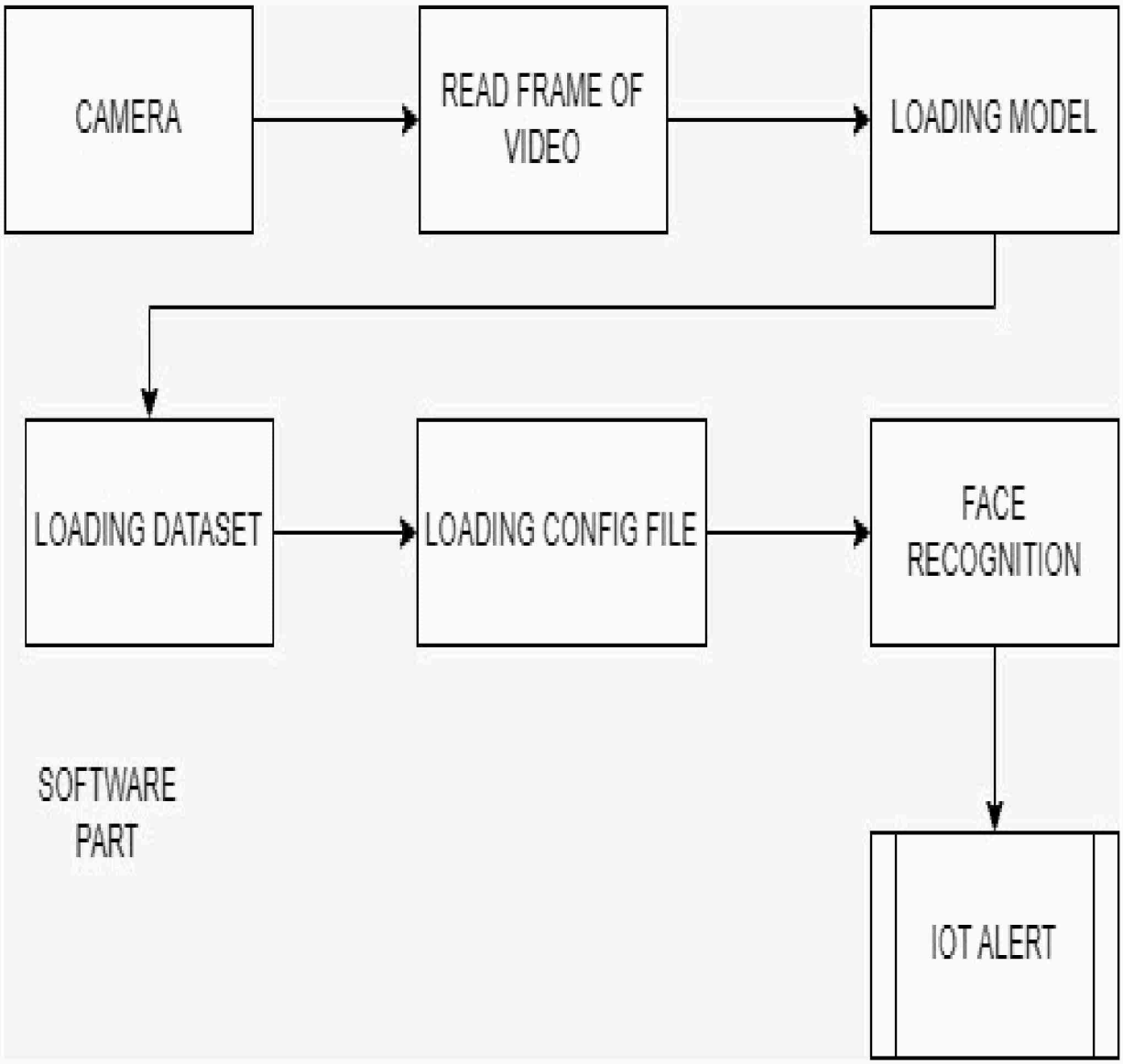

Figure 6.1.1 Software Block Diagram Of IDS

6.2 HARDWARE BLOCK DIAGRAM OF IDS

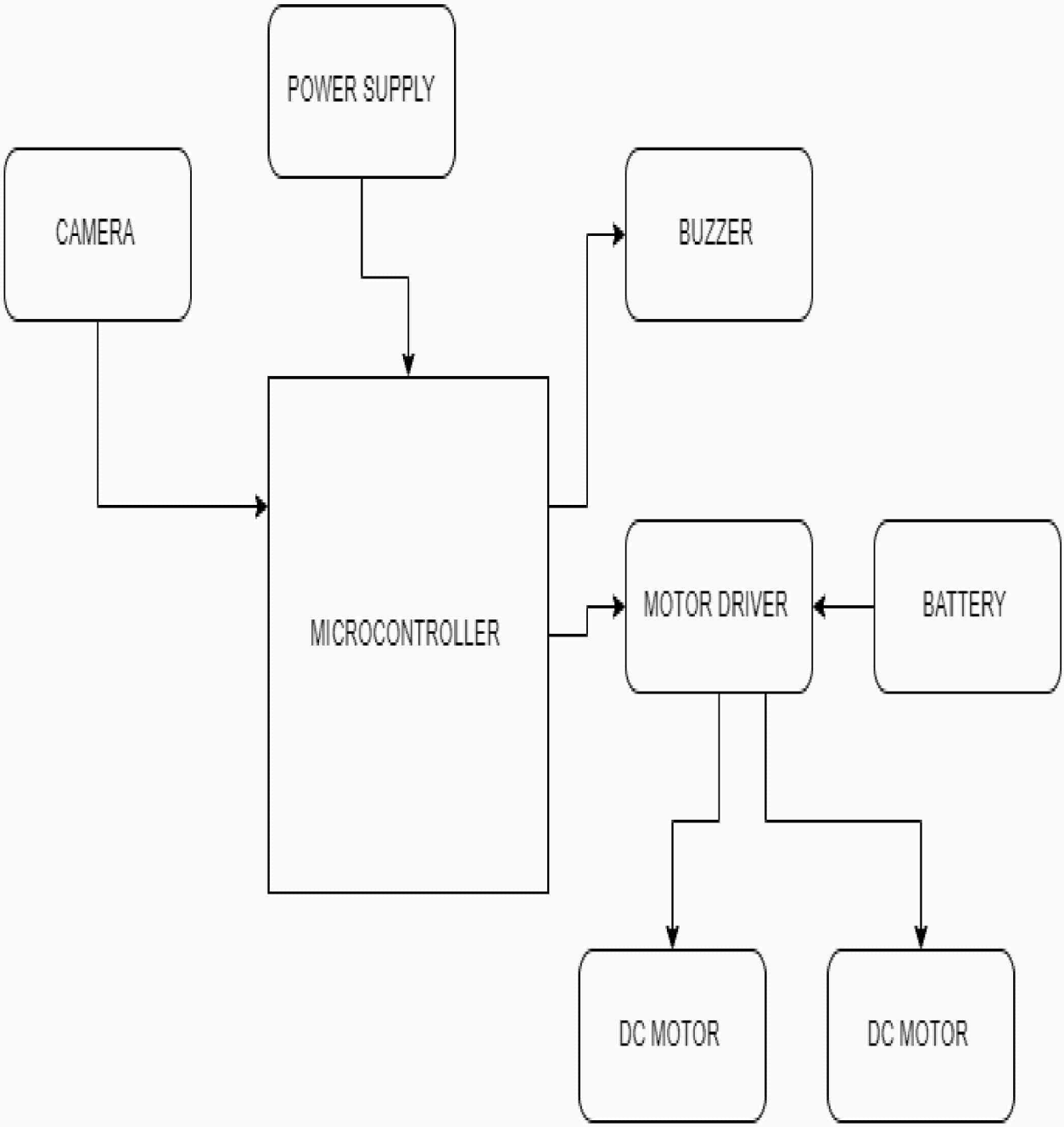

Figure 6.2.1. Hardware Block Diagram Of IDS

6.3 BLOCK DIAGRAM EXPLANATION OF IDS

The camera is the input device responsible for capturing images or video footage of the monitored area. It serves the purpose of detecting both known and unknown persons using deep learning algorithms for person identification. The microcontroller acts as the central processing unit of the system, receiving input from the camera and making decisions based on the detected individuals. It processes the incoming video data, executes the deep learning algorithms for person identification, and determines whether the detected person is known or unknown. The motor driver is a crucial component for controlling the movement of the regulated patrolling robots. It interprets commands from the microcontroller and regulates the power supplied to the motors of the robots, enabling precise control over their movement. The buzzer serves as an alerting device, generating audible signals to notify residents or security personnel of an intrusion. It is activated by the microcontroller when an unknown person is detected, contributing to a multi-modal alerting system.

CHAPTER 7
METHODOLOGY

7.1 BLOCK DIAGRAM OF IDS

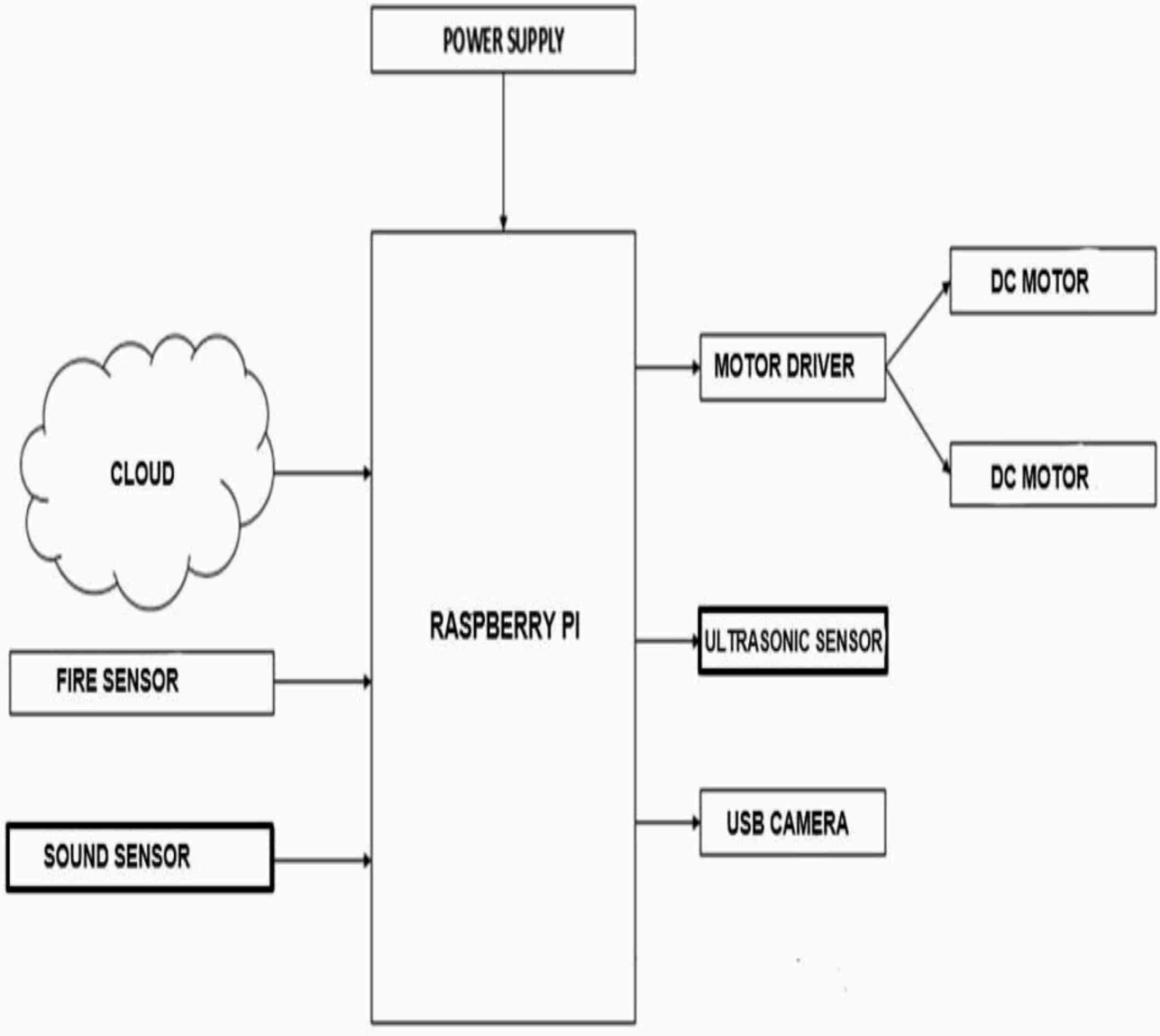

Figure 7.1.1. Block Diagram of the IDS Module

We have developed a surveillance robot specifically designed to patrol apartment premises continuously day and night. The robot follows a predefined

route for navigation. Equipped with an ultrasonic sensor and a movable neck mounted USB camera, our robot can capture images from various angles. To detect living objects or human faces, the robot is equipped with a sound sensor. The ultrasonic sensor enables obstacle detection and avoidance on flat surfaces. In case of any detected sound or movement using the sound sensors, and if any obstacles are detected using an ultrasonic sensor the robot responds accordingly. Additionally, the robot utilizes its camera to capture pictures and store them in a database for further analysis and reference.

The captured images will be sent for Image Processing. Which uses Haar Cascade Algorithm. The total system is divided into 3 sections:

Database creation: Using the camera the pictures of the people residing in apartments is collected. Providing the user ID to each person's image. Convert the image into grayscale, and detect the face. Store it in the database and later used it for comparison.

Training: Initialize LBPH face recognizer. Get faces and IDs from the database folder to train the LBPH recognizer. Save the trained data as XML or YML files.

Testing: Load Haar classifier, LBPH face recognizer, and trained data from XML or YML files. Capture the image from the camera. Convert it into grayscale. Detect the face in it. Predict the face using the above recognizer. When there is any mismatch in the face recognition the algorithm will alert that unidentified user entered our premises. when the sound or movement or obstacle is detected by the robot it will run this image processing and sends an alert notification that an unidentified person is on our premises if the face is mismatched.

7.2 IMAGE PROCESSING

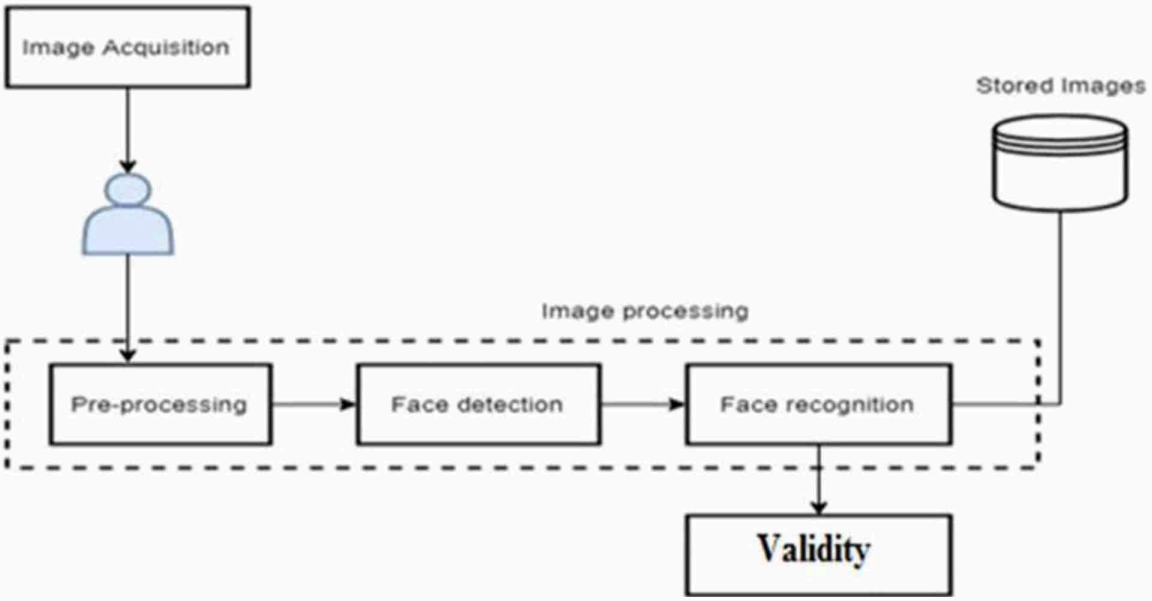

Figure 7.2.1. Face Identification Methodology

7.2.1 IMAGE PRE-PROCESSING

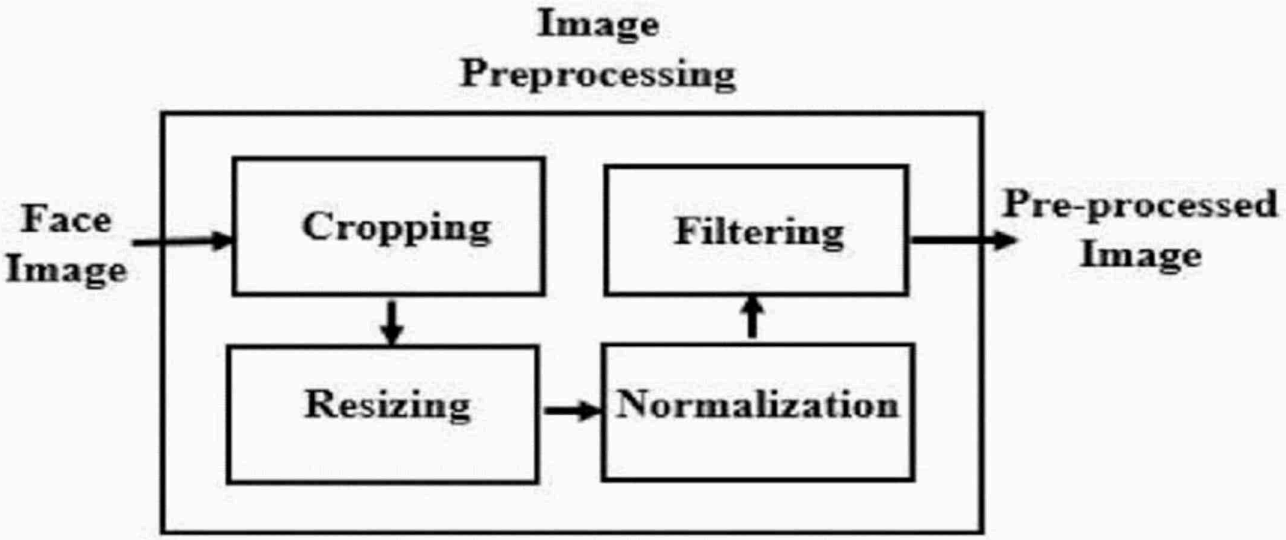

Figure 7.2.1.1. Image Pre - Processing

Image preprocessing refers to a series of techniques and operations applied to an image before it undergoes further analysis or processing. The purpose of image preprocessing is to enhance the quality, improve interpretability, or extract relevant information from the image. Common Image

Preprocessing Techniques Include:

Image resizing: Changing the dimensions of an image to a desired size, which can be useful for standardization or compatibility purposes.

Image cropping: Removing unwanted portions of an image to focus on the region of interest or remove irrelevant background**.**

Image denoising: Reducing noise or unwanted disturbances present in the image, which can be caused by factors such as sensor limitations or environmental conditions.

Image enhancement: Techniques like contrast adjustment, brightness correction, or histogram equalization to improve the visual quality and emphasize important image details.

Image normalization: Adjusting the pixel values of an image to a standardized scale or range to facilitate further processing or comparison.

Image smoothing: Reducing high-frequency noise or sharp edges in the image using filters or smoothing algorithms, such as Gaussian or median filtering.

Image sharpening: Enhancing the sharpness or edge definition of an image to improve visual clarity or facilitate feature extraction.

Image color correction: Adjusting color balance, saturation, or hue to improve color fidelity or correct for lighting variations.

Image registration: Aligning multiple images to a common coordinate system for comparison, fusion, or analysis.

Image segmentation: Dividing an image into meaningful regions or objects based on characteristics such as color, texture, or intensity for further analysis or object

recognition. These preprocessing techniques help improve the quality, usability, and interpretability of images, ensuring that subsequent analysis or processing tasks can be performed more effectively

7.3 FACE DETECTION

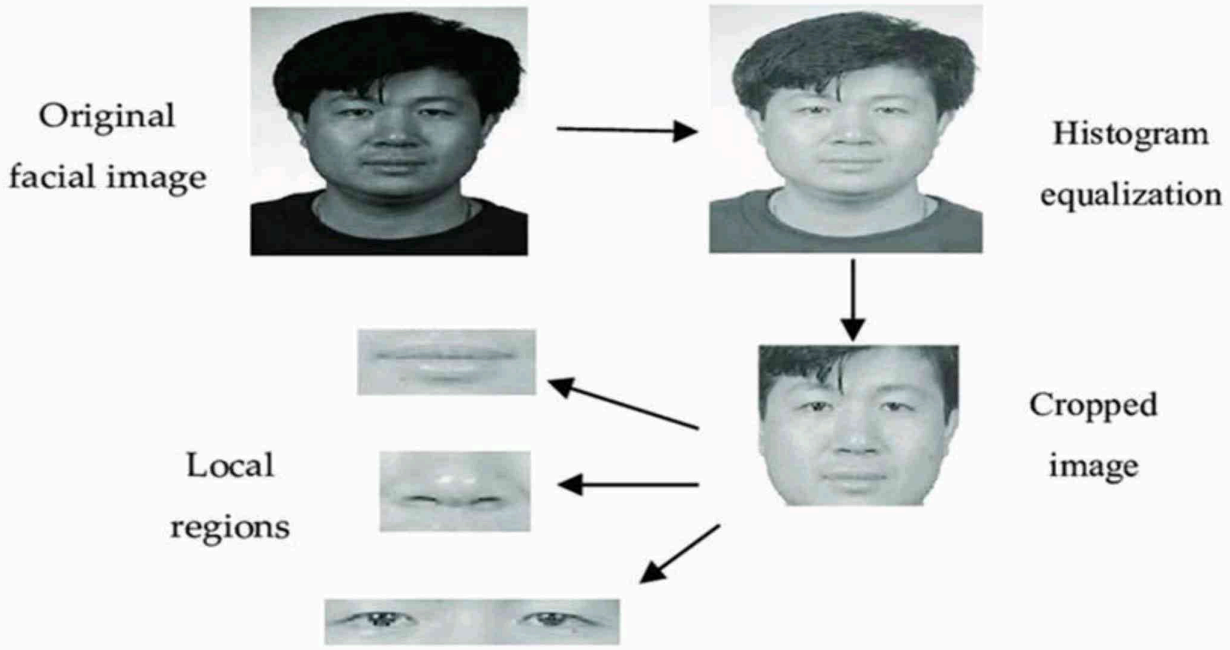

Figure 7.3.1. Face Detection

Face detection refers to the process of identifying and locating human faces within an image or a video frame. It is a fundamental task in computer vision and has numerous applications, including facial recognition, emotion analysis, biometrics, surveillance, and augmented reality. The objective of face detection is to detect human faces in an image or video. The process typically involves the following steps: Input image or video frame: The system receives an input image or video frame that contains one or more human faces. Preprocessing: Preprocessing techniques may be applied to enhance the image

quality, such as resizing, normalization, or noise reduction. Feature extraction: Various visual features are extracted from the image, such as color, texture, or shape, to identify potential face regions. Face region localization: Once faces are detected, bounding boxes or contours are typically drawn around the detected face regions to indicate their positions within the image. Post-processing: Additional post- processing steps may be applied to refine the face detection results, remove false positives, or improve the accuracy of the detections.

7.4 FACE RECOGNITION

7.4.1 HAAR CASCADE CLASSIFIER

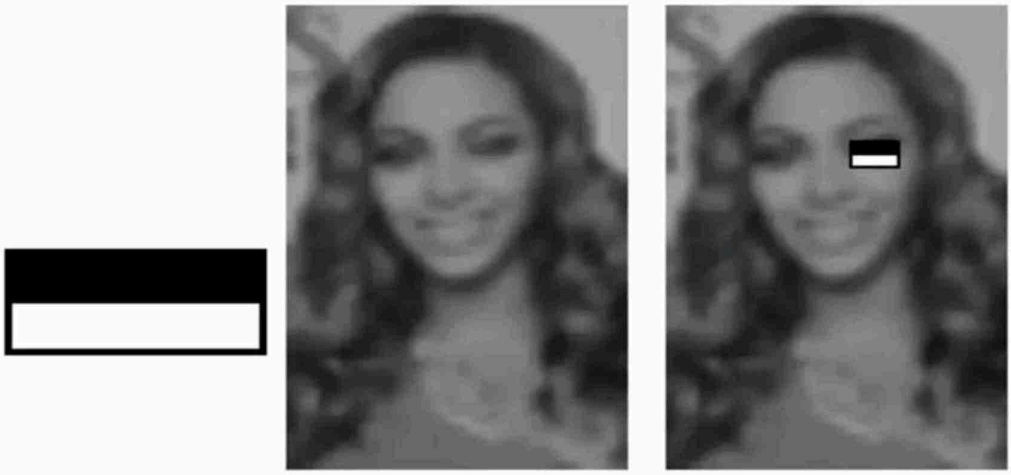

Figure 7.4.1.1. Haar-Feature Extraction

The Haar Cascade classifier utilizes the Haar Wavelet technique to analyze pixels within an image, dividing them into square regions based on their function. This approach incorporates the concept of "integral images" to calculate the detected features. Haar Cascades use the Ada-boost learning algorithm which selects a small number of important features from a large set to give an efficient

result of classifiers then use cascading techniques to detect a face in an image. Here are some Haar-Feature

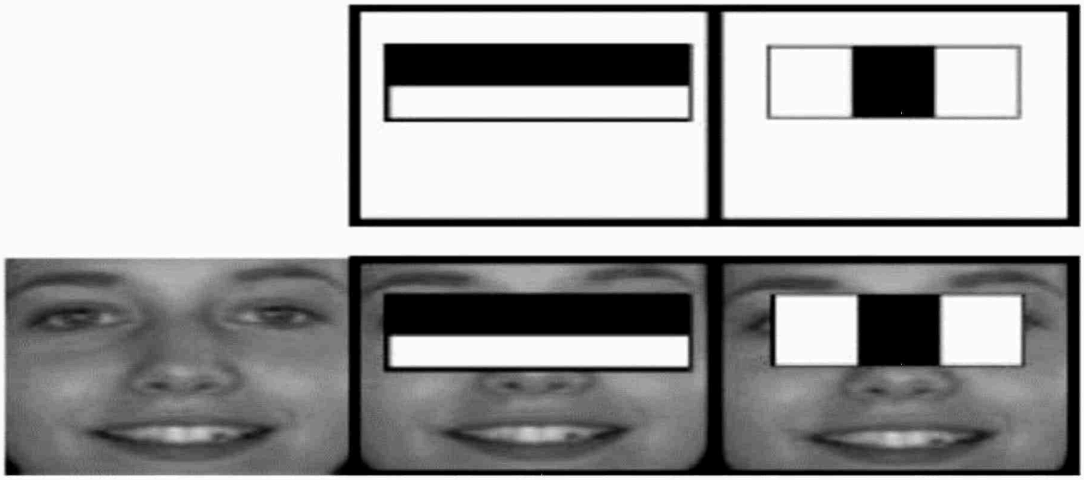

Figure 7.4.1.2. represents the Haar-like feature.

It consists of an edge feature and a line feature. Within the grayscale image, the presence of a white bar indicates the pixels that exhibit proximity to the light source. This assessment is derived through the process of Haar value calculation.: Pixel value = (Sum of the Dark pixels / Number of Dark pixels) − (Sum of the Light pixels / Number of Light pixels) Haar cascade Classifier is an object detection algorithm. To facilitate object detection and identification, the image will undergo feature extraction, Using the above equation Haar pixel value can be calculated.

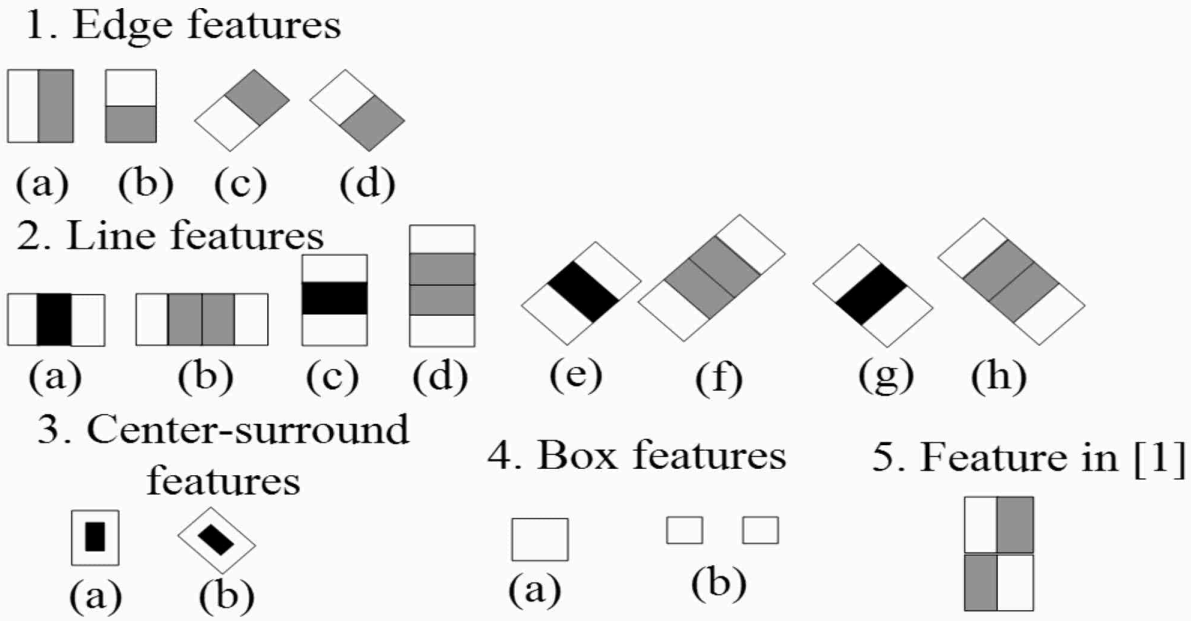

Figure 7.4.1.3 Different Haar-Features

While Haar cascades can be used as a preliminary step for face recognition, it involves a separate process that uses different algorithms and techniques. Face recognition typically involves the following steps: Face detection: Initially, a face detection algorithm like the Haar cascade is used to locate and extract face regions from an image or video frame. Face alignment: Once the faces are detected, facial landmarks or key points are identified to align the face regions and normalize their orientation and scale. This step ensures consistent feature extraction for accurate recognition. Feature extraction: Various methods can be employed to extract discriminative features from the aligned face regions. Popular techniques include Eigenfaces, Local Binary Patterns (LBP), or deep learning-based approaches like Convolutional Neural Networks (CNNs).

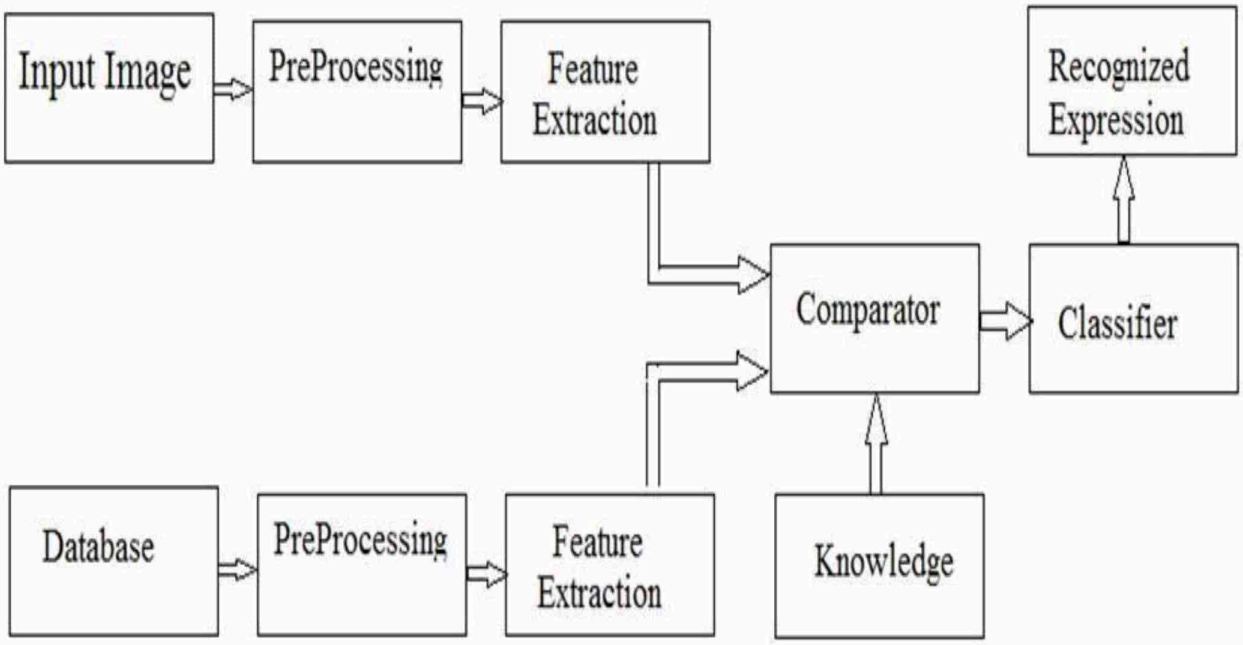

Figure 7.4.1.4. Flow Diagram of Haar Cascade

These features aim to capture unique characteristics of the face that can distinguish individuals. Feature matching and comparison: In the recognition phase, the extracted facial features are compared with a database of known identities. The comparison can be performed using various techniques, such as Euclidean distance, Mahalanobis distance, or similarity measures like cosine similarity or correlation coefficients. Recognition decision: Based on the similarity or distance metrics, a decision is made to determine the identity of the individual. A threshold or classification algorithm may be used to classify the face into a specific identity or determine if it is an unknown face.

CHAPTER 8
RESULTS AND DISCUSSION

8.1 RESULTS AND DISCUSSION

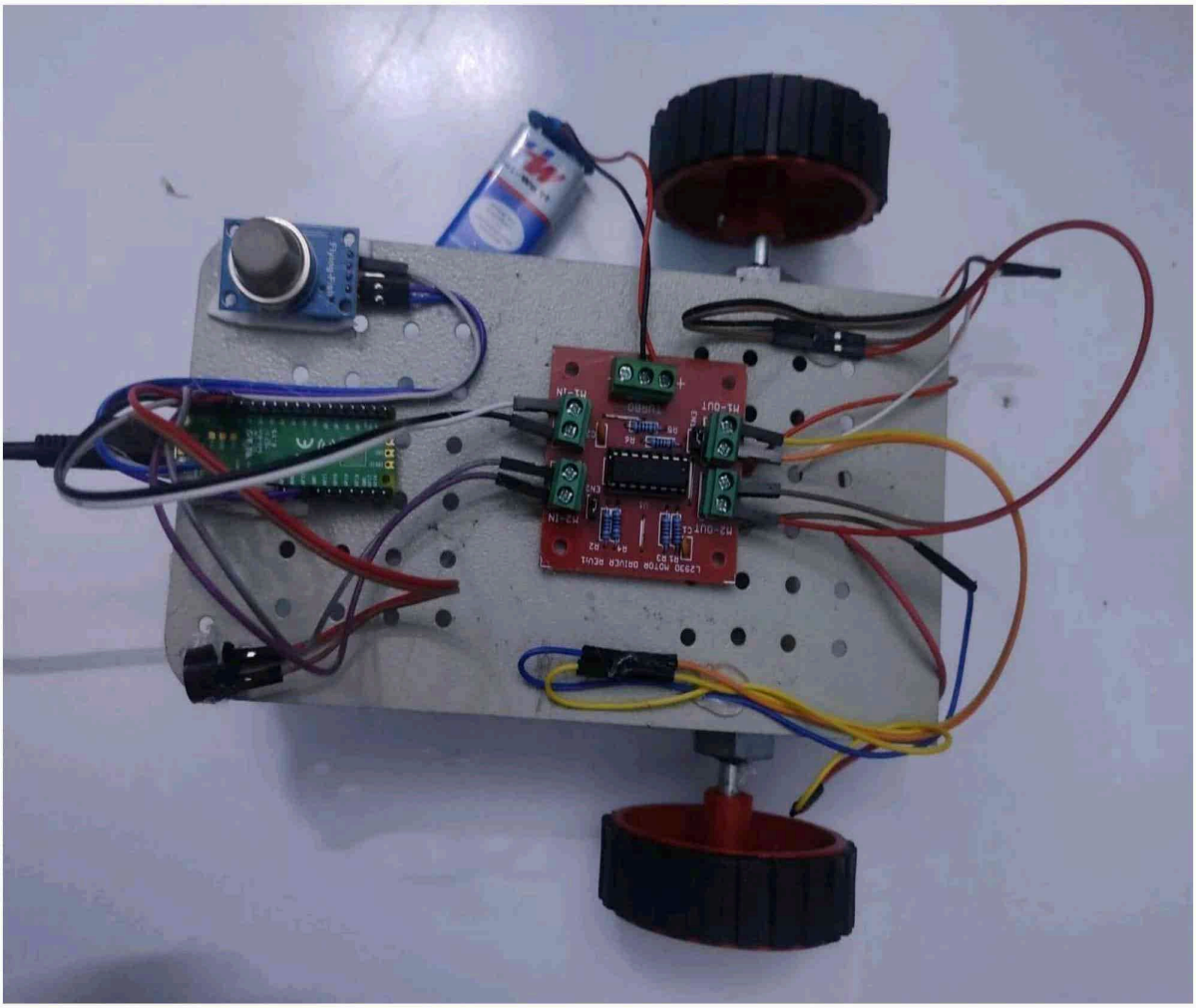

Figure 8.1.1. Completed model picture

When any unknown person is detected in the surrounding premises of the robot then it will be displayed in the Python IDE.

After the detection of unknown person, the robot will capture the situation using the USB camera and detects the person whether the person is matched to

the defined dataset, if not it displays the UNKNOWN status in the Python IDE as shown in figure 8.1.4 below.

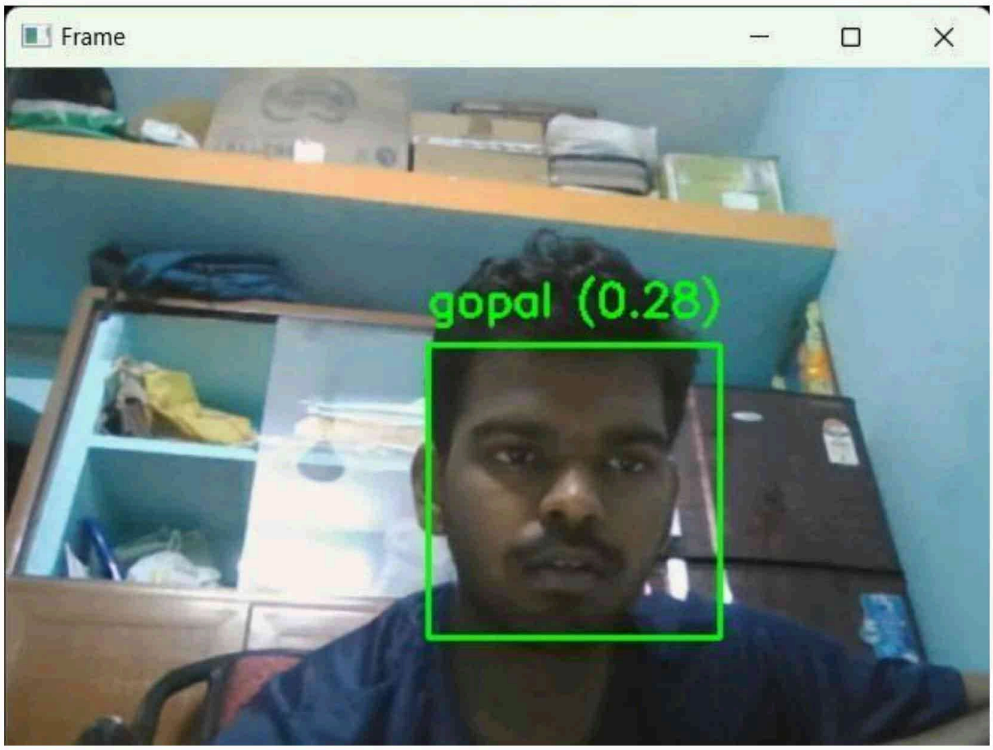

Figure 8.1.2. Status of the known person with name

Figure 8.1.3. Status of the Unknown person

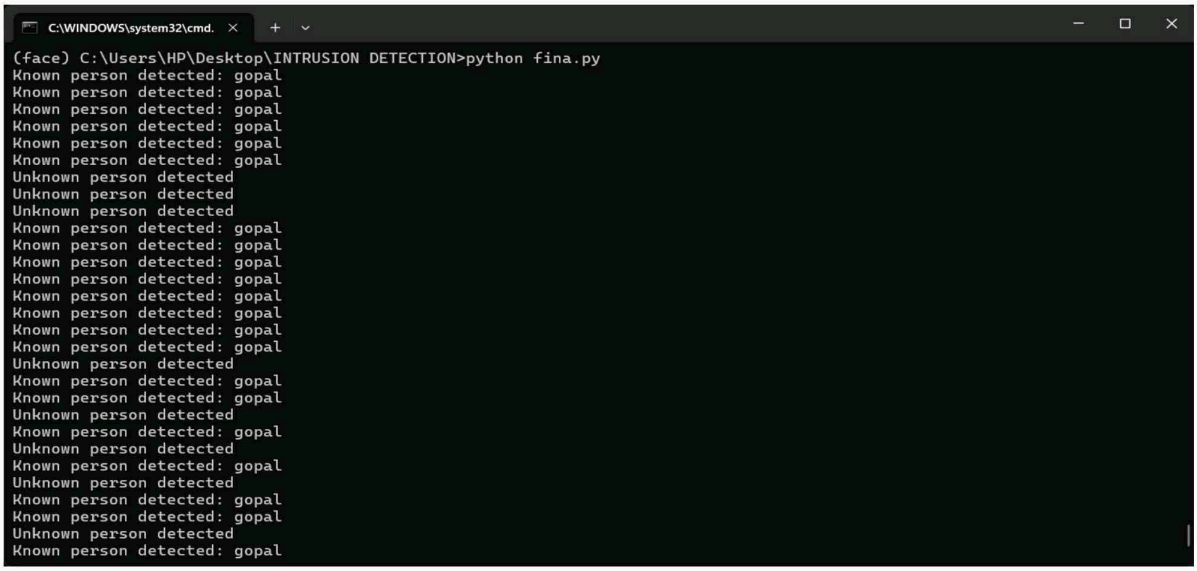

Figure 8.1.4. Status of face recognition

When the patrolling robot detects the unknown person, then it sends an alert message to the authority as shown in the figure 8.1.5.

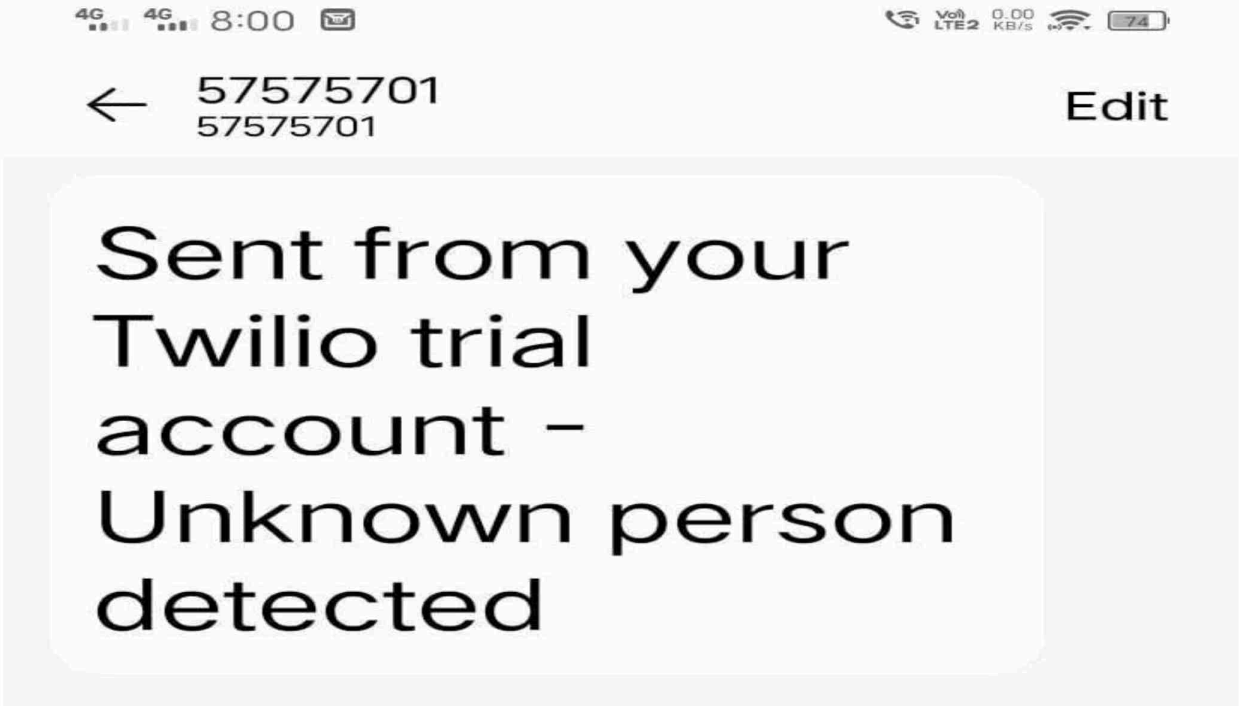

Figure 8.1.5. Alert Message

CHAPTER 9
CONCLUSION

9.1 CONCLUSION

The implementation of an intrusion detection system using regulated patrolling robots for apartments presents a promising solution to enhance security and safety measures. The integration of robotic technology with advanced surveillance anddetection mechanisms offers a proactive and efficient approach to combat potential threats. By deploying regulated patrolling robots equipped with sophisticated sensors and surveillance capabilities, apartments can benefit from continuous and reliable monitoring. These robots can navigate through the premises, detecting any unusual activities or unauthorized access in real- time. The timely detection of intrusions enables prompt response and intervention, minimizing the risk of property damage or harm to residents.

The aim of this project was to design and implement a security robot that is capable of performing security tasks related to real-time monitoring and capturing and alerting the officials. In general, autonomous robots can serve as more reliable and efficient security agents when compared with existing security solutions and humans. As a result, a specific user robot was configured by enabling the sound sensors to detect movement in objects and can detect the persons belonging to the apartment or not. If not it will alert the official by proving the live video streaming.

APPENDIX 1
SOURCE CODE

Program to Create DataSet :

```
import cv2
import os

haar_file = cv2.data.haarcascades + 'haarcascade_frontalface_default.xml'

datasets = 'datasets'  # All the faces data will be present in this folder
sub_data = 'gopal'  # Sub-directory for your data

path = os.path.join(datasets, sub_data)
if not os.path.isdir(path):
    os.mkdir(path)

(width, height) = (130, 100)  # Defining the size of images

face_cascade = cv2.CascadeClassifier(haar_file)
webcam = cv2.VideoCapture(0)  # '0' is used for the default webcam, you can change it if you have other cameras

# The program loops until it has 100 images of the face.
count = 1
while count <= 100:
    (_, frame) = webcam.read()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, 1.3, 4)
    for (x, y, w, h) in faces:
```

```
        cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
        face = gray[y:y + h, x:x + w]
        face_resize = cv2.resize(face, (width, height))
        cv2.imwrite(f'{path}/{count}.png', face_resize)
        count += 1
    cv2.imshow('OpenCV', frame)
    key = cv2.waitKey(10)
    if key == 27:
        break
webcam.release()
cv2.destroyAllWindows()
```

Program for Encoding :

```
from imutils import paths
import face_recognition
import argparse
import pickle
import cv2
import os

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--dataset", required=False, default="../datasets/",
        help="path to input directory of faces + images")
ap.add_argument("-e", "--encodings", required=False,
default="encodings.pickle",
        help="path to serialized db of facial encodings")
ap.add_argument("-d", "--detection-method", type=str, default="hog",
        help="face detection model to use: either `hog` or `cnn`")
```

```
args = vars(ap.parse_args())

# grab the paths to the input images in our dataset
print("[INFO] quantifying faces...")
imagePaths = list(paths.list_images(args["dataset"]))

# initialize the list of known encodings and known names
knownEncodings = []
knownNames = []

# loop over the image paths
for (i, imagePath) in enumerate(imagePaths):
	# extract the person name from the image path
	print("[INFO] processing image {}/{} - {}".format(i + 1,
		len(imagePaths),
		imagePath))
	name = os.path.basename(os.path.dirname(imagePath))

	# load the input image and convert it from RGB (OpenCV ordering)
	# to dlib ordering (RGB)
	image = cv2.imread(imagePath)
	rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

	# detect the (x, y)-coordinates of the bounding boxes
	# corresponding to each face in the input image
	boxes = face_recognition.face_locations(rgb,
		model=args["detection_method"])

	# compute the facial embedding for the face
```

```
encodings = face_recognition.face_encodings(rgb, boxes)

# loop over the encodings
for encoding in encodings:
    # add each encoding + name to our set of known names and
    # encodings
    knownEncodings.append(encoding)
    knownNames.append(name)
```

Program for Face Detection :

```
# Import the necessary packages
import cv2
import argparse
import imutils
import face_recognition
import pickle
import numpy as np
import serial
import time
port=serial.Serial("COM3",9600,timeout=0.1)

# Define the argument parser
ap = argparse.ArgumentParser()
ap.add_argument("-c", "--cascade", type=str, required=False,
default="haarcascade_frontalface_default.xml",
    help="path to where the face cascade resides")
ap.add_argument("-e", "--encodings", type=str, required=False,
default="encodings.pickle",
    help="path to serialized db of facial encodings")
```

```
ap.add_argument("-s", "--source", required=False, default=0,
        help="Use 0 for /dev/video0 or 'http://link.to/stream'")
ap.add_argument("-o", "--output", type=int, required=False, default=1,
        help="Show output")
ap.add_argument("-t", "--tolerance", type=float, required=False, default=0.4,
        help="How much distance between faces to consider it a match. Lower
is more strict")
args = vars(ap.parse_args())

# Load the facial encodings and the face detector
data = pickle.loads(open(args["encodings"], "rb").read())
detector = cv2.CascadeClassifier(args["cascade"])

# Open the video source
vs = cv2.VideoCapture(args["source"])

tolerance = float(args["tolerance"])

while True:
    ret, frame = vs.read()
    frame = imutils.resize(frame, width=500)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    # Detect faces using the cascade classifier
    rects = detector.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(30, 30), flags=cv2.CASCADE_SCALE_IMAGE)
    boxes = [(y, x + w, y + h, x) for (x, y, w, h) in rects]
    encodings = face_recognition.face_encodings(rgb, boxes)
```

```
    names = []  # Initialize names list

    for encoding in encodings:
        distances = face_recognition.face_distance(data["encodings"], encoding)
        minDistance = 1.0
        if len(distances) > 0:
            minDistance = min(distances)
        if minDistance < tolerance:
            idx = np.where(distances == minDistance)[0][0]
            name = data["names"][idx]
            print("Known person detected: " + name)
            port.write(str.encode("A"))
            time.sleep(2)
        else:
            name = "unknown person"
            print("Unknown person detected")
            port.write(str.encode("B"))
            time.sleep(2)
        names.append(name)  # Append the detected name to the names list

    # Loop through boxes and names to draw rectangles and labels
    for ((top, right, bottom, left), name) in zip(boxes, names):
        if name != "unknown person":
            cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)  # Display
a green boundary for known persons
        y = top - 15 if top - 15 > 15 else top + 15
        txt = name + " (" + "{:.2f}".format(minDistance) + ")"
        cv2.putText(frame, txt, (left, y), cv2.FONT_HERSHEY_SIMPLEX, 0.75,
(0, 255, 0), 2)
```

```
    if args["output"] == 1:
        cv2.imshow("Frame", frame)

    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
        break
cv2.destroyAllWindows()
vs.release()
```

IDS

ORIGINALITY REPORT

13%	11%	6%	9%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijsart.com Internet Source	3%
2	journalstd.com Internet Source	2%
3	Submitted to Bilkent University Student Paper	<1%
4	www.coursehero.com Internet Source	<1%
5	Submitted to University of Central England in Birmingham Student Paper	<1%
6	Submitted to Napier University Student Paper	<1%
7	Houda Meddeb, Zouhaira Abdellaoui, Firas Houaidi. "Development of surveillance robot based on face recognition using Raspberry-PI and IOT", Microprocessors and Microsystems, 2022 Publication	<1%

8	pure.unileoben.ac.at Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		

REFERENCES

[1] S R Madkar, Vipul Mehta, Nitin Bhuwania, Maitri Parida, International Journal of Advanced Research in Computer Science and Software Engineering, Robot Controlled Car Using Wi-Fi Module, Volume 6, Issue 5, May 2016.

[2] Chinmay Kulkarni, Suhas Grama, Pramod Gubbi Suresh, Chaitanya Krishna, Joseph Antony, First International Conference on Systems Informatics, Modelling and Simulation, Surveillance Robot Using Arduino Microcontroller, Android APIs and the Internet, 2014 IEEE.

[3] Minni Mohan And Siddharth Shelly, International Journal on Cybernetics & Informatics (IJCI),Border Security Robot Vol. 5, No. 2, April 2016.

[4] Android Developers Site [Online]. Available: http://developer.android.com/index.html, 2012

[5] Singoee Sylvestre Sheshai, f17/1454/2011, university of Nairobi department of electrical and information engineering, raspberry pi based security system, Prj index 156, 17th May, 2016.

[6] M. Peter and H. David, "Learn Raspberry Pi with Linux," Apress, 2012

[7] Rahul Kumar, Ushapreethi P, Pravin R. Kubade, Hrushikesh B. Kulkarni, Android Phone controlled Bluetooth Robot, International Research Journal of Engineering and Technology (IRJET), Volume: 03 Issue: 04 | Apr-2016 p-ISSN: 2395-0072 © 2016, IRJET

[8] Tahzib Mashrik, Hasib Zunair, Maofic Farhan Karin, Asia Modelling Symposium 2017 Eleventh Asia International Conference on Mathematical Modelling and Computer Simulation (AMS2017), At Kota Kinabalu, Sabah, Malaysia, Design and Implementation of Security Patrol Robot using Android Application, DOI: 10.1109/AMS.2017.20, 11th October 2017, published 12th september 2018.

[9] K. Pooventhan, R. Achuthaperumal and C. Manoj Balajee, "Surveillance Robot Using Multi Sensor Network", International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control, vol. 3, no. 2, pp. 113-115, 2015. 2

[10] Sneha Singh1, PradnyaAnap, YogeshBhaigade, International Journal of Advanced Research in Computer and Communication Engineering, “IP Camera Video Surveillance using Raspberry Pi.,” Vol. 4, Issue 2, February 2015.

[11] P. Sanjana, J. S. Clement, and S. R., International Journal of Computer Science and Information Technologies “Smart Surveillance Monitoring System Using Raspberry PI and PIR Sensor.” , Vol. 5 (6) , 2014, 7107-7109,

[12] S. Chia, J. Guo, B. Li and K. Su, "Team Mobile Robots Based Intelligent Security System", Applied Mathematics & Information Sciences, vol. 7, no. 2, pp. 435-440, 2013.

[13] H. Lee, W. Lin and C. Huang, "Indoor Surveillance Security Robot with a Self-Propelled Patrolling Vehicle", Journal of Robotics, vol. 2011, Article ID 197105, 2011.

[14] Y. Shimosasa, J. Kanemoto, K. Hakamada, H. Horii, T. Ariki, Y. Sugawara, F. Kojio, A. Kimura, S. Yuta, “Some 6481results of the test operation of a security service system with autonomous guard robot”, 26th Annual Conference of the IEEE on Industrial Electronics Society, Vol.1, pp.405-409, 2000.

[15] “Robot chefs take over Chinese restaurant,” [Online]. Available: http://story.newscloud.co.uk/BbcNewsWorld/2014/04/22/VideoRobotChefsTakeOverRestaurant. html. [Accessed 19 April 2014]. [16] Raspberry Pi for Beginners, 2014th ed. London UK: Imagine Publishing Ltd