A

Capstone Project Report

On

Face Detection and Human Tracking

Submitted

in Partial Fulfilment of the Requirements for the Degree of

Bachelor of Technology

in

Computer Science and Information Technology

by

Mr. Ritesh Arun Arage 2110026

Mr. Abhiram Maharudra Savale 2110031

Mr. Atharv Vijay Thanekar 2110039

Mr. Prathmesh Deepak Patil 2110063

Under The Guidance Of

Prof. P. T. Sawant

Information Technology Department

K.E. Society's

Rajarambapu Institute of Technology, Rajaramnagar

(An Empowered Autonomous Institute, Affiliated to Shivaji University, Kolhapur)

2024-2025

CERTIFICATE

This is to certify that Mr. Ritesh Arage, Mr. Abhiram Savale, Mr. Atharv Thanekar and Mr. Prathmesh Patil has successfully completed the project work and submitted project report on **"Face Detection and Human Trakking"** for the partial fulfillment of the requirement for the degree of **Bachelor of Technology** in **Computer Science and Information Technology** at **Rajarambapu Institute of Technology, Rajaramnagar, Dist: Sangli.** This final report is the record of the students work carried out under my supervision and guidance.

Prof. P. T. Sawant	Dr. A. C. Adamuthe	Dr. P. V. Kadole
Project Guide	**Head IT Dept.**	**Director**

Name and Sign of External Examiner:-

Date:

Place: RIT, Rajaramnagar

DECLARATION

We declare that this report reflects our thoughts about the subject in our own words. We have sufficiently cited and referenced the original sources, referred or considered in this work. We have not misrepresented or fabricated or falsified any idea/data/fact/source in this submission. We understand that any violation of the above will be cause for disciplinary action by the Institute.

Sr. No	Student Name	Roll No	Signature
1	Ritesh Arun Arage	2110026	
2	Abhiram Maharudra Savale	2110031	
3	Atharv Vijay Thanekar	2110039	
4	Prathmesh Deepak Patil	2110063	

Date:

Place: RIT, Rajaramnagar

ACKNOWLEDGEMENT

It is our foremost duty to express our deep sense of gratitude and respect to the guide Prof. P. T. Sawant mam for her uplifting tendency and inspiring us for taking up this project work successful. We are also grateful to Dr. A. C. Adamuthe (Head of Department, Information Technology) sir for providing all necessary facilities to carry out the project work and whose encouraging part has been a perpetual source of information.

We are thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staff of the Information Technology Department which helped us in successfully completing our project work. Also, we would like to extend our sincere esteems to all staff in the laboratory for their timely support.

ABSTRACT

Face Detection and Human Tracking System is useful in security and surveillance-based applications, child and elderly care and medical automation as a monitoring system. The project focuses on designing a cost-effective solution that can accurately identify individuals under various environmental conditions, such as changes in lighting, occlusions, and complex backgrounds. The system consists of multiple modules human face detection, motion tracking, motion control and user-friendly interface. By utilizing low-cost hardware components such as Raspberry Pi coupled with a web application for computation, the system aims to be accessible and adaptable to various applications including security, surveillance, and human-computer interaction. Simulation based Hardware support packages will be used to deploy this algorithm on to hardware. The project outcome includes prototype development, research paper publication, and participation in project competitions and conferences, contributing to advancements in the field of machine learning-based human tracking systems.

Keywords: : Machine Learning, Autonomous Human Tracking System.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation For Present Work

In today's technological world requirement for robust systems for detection of human faces is necessary and it is ever-growing. Applications ranging from security and surveillance to human-computer interaction require efficient solutions that can operate in various environmental conditions and scenarios. The primary objective of this project is to design and implement a cost-effective solution capable of detecting and tracking human faces in real-time. It is also useful in home and hospitals for child or elderly care and patient care, etc. Face detection and human tracking is widely used in domains like security, surveillance, and human-computer interaction. This project focuses on detecting faces and tracking human, exploring the underlying algorithms, methodologies, and real-world applications that contribute to the seamless integration of these technologies. It emphasizes the integration of cutting-edge deep learning techniques to facilitate real-time identification and tracking of individuals under diverse environmental conditions. Furthermore, the document highlights the creation of a user-friendly interface, enabling seamless interaction with the system and providing feedback on detected faces or humans. This interface enhances accessibility and usability, catering to a wide range of potential users across various sectors.

1.2 Problem Statement

Create a system to keep an eye on children and elderly individuals in houses or medical wards, as children tend to wander out when no one is around and elderly people with mental illnesses may leave their home or medical wards without warning and also can be used in various industries to monitor workers without the need for human intervention.

1.3 Objective

1. Design and develop a real time autonomous human tracking system using low-cost hardware.

2. To create a solution that can handle various environmental conditions, such as changes in lighting, occlusions, and complex backgrounds.

3. To Create a user-friendly interface that provides feedback on detected faces or humans and allows users to interact with the system easily

1.4 Layout Of Work

1.4.1 Fetching Real Time image,video and processing:

Capturing of live images using a Raspberry pi webcam is done. The images serve as input data for further analysis. And the captured real-time images and videos are processed to follow humans .

1.4.2 Detection Human:

In this phase, human detection is performed. Using deep learning algorithms like YOLO (You Only Look Once), the system identifies and locates humans within images or video frames. YOLO's convolutional neural network analyzes visual features, such as body shape, posture, and other contextual clues, to detect humans in real-time. The model is trained on a large, labeled dataset that contains various images of humans in different environments, positions, and lighting conditions, allowing it to accurately recognize and classify human figures in diverse scenarios. YOLO provides high-speed detection by predicting bounding boxes and class labels for humans in a single pass, making it effective for real-time applications

Chapter 2

Literature Review

The Design of Face Recognition and Tracking for Human-Robot Interaction .

This paper discusses the development of Social Robot named SyPEHUL (System of Physic, Electronic, Humanoid Robot and Machine Learning) which can detect and tracking human face. Face detection and tracking process use Cascade Classification and LBPH (Local Binary Pattern Histogram) Face detection method based on OpenCV library and Python Author: W.S. Mada Sanjaya et al.

Robust Real-Time Face Detection

This paper describes a face detection framework that is capable of processing images extremely rapidly while achieving high detection rates. A set of experiments in the domain of face detection is presented. The system yields face detection performance comparable to the best previous systems. Author: Paul Viola, Michael Jones.

A Fast and Accurate System for Face Detection, Identification, and Verification:

This Paper provide the design details of the various modules involved in automatic face detection: face detection, landmark localization and alignment, and face identification/verification. We propose a novel face detector, Deep Pyramid Single Shot Face Detector (DPSSD), which is fast and detects faces with large scale variations (especially tiny faces).

Additionally, we propose a new loss function, called the Crystal Loss, for the tasks of face verification and identification.

Author: Rajeev Ranjan, Ankan Bansal et al.

Face Recognition and Tracking Framework for Human–Robot Interaction :

Author developed a framework based on lightweight CNNs for all face detection stages, including face detection, alignment and feature extraction, to achieve higher accuracies in these challenging circumstances while maintaining the real-time capabilities required for HRI systems. To maintain the accuracy, a single-shot multi-level face localization in the wild (RetinaFace) is utilized for face detection, and additive angular margin loss (ArcFace) is employed for detection. Author: Aly Khalifa et al.

A comparative study on face detection and tracking algorithms :

Face localization is the first stage in many vision-based applications and in human–computer interaction. The problem is to define the face location of a person in a colour image. The four boosted classifiers embedded in OpenCV, based on Haar-like features, are compared in terms of speed and efficiency. Skin colour distribution is estimated using a non-parametric approach.

Chapter 3

Preliminaries

To establish a solid foundation for the successful implementation of the project on Face Detection and Human Tracking using YOLO Algorithm, several essential preparatory steps were meticulously carried out:

3.1 Software Setup

To develop the human tracking system, essential software tools and libraries were installed and configured. Additionally, libraries such as picamera were used to interface the Raspberry Pi camera module, enabling real-time video capture and processing. Optimizations for Raspberry Pi, such as reduced model sizes and efficient inference methods, were also implemented to ensure smooth operation on limited hardware.

3.2 Hardware Resources

The core hardware for the system included a Raspberry Pi 4, a compatible camera module, and the Coral USB Accelerator. The Raspberry Pi provided the computing platform, while the Coral Edge TPU USB Accelerator, which is designed to run machine learning models at the edge, significantly boosted the inference speed. The combination of the Raspberry Pi and Coral Edge TPU allowed the system to efficiently run the YOLO model for human detection and tracking, meeting real-time performance requirements while using minimal power

Chapter 4

Implemented Work

4.1 Proposed System

The proposed system focuses on real-time face detection and human tracking using Raspberry Pi, Google Coral, and the YOLO algorithm. The system integrates hardware and deep learning to achieve seamless human-following capabilities.

A webcam is used as the primary input source, capturing live video feeds. The YOLO (You Only Look Once) algorithm processes these feeds to detect human faces and track their movements. The Raspberry Pi handles image acquisition and preliminary processing, while Google Coral accelerates the inference of the YOLO model, ensuring low-latency detection.

The detected human's position is translated into motion commands for the robot, which uses motors and wheels for navigation. The robot continuously adjusts its movements to maintain a predefined distance from the person, effectively following them in real-time. The use of a lithium-ion battery provides sufficient power for prolonged operation.

This system demonstrates a practical application of deep learning in robotics, achieving robust and efficient human tracking in dynamic environments. Future enhancements could include obstacle avoidance and improved accuracy in crowded scenarios.

4.2 End Users

- **Patients in Hospitals:** The system ensures real-time monitoring and mobility support for hospital patients. By following them autonomously, it assists medical staff in tracking patient movements and ensuring their safety, particularly for those requiring close supervision. For patients with mobility issues, the system can also serve as a companion device, offering proximity-based assistance.
- **Elderly Individuals:** Elderly individuals often require consistent monitoring and assistance, especially those prone to wandering or who have mobility challenges. This system acts as a companion robot, tracking their movements and staying nearby, ensuring their safety and reducing the burden on caregivers. Its ability to follow humans autonomously provides an added layer of security in both hospital and home environments.

4.3 Functional Requirements

1. **Human Detection and Tracking**
    - The system should detect and identify humans in real-time using the YOLO algorithm.
    - It should accurately track a human's position and movement.
2. **Human Following**
    - The system should control motors and wheels to follow a detected human while maintaining a safe distance.

- It should dynamically adjust to changes in the human's movement, such as direction or speed.

3. **Real-Time Performance**

   - The system should process input from a webcam and provide real-time tracking and following capabilities without significant delay.
   - It should ensure continuous and smooth operation in dynamic environments.

4.4 Non Functional Requirements

1. **Performance**

   - The system must deliver real-time human detection and tracking with minimal latency to ensure smooth operation.

2. **Accuracy**

   - The tracking system should maintain high precision in detecting and following humans, minimizing errors in movement prediction or misidentification.

3. **Security**

   - User data, including any video or image inputs, must be processed securely to maintain confidentiality and prevent unauthorized access.

4. **Usability**

   - The system should be intuitive and easy to use, requiring minimal technical expertise for setup and operation.

5. **Compatibility**

- It should operate seamlessly across various platforms, including Linux-based systems like Raspberry Pi, and be integrable with other hardware components.

4.5 Scope of Project

The proposed human detection and tracking system leverages advanced technology to assist users by autonomously following them based on real-time facial detection. Using a camera, the system identifies a human, processes their movement, and navigates accordingly. This system is particularly useful for enhancing mobility and providing support in settings like hospitals, homes, and childcare environments.

The project aims to deliver a reliable, user-friendly solution with smooth and accurate tracking capabilities. By focusing on real-time detection and tracking, the system ensures consistent performance, adapting to changes in the user's motion. It is designed to be intuitive, with results and operations that are straightforward and accessible for a wide range of users.

4.6 Deployment Requirements

4.6.1 Hardware Requirements

Table 4.1: Hardware Requirements

Requirement	Specification
Raspberry Pi	IoT OS Model
Raspberry pi cam	Web Camera 640*480 resolution or higher
DC Motor	200 rpm
Chassis	Car chassis infrastructure
Google Coral	USB Accelerator
Ultrasonic sensor	Ultrasonic sensor

4.6.2 Software Requirements

Table 4.2: Software Requirements

Requirement	Specification
Chrome Browser	latest
Python	Version: 3.10.0

Chapter 5

Design

5.1 System Architecture

A system architecture is a conceptual framework that describes a system's behavior, structure, and other features. A formal description and representation of a system's behavior and structure, designed to facilitate reasoning about such behaviors and structures, is referred to as an architecture description.

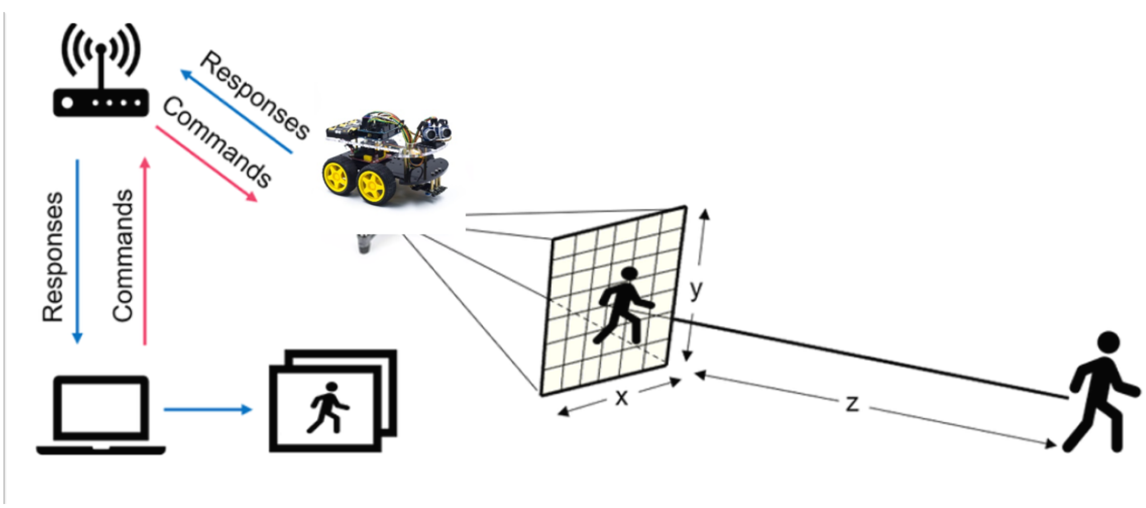

Figure 5.1: System Architecture

5.2 Flow Chart

A well-defined pictorial representation of a logical process, a labor-intensive process, or an industrial process is a flowchart. Giving users a common language or informative source when managing a project or process is the major goal of a flow chart.

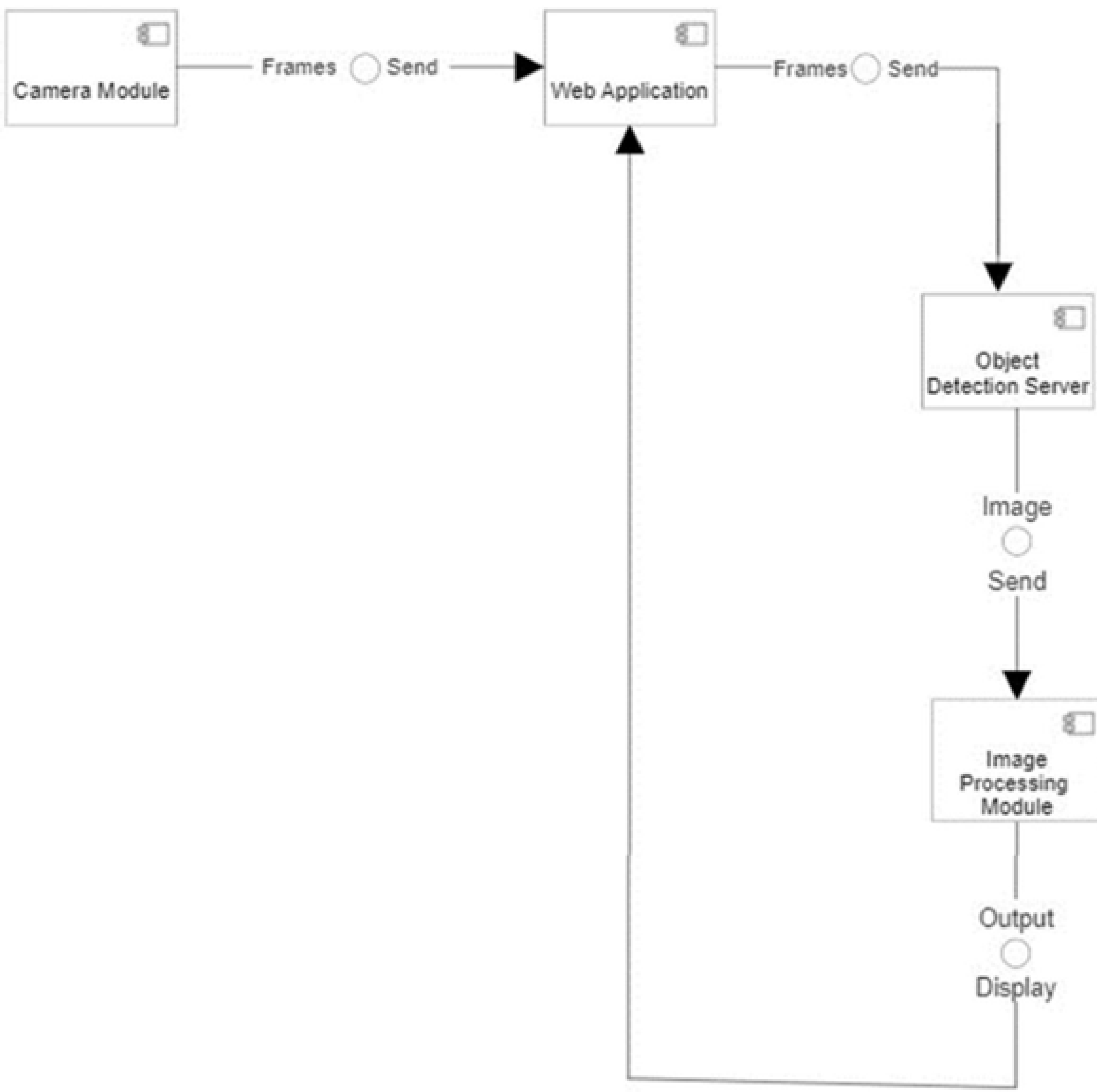

Figure 5.2: Flowchart

5.3 Data Flow Diagrams

DFD is the appearance of the overall system. Helps you to easily understand the system. The line with the arrow denotes the direction of data flow. DFD diagrams display the process as well as the inputs and outputs of each entity.

5.3.1 DFD Level-0

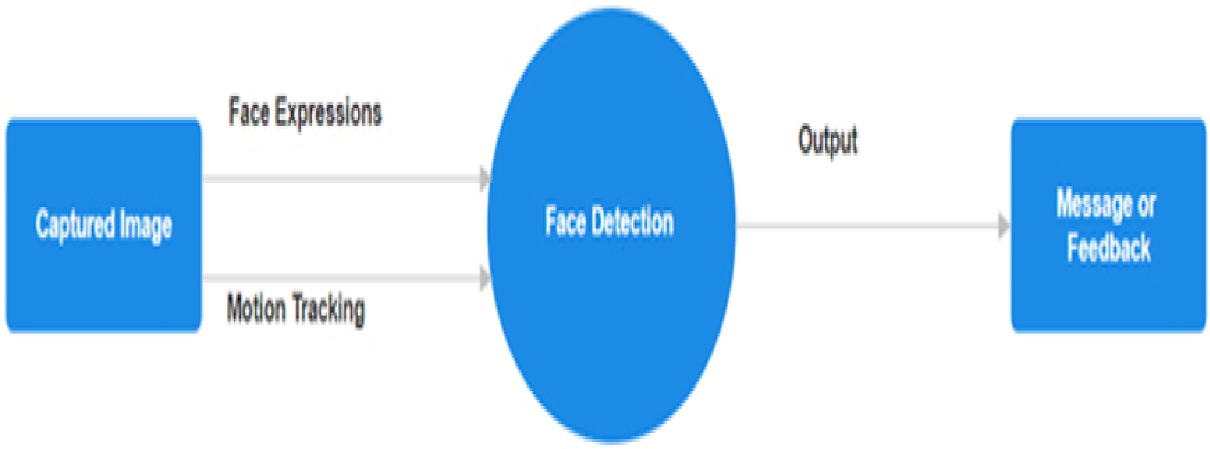

Figure 5.3: DFD Level-0

In figure 5.3, DFD level-0 outlines the high-level structure of a Prediction and Recommendation System, illustrating the primary processes and the flow of information between the Admin and User. The central process is the "Prediction and Recommendation System," which performs the core functions of generating predictions and recommendations. The Admin entity is responsible for system administration, including configuration and oversight, while the User entity interacts with the system to receive personalized predictions and recommendations.

5.3.2 DFD Level-1

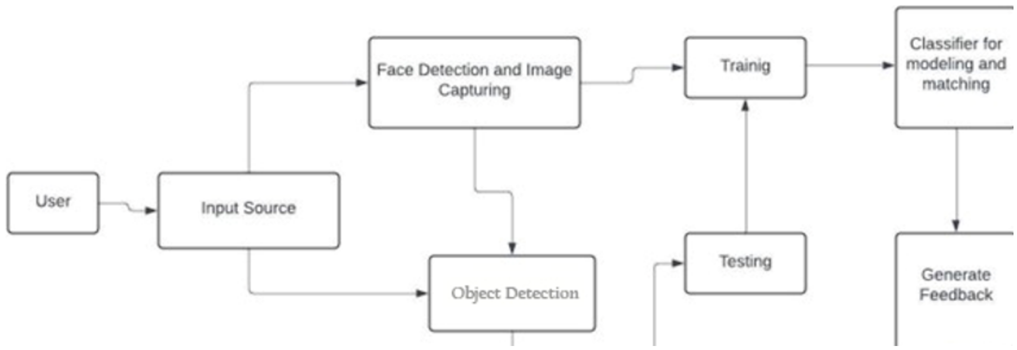

Figure 5.4: DFD Level-1

In figure 5.4, DFD level-1 provides a more detailed view of the Prediction and Recommendation System, focusing on specific processes. It encompasses the user registration process, user database management, and login functionalities for both users and administrators. Users register by providing input details, and their information is stored in the user database. Users input details for personalized interactions, and the system generates outputs such as recommendations and predictions based on these inputs.

5.4 Class Diagram

The classes are represented by the rectangles. Rectangle has three sub-sections: class name, attribute, and operation. A connection between upper elegance and its subclass can be seen in the generalisation. The relationship is indicated by the arrow. Subclasses are used to connect interfaces.

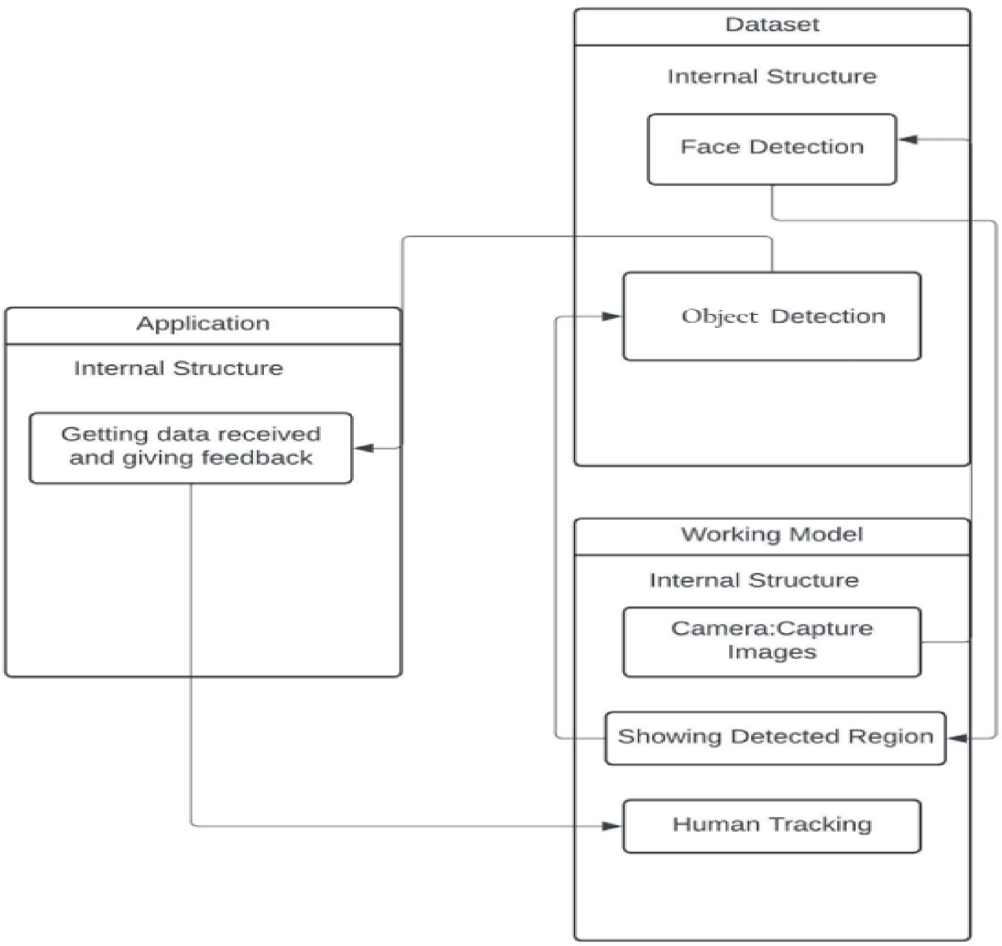

Figure 5.5: Class Diagram

In figure 5.5, The class diagram for the system has dataset with internal structure face detection and object detection with the camera capturing images and videos. It shows detected region and shows on interface.

5.5 Use Case Diagram

A use case diagram, which is typically accompanied by other diagrams, illustrates the system's various use cases and various types of users. Use cases are shown as an oval with arrows pointing in the direction of relationships.

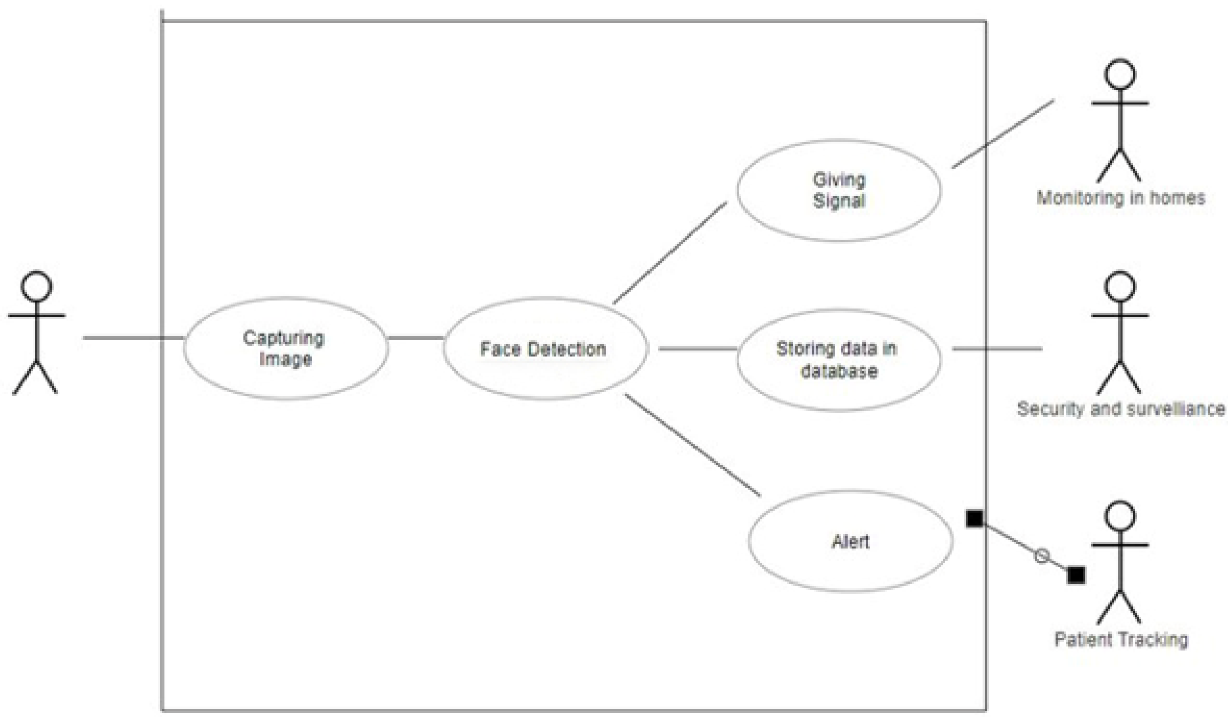

Figure 5.6: Use Case Diagram

In figure 5.6, The use case diagram for the system involving end users like home security, patient monitoring etc.

Chapter 6

Development/Implementation Details

6.1 Setup and Environment

Install Python: Make sure you have Python installed on your system. Import Libraries: Import the required libraries, including OpenCV (for YOLO).

6.2 Capture Image

Initialize Image Capture: Create an Image capture object to access the camera feed. Read Frames: Continuously read frames from the Image capture object.

6.3 Preprocessing

Convert Frames: Convert the captured frames to grayscale or apply other preprocessing techniques, if needed, to enhance the image quality or simplify the processing.

6.4 Hand/Object Detection:

Implement Hand/Object Detection: Apply hand or object detection algorithms to locate the user's hand or object of interest within the captured frame. This can be achieved using techniques like background subtraction, contour detection, or machine learning-based methods.

Chapter 7

Testing

7.1 Test Cases

Testing is the process of debugging a programme, which is one of the most important components of computer programming. It is the process of evaluating a system or its component(s) with the goal of determining whether or not it meets the defined requirements. It is putting in place a system to discover any gaps, flaws, or missing requirements between what it wants and what it needs.
Features to be tested

- Web interface module
- Face detection and human tracking module

7.1.1 Unit Testing

Unit testing is the process of testing each unit or component of a software application separately. The goal of unit testing is to ensure that unit components work as expected. The purpose of unit testing is to separate each component of a program and demonstrate that each component is working properly. A unit test establishes a clear, written contract that the code must meet. As a result, it provides a number of advantages. Unit testing identifies issues early in the development process.

Web interface Module

Test Scenario:

1. Wifi Connectivity with web interface

Table 7.1: Test Case: Raspberry Pi Module

Test Case ID	Test Requirement	Precondition	Steps	Test Data	Expected Result	Status
TC01	Verify the hardware connection.	No connection error between raspberry pi and localhost.	1. Connect the batteries to the raspberry pi. 2. Turn on the system. 3. Open chrome and connect to localhost.	Localhost ip address: http://192.168.255.103/ controlpanel	Connection successful and web interface is accessible.	Pass

Face Detection and Human Tracking module

Test Scenario:

1. Live feed with one person available.

2. The system should work on different lighting conditions

Table 7.2: Test Case: Face Detection and Human Tracking Module

Test Case ID	Test Requirement	Precondition	Steps	Test Data	Expected Result	Status
TC01	Provide a live video feed with one person available	1. System should be on for tracking. 2. System should not lag behind	1. Human face detection. 2. Storing it in database. 3. Tracking	Data is collected from live feed (captured images/videos dataset)	The system should detect the face and display a bounding box around it.	Pass
TC02	Performance Under Low Lighting.	Provide video feed in low lighting conditions	Switch on the lights	-	The system should accurately detect and track faces.	Pass

Chapter 8

Deployment

8.1 User manual

8.1.1 Hardware module

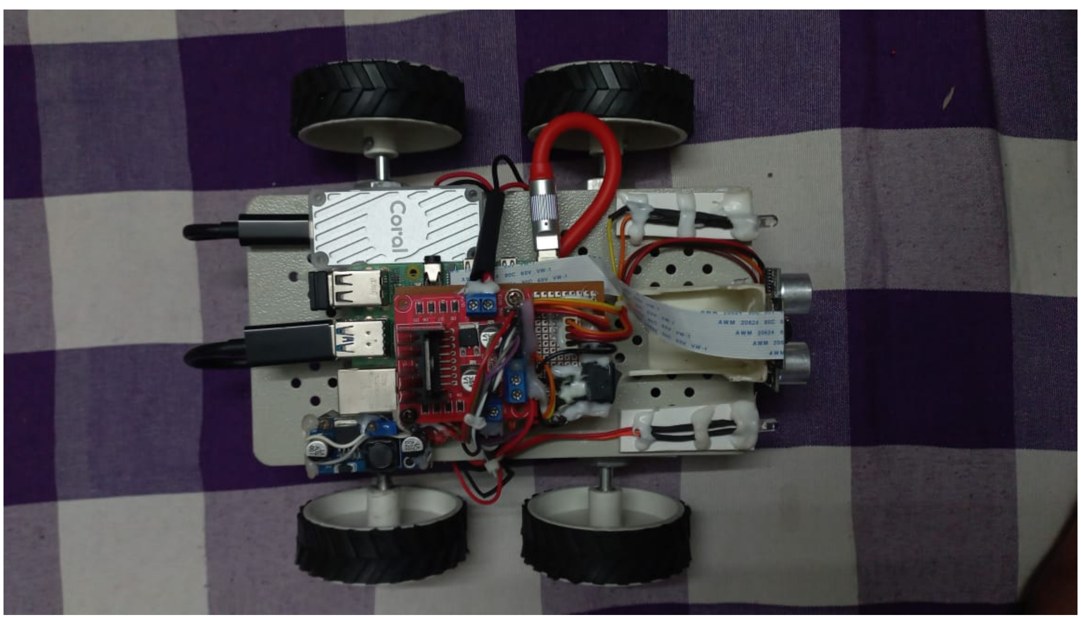

Figure 8.1: Hardware Module

The Figure 8.1 shows hardware of face detection and human tracking system.

8.1.2 Web interface module

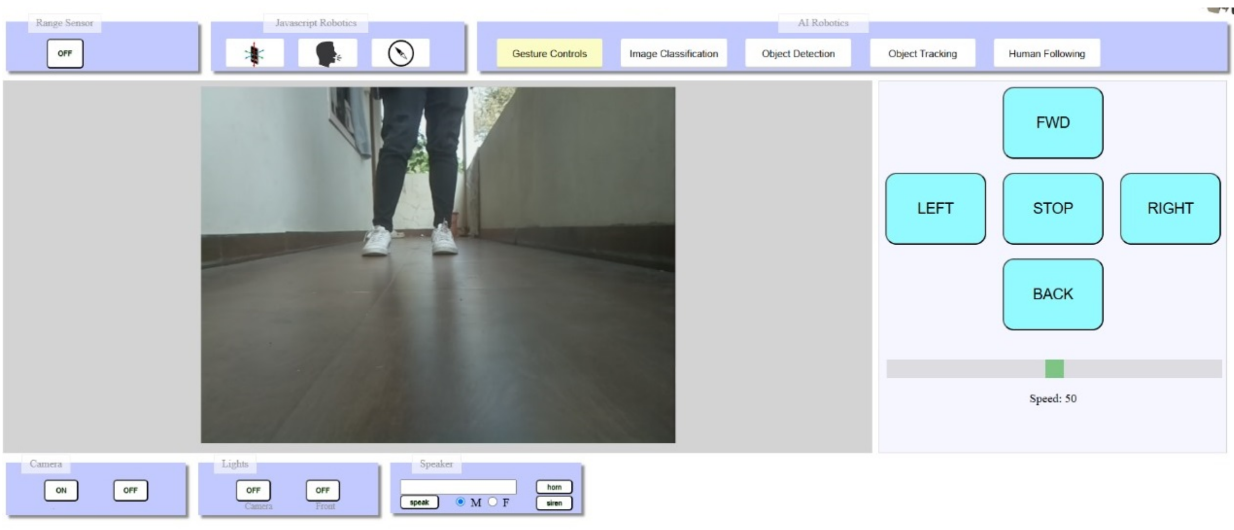

Figure 8.2: Web interface module

The Figure 8.2 shows the web interface between raspberry pi and user using localhost ip.

8.1.3 Human Tracking Module

In Figure 8.4, system tracks the human and follows them.

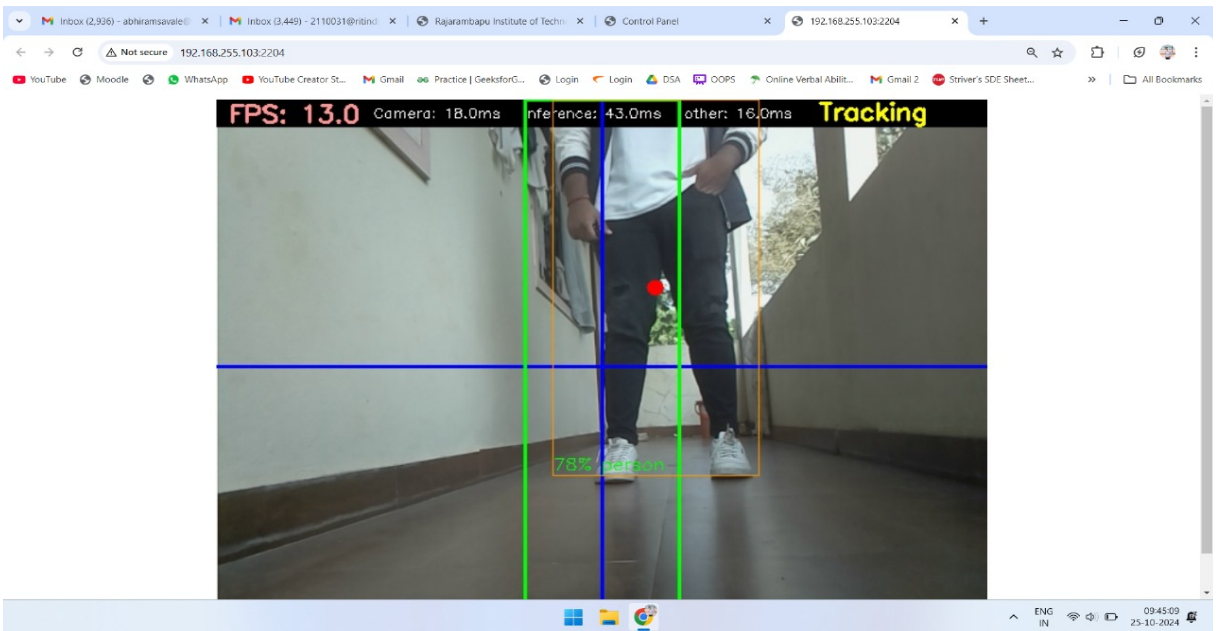

Figure 8.3: Human Tracking module

Chapter 9

Results and Discussion

The system integrates advanced face detection and human tracking techniques to achieve real-time performance and user interactivity. Face detection leverages deep learning-based methods, such as Convolutional Neural Networks (CNNs), to accurately identify human faces within video frames. Following detection, tracking algorithms like YOLO, openCV, haar-cascade are employed to monitor movement while addressing challenges such as occlusions, scale variations, and abrupt motion changes. A user-friendly interface complements the system by providing live video feeds with overlaid bounding boxes for detected faces and interactive features like starting or stopping tracking, adjusting parameters, and viewing system status. This cohesive approach ensures robust functionality and accessibility.

Chapter 10

Conclusion and Future Work

10.1 Conclusion

In summary, this project demonstrates a cost-effective solution for real-time human face detection and tracking, emphasizing its relevance in areas such as security, surveillance, and human-computer interaction. By exploring key technologies and practical applications, it lays the groundwork for integrating these systems effectively across various domains. The future of this work includes enhancing detection features by incorporating facial gestures, emotions, and expressions, addressing privacy concerns with robust anonymization and encryption methods, and advancing algorithms for adaptive learning and self-evolution. These developments aim to make face detection and tracking systems more versatile, secure, and responsive to dynamic environments, paving the way for broader adoption and impactful innovations.

10.2 Future Work

- Link the system to a wider connectivity like cloud.
- Enhancing accuracy of detection and in crowded and complex environments with multiple subjects.

Chapter 11

References

[1] H. E. Dursun, E. C. Güven, B. Avci and T. Kumbasar, "Recognizing and Tracking Person of Interest: A Real-Time Efficient Deep Learning based Method for Quadcopters," 2023

[2] J. Zheng, R. Ranjan, C. -H. Chen, J. -C. Chen, C. D. Castillo and R. Chellappa, "An Automatic System for Unconstrained Video-Based Face Recognition," in IEEE Transactions on Biometrics, Behavior, and Identity Science, vol. 2, no. 3, pp. 194-209, July 2020.

[3] J. Qian, S. Ma, Z. Hao and Y. Shen, "Face detection and recognition method based on skin color and depth information," 2011 International Conference on Consumer Electronics, Communications and Networks (CECNet), Xianning, China, 2011, pp. 345-348, doi: 10.1109/CECNET.2011.

[4] P. Viola and M. Jones, "Robust real-time face detection," Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001, Vancouver, BC, Canada, 2001.

[5] X. Wenkai, L. SukHwan, O. Soo-Yol and L. Eung-Joo, "Motion recognition based on face and hand activity detection," 2012 IEEE Symposium on Computers Informatics (ISCI), Penang, Malaysia, 2012.

[6] M.R. Islam and K. Horio, "Real Time-Based Face Recognition, Tracking, Counting, and Calculation of Spent Time of Person Using OpenCV and Centroid Tracker Algorithms,"

2023 5th International Conference on Computer Communication and the Internet (ICCCI), Fujisawa, Japan, 2023.

[7] Tayal, Anmol. "An Interactive Alphabet and Number learning system using OpenCV and CNN." In 2021 Asian Conference on Innovation in Technology (ASIANCON), pp. 1-4. IEEE, 2021.

[8] Mohanty, Ronak R., Ricardo M. Castillo, Eric D. Ragan, and Vinayak R. Krishnamurthy. "Investigating force-feedback in mid-air sketching of multi-planar three-dimensional curve-soups." Journal of Computing and Information Science in Engineering 20, no. 1 (2020): 011010

[9] Saoji, S. U., Nishtha Dua, Akash Kumar Choudhary, and Bharat Phogat. "Air canvas application using Opencv and numpy in python." IRJET 8, no. 08 (2021).

[10] Younas, J., Margarito, H., Bian, S., Lukowicz, P. (2020, December). Finger air writing-movement reconstruction with low-cost imu sensor. In MobiQuitous 2020-17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (pp. 69-75).