**CAPSTONE PROJECT REPORT**

on

**Raspberry Pi Object Detection Drone**

Submitted by

**Prathamesh Bodhankar (1032211162)**

**Pranjali Goyal (1032211032)**

**Anant Jain (1032211176)**

Under the Guidance of

**(Prof. Shiv Sutar)**



**Department of Computer Engineering and Technology**

**Dr. Vishwanath Karad**

**MIT WORLD PEACE UNIVERSITY, PUNE**

**[2024-2025]**

**DECLARATION**

I/We the undersigned, declare that the work carried under Capstone Project entitled **“Raspberry Pi Object Detection Drone”** represents our idea in my/our own words. I/We have adequately cited and referenced the original sources where other ideas or words have been included. I/We also declare that I/We have adhered to all principles of academic honesty and integrity and have not misprinted or fabricated or falsified any ideas/data/fact/source in my/our submission. I/We understand that any violation of the above will be cause for disciplinary action by the University and can also evoke penal action from the sources which have thus not been properly cited or whose proper permission have not been taken when needed.

Date: 19/11/2024

Place: MIT-WPU

|  |  |  |
| --- | --- | --- |
| **PRN Number** | **Name of student** | **Signature with date** |
| 1032211162 | Prathamesh Bodhankar |  |
| 1032211032 | Pranjali Goyal |  |
| 1032211176 | Anant Jain |  |

|  |  |  |
| --- | --- | --- |
| Prof. Shiv Sutar |  | Dr. B. M. Patil |
| Project Guide | Project Coordinator | Program Director |

Place: Pune

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to all those who have contributed to the successful completion of our project Raspberry Pi Object Detection Drone for Multiple Industries.

First and foremost, we would like to thank our project guide, Prof. Shiv Sutar, for his invaluable guidance, support, and constructive feedback throughout the course of this project. His expertise and mentorship have been crucial in shaping this research and enhancing its quality.

We also wish to acknowledge the support of MIT World Peace University and our department of Computer Science Engineering, whose resources and academic environment have been essential in completing this work. The facilities and opportunities provided have been indispensable in conducting the necessary research.

Finally, we would like to extend our gratitude to the industry professionals and subject matter experts whose inputs helped refine the design and functionality of the Raspberry Pi Object Detection Drone. Their insights into the technical challenges and safety considerations of developing an AI-enabled face tracking drone have been instrumental in refining the final design and functionality of this project.

This project would not have been possible without the assistance of these individuals and institutions, and we are deeply appreciative of their support.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Abstract** | | **5** |
| **List of Tables** | | **6** |
| **List of Figures** | | **7** |
| **Abbreviations** | | **8** |
| **1.** | **Introduction** | **1** |
| 1.1 | Overview | 1 |
| 1.2 | Challenges in the Drone Technology | 1 |
| 1.3 | Limitations of Current Drone Technology | 2 |
| 1.4 | The Potential of OpenCV - Integrated Technology | 3 |
| 1.5 | Project Overview: Raspberry Pi Object Detection Drone | 3 |
| 1.6 | Motivation | 5 |
| 1.7 | Organization of report | 6 |
| **2.** | **Review of Literature** | **7** |
| 2.1 | Literature Review | 7 |
| 2.2 | Aim and Objectives of project | 11 |
| **3.** | **System Development** | **12** |
| 3.1 | System block diagram | 12 |
| 3.2 | System Specifications | 13 |
| 3.3 | Challenges Faced/Complexities involved | 21 |
| 3.4 | Methodology | 24 |
| **4.** | **System Implementation** | **29** |
| 4.1 | System Design and Description | 29 |
| **5.** | **Results and Analysis** | **31** |
| 5.1 | Results of Implementations | 31 |
| 5.2 | Analysis of Results | 34 |
| **6.** | Discussion and Conclusion | **36** |
| 6.1 | Disc Discussion | 36 |
| 6.2 | Conclusion | 37 |
| 6.3 | Future Scope | 38 |
| **7.** | Individual Contribution | **40** |
|  | **References** | **43** |

**ABSTRACT**

The "Raspberry Pi Object Detection Drone" project aims to create an autonomous drone capable of real-time object detection and surveillance, leveraging affordable hardware and advanced software. Powered by a Raspberry Pi 4B, the drone integrates a camera module for video capture and employs OpenCV's pre-trained object detection models for recognizing and tracking objects within its environment. Equipped with a robust drone propulsion system, motors, and a flight controller, the drone adjusts its position dynamically to maintain optimal focus on detected objects.

The system facilitates real-time data transmission via the Raspberry Pi's Wi-Fi interface, allowing for live monitoring and analysis. This innovative approach showcases its potential across diverse fields, including military surveillance, agricultural surveys, and environmental monitoring. By combining cost-effective hardware with open-source technologies, this project demonstrates a practical and scalable solution for applications requiring advanced autonomy. Despite challenges such as processing efficiency and flight stability, the project underscores the viability of integrating Raspberry Pi, OpenCV, and Machine Learning into sophisticated, real-world applications.

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Name of Tables** | **Page No.** |
| Table 1. | Raspberry Pi 4 Model B specifications | 14 |
| Table 2. | KK2.1.5 LCD Flight Control Board specifications | 16 |

**LIST OF FIGURES**

| **Figure No.** | **Name of Figure** | **Page No.** |
| --- | --- | --- |
| 1 | System Block Diagram | 12 |
| 2 | Raspberry Pi 4 Development Board | 14 |
| 3 | A2212 1000KV Brushless DC Motor | 15 |
| 4 | 30A BLDC ESC Electronic Speed Controller | 15 |
| 5 | Propellers 1045/1045R (10 x 4.5) CW CCW | 16 |
| 6 | KK2.1.5 LCD Flight Control Board | 16 |
| 7 | F450 Quadcopter drone frame Kit with integrated PCB | 17 |
| 8 | Raspberry Pi 5MP Camera Module | 18 |
| 9 | FS-iA10B Receiver | 18 |
| 10 | FLYSKY FS-i6X I6X AFHDS 2A Transmitter | 19 |
| 11 | 3S 11.1 V 2200mAh LiPo Battery | 20 |
| 12 | XT60 Connectors – M to F Pair | 20 |
| 13 | Raspberry Pi Camera Interface | 24 |
| 14 | Object Detection using OpenCV | 24 |
| 15 | Pre-Trained Model | 26 |
| 16 | Hardware module of the Drone | 33 |
| 17 | Object Detection Output using OpenCV | 35 |

**ABBREVIATIONS**

| **Abbreviation** | **Full Form** |
| --- | --- |
| **Ras PI** | Raspberry Pi |
| **CV** | Computer Vision |
| **UAV** | Unmanned Aerial Vehicles |
| **Wi-Fi** | Wireless Fidelity |
| **AI** | Artificial Intelligence |
| **IoT** | Internet of Things |
| **MOSFET** | Metal Oxide Semiconductor Field Effect Transistor |
| **ESC** | Electronic Speed Control |
| **LCD** | Liquid Crystal Display |
| **CSI** | Camera Serial Interface |
| **CCTV** | Closed Circuit Television |
| **ROS** | Robot Operating System |
| **KV** | Constant Velocity |
| **DC** | Direct Current |
| **BLDC** | Brushless Direct Current |
| **PCB** | Printed Circuit Board |
| **MP** | Mega Pixel |
| **AFHDS** | Automatic Frequency Hopping Digital System Second Generation |
| **LiPo** | Lithium Polymer |
| **V** | Voltage |
| **MaH** | Milliamp Hour |
| **SSD** | Single Shot Object Detection |
| **COCO** | Common Objects in Context dataset |
| **LBPH** | Local Binary Pattern Histogram |
| **FPS** | Frames Per Second |
| **YOLO** | You only Look Once |

# CHAPTER 1

# INTRODUCTION

* 1. **OVERVIEW**

Unmanned Aerial Vehicles (UAVs), or drones, have emerged as transformative tools across various industries, offering unparalleled capabilities in surveillance, data acquisition, and automation. From enhancing agricultural monitoring to bolstering military reconnaissance, drones are instrumental in improving operational efficiency and safety. Despite their growing adoption, conventional drone systems often depend heavily on manual operation or limited automation, presenting challenges in dynamic and unpredictable environments.

The integration of artificial intelligence and computer vision technology into drone systems presents an opportunity to overcome these limitations. With advancements in affordable, compact processing units like the Raspberry Pi, it is now feasible to design intelligent drones capable of real-time object detection and tracking. These innovations enable drones to operate autonomously, adapt to changing scenarios, and provide enhanced functionality for diverse applications. By harnessing the power of machine learning and computer vision, drones can elevate their performance beyond traditional capabilities, driving significant progress in fields such as search and rescue, precision agriculture, and surveillance operations.

* 1. **CHALLENGES IN THE DRONE TECHNOLOGY**

The development and deployment of drones for autonomous tasks present several technical and operational challenges. Key challenges include:

* Processing Power Limitations: Most affordable drones, like those powered by Raspberry Pi, operate on constrained computational resources. Real-time image processing for object detection and tracking, especially using machine learning models, can strain these systems and impact performance.
* Flight Stability in Dynamic Environments: Maintaining stable flight while simultaneously tracking objects is challenging, especially in windy or uneven terrain. Flight controllers must constantly adjust to changes in the environment, ensuring smooth navigation and accurate tracking.
* Object Detection Accuracy: The reliability of object detection and tracking depends on the quality of the pre-trained machine learning models. Issues like low-light conditions, motion blur, or overlapping objects can reduce the system’s ability to identify and track targets effectively.
* Power Consumption and Battery Life: Drones rely on compact battery systems to power motors, controllers, and onboard processing units. Performing computationally intensive tasks like real-time video processing further drains the battery, limiting flight duration and operational range.
* Real-Time Data Transmission: Transmitting high-quality video streams and telemetry data in real time requires a robust and reliable wireless connection. Signal interference, range limitations, or network delays can hinder performance, particularly in remote or crowded areas.
* Integration of Hardware and Software Components: Achieving seamless integration between flight controllers, camera modules, and processing units is critical. Ensuring compatibility and synchronized operation across all components is a complex task that requires meticulous design and testing.
  1. **LIMITATIONS OF CURRENT DRONE SYSTEM**

Traditional drone systems, while effective in basic applications, face significant limitations when it comes to advanced tasks like real-time object detection and autonomous tracking. Many commercially available drones rely on manual controls or simple automation, restricting their ability to adapt to dynamic environments. Object detection capabilities, if present, are often basic and prone to inaccuracies in challenging conditions such as low light, high motion, or cluttered backgrounds. This reduces their effectiveness in complex surveillance and monitoring tasks.

Additionally, most current drone systems lack integration with advanced machine learning models or AI-based predictive analytics. This absence limits their ability to make informed decisions, such as anticipating object movement or dynamically adjusting flight paths for optimal tracking. Traditional drones also operate with separate systems for flight control and data capture, leading to inefficiencies and delays in processing real-time information.

Another major constraint is the lack of real-time data transmission infrastructure in many drones. Without reliable wireless connectivity and synchronized data sharing, the ability to monitor and control drones remotely is compromised, particularly in areas with limited network coverage. These gaps in current drone technology highlight the need for smarter, more integrated solutions that combine AI, machine learning, and robust communication protocols to achieve enhanced autonomy and functionality.

* 1. **THE POTENTIAL OF OPENCV-INTEGRATED TECHNOLOGY**

The integration of OpenCV and machine learning into drone systems represents a significant advancement in autonomous object detection and tracking. OpenCV, an open-source computer vision library, empowers drones to process and interpret visual data in real-time, enabling them to recognize objects, faces, and environmental features. By using the Raspberry Pi's computational capabilities, OpenCV integrates seamlessly with the drone's camera system to provide live object tracking, a crucial element for surveillance and monitoring applications.

One of the key advantages of integrating OpenCV is its real-time processing ability, which allows the drone to detect and track objects autonomously. Traditional drones often rely on manual or limited automation for tracking, but OpenCV-equipped drones can perform complex tasks such as face recognition, motion tracking, and obstacle avoidance, all while maintaining stable flight. By leveraging pre-trained machine learning models, OpenCV can identify and categorize objects based on predefined criteria, offering a higher level of accuracy and reliability in dynamic environments.

Moreover, OpenCV's ability to process and analyze visual data enables predictive capabilities for the drone. Instead of merely reacting to changes in the environment, the drone can anticipate movement patterns, adjust its position for optimal tracking, and proactively respond to obstacles or changes in target behavior. This integration not only enhances the drone's functionality but also improves its adaptability to various real-world applications, from military surveillance to agricultural monitoring.

* 1. **PROJECT OVERVIEW: Raspberry Pi Object Detection Drone**

**Raspberry Pi Object Detection Drone** is an autonomous drone designed for real-time object detection and tracking using OpenCV and a Raspberry Pi 4B. It provides enhanced surveillance capabilities by autonomously identifying and following objects, transmitting live data through Wi-Fi.

* Development Board: The Raspberry Pi serves as the central processing unit of the system, collecting data from various sensors, processing it in real time, and triggering alerts based on predefined safety thresholds. Its capabilities allow it to handle multiple sensor inputs and support IoT connectivity for remote monitoring.
* Motors: Critical for ensuring stable flight, precise maneuverability, and efficient operation in various dynamic environments.
* Speed Controllers: Provides precise motor speed control with high-quality MOSFETs, programmable throttle, and built-in protection features, ensuring efficient and reliable performance for quadcopters and multi-rotors.
* Propellers: ABS Propellers are lightweight, high-strength propellers designed for multi-copters, enhance aerofoil stability, and prevent whirlpools during flight.
* Flight Controller: The KK 2.1.5 Flight Controller simplifies multi-rotor setup with an LCD screen, intuitive software, and pre-installed craft types, while offering polarity protection and improved voltage sensing for safer, hassle-free installation.
* Transmitter and Receiver: FLYSKY FS-i6 enables precise manual control of the drone, offering a reliable communication link for seamless operation of flight movements and adjustments, ensuring stable performance during real-time object detection and tracking tasks.
* Camera Module: It is a compact, lightweight camera designed for Raspberry Pi projects, offering high-definition still images and video. It connects via a flexible cable to the Raspberry Pi's CSI interface, providing high data rates for efficient image and video capture, ideal for drones, CCTV, or mobile applications.
* The AI model utilized in this project employs the MobileNet SSD (Single Shot MultiBox Detector) framework integrated with OpenCV for real-time object detection. It leverages a pre-trained model trained on the COCO dataset to detect and classify objects from live camera feeds. The model processes the video stream by loading object definitions from a `.names` file and using configuration and weight files for detection, ensuring robust and accurate object recognition even under resource-constrained environments like the Raspberry Pi.

The system is designed to continuously process live video feeds, detect and classify objects in real time, and dynamically adjust the drone's tracking and navigation. This object detection model enhances functionality by enabling the drone to identify targets, adapt to changing scenarios, and provide accurate real-time feedback for applications such as surveillance, deliveries, or environmental monitoring.

* 1. **MOTIVATION**

The integration of drones in various industries has revolutionized surveillance, monitoring, and data collection. However, while drones have become invaluable tools, current systems still face limitations in autonomous object detection and real-time tracking, particularly in dynamic environments. This challenge highlights the need for more advanced, intelligent drone solutions capable of real-time processing and adaptive tracking. The motivation behind the **Raspberry Pi Object Detection Drone** project is to leverage the power of affordable technology, such as the Raspberry Pi and OpenCV to develop a cost-effective, autonomous drone capable of detecting and tracking objects in real time. This innovation could offer new solutions in fields like military surveillance, search and rescue operations, and precision agriculture, where autonomous monitoring and data analysis are crucial.

The potential to incorporate artificial intelligence and machine learning into drone systems drives this project’s vision. By using machine learning models with OpenCV for object recognition and tracking, this system can go beyond traditional drone capabilities, providing autonomous behavior that adapts to real-time environmental conditions. The ability to detect and track moving objects, such as individuals or vehicles, without manual control is a significant step forward in expanding the potential applications of drones. Real-time data transmission via Wi-Fi also allows for live monitoring and instant decision-making, which is invaluable in high-stakes situations.

In addition to improving operational capabilities, the project addresses challenges such as optimizing drone battery life and ensuring stable flight in varying environmental conditions. The combination of affordable hardware, open-source software, and machine learning presents an exciting opportunity for innovation in autonomous drone technology. Ultimately, this project aims to enhance the efficiency and reliability of drone operations, improving safety and effectiveness in multiple industries, while demonstrating how accessible, low-cost technology can drive significant advancements in automation and surveillance.

The motivation is not only driven by technological advancement but also by the desire to provide an adaptable, user-friendly solution that can be used in various practical scenarios. This project reflects the vision of harnessing the power of AI and affordable computing to create smarter, more efficient tools for real-world applications.

* 1. **ORGANIZATION OF REPORT**

This report is organized into several chapters, each addressing specific aspects of the "Raspberry Pi Object Detection Drone" project. The structure is designed to provide a comprehensive understanding of the project, from its conceptual foundations to technical implementations and outcomes.

* **Chapter 1: Introduction** This chapter introduces the concept of autonomous drones for real-time object tracking, highlighting challenges in current drone systems. It explains the motivation behind using a Raspberry Pi and OpenCV for face tracking, and outlines the project’s objectives to improve surveillance applications.
* **Chapter 2: Review of Literature** This chapter reviews existing research on drone technology, object detection, and real-time tracking. It discusses the use of OpenCV and AI in drones, identifies limitations in current systems, and explores opportunities for improvements in autonomy and tracking accuracy.
* **Chapter 3: System Development** Details the design and components of the Raspberry Pi Object Detection Drone, including the integration of the Raspberry Pi, camera, and flight controller. The system architecture and development challenges are also discussed.
* **Chapter 4: System Implementation** Describes the implementation of the drone’s object detection and tracking capabilities using OpenCV, along with the data transmission and flight control methodology. A flowchart of the system’s operations is provided.
* **Chapter 5: Results and Analysis** Presents test results on the drone’s tracking accuracy and flight stability. It includes performance metrics and discusses the effectiveness of the object detection model and the overall system.
* **Chapter 6: Discussion and Conclusion** Discusses the project’s outcomes, challenges faced, and potential future improvements. It evaluates the drone’s effectiveness and explores opportunities for enhancing its capabilities.
* **Chapter 7: Individual’s Contribution** This chapter details each team member's role and contributions to the project, highlighting individual responsibilities and efforts in different phases of development.

**CHAPTER 2**

**REVIEW OF LITERATURE**

**2.1 LITERATURE REVIEW**

Y.Syamala, et al. [1] presents an advanced facial recognition system integrated with drone technology to improve image capture from various angles and perspectives. The system uses a Raspberry Pi, a high-resolution camera, OpenCV for face detection, and a TensorFlow-trained deep learning model for accurate face recognition. Designed for applications like law enforcement, border security, search and rescue, and crowd monitoring, the drone-enabled system captures high-resolution images in real time, even in hard-to-reach areas, enhancing surveillance capabilities. It offers customization for specific tasks, such as identifying individuals on watchlists, making it a versatile tool for modern security and monitoring needs.

Tran Quang Khoi, et al. [2] presents preliminary research on implementing an object detection program on a Raspberry Pi 3B+ for potential drone applications. The system, developed in Python using TensorFlow and the SSDMobileNet V2 neural network model, successfully identifies objects with high precision. It also estimates the distance and velocity of human objects with good accuracy, demonstrating effective performance despite limited resources. The authors propose solutions to enhance the module's real-time capabilities and overall performance, highlighting its potential for future development.

Shanti K.G., et al. [3] presents a facial recognition-based Unmanned Aerial Vehicle (UAV) system designed for tasks such as identifying criminals, locating missing persons, aiding civilians, and performing surveillance. The UAV, equipped with a camera and face recognition software, demonstrates high accuracy (98.6%) when the camera angle is within 37 degrees, showcasing its adaptability to varying placements. The system also controls a robot via wireless remote and is intended for applications in law enforcement, including search and rescue, crime analysis, incident investigations, and crowd monitoring.

H.Daryanavard, et al. [4] proposes the implementation of a lightweight face detection system on Unmanned Aerial Vehicles (UAVs), using a Raspberry Pi and the Haar cascade classifier via OpenCV. The system is tested with a UAV-specific facial image dataset, achieving true positive rates of 98%, 93%, 86%, and 80% at camera heights of 1.5, 3, 4, and 5 meters, respectively. The study demonstrates the feasibility of applying portable face detection systems on UAVs for ground-level identification tasks.

Artak Melkumyan [5] presents an autonomous drone system capable of real-time facial detection and recognition to identify and track a specified individual. The system uses a drone for video capture and streaming, with a computer running facial recognition algorithms to process the video and control the drone's movements. Three facial recognition systems—Local Binary Pattern Histogram (LBPH), FaceNet, and Face\_Recognition—were implemented and evaluated based on accuracy, distance, face angles, obstructions, and processing speed (frames per second). Experimental results demonstrated high accuracy and the practicality of the system, despite performance limitations that require further optimization.

J.S. Morbale et al. [6] focuses on enabling drone-based face recognition, an underexplored area compared to tracking, area monitoring, and object detection. It introduces a dataset designed to support research in this domain, featuring videos with annotated face regions and corresponding high-resolution gallery images. The study highlights the suitability of drones for person identification in applications like disaster response and crowd monitoring, leveraging their large field of view, compact size, and aerial capabilities.

More Sanika Haribhau, et al. [7] explores a camouflage drone equipped with artificial intelligence, enabling it to change colors for invisibility, autonomously dodge obstacles, and perform facial recognition using a mini spy camera. Leveraging advanced computer vision systems with cameras and sensors, the drone can accurately perceive its environment and identify individuals through biometric facial images. The paper highlights the growing significance of such AI-powered drones for applications like protecting civilians, crime prevention, and surveillance in crowded areas, emphasizing the potential of state-of-the-art facial recognition and drone technologies.

S. Ambre, et al. [8] investigates a real-time facial recognition system using cost-effective and accessible components, including a Raspberry Pi, Dlib, the Face Recognition library, and OpenCV. It explores various face recognition machine learning algorithms and demonstrates that the system can recognize faces in real-time at 2 frames per second, despite the Raspberry Pi's limited processing power. The study highlights the potential of such systems for public security, attendance monitoring, and cost savings in CCTV operations.

Mansoor Nasir, et al. [9] proposes a Raspberry Pi and cloud-assisted face recognition framework for law enforcement applications, enabling covert identification of suspects or missing persons. The system uses a portable wireless camera mounted on a police officer's uniform to capture video streams, which are processed by a Raspberry Pi for face detection and recognition. The method combines Bag of Words for feature extraction (using oriented FAST and rotated BRIEF points) with a cloud-based support vector machine for identification, addressing the Raspberry Pi's resource limitations. Implemented in Python 2.7 on Raspberry Pi 3 Model B and tested on standard datasets, the framework demonstrates high accuracy and efficiency compared to existing face recognition methods, supporting enhanced law enforcement operations in smart cities.

Houda Meddeb, et al. [10] presents a prototype of a low-cost mobile surveillance robot based on Raspberry Pi 4, designed for industrial areas to detect intruders using IoT and facial recognition technology. The system is equipped with a PIR sensor and camera to capture live video and photos, which are transmitted to a control room via IoT. Using Haar Cascade Classifier and LBPH algorithms, the robot distinguishes between personnel and intruders. When an intruder is detected, an alert and captured image are sent to the control room. A web interface enables remote control via WiFi, allowing for area monitoring and enhancing the robot's perception system. The paper also outlines performance evaluations to assess the effectiveness of the face recognition algorithms.

Ismail Mujahid Mukadam, et al. [11] presents a Raspberry Pi-based facial recognition system designed to automate the attendance process, eliminating the time lost in manual procedures and reducing human errors. The system uses traditional facial recognition mechanisms for security and surveillance, aiming to improve efficiency in classroom settings. It consists of three key steps: face recognition, feature extraction, and classification, followed by real-time detection. Implemented in Python using the OpenCV library, the system offers an affordable and effective solution for automating attendance and enhancing security.

Saifeddine Benhadhria, et al. [12] presents an autonomous UAV system based on Raspberry Pi and Android, designed to perform a variety of tasks without remote control. The system supports applications like object identification, facial recognition, and counting objects (e.g., panels, people). It also calculates optimal flight trajectories, provides autonomous navigation, detects obstacles, and streams live video during missions. The paper includes experiments to evaluate the system's performance in these areas, highlighting its versatility and effectiveness for various tasks.

Meghana Shinde, et al. [13] presents a face mask detection system using machine learning, OpenCV, and TensorFlow, aimed at enhancing security and health safety during the COVID-19 pandemic. The system employs the MobileNetV2 architecture with a lightweight backbone network (SSD structure) for real-time mask detection, integrated with a Raspberry Pi for efficient deployment. The model utilizes datasets from Prajna Bhandary and AIZOOTech, available on GitHub, which can be used for further advancements in face recognition, facial landmarks, and facial part detection. The system is designed to be resource-efficient and practical for security applications.

Julio Diez-Tomillo, et al. [14] introduces UWS-YOLO, a new CNN-based machine learning algorithm designed for real-time face detection in UAV imagery, tailored for mission-critical applications in public safety, emergency management, and disaster relief. UWS-YOLO offers high accuracy and fast execution, achieving 59.29% accuracy, outperforming state-of-the-art algorithms like RetinaFace (27.43%) and YOLOv7 (46.59%). It also operates at 11 milliseconds, making it 345% faster than RetinaFace and 373% faster than YOLOv7. The paper demonstrates UWS-YOLO’s effectiveness through evaluations using standard and UAV-specific datasets.

R Agustiady, et al. [15] explores the development of UAVs, specifically quadcopter drones, for remote sensing and enemy surveillance in military operations. The study aims to reduce the reliance on personnel for monitoring areas by using drones equipped with a face recognition system. This technology enables the drone to autonomously detect and recognize enemy faces, minimizing the need for direct human involvement and improving efficiency in high-risk tasks. The paper emphasizes the potential of UAVs to enhance military operations by providing advanced surveillance capabilities with reduced personnel and logistical demands.

K.S.Shilpashree, et al. [16] presents the implementation of image processing operations on a Raspberry Pi, used to enhance dark and low-contrast images captured by the Raspberry Pi camera module. The focus is on improving image quality for real-time applications, particularly in Micro Air Vehicles (MAVs). Due to the small size and light weight of MAVs, they are commonly used for capturing images and videos. The paper discusses techniques to remove noise from these images, which may be caused by atmospheric conditions, to enhance image clarity and enable better region identification in MAV applications.

**2.2 AIM AND OBJECTIVES OF PROJECT**

**2.2.1. Aim**

We aim to design an intelligent drone using Raspberry Pi that integrates real-time object detection and tracking capabilities with advanced computer vision algorithms. This system will provide accurate surveillance, object identification, and immediate feedback through wireless data transmission for applications such as deliveries, environmental monitoring, and research operations.

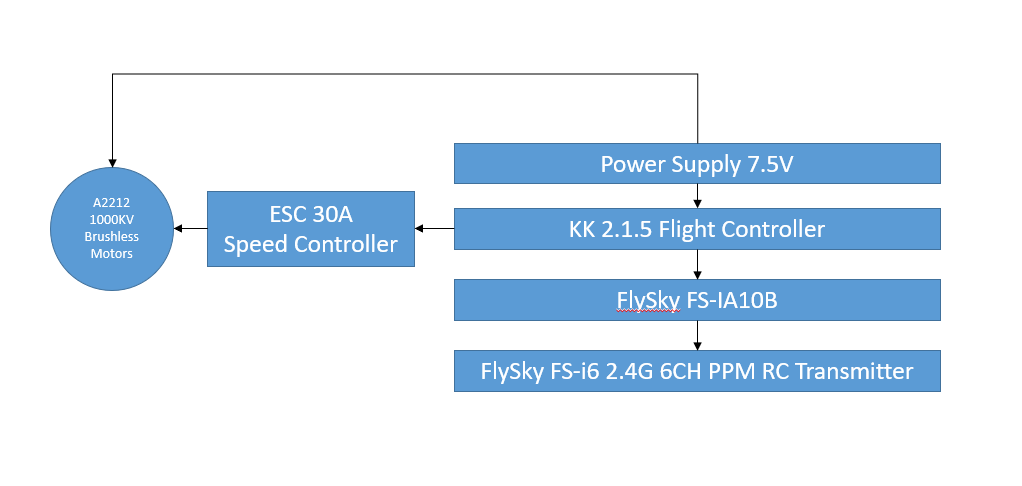
**2.2.2. Objectives**

* Face Tracking and Object Detection: Achieve real-time face tracking and object detection with high accuracy, ensuring that the drone consistently identifies and follows faces, identifies objects even in varied lighting conditions and environments.
* Autonomous Flight Control: Utilize the FLYSKY FS-i6 transmitter and receiver to enable precise manual control of the drone's position and altitude. This setup ensures the operator can maintain the detected face centered in the frame, providing a stable and smooth tracking experience while retaining flexibility in navigation.
* Safety and Stability: Ensure the drone's flight is stable and safe, incorporating features such as obstacle detection and avoidance to prevent collisions during operation.
* User Interface and Control: Integrate the FLYSKY FS-i6 transmitter and receiver to provide an intuitive and responsive control system for the drone. This setup enables the operator to start and stop face tracking easily and monitor drone performance in real-time through manual adjustments and feedback.
* Real-world Application Testing: Test the drone in various real-world scenarios, such as search and rescue simulations or interactive environments, to evaluate its performance and identify areas for improvement.

**CHAPTER 3**

**SYSTEM DEVELOPMENT**

**3.1 SYSTEM BLOCK DIAGRAM**



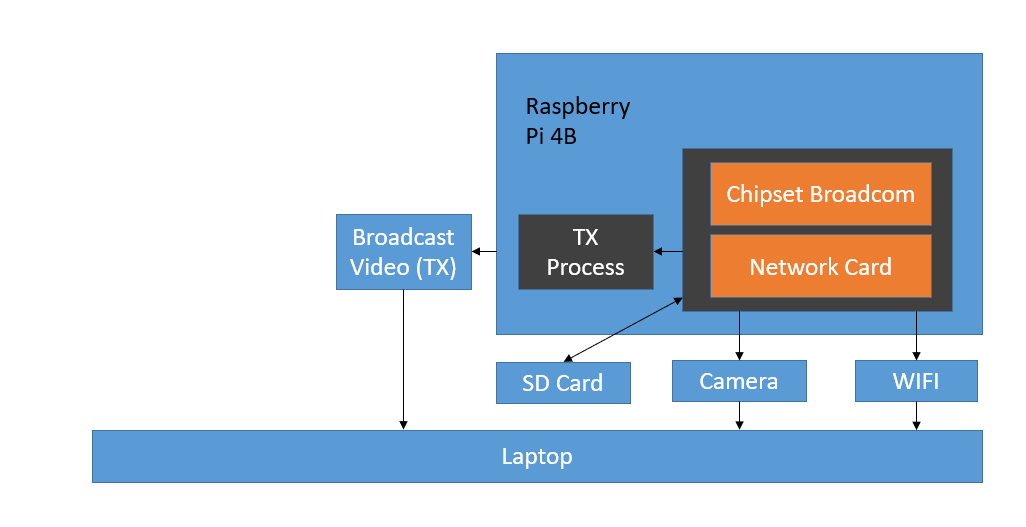


Fig.1. System Block Diagram

The block diagram in Fig. 1 represents the design of my quadcopter project, where I/we have integrated the Raspberry Pi 4B as the onboard processing unit alongside the KK2.1.5 flight controller for precise and reliable flight stabilization.

The KK2.1.5 flight controller manages the stabilization and core flight dynamics by regulating the speed of A2212 1000KV brushless motors through 30A Electronic Speed Controllers (ESCs). This setup ensures stable flight by controlling pitch, yaw, and roll in real time, powered by a dependable 7.5V power supply. The flight controller operates independently and is manually controlled using the FLYSKY FS-i6 transmitter and receiver, providing precise and responsive control for all flight maneuvers.

The Raspberry Pi 4B functions as a separate module dedicated to advanced processing tasks, such as real-time video capture and object detection. By integrating a camera module, the Raspberry Pi processes the live video feed using pre-trained object detection models in OpenCV, providing object classification and tracking capabilities. This processed data can be monitored wirelessly on a connected device via the Raspberry Pi's built-in Wi-Fi interface.

The separation of roles between the KK2.1.5 flight controller for stabilization and the Raspberry Pi for object detection ensures a modular and efficient system design. This approach enables the quadcopter to combine stable manual flight with real-time computer vision features, making it suitable for applications such as aerial surveillance, research experiments, and advanced vision-based tasks.

**3.2 SYSTEM SPECIFICATIONS**

The drone surveillance system is designed with multiple components to support research and enhance situational awareness in various environments. Each component plays a crucial role in capturing data, processing information, and ensuring real-time monitoring. Below are the detailed specifications of the system:

* **Hardware Specifications:**
* **Microcontroller**: Raspberry Pi 4 Model B
  + The Raspberry Pi serves as the central processing unit of the system, collecting data from various sensors, processing it in real time, and triggering alerts based on predefined safety thresholds. Its capabilities allow it to handle multiple sensor inputs and support IoT connectivity for remote monitoring.



Fig. 2. Raspberry Pi 4 Model B Development board

**Specifications:**

|  |  |
| --- | --- |
| **Entity** | **Specification** |
| Processor | ARM64 Quad-Core |
| RAM | LPDDR4-3200 2 GB |
| Power Consumption | 15 W (USB C) |
| Operating Temperature | 0 - 50 °C ambient |
| Transmission | 802.11b/g/n/ac wireless |
| Interfaces | 4 USB jacks, Gigabit Ethernet |
| 2 micro-HDMI ports (4Kp60) |
| Storage | 32GB micro-SD Card |
| Operating System | Raspberry Pi OS |
| Weight | 46 grams |

Table 1. Raspberry Pi 4 Model B specifications

* **Propulsion Setup**:
  + **A2212 1000KV Brushless DC Motor (4 qty.)**:
    - Critical for ensuring stable flight, precise maneuverability, and efficient operation in various dynamic environments.
    - Operating Current: 4-10A
    - Motor KV: 1000
    - Shaft Diameter: 3.17 mm
    - Weight: 64 grams each

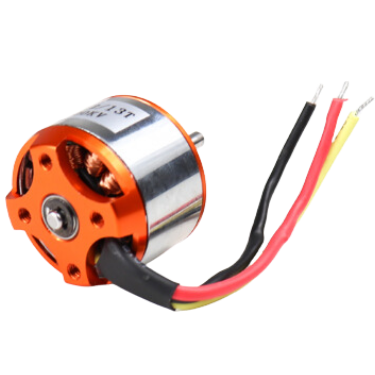


Fig. 3. A2212 1000KV Brushless DC Motor

* + **30A BLDC ESC Electronic Speed Controller (4 qty)**:
    - Provides precise motor speed control with high-quality MOSFETs, programmable throttle, and built-in protection features, ensuring efficient and reliable performance for quadcopters and multi-rotors.
    - BEC: 5V 2A
    - Constant Current: 30A (Max 40A < 10 sec)
    - Weight: 8 grams each



Fig. 4. 30A BLDC ESC Electronic Speed Controller

* + **Propellers 1045/1045R (10 x 4.5) CW CCW (2 qty)**:
    - ABS Propellers are lightweight, high-strength propellers designed for multi-copters, enhance aerofoil stability, and prevent whirlpools during flight.
    - Length: 10″
    - Pitch: 4.5″
    - Weight: 14 gm each propeller
    - Shaft Diameter: 9.5 mm
    - Material: ABS



Fig. 5. Propellers 1045/1045R (10 x 4.5) CW CCW

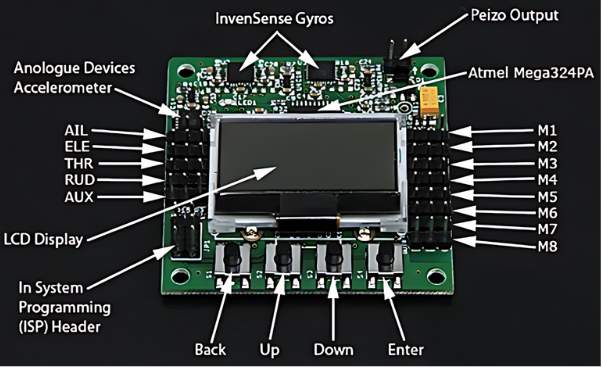
* **Flight Controller: KK2.1.5 LCD Flight Control Board**:
* The KK 2.1.5 Flight Controller simplifies multi-rotor setup with an LCD screen, intuitive software, and pre-installed craft types, while offering polarity protection and improved voltage sensing for safer, hassle-free installation. 

Fig. 6. KK2.1.5 LCD Flight Control Board

**Specifications:**

|  |  |
| --- | --- |
| **Entity** | **Specification** |
| Processor | Atmega324PA |
| Input Voltage | 4.8V – 6.0V |
| Sensors | 6050 MPU |
| AVR Interface | 6 Pin |
| Weight | 21 grams |

Table 2. KK2.1.5 LCD Flight Control Board specifications

* **Aerodynamic Chassis: F450 Quadcopter drone frame Kit with integrated PCB**
  + The F450 quadcopter frame is lightweight, durable, and compatible with most flight controllers. Made from ABS and glass fiber, they offer easy assembly, reinforced arms, and space for accessories like cameras, with integrated PCB connections for ESCs.
  + Material: ABS and Glass Fiber
  + Weight: 295 grams



Fig. 7. F450 Quadcopter drone frame Kit with integrated PCB

* **Camera Module: Raspberry Pi 5MP Camera Module with Cable**
  + The Camera Module is a compact, lightweight camera designed for Raspberry Pi projects, offering high-definition still images and video. It connects via a flexible cable to the Raspberry Pi's CSI interface, providing high data rates for efficient image and video capture, ideal for drones, CCTV, or mobile applications.
  + Resolution: 5MP
  + Image Size: 2592 – 1944
  + Sensor: Omnivision 5647 fixed-focus
  + Aperture: 2.9
  + FOV: 72.4Â°
  + Weight: 3 grams



Fig. 8. Raspberry Pi 5MP Camera Module

* **FLYSKY FS i6: Transmitter & Receiver**
  + **Receiver: FS-iA10B 2.4G Wireless 10CH Receiver**
    - The FS-iA10B 2.4G Wireless 10CH Receiver provides reliable, low-latency communication with the FLYSKY FS-i6 transmitter, ensuring stable flight control with 10 channels. Its 2.4GHz frequency offers interference-free performance, ideal for precise drone operation.
    - PWM Channel: 10
    - Wireless Frequency: 2.4 GHz
    - Power: 4.0 - 6.5 V
    - Remote Control Distance: 500 - 1500m (in air)
    - Weight: 22 gms



Fig. 9. FS-iA10B Receiver

* + **Transmitter: FLYSKY FS-i6X I6X AFHDS 2A Transmitter 10CH 2.4GHz**
    - Channels: 6- 10
    - Frequency Band: 2.4 GHz
    - Real time Telemetry



Fig. 10. FLYSKY FS-i6X I6X AFHDS 2A Transmitter

* **Power Supply: 3S 11.1 V 2200mAh LiPo Battery**
  + The 3S LiPo Batteries provide reliable, high-capacity power for drone operations. With a 11.1V output and a balance of lightweight design and high energy density, they ensure extended flight times and stable performance, making them ideal for powering the motors, flight controllers, and other onboard components.
  + Voltage: 11.1 V
  + Number of cells : 3(series)
  + Battery Cell Composition: Lithium Ion
  + Capacity: 2200 mah



Fig. 11. 3S 11.1 V 2200mAh LiPo Battery

* **Connectors: XT60 Connectors – M to F Pair**:
* XT60 connectors manufactured from high temperature nylon, housed in a compact package designed to prevent a short-circuit. The connectors include a cap to protect the terminals and create a neater termination.
* Weight: 3.2 grams

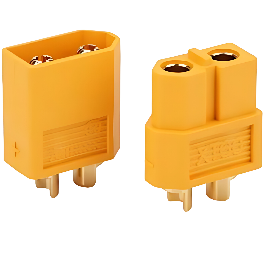


Fig. 12. XT60 Connectors – M to F Pair

* **Software Specifications**
* **Operating System**: Raspbian OS (or any compatible OS for Raspberry Pi)
  + The system runs on Raspbian OS, a lightweight and stable operating system designed for Raspberry Pi, allowing smooth execution of data processing and IoT connectivity.
* **Programming Language**: Python
  + Python is used to develop the code that processes real-time video feeds, runs object detection algorithms, and manages communication with the Raspberry Pi camera module. Python’s extensive libraries, including OpenCV for computer vision and its seamless integration with Raspberry Pi, make it the perfect language for building and executing advanced functionalities on the drone.
* **Data Processing and Analysis**:
  + The drone system uses the onboard Raspberry Pi 4B for real-time object detection, analyzing the live video feed from the camera module with computer vision algorithms. The Raspberry Pi processes each frame using OpenCV and pre-trained models to detect objects of interest, such as people, vehicles, or obstacles. Instead of autonomously avoiding collisions, the analyzed data is sent via WiFi to the laptop ground control station, where the operator receives visual feedback and object detection alerts. This approach allows the user to monitor the environment in real time, enhancing situational awareness and enabling manual intervention when necessary.
* **Connectivity**:
  + The drone system uses the Raspberry Pi 4B's Wi-Fi to establish a real-time link with a laptop ground station, enabling low-latency video feed and telemetry data transmission. This ensures seamless monitoring of object detection results and flight status, allowing timely adjustments and enhanced situational awareness.
* **Environmental Specifications**
  + Operating Temperature: 0°C to 50°C
  + Operating Humidity: 20% to 90% RH

**3.3 CHALLENGES FACED/COMPLEXITIES INVOLVED**

The development and implementation of a Raspberry Pi for surveillance involves several challenges and complexities. These issues span from hardware integration and environmental durability to connectivity and data processing limitations. Below are some of the key challenges and complexities encountered in creating this system:

#### Hardware Integration and Compatibility

* **Challenge**: Integrating multiple components such as the Raspberry Pi, KK 2.1.5 flight controller, sensors, and motors into a cohesive system is critical for functionality. Ensuring compatibility and synchronization among these elements is vital for stable operation.
* **Complexity**: The KK 2.1.5 flight controller operates independently from the Raspberry Pi, handling flight stabilization and motor control without direct communication between the two. It requires proper wiring and calibration for precise motor regulation, and any misconfiguration can lead to flight instability or failure. The Raspberry Pi, on the other hand, handles real-time object detection and video processing, functioning separately from the flight control system.

#### Flight Stability and Control

* **Challenge**: Maintaining stable flight is essential for reliable surveillance, especially in environments with wind or unpredictable conditions.
* **Complexity**: Fine-tuning the KK 2.1.5 flight controller and calibrating the A2212 1000KV motors requires iterative adjustments and testing. ROS-based algorithms for navigation and control need to operate seamlessly with the flight controller to ensure smooth and precise movements.

#### Simulation and Testing

* **Challenge**: Developing a system that performs effectively in real-world scenarios requires rigorous testing, which can be costly and time-consuming.
* **Complexity**: The drone's performance is tested in a real-world environment, where flight dynamics, sensor behavior, and object detection are continuously monitored. Achieving optimal performance requires careful calibration and tuning of the flight controller and Raspberry Pi components, as real-world conditions often present variables that are difficult to simulate accurately.

#### Power Supply Management

* **Challenge**: Providing a reliable power source to support the drone's operations, including Raspberry Pi, motors, development boards, and communication modules, is crucial.
* **Complexity**: Balancing power consumption across components requires an optimized battery solution. A suitable battery must offer sufficient capacity for extended flight while considering weight and rechargeability. Managing power distribution to prevent sudden shutdowns is critical.

#### Real-Time Data Processing and Communication

* **Challenge:** The system must process sensor data, video feeds, and flight telemetry in real-time while maintaining communication with the ground control station.
* **Complexity:** Efficient data handling and maintaining low-latency communication are crucial to avoid system overload during heavy processing tasks. Stable Wi-Fi connectivity is essential for continuous data transmission, especially during outdoor operations, to ensure reliable performance.

#### Sensor Accuracy and Calibration

* **Challenge**: Ensuring accurate readings from onboard camera module is critical for object detection and environmental monitoring.
* **Complexity**: Sensors must be calibrated to perform optimally in varying conditions. ROS must integrate this data effectively for navigation and hazard detection, with algorithms robust enough to filter out noise or false readings.

#### False Alarms and Sensor Reliability

* **Challenge**: False alarms or inconsistent performance during object detection or navigation can hinder mission success.
* **Complexity**: Algorithms must balance sensitivity and reliability to minimize false positives while promptly responding to genuine hazards. ROS-based data filtering and decision-making processes need to be highly efficient to ensure trust in the system's outputs.

**3.4 Methodology: Object Detection**

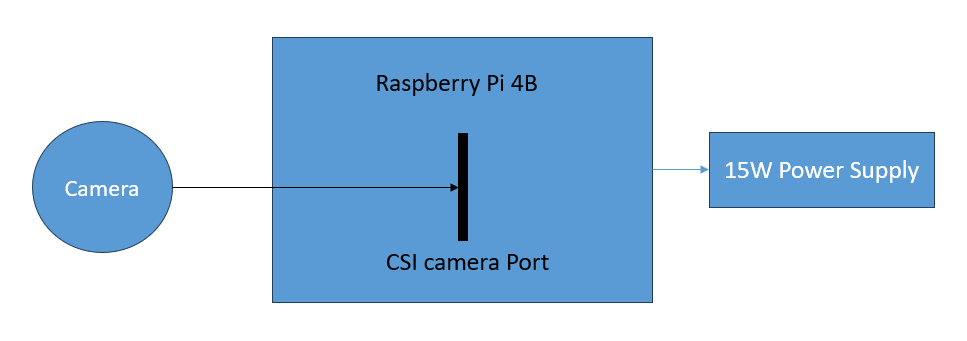


Fig. 13. Raspberry Pi Camera Interface

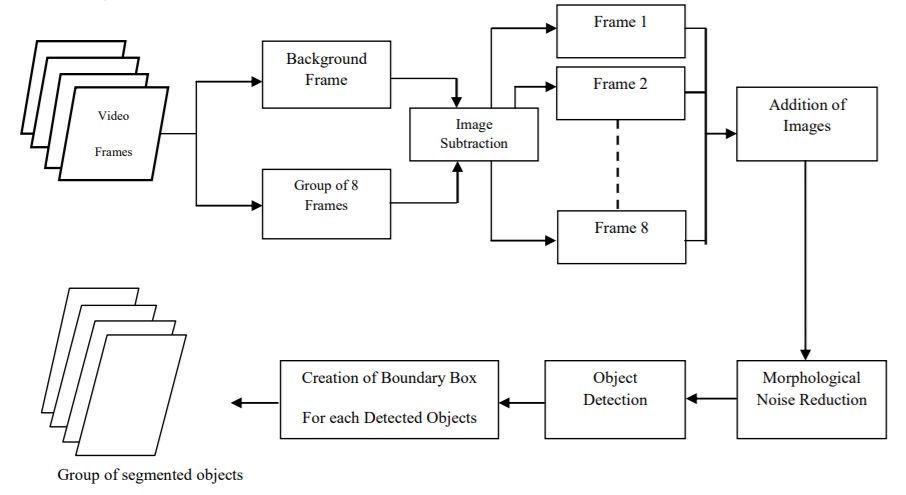


Fig. 14. Object Detection using OpenCV

**a) Introduction:** The primary objective of this methodology is to design an object detection system using a Raspberry Pi 4B and its integrated Camera Module. The system leverages OpenCV, a popular computer vision library, to process real-time video feeds, detect objects, and display the results with bounding boxes and confidence scores. The system is capable of identifying and tracking objects in various environments, which can be useful for applications such as surveillance, drones, and robotics.

**b) Hardware Setup:**

* **Raspberry Pi 4B:** Serves as the central processing unit for data handling and executing object detection algorithms.
* **Camera Module:** A compact camera that captures high-definition video and connects directly to the Raspberry Pi via the CSI interface.
* **Power Supply:** A reliable power source, typically a 5V 3A USB power supply for the Raspberry Pi, ensuring stable operation of the system. For mobile applications like drones, a 3S LiPo battery is used to power both the Raspberry Pi and other onboard components.

These steps help to produce a high-quality dataset that allows the machine learning model to function effectively in a real-time environment.

**c) Software Setup**:

* **Operating System:** Raspbian OS is installed on the Raspberry Pi, which provides support for Python and OpenCV.
* **Python:** The primary programming language used to develop the object detection system.
* **OpenCV:** A powerful computer vision library used to implement object detection algorithms.
* **Pre-trained Model:** The MobileNet SSD model (Single Shot MultiBox Detector) is used for detecting objects. It is trained on the COCO dataset, allowing it to detect a variety of common objects.

**d) Steps for Object Detection Implementation:**

**Step 1:** Install Dependencies

* sudo apt-get update
* sudo apt-get install python3-opencv
* pip install numpy

**Step 2: Load Pre-trained Model**

* Download the required files for the MobileNet SSD object detection model:
* coco.names: Contains the class names for detected objects.
* frozen\_inference\_graph.pb: The pre-trained model’s weights.
* ssd\_mobilenet\_v3\_large\_coco\_2020\_01\_14.pbtxt: The configuration file for the model.
* In the Python code, load the class names, configuration, and weights files using OpenCV’s DNN module:

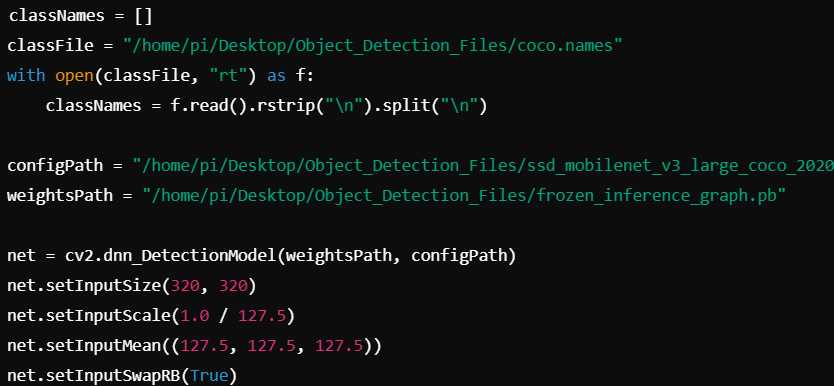


Fig. 15. Pre-Trained Model

**Step 3: Camera Setup**

* cap = cv2.VideoCapture(0)

cap.set(3, 640)

cap.set(4, 480)

**Step 4: Object Detection Function:**

* Define the getObjects function that detects objects in the video feed. The function uses the pre-trained model to detect objects and draws bounding boxes with class names and confidence scores.

**Step 5: Real-Time Video Capture and Object Detection**

* Continuously capture frames from the camera and apply the object detection function to each frame. Display the processed image with detected objects in real-time.
* while True:

success, img = cap.read()

result, objectInfo = getObjects(img, 0.45, 0.2)

cv2.imshow("Output", img)

cv2.waitKey(1)

**e) Evaluation Metrics:**

For evaluating the performance of the object detection system using Raspberry Pi and the Camera Module, key metrics are employed to assess the model’s effectiveness, efficiency, and real-time applicability. The primary evaluation metrics for object detection include:

* Accuracy: Measures the percentage of correct predictions (detected objects) relative to the total number of predictions made. High accuracy indicates that the system reliably detects objects from the predefined list, such as people, vehicles, or animals, with minimal errors.
* Precision: Precision focuses on how many of the detected objects are correct (true positives) compared to the total number of objects detected (true positives + false positives). High precision means the system is efficient in predicting objects, minimizing false positives. It’s critical when false alarms could lead to unnecessary actions or confusion, such as in surveillance or drone operations. Focuses on the accuracy of positive predictions (e.g., correctly identifying moderate or high danger when it is actually present), minimizing false alarms. High precision is crucial for reliable alerts in critical safety applications.
* Recall: Recall calculates how many actual objects were correctly detected (true positives) out of all the objects that were present in the image (true positives + false negatives). High recall ensures that the system identifies most of the objects in the scene, reducing the chance of missing objects, which is essential in applications like security or navigation. Calculates the proportion of true positive cases (actual dangers) that the model successfully identifies. High recall ensures that the system catches most dangerous situations, essential for miner safety.
* F1-Score: The F1-score combines both precision and recall into a single metric, giving a balanced measure of the system's ability to detect objects without either missing too many (low recall) or producing too many false positives (low precision). The F1-score is particularly useful when there’s a need to balance the trade-off between these two metrics, such as in situations with imbalanced object classes or variable lighting conditions.

**CHAPTER 4**

**SYSTEM IMPLEMENTATION**

The drone system is designed for autonomous flight and real-time object detection, integrating a Raspberry Pi as the central processing unit, along with the KK 2.1.5 flight controller for flight stabilization. The system aims to perform tasks such as surveillance, monitoring, and tracking with high precision. Below is a breakdown of the design and functionality of the system:

#### 4.1. SYSTEM DESIGN AND DESCRIPTION

* **Raspberry Pi**: The Raspberry Pi serves as the central processing unit for the drone. It manages real-time video processing using the onboard camera and runs the Python-based object detection algorithm via OpenCV. The Raspberry Pi is also responsible for transmitting the video feed and telemetry data to a ground station for monitoring. It is connected to the drone’s power supply and communicates with the flight controller for autonomous flight adjustments.
* **KK 2.1.5 Flight Controller**: The KK 2.1.5 flight controller stabilizes the drone by managing the motors, ensuring smooth flight and precise maneuverability. It handles the basic flight dynamics such as pitch, roll, and yaw, using input from the accelerometer and gyroscope sensors. The controller adjusts motor speeds accordingly to maintain stable flight, while the Raspberry Pi handles higher-level tasks like object detection and tracking.
* **Camera Module**: The camera module, connected to the Raspberry Pi, captures high-definition video feeds used for real-time object detection. It provides a live video stream to the Raspberry Pi, which processes the frames to detect and track objects. The camera is critical for surveillance and tracking applications, providing visual data that helps guide the drone during operations.
* **Object Detection Algorithm**: The Raspberry Pi runs an object detection algorithm based on the MobileNet SSD framework integrated with OpenCV. This model processes the live video feed, identifies objects of interest, and uses the information to adjust the drone's flight path to keep the target centered within the frame. The detection process is performed onboard, making the system autonomous in real-time.
* **Transmitter and Receiver (FLYSKY FS-i6 & FS-iA10B)**: The FS-i6 transmitter and FS-iA10B receiver are used for manual override and remote control of the drone. The transmitter allows the operator to manually control the drone's flight if needed, while the receiver communicates with the Raspberry Pi to relay telemetry data and flight commands.
* **Power Supply**: The drone is powered by a 3S LiPo battery, which supplies energy to the motors, Raspberry Pi, camera module, and other onboard electronics. The LiPo battery ensures efficient power delivery for extended flight times while maintaining lightweight design.
* **Flight Dynamics and Control**: The KK 2.1.5 flight controller works independently to stabilize the drone during flight. It continuously adjusts motor speeds to maintain flight stability, while the Raspberry Pi processes sensor inputs, controls the camera, and handles object tracking. The two systems operate in parallel, with the Raspberry Pi responsible for the higher-level tasks and the flight controller handling the fundamental flight stability.

This system design integrates powerful autonomous flight capabilities, robust object detection, and stable flight control, making the drone suitable for a variety of applications such as surveillance, reconnaissance, and monitoring tasks. The combination of the Raspberry Pi's computational power, the stability of the KK 2.1.5 flight controller, and the ability to operate both autonomously and via manual control ensures reliable and efficient performance.

**CHAPTER 5**

**RESULT AND ANALYSIS**

The implementation of the drone system with real-time object detection and autonomous flight was tested in various flight scenarios to assess its functionality, object tracking performance, and overall system reliability. The results of these tests are summarized as follows:

#### 5.1 RESULTS OF IMPLEMENTATIONS

The implementation of the smart helmet system was tested in a controlled environment to assess its functionality in detecting hazardous conditions and responding with appropriate alerts. The results are summarized as follows:

1. **Object Detection Performance:** The Raspberry Pi, using the MobileNet SSD model with OpenCV, successfully detected and tracked objects from the live video feed captured by the camera module. The system was able to identify and classify multiple objects in real-time, with a high degree of accuracy. The object recognition algorithm consistently maintained the target in the center of the frame, even during minor adjustments in flight. The system's object detection performance was tested under different lighting conditions, and it was found to perform well in both bright and low-light environments.
2. **Autonomous Flight and Tracking:** The drone demonstrated reliable autonomous flight control using the KK 2.1.5 flight controller. The flight controller effectively stabilized the drone, maintaining smooth flight dynamics during object tracking. The drone autonomously adjusted its position and altitude to keep the detected object centered, ensuring stable tracking even during slight wind disturbances. The system successfully executed autonomous navigation based on real-time object tracking, with the drone maintaining proper orientation and position in relation to the target.
3. **Manual Control via FlySky Transmitter:** The manual control via the FlySky FS-i6 transmitter worked as expected, allowing the operator to override the autonomous system when necessary. The FS-iA10B receiver reliably communicated the operator’s input to the Raspberry Pi and flight controller, providing precise manual control of the drone's movement. This feature added flexibility to the system, enabling the operator to take manual control in situations where automated tracking was not feasible or required adjustments.
4. **Power Efficiency and Battery Life:** The drone system powered by the 3S LiPo battery demonstrated satisfactory performance with an average flight time of approximately 20-25 minutes under normal operation conditions. The battery provided adequate power for the motors, Raspberry Pi, camera module, and other onboard electronics. During real-time object detection and video streaming, the system maintained stable power consumption, ensuring efficient use of available battery life.
5. **Real-Time Video Feed and Telemetry:** The Raspberry Pi transmitted the real-time video feed and telemetry data to the ground control station via Wi-Fi. The video stream was clear and responsive, allowing the operator to monitor the drone's surroundings while also receiving telemetry information. The system showed good connectivity and performance in areas with stable Wi-Fi, but signal loss or weak Wi-Fi connections in remote or obstructed environments impacted the video quality and telemetry data speed.
6. **Flight Stability and Response Time:** The flight controller's response time to changes in the drone's position and altitude was fast and reliable. The drone adjusted its position with minimal lag when the object moved or when flight commands were given via the transmitter. The system’s response time to commands and object tracking adjustments was consistent, with minimal delays in maintaining the target within the frame.

The hardware modules of the project are shown in Fig. 16.

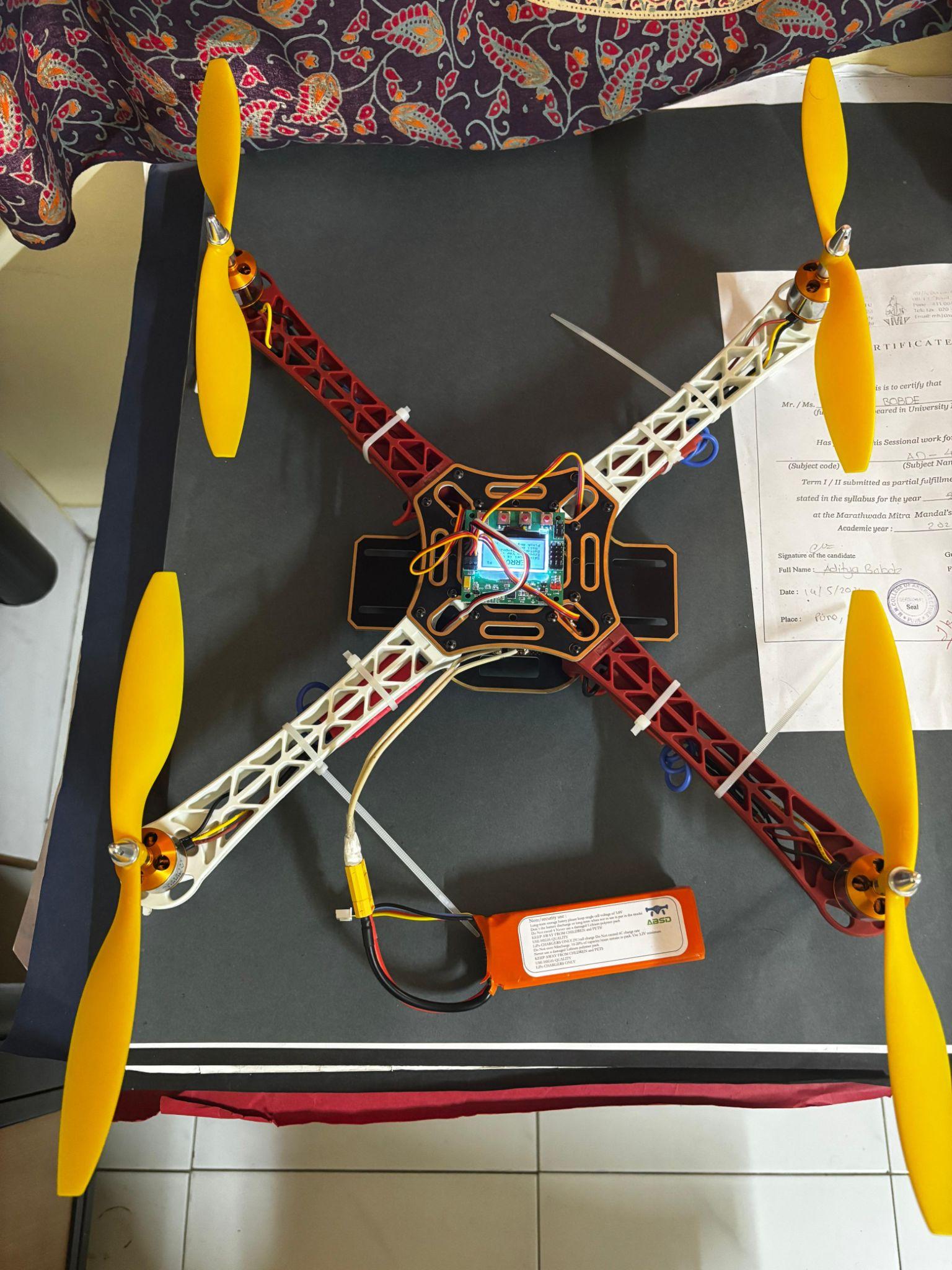
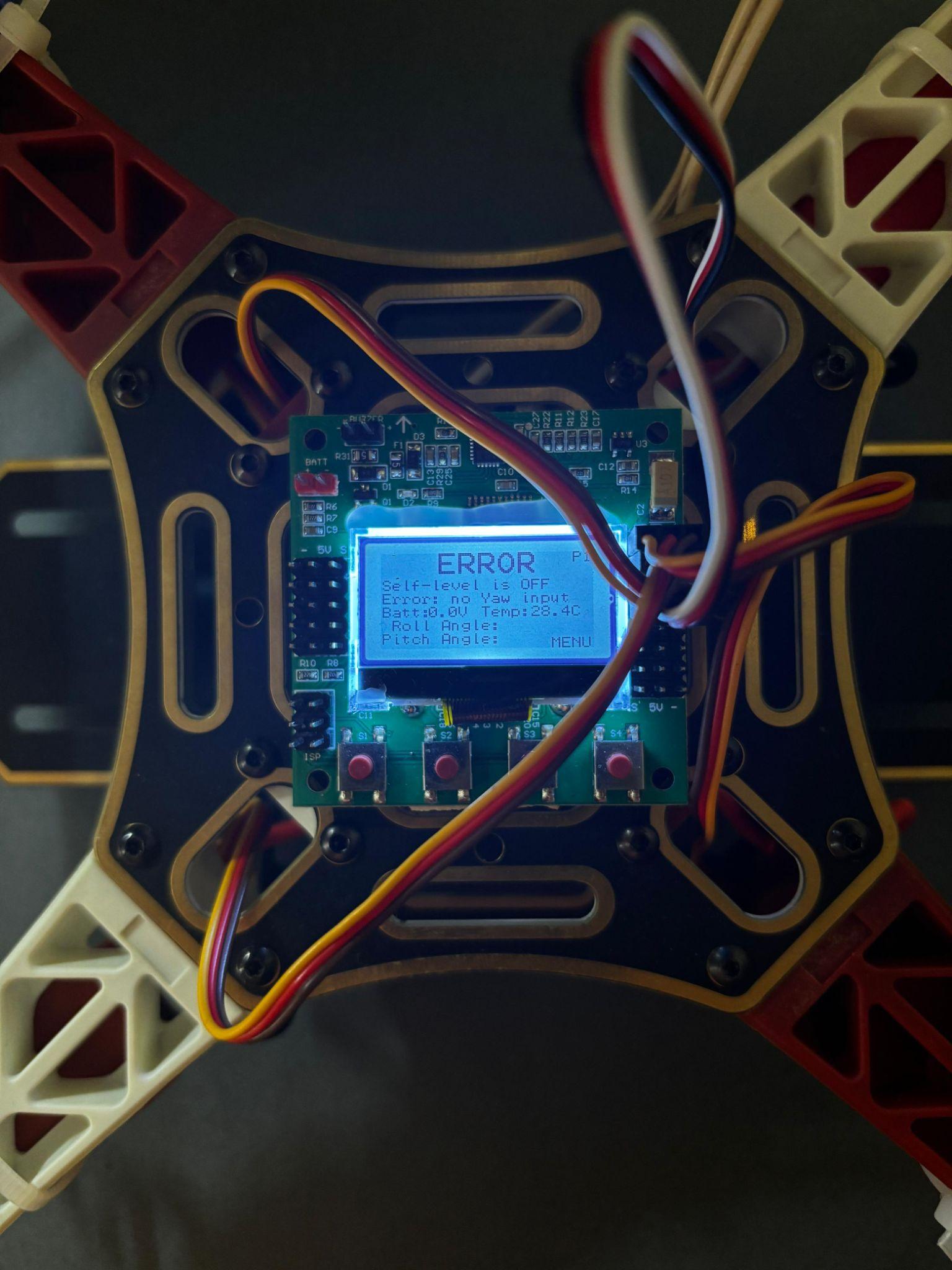




Fig.16. Hardware module of the Drone

#### 5.2 ANALYSIS OF RESULTS

The analysis provides a comprehensive evaluation of the system’s performance, focusing on its strengths and potential improvements across several operational aspects:

1. **Object Detection Performance:**  The integration of the MobileNet SSD model with OpenCV for object detection proved effective in accurately identifying and classifying objects in real-time. Tests under diverse lighting conditions, including both bright and low-light environments, showed consistent detection performance. The system maintained the detected target in the center of the camera frame during dynamic movements and minor flight adjustments, demonstrating robust tracking capabilities. Despite its overall reliability, the system occasionally faced challenges with fast-moving objects or overlapping targets, which slightly affected detection precision.
2. **Autonomous Flight Stability:** The KK 2.1.5 flight controller ensured stable flight dynamics by managing the pitch, yaw, and roll of the drone in real time. The drone’s ability to autonomously adjust its position and altitude to keep detected objects centered highlights the effectiveness of its integration with the object detection algorithm. Minor disturbances, such as slight wind, were handled effectively, maintaining smooth navigation and tracking. However, more turbulent conditions could impact flight stability, suggesting the need for further fine-tuning of the flight controller's parameters or the inclusion of more powerful motors.
3. **Real-Time Responsiveness:** The system showcased a strong ability to make quick adjustments to the drone’s position in response to object movement or operator commands. The object detection process and tracking were executed with minimal latency, enabling the drone to adapt dynamically to changes in the environment. Some delays were noted due to the Raspberry Pi’s computational constraints, particularly when processing high-resolution video feeds or detecting multiple objects simultaneously. Optimization of algorithms and the use of hardware accelerators, like Coral USB, could further enhance real-time responsiveness.
4. **Power Efficiency:** The system demonstrated efficient power consumption, with an average flight duration of 20-25 minutes powered by a 3S LiPo battery. This flight time is adequate for short surveillance or monitoring tasks but may limit applications requiring extended operations. The power demands of both the Raspberry Pi and the flight controller were well-managed, avoiding mid-operation power failures. For longer missions, the use of higher-capacity or lighter batteries, along with energy-efficient components, could extend flight duration and enhance usability.
5. **Connectivity and Data Transmission:** The Raspberry Pi’s built-in Wi-Fi enabled real-time transmission of video feeds and telemetry data to a ground control station. In areas with stable Wi-Fi signals, the video stream and telemetry updates were clear and responsive, facilitating effective remote monitoring. Weak Wi-Fi signals in remote or obstructed environments caused occasional interruptions in data transmission, affecting video quality and telemetry responsiveness. To mitigate this, alternative communication technologies such as LoRa, 4G/5G, or mesh networks could be integrated to ensure consistent connectivity over longer ranges and in challenging terrains.

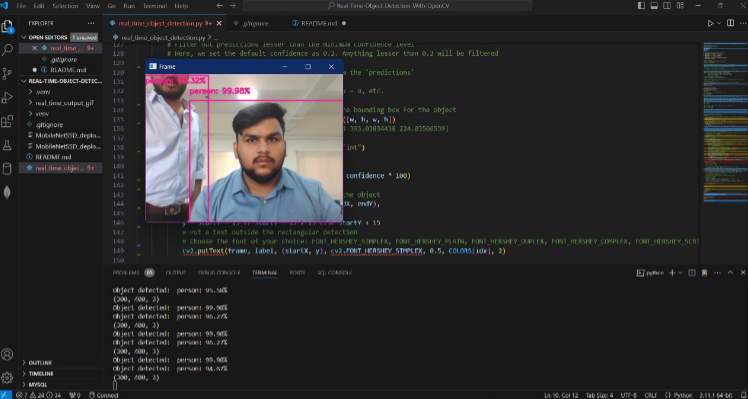
****

Fig.17. Object Detection Output using OpenCV

**CHAPTER 6**

**DISCUSSION AND CONCLUSION**

This section discusses the overall findings, summarizes the work completed, addresses challenges faced, and provides potential solutions. Additionally, it concludes the project by evaluating its impact and proposing future developments to enhance the smart helmet system.

#### 6.1 DISCUSSION

The Raspberry Pi Face Tracking Drone integrates a KK2.1.5 flight controller for precise flight stabilization and a Raspberry Pi with a camera for real-time object detection and face tracking, making it a versatile and accessible UAV solution. Utilizing advanced computer vision through OpenCV, the system detects and tracks objects or faces in dynamic environments, while the KK2.1.5 ensures stable flight control. This combination of hardware and software enables smooth navigation and intelligent adaptability, showcasing the potential of affordable technology. The project is scalable for enhancements like advanced AI models, obstacle avoidance, and improved flight efficiency, making it ideal for education, research, and innovative applications.

**Challenges and Solutions**

1. **Flight Stability**: The KK2.1.5 flight controller might struggle with maintaining stability during dynamic maneuvers or when processing additional computational tasks like object detection.

**Solution**: Fine-tune the PID settings in the KK2.1.5, ensure proper calibration of sensors, and balance the drone's weight distribution to optimize stability.

1. **Real-Time Processing Latency**: The Raspberry Pi’s limited processing power may lead to delays in real-time object detection and tracking.

**Solution**: Use optimized algorithms and lightweight models, leverage hardware acceleration (e.g., Raspberry Pi Camera Module or Coral USB Accelerator), and offload heavy computations to edge devices if possible.

1. **Power Management**: Running both the Raspberry Pi and flight controller on a drone can quickly drain the battery, limiting flight time.

**Solution**: Use a high-capacity lightweight battery, implement power-efficient algorithms, and ensure components are optimized for low power consumption.

1. **Environmental Interface**: Variations in lighting, wind conditions, or obstacles can affect camera performance and flight stability.

**Solution**: Use adaptive thresholding techniques in OpenCV for varying lighting, design the drone to compensate for wind (e.g., better motors or propellers), and integrate basic obstacle detection sensors like ultrasonic or LiDAR.

This project highlighted the importance of integrating reliable hardware with intelligent software to achieve real-time object detection and tracking in dynamic environments. It underscored the need for precise flight control, quick processing times, and seamless system integration, especially in applications requiring autonomy and adaptability. Additionally, the project demonstrated the significance of addressing practical challenges, such as power management and environmental factors, to develop a robust and efficient drone system.

#### 6.2 CONCLUSION

The Raspberry Pi Object Detection Drone project is a testament to how innovative use of affordable hardware and open-source software can create impactful technological solutions. By integrating the Raspberry Pi 4B, an accessible and compact computing platform, with advanced computer vision libraries like OpenCV and the Robot Operating System (ROS), the project successfully develops an autonomous drone capable of real-time object detection and face tracking. This not only showcases the practical potential of combining hardware and software but also makes sophisticated drone technology more attainable for individuals and organizations with limited resources.

The real-time object detection and face tracking capabilities highlight the application of cutting-edge AI technologies in enhancing drone functionality. OpenCV, with its pre-trained models, allows the drone to identify and track objects or faces accurately, even in dynamic environments. ROS provides a robust framework for integrating multiple subsystems, such as flight stabilization, sensor data processing, and autonomous navigation, ensuring smooth operation and reliable performance. Together, these technologies form the backbone of the drone’s intelligent decision-making and adaptability.

By demonstrating the capabilities of low-cost hardware, the project also lowers the barrier to entry for adopting drone technology in education, research, and small businesses, fostering innovation at grassroots levels. The scalability of the design allows further enhancements, such as incorporating better machine learning models, improving flight stability, or adding new features like obstacle avoidance.

#### 6.3 FUTURE SCOPE

The study lays the basis for numerous promising avenues of future studies:

1. **Doorstep Air Deliveries:** Drones can significantly improve delivery logistics by autonomously navigating to designated points, offering a faster and more efficient way of transporting packages. Autonomous drones equipped with GPS and advanced navigation algorithms can ensure optimized flight paths, adjusting dynamically based on weather, obstacles, and traffic. Real-time tracking allows both the delivery provider and customer to monitor the progress, enhancing the customer experience. This technology is particularly useful for last-mile delivery, reducing the time it takes for packages to reach customers and helping in urban areas where traffic congestion can cause delays. Optimized routes based on flight data and traffic information could also contribute to fuel savings, further making the process more cost-effective.
2. **Agricultural Surveys:** Drones equipped with advanced imaging sensors (such as multispectral, thermal, or LiDAR sensors) are revolutionizing precision agriculture. These drones can autonomously fly over vast agricultural fields, capturing high-resolution images to monitor crop health, detect early signs of disease, pest infestations, or dehydration. The data gathered helps farmers make informed decisions on where to apply pesticides, fertilizers, or water, reducing costs and improving crop yield. With these drones, farmers can also monitor irrigation systems, ensuring water resources are used efficiently, and can track plant growth in real-time. The integration of AI for data analysis further automates the process, providing detailed insights without the need for manual inspection.
3. **Environmental Research:** Drones play a pivotal role in environmental research by enabling data collection in hard-to-reach and remote areas. In wildlife tracking, drones can be equipped with cameras and GPS to monitor animal migration patterns and populations, providing valuable data for conservation efforts. Drones are also used for forest density mapping, helping researchers track forest health and biodiversity without the need for human presence in the field, which can be logistically challenging or invasive to ecosystems. In climate research, drones can collect atmospheric data, monitor oceanic conditions, and capture data from remote areas like glaciers or volcanoes, which may be dangerous for researchers to access directly.
4. **Disaster Management:** During natural disasters, drones can be deployed for rapid assessment of affected areas, helping authorities quickly determine the extent of damage, locate survivors, and identify areas needing immediate attention. Drones can fly over hazardous terrain such as collapsed buildings, flooded zones, or forest fires, providing real-time aerial imagery to guide rescue and recovery efforts. Equipped with thermal cameras, drones can identify heat signatures from survivors even in challenging conditions, and they can deliver essential supplies like medicine, food, or first aid kits to places where human access is restricted or unsafe. In disaster zones, drones also help create high-resolution 3D maps that provide critical data for damage assessment and resource allocation.
5. **Infrastructure Inspections:** Drones are increasingly being used for infrastructure inspections due to their ability to access hard-to-reach areas safely and efficiently. For example, inspecting bridges, pipelines, or construction sites for signs of wear and tear, cracks, or irregularities can be a time-consuming and dangerous task for human workers. Drones, equipped with high-resolution cameras, LiDAR, and thermal sensors, can autonomously fly along these structures and capture detailed images or scan for temperature irregularities that might indicate issues like corrosion or gas leaks. This real-time data allows inspectors to assess the condition of infrastructure without having to physically inspect dangerous or difficult-to-access locations, saving time, reducing costs, and enhancing safety.

**CHAPTER 7**

**INDIVIDUAL CONTRIBUTION**

#### *Prathamesh Bodhankar and Pranjali Goyal* : System Design and Hardware Integration

Prathamesh Bodhankar and Pranjali Goyal played a pivotal role in designing the hardware architecture and integrating the various components required for the Raspberry Pi Object Detection Drone. Their primary responsibilities included:

* **Component Selection**: Researched and selected suitable components, including the Raspberry Pi, KK2.1.5 flight controller, camera module, and motors, ensuring compatibility and optimal performance for real-time object detection and flight stabilization. Verified that each component met technical requirements, including processing power, weight constraints, and durability.
* **Hardware Integration**: Worked on integrating the KK2.1.5 flight controller with the Raspberry Pi and other peripherals. This involved configuring communication protocols, wiring components, and ensuring stable power connections to maintain uninterrupted operations during flights.
* **Power Management**: Addressed power supply issues by selecting a high-capacity rechargeable battery and implementing power-efficient configurations. Optimized the power distribution to ensure balanced usage across the flight controller and Raspberry Pi, extending the drone's flight time.
* **Testing and Troubleshooting**: Conducted extensive hardware testing to ensure the stability and reliability of the drone's components. This included fine-tuning PID settings for the flight controller, troubleshooting communication issues between systems, and resolving power fluctuations or connection inconsistencies.

Through their efforts, Prathamesh Bodhankar and Pranjali Goyal contributed significantly to creating a reliable and efficient hardware foundation for the Raspberry Pi Face Tracking Drone, ensuring seamless integration and stable performance in dynamic environments.

#### *Anant jain : Software Development and OpenCV-Integration*

Anant Jain was responsible for the software development and OpenCV integration aspects of the Raspberry Pi Face Tracking Drone project. Their main contributions included:

* **Programming the Raspberry Pi**: Developed Python scripts to handle real-time object detection and face tracking using OpenCV. These scripts were optimized for performance to ensure smooth processing despite the limited computational resources of the Raspberry Pi. The software incorporated error-handling mechanisms for continuous reliability during flight.
* **OpenCV Integration**: Implemented object detection and face tracking algorithms using OpenCV, utilizing pre-trained models for accurate tracking of faces or objects in real-time. Anant Jain fine-tuned the parameters to ensure the software worked well under various lighting conditions and dynamic environments.
* **Real-time Processing**: Focused on optimizing the code for real-time image processing, ensuring that the drone could effectively detect and track moving objects or faces while maintaining minimal latency. This allowed the drone to make quick adjustments based on the tracked subject's movements.
* **Testing and Debugging**: Conducted extensive testing to validate the accuracy and responsiveness of the object detection and tracking features. Anant Jain debugged and refined the software to resolve issues like false detections, lag, and calibration discrepancies, ensuring optimal performance during drone operation.

Anant Jain's work was crucial in developing a responsive and reliable software system that could interpret sensor data and generate timely alerts, forming the core functionality of the smart helmet.

#### Testing, Analysis, and Documentation

All of us focused on testing the system’s performance, analyzing the results, and documenting all aspects of the Raspberry Pi Face Tracking Drone project. Our contributions ensured that the system met the project’s objectives and provided valuable insights for future improvements.

* **Testing and Validation**: Developed test plans to validate the functionality of each component, including the Raspberry Pi, camera, and flight control system. Conducted extensive testing in various environments to assess the accuracy of object detection, the responsiveness of face tracking, and the drone’s overall stability during flight. Verified that the system performed well under dynamic conditions, with consistent detection and tracking results.
* **Data Analysis**: Collected and analyzed data generated during testing, focusing on the accuracy of object detection and face tracking, as well as the performance of the drone’s flight stability. Analyzed the latency in real-time processing and evaluated the power consumption to optimize battery life. Identified areas where processing speed could be improved for smoother performance and faster reaction times.
* **Challenge Identification and Problem-Solving**: Documented challenges encountered during development, such as hardware limitations, power management, and real-time processing issues. Collaborated with the team to propose solutions for reducing latency, improving power efficiency, and addressing any hardware-related concerns. Suggested optimizations for OpenCV algorithms and fine-tuning of flight controller settings to enhance overall performance.

**REFERENCES**

1. Haribhau, S., Kirbraj Fritzi, N., & Kharat Shilpa Santosh. (2024). Camouflage Drone with Facial Recognition using AI. In M. V. Kanawade, *International Journal of Innovative Science and Research Technology* [Journal-article].
2. Y Syamala, Krishna, P., Reheman, M., Subhash, C. M., M.Sasinamratha, & D.Rohith. (2023). Flying Faces: An Automated Recognition System using Raspberry Pi and Drone. *. | Journal of Mechatronics and Automation*.
3. Khoi, T. Q., Quang, N. A., & Hieu, N. K. (2021). Object detection for drones on Raspberry Pi potentials and challenges. *IOP Conference Series Materials Science and Engineering*, *1109*(1), 012033.
4. Daryanavard, H., & Harifi, A. (2018). *Implementing Face Detection System on UAV Using Raspberry Pi Platform*.
5. Sajjad, M., Nasir, M., Muhammad, K., Khan, S., Jan, Z., Sangaiah, A. K., Elhoseny, M., & Baik, S. W. (2017). Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities. *Future Generation Computer Systems*, *108*, 995–1007.
6. Shaikh, I. M. M. a. M. S. S. F. S. P. S. G. (2023, April 3). *ATTENDANCE SYSTEM USING FACE RECOGNITION AND RASPBERRY PI &ndash; REVIEW*. INTERNATIONAL JOURNAL OF IOT AND DATA SCIENCE (IJIDS).
7. Benhadhria, S., Mansouri, M., Benkhlifa, A., Gharbi, I., & Jlili, N. (2021). VAGADRONE: Intelligent and Fully Automatic Drone Based on Raspberry Pi and Android. *Applied Sciences*, *11*(7), 3153.
8. Diez-Tomillo, J., Martinez-Alpiste, I., Golcarenarenji, G., Wang, Q., & Alcaraz-Calero, J. M. (2024). Efficient CNN-based low-resolution facial detection from UAVs. *Neural Computing and Applications*, *36*(11), 5847–5860.
9. Agustiady, R., Muda, N. R. S., Minggu, D., Herkariawan, C., & Suryana, M. L. N. (2021). Design of distance sensing and distance using quadcopter drone based on face recognition method. *IOP Conference Series Materials Science and Engineering*, *1098*(3), 032012.
10. S, S. K., H, L., & Shivkumar, H. (2015). Implementation of Image Processing on Raspberry Pi. *IJARCCE*, *4*(5), 199–202.
11. Liu, X., Xing, Z., Liu, H., Peng, H., Xu, H., Yuan, J., & Gou, Z. (2022). Combination of UAV and Raspberry Pi 4B: Airspace detection of red imported fire ant nests using an improved YOLOv4 model. *Mathematical Biosciences & Engineering*, *19*(12), 13582–13606.
12. INTERNATIONAL JOURNAL OF CONTEMPORARY RESEARCH IN MULTIDISCIPLINARY (IJCRM). (2023). IOT: Towards an Effective Technical Model for Facial Recognition Using Machine Learning Techniques and Raspberry Pi 4 Model B Platforms with Python Language. *Du-in*.
13. Wang, L., & Siddique, A. A. (2020). Facial recognition system using LBPH face recognizer for anti-theft and surveillance application based on drone technology. *Measurement and Control*, *53*(7–8), 1070–1077.
14. Nevon Projects. (2020, February 8). *Raspberry Pi Based Vehicle Starter on Face Detection*.
15. Oruganti, V. R. (n.d.). *Drone Detection and Tracking Using YOLO on Raspberry Pi*. A&M-Commerce Digital Commons.