AUTONOMOUS CAMPUS
SURVEILLANCE DRONE

Department of Electrical and Electronics Engineering

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY, PUNE
[2021-2025]

ACKNOWLEDGEMENT

It gives us immense pleasure to present our Project Report on **"Autonomous Campus Surveillance Drone"**. This project has been a remarkable journey that provided us with a platform to explore real-world engineering challenges and enhance our technical, analytical, and collaborative skills. The experience has not only deepened our understanding of drone technology and intelligent surveillance systems but has also instilled in us a strong sense of innovation and responsibility.

Table of Contents

ABSTRACT

In current times, the fast-changing academic scenarios, safety and security of University campuses is of utmost importance. Traditional security surveillance systems have constraint over their static range and fail to respond dynamically in vulnerable situations which rise the probability of missing security threats as well as fire mishaps.

In an attempt to arrive with a solution to these challenges, we propose the development of an Autonomous Campus Surveillance Drone, designed to improve surveillance capabilities through intelligent aerial monitoring. The system integrates a Raspberry Pi 5 with 8 GB RAM, Pixhawk 2.4.8 flight controller, ultrasonic sensors, and a Raspberry Pi Camera V2 for real-time detection of fire and smoke, crowd head counting and person detection using YOLOv5 Nano. It can work with manual mode and also in autonomous mode. In autonomous modes, it navigates through GPS, and data transmission from a secure IOT channel. Our test performance implicate that such drones would be a great replacement or an addition to the traditional surveillance systems. In addition, the drone has dual operational modes, both for manual operation and precision as well as testing, and autonomous flight through GPS waypoint mapping. It supports obstacle detection in real time through ultrasonic sensors, allowing for safe navigation through crowded environments. The system is also equipped with a YOLOv5 Nano model that has been trained on local datasets for strong identification of crowd density and fire/smoke anomalies. An IoT-driven alert mechanism ensures that there are real-time notifications sent to monitoring stations, and anomaly snapshots and GPS coordinates are recorded locally. This solution not only adds situational awareness but also aids proactive and automated campus security management.

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

IoT	Internet of Things
YOLO	You Only Look Once
IMU	Inertial Measurement Unit
GPS	Global Positioning System
CNN	Convolutional Neural Network
DIY	Di It Yourself
IEEE	Institute of Electrical and Electronics Engineers
UAV	Unmanned Aerial Vehicle
MCNN	Masked Convolutional Neural Network
ESC	Electronic stability control
CPU	Central Processing Unit
GPU	Graphical Processing Unit
UART	Universal Asynchronous Receiver/Transmitter
I2C	Inter-Integrated Circuit
SPI	Serial Peripheral Interface
CAN	Controller Area Network
USB	Universal Serial Bus
PPM	Pulse Position Modulation
RTL	Return To Launch
REST	Representational State Transfer

CHAPTER 1
INTRODUCTION

1.1 Motivation/Introduction/Relevance

Ensuring campus safety is a priority in modern educational institutions. Fixed security systems for surveillance lack mobility leading to its inability to adapt in real-time. Our aerial monitoring-based solution provides a dynamic, intelligent and responsive manner to monitor large and complex areas autonomously. The technology we have used is based upon Pixhawk flight controller helping the drone have a safely managed autonomous flight. The drone is mounted with a camera on its lower surface. It helps detect anomalies in crowded environments. The anomalies are set to be fire, smoke and excess head count detection. The ultrasonic sensors are place on the drone chassis providing it with obstacle avoiding capabilities. The increasing security threats during huge political events being held in the campus as well as the cultural competitions, musical concerts and campus events led to the idea of developing this drone-based surveillance system. There are various factors that give rise to use of such aerial security systems which are shown in figure.

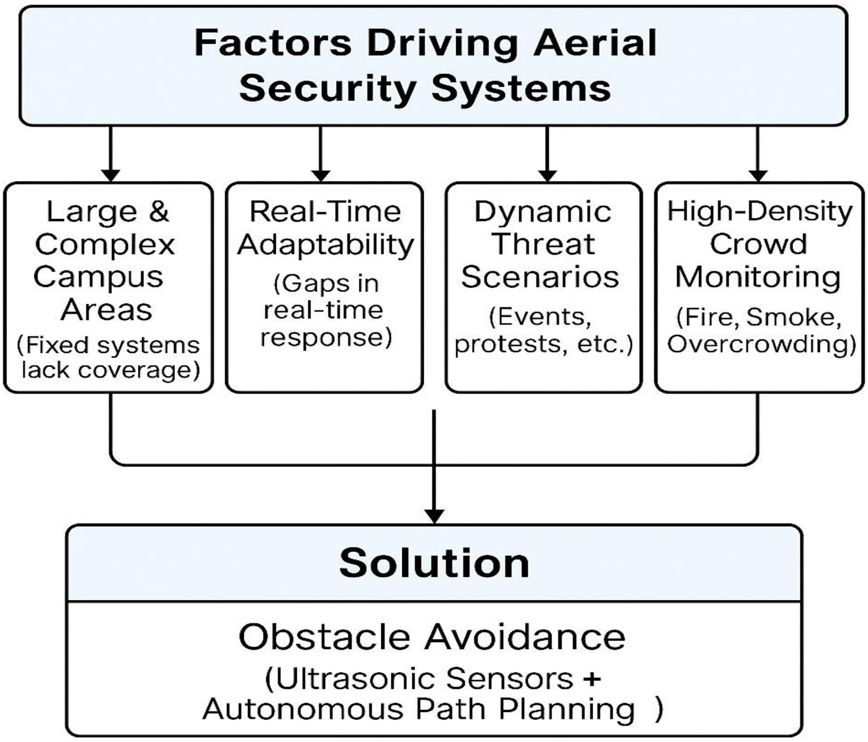

Figure 1.1

The use of air security systems is motivated by numerous key parameters in contemporary surveillance contexts. A significant number of sprawling and congested campus sites often have areas devoid of complete visibility by immovable security solutions, making them depend on deployable aerial technology. Adaptation to real-time threats is a leading consideration as well; established solutions are insensitive to quickly developing environments, but airborne systems may rapidly re-allocate to scan novel threats emerging as they materialize. Such threats often encompass uncertain and high-risk scenarios like protests or emergencies that require adaptive and swift monitoring. In addition, dense crowd settings such as those found around festivals, sports games, or college events necessitate keen monitoring for possible dangers like fires, smoke, or overcrowding. These issues highlight the need for drone-based security systems with superior features such as autonomous navigation and obstacle avoidance to guarantee effective and timely campus safety.

1.2 Organization of report

1.2.1 Chapter 2 presents existing literature and the aim of the project

1.2.2 Chapter 3 discusses the development and design of the drone

1.2.3 Chapter 4 focuses on implementation and methodology

1.2.4 Chapter 5 analyzes results from real-world testing

1.2.5 Chapter 6 concludes with discussions, challenges, and future scope

CHAPTER 2

REVIEW OF LITERATURE

2.1 LITERATURE REVIEW

S.No	Title	Objective	Methodology	Relevance to Project
1	Raspberry Pi-based Surveillance Drone (Rajarajeswari College)	Develop a low-cost surveillance drone using Raspberry Pi	Uses Pi Camera, GPS, Python-based flight control	Forms base for a budget-friendly smart drone platform
2	Pi Drone: An Autonomous Educational Drone (Brown University)	Educational, modular drone for autonomy experiments	GPS, IMU, ultrasonic sensors + Python control scripts	Demonstrates integration of sensors with Pi for autonomy
3	UAV Path Planning (Junhai Luo)	Explore efficient UAV route algorithms	A*, Dijkstra, Ant Colony Optimization	Useful for optimizing campus patrol routes dynamically
4	Raspberry Pi-based Autonomous Drone	Build DIY drone with basic autonomy features	Open-source software, GPS, camera streaming	Validates feasibility of Pi-powered autonomous surveillance
5	Image Processing Based Fire Detection (Navdeep et al.)	Detect fire using basic vision algorithms	HSV color filtering, contour & motion detection	Light-weight method for fire detection on Pi
6	Real-Time Fire and Smoke Detection Using DL	Enhance safety using CNN-based fire detection	Trained YOLO models on fire/smoke datasets	High accuracy fire/smoke detection in real time
7	Deep Learning for Crowd Density Estimation (IEEE)	Review DL models for crowd analysis	MCNN, CSRNet, regression-based CNNs	Supports crowd anomaly detection in campus areas
8	Realtime Crowd Monitoring using Hybrid YOLOv4	Detect and track people, estimate direction/speed	YOLOv4 + tracking modules, real-time processing	Enables dynamic crowd flow monitoring via drone
9	Autonomous Drone with Obstacle Detection (Ultrasonic + GPS)	Obstacle avoidance using low-cost sensors	Distance thresholds with ultrasonic sensors + GPS	Prevents collisions in autonomous navigation
10	Obstacle Detection & Avoidance for UAVs (IEEE Access)	Review of UAV obstacle detection strategies	Compares vision-based, sensor-based, hybrid methods	Informs design of robust navigation with minimal hardware

11	IoT & AI-Based Smart Surveillance Drone (IJITEE)	Integrate drone AI with IoT for safety alerts	YOLO for detection + IoT sensors + cloud dashboard	Matches goal of real-time alerts for fire/crowd detection
12	Very Deep CNNs (VGGNet – Simonyan & Zisserman)	Propose deep network (VGG) for vision tasks	Deep convolutional layers, ImageNet classification	Backbone for YOLO-based object detection architectures
13	Autonomous Fire Extinguishing System (Rehman et al., 2012)	Autonomous fire suppression	Embedded sensors + robotics	Inspires drone-based fire response systems
14	Anomaly Detection via Edge Computing (Patrikar & Parate, 2022)	Video surveillance anomaly review	Edge computing + AI-based systems	Fits low-power edge AI for UAVs
15	Abnormal Event Detection at 150 FPS (Lu et al., 2013)	Real-time abnormal event detection	MATLAB + visual pattern analysis	Inspires fast processing pipeline for drones
16	Real-World Anomaly Detection in Videos (Sultani et al., 2018)	Detect anomalies in long videos	MIL + deep CNN-LSTM	Enhances crowd behavior anomaly detection
17	Crowd Anomaly Detection – Drone & Ground (Ito et al.)	Combine aerial and ground analytics	YOLO + perspective mapping	Assists hybrid surveillance approach
18	Real-Time UAV Path Planning (Liu et al., 2013)	Optimize UAV navigation	Bi-level programming + constraints	Suitable for dynamic campus navigation
19	ICA-ANN Path Planning for UCAVs (Duan & Huang, 2012)	Path planning using AI	ICA optimization + neural networks	Suggests hybrid AI for UAV routes
20	Genetic Algorithm & Voronoi Path Planning (Pehlivanoglu, 2011)	Smart autonomous UAV routing	Vibrational GA + Voronoi diagram	Inspires efficient route-planning design
21	UAV Capsule Delivery for Pest Control (Freitas et al., 2020)	Capsule delivery + smart planning	UAV + GIS + optimized paths	Cross-domain use case of UAV smart planning

22	CI-Based UAV Path Planning Survey (Zhao et al., 2018)	Survey CI in path planning	ANN, GA, PSO comparative study	Benchmarks intelligent routing techniques
23	Raspberry Pi Surveillance Drone – IJMRSET (2024)	Updated Pi-based surveillance drone	GPS, camera, real-time alerts	Real-world validated build for smart surveillance
24	Autonomous Navigation of Flying Quadcopter (Kumar et al.)	Navigation of quadcopter drones	GPS, sensors, embedded systems	Supports development of reliable drone navigation

Table 2.1 Literature Survey

2.2 AIM AND OBJECTIVES OF PROJECT

Aim:

To develop, design, and deploy a sturdy and adaptable autonomous drone system with the ability to execute tasks demanding manual control, GPS navigation with obstacle avoidance, and autonomous anomaly detection (fire, smoke, and crowd density), with integrating critical emergency fail-safe mechanisms for safe operation.

Objectives:

To accomplish the aforementioned aim, the following objectives will be undertaken:

- Establish Stable and Reliable Flight:
    - Assemble and set up the drone frame, ESCs, and brushless motors to provide stable quadcopter flight under manual as well as autonomous control.
    - Design a flight controller system that can hold desired attitude and altitude.

- Implement Precise GPS-Based Autonomous Navigation:
    - Setup the GPS module to obtain accurate location information.
    - Create software for defining, uploading, and following autonomous flight trajectories using GPS waypoints.
    - Provide smooth and accurate following of predefined waypoints during autonomous missions.

- Implement a Strong Real-Time Obstacle Avoidance System:
    - Connect and calibrate the ultrasonic sensor array to properly sense obstacles in the drone's environment (front, rear, left, right, top, bottom) within a 1.5-meter radius.
    - Develop an obstacle avoidance algorithm that invokes proper responses (halting, altitude correction, path recalculation) when obstacles are detected during autonomous flight.

- Implement Real-Time Anomaly Detection Capabilities:
    - Connect the camera module to the Raspberry Pi 5 for real-time video feed capture.
    - Deploy and scale the YOLOv5 Nano model to detect fire and smoke in real-time within the video stream.
    - Create and embed a crowd count algorithm (drawing on density classification: sparse, moderate, dense) based on the camera image.
    - Install an alert generation system and send alert signals to a base station on detection of fire, smoke, or designated crowd density thresholds.

- Install Comprehensive Emergency Fail-Safe Mechanisms:
    - Design and implement a low battery voltage monitoring system that initiates an automatic emergency landing when the battery voltage is at a critical level.
    - Incorporate a GPS signal loss detection system that invokes a "Return to Launch" (RTL) fail-safe mode.
    - Design and implement an emergency landing protocol that involves a short hover, obstacle sensing on descent, and safe landing in the closest open space.

- Provide Seamless Manual Override Capability:
    - Implement the FlySky FS-i6X transmitter and receiver in order to enable immediate manual drone control for path recording, testing, or emergency use.
    - Provide a seamless and fault-tolerant mode switch from autonomous to manual mode.

- Implement a Modular and Efficient Software Architecture:
    - Use Python as the main programming language for flight logic and sensor integration.

- Use OpenCV for the fast processing of camera streams.
- Utilize the Multiprocessing library to execute important operations like camera feed processing, GPS and sensor data logging, and flight control adjustments in parallel for maximum performance.

- Validate System Performance and Safety:
  - Test each individual hardware and software component extensively.
  - Execute integrated system tests in test labs to assess the performance of autonomous navigation, obstacle avoidance, anomaly detection, and emergency fail-safe modes.
  - Evaluate the safety and reliability of the entire system under different operating conditions.

2.3 BLOCK DIAGRAM

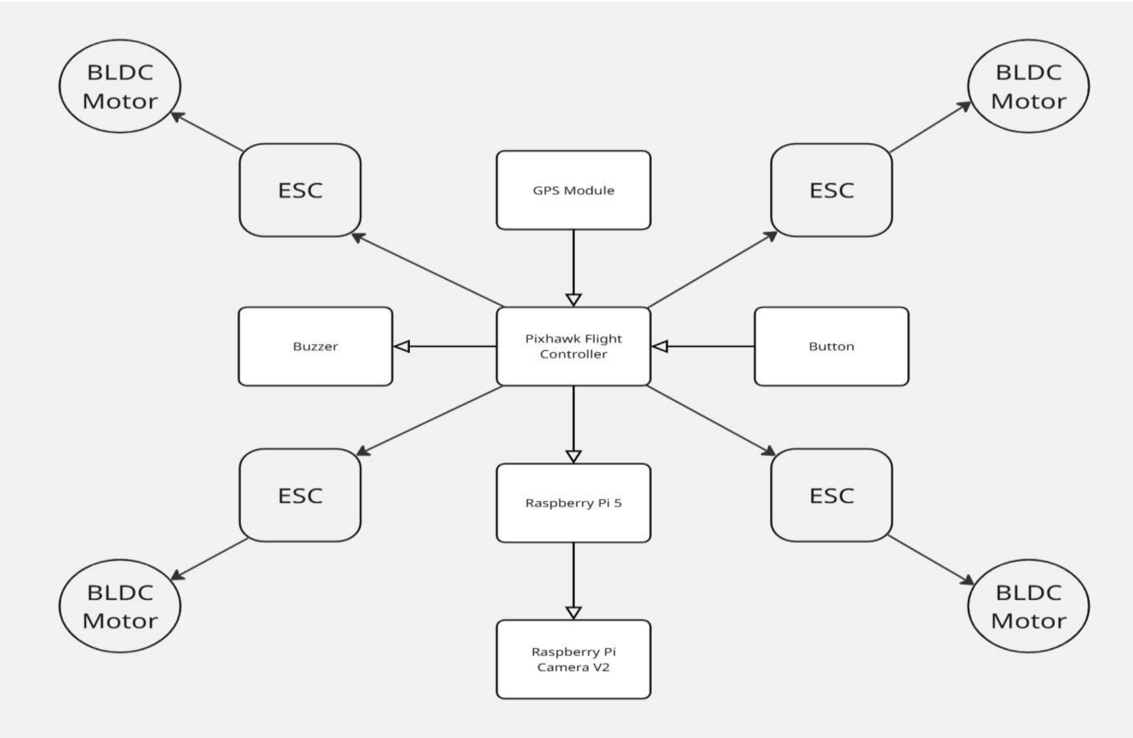

Figure 2.1 System Block Diagram

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 System Block Diagram

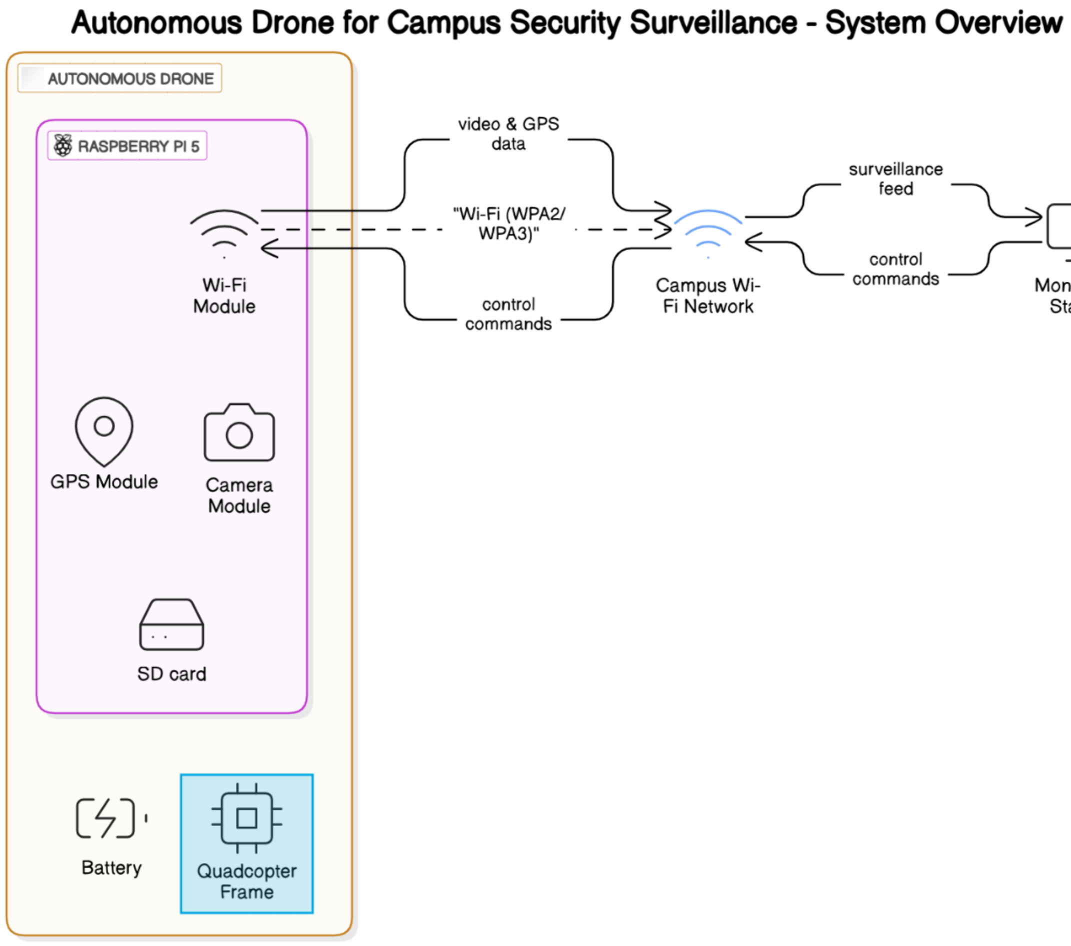

Figure 3.1 System Overview

This system overview illustrates a smart, autonomous drone setup built specifically for enhancing campus security. At the heart of the drone lies a Raspberry Pi 5, which acts as the central processing unit, managing communication, navigation, and video capture. Equipped with a GPS module, the drone can track and report its exact location, which is crucial for surveillance over large areas. The camera module continuously captures real-time footage, while an SD card handles local data storage. To stay connected, the drone uses a Wi-Fi module that links directly to the campus's secure network (WPA2/WPA3), allowing it to send live video and location data to a remote monitoring station. In turn, the station can issue control commands to guide or reroute the drone when needed. This two-way communication ensures dynamic responsiveness during patrols. The drone operates on a rechargeable battery and is mounted on a stable quadcopter frame, allowing for smooth flight and maneuverability. This system not only reduces the need for constant human patrols but also ensures timely response to unusual or suspicious activity, making the campus environment significantly safer and smarter.

3.2 System Specifications

Drone frame with ESCs and motors:

Figure 3.2 Drone Frame

We have used a Quadcopter frame with an arm size of 220 x 40 (LxW) mm, wheelbase of 450 mm.

We used a quadcopter frame with 220 x 40 mm arms and a 450 mm wheelbase, providing a solid and stable structure ideal for surveillance tasks. The frame's size supports smooth flight, can handle our equipment like the Raspberry Pi, camera, and GPS module, and ensures stability even during hover or mild wind, making it perfect for campus monitoring.

- We have used a 920KV Brushless DC Motor with 30A ESCs (Electronic Speed Controller)
  Raspberry Pi 5(8gb):

Figure 3.3 BLDC Motor

Figure 3.4 ESC

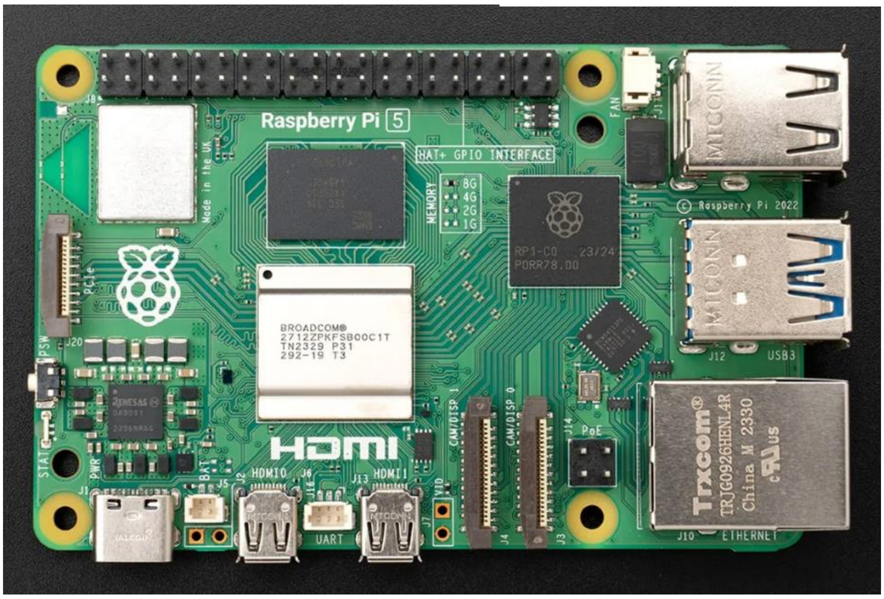

Figure 3.5 Raspberry PI 5

For our drone system, we chose the Raspberry Pi 5 with 8GB of RAM, which acts as the central control unit for all onboard functions. It's equipped with a powerful quad-core Cortex-A76 processor running at 2.4GHz, giving it the speed and responsiveness needed to handle live video processing, GPS tracking, and wireless communication—all at the same time. The built-in VideoCore VII graphics unit supports high-resolution output, though in our case, it ensures smooth handling of the live camera feed and visual data from the drone.

To store video footage, image captures, and navigation logs, we use a 128GB microSD card, which gives us plenty of space without slowing down performance. Connectivity-wise, the Pi

5 supports dual-band Wi-Fi (802.11ac) and Bluetooth 5.0, making it easy to maintain stable and fast communication with the monitoring station. It also offers Gigabit Ethernet for high-speed wired connections during initial setup or troubleshooting.

In terms of ports, it includes two USB 3.0 and two USB 2.0 for external modules or debugging tools, a USB-C power input, and a 40-pin GPIO header for custom sensor integration. It runs on a standard 5V/5A USB-C power supply and has a compact footprint (85.6mm x 56.5mm), allowing it to fit snugly within our drone frame without adding unnecessary bulk.

For visual data collection, we've paired the Raspberry Pi with the Pi Camera v2. Despite its small size—just 25mm x 23mm x 9mm and weighing barely over 3 grams—it's highly capable. The camera features an 8-megapixel sensor that can capture still images in high detail (up to 3280 x 2464 resolution). It also supports different video modes including full HD (1080p at 30fps), HD (720p at 60fps), and lower resolutions for faster frame rates (640x480 at 90fps). The camera connects directly to the Pi using a simple ribbon cable, making the setup clean and efficient.

Together, this combination of Raspberry Pi 5 and Pi Camera v2 gives our drone the intelligence and vision it needs for real-time campus surveillance, all in a lightweight and compact form ideal for flight.

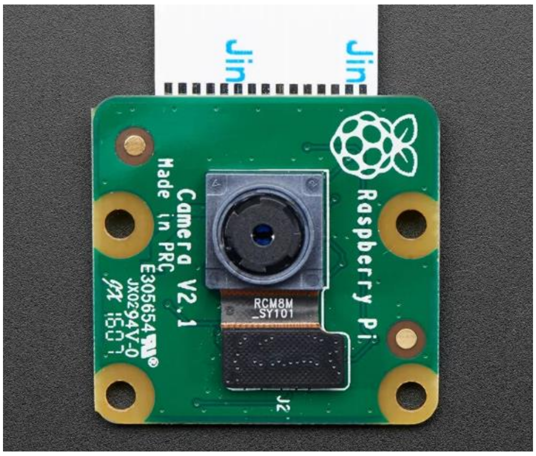

Figure 3.6 Raspberry PI Camera V2

Pi Camera v2:

The Pi Camera v2 is a compact yet highly capable imaging module that perfectly complements the Raspberry Pi for visual data collection. At the heart of this camera is an 8-megapixel sensor, which allows it to capture crisp and detailed still images at a resolution of 3280 by 2464 pixels. This level of clarity is particularly valuable in surveillance applications, where visual precision can make all the difference in detecting and analyzing events.

In terms of video capabilities, the Pi Camera v2 supports a range of resolutions and frame rates, making it versatile for both high-definition and faster-paced recordings. It can record in full HD (1080p) at 30 frames per second, HD (720p) at 60 frames per second, and even lower resolutions like 640x480 at a smooth 90 frames per second—ideal for capturing fast motion.

Physically, the module is incredibly lightweight, weighing just over 3 grams, and has compact dimensions of 25mm by 23mm by 9mm. This makes it easy to integrate into tight spaces, such as the frame of a drone, without adding unnecessary bulk or weight. The camera connects to the Raspberry Pi using a short, flat ribbon cable, which plugs directly into the dedicated camera interface on the Pi board, ensuring a secure and efficient data transfer.

Overall, the Pi Camera v2 is a reliable and efficient component for capturing real-time video and still images, particularly in projects like drone-based surveillance where space, weight, and quality are all critical factors.

Figure 3.7 PIXHAWK 2.4.8

Pixhawk 2.4.8 flight controller

This system is powered by a 32-bit STM32F427 microprocessor, clocked at 168 MHz, which ensures high-speed processing for efficient handling of complex operations. It comes with 2MB of Flash memory, providing ample space for essential firmware and configurations, and 256KB of RAM, enabling smooth real-time data processing and control during flight.

For precise flight control and navigation, the system integrates several key sensors: a gyroscope, accelerometer, magnetometer, and barometer. Together, these sensors provide vital data on the aircraft's orientation, movement, altitude, and pressure changes, ensuring accurate stability and positioning during flight in different environments.

Connectivity is made easy with a variety of ports, including five UART ports for communication with multiple devices, along with I2C and SPI interfaces for connecting additional sensors or peripherals. The system also features a CAN bus for fast data exchange, a USB port for firmware updates and file transfers, and specialized connections such as PPM, SBUS, and RSSI to interface with remote control receivers and telemetry systems.

To control external components like servos or motors, the system offers 14 PWM channels, which can be used for various actuators, including controlling flight surfaces or additional peripherals.

The system also has a **4GB microSD card**, which is used for logging flight data. This allows for storing important information, such as telemetry data and sensor readings, for later review, diagnostics, or performance enhancements after flights.

With compact dimensions of **82×50×16 mm** and weighing in at around **40g**, this system is lightweight and compact, making it an ideal choice for drones and other small UAVs where weight and space efficiency are critical.

Figure 3.8 GPS

GPS module

This system integrates the uBlox NEO-M8N GPS module, which not only provides precise GPS tracking but also includes a built-in compass. The GPS module allows the system to pinpoint its exact location, offering accurate navigation and ensuring the device can follow set paths or adjust its movement in real-time. The embedded compass works seamlessly with the GPS, enabling the system to maintain orientation, even in situations where directional information may be less clear, enhancing overall navigation reliability.

To detect nearby obstacles and measure distances, the system features six ultrasonic sensors, each using the HC-SR04 model. These sensors are designed to emit sound waves and measure how long it takes for the sound to bounce back after hitting an object. With a range of 2 cm to 400 cm, these sensors offer a wide range of capabilities for detecting objects both close by and at a greater distance, providing crucial information for avoiding obstacles and navigating complex environments.

The system operates on a simple 5V power supply, making it compatible with a variety of other components and ensuring efficient power usage for continuous operation.

Figure 3.9 HC-SR04

HC-SR04 ultrasonic sensor

The HC-SR04 ultrasonic sensor is an affordable and widely used device for measuring distances in a variety of electronic and robotics applications. It operates by sending out high-frequency sound waves (ultrasonic pulses) from a transmitter and waiting for these waves to bounce back after hitting an object. The sensor has a receiver that listens for the returning sound waves, and by calculating how long it takes for the sound to return, the sensor can determine the distance between itself and the object.

This sensor is capable of measuring distances from 2 cm to 400 cm, which makes it incredibly versatile for tasks like obstacle detection in robots, range-finding for drones, or even proximity sensing in automation systems. The basic principle is simple: the transmitter emits a pulse, and the receiver listens for the echo. Based on how much time passes before the echo is received, the sensor computes the distance to the object, all using the speed of sound.

The HC-SR04 works on a 5V power supply, which is standard for most electronic projects, making it an easy and inexpensive component to integrate into a variety of systems. Its simplicity is a big draw—developers and hobbyists alike love how easy it is to connect and use, without needing complex setup or calibration.

The sensor is small but effective, with a detection range that allows it to be used for everything from measuring the height of objects to avoiding collisions in autonomous vehicles. Whether you're building a robot, a smart parking sensor, or a security system, the HC-SR04 offers reliable, real-time feedback for your project.

Figure 3.10 Lithium Polymer Battery

Lithium Polymer Battery

A lithium polymer (LiPo) battery is a lightweight, rechargeable battery that uses a gel-like electrolyte, making it flexible and compact. Known for high energy density, it’s widely used in devices like drones and smartphones. LiPo batteries offer fast charging, longer battery life, and customizable shapes. They are safer than traditional lithium-ion batteries but require careful handling to avoid overcharging or physical damage. With a voltage of 3.7V per cell, they provide efficient power in a compact form, ideal for performance-sensitive applications.

Figure 3.11 Propellers

Propellers

The 1045 propellers (10×4.5) are a popular choice for drones, offering a great balance of performance and efficiency. The 10 inches represents the diameter, and 4.5 inches is the pitch, which indicates how much distance the propeller would theoretically travel forward in one complete revolution. These propellers are available in Clockwise (CW) and Counterclockwise (CCW) rotations. The use of both CW and CCW propellers in a drone is essential for achieving balanced thrust and preventing unwanted spinning, as each pair works together to stabilize the flight.

The 1045 propellers are ideal for medium-sized drones and are particularly appreciated for their ability to provide sufficient lift without compromising on efficiency. The 4.5-inch pitch gives these propellers the ability to generate enough thrust for various types of flights, including stable, slow-motion captures or fast-paced drone races. They are commonly used for aerial photography, racing drones, and hobbyist drones, as they strike the right balance between speed and stability.

In terms of material, these propellers are lightweight, which contributes to better overall battery efficiency and extended flight times. They can handle moderate to strong winds, making them versatile for different flying conditions. Their combination of size, pitch, and rotational direction ensures that the drone has the necessary power and control to perform a wide range of tasks, from capturing smooth video footage to executing high-speed.

Overall, the 1045 CW and CCW propellers are a reliable choice for drone enthusiasts and professionals looking for high performance and durability while maintaining an optimal flight experience.

Figure 3.12 Flysky FS-i6x

Flysky FS-i6X 2.4GHz 6CH Transmitter

The Flysky FS-i6X 2.4GHz 6CH Transmitter and FS-iA10B 2.4GHz 10CH Receiver offer a reliable and versatile solution for controlling RC models like drones, planes, and cars. The FS-i6X transmitter features a user-friendly design with a 6-channel setup, a rechargeable LiPo battery, and the ability to store up to 10 models. It offers adjustable settings such as servo reversing, expo settings, and dual rates. The FS-iA10B receiver provides 10 channels, ensuring stable long-range control and features failsafe technology for safety. Together, this system delivers excellent range, stability, and precision for both beginners and experienced hobbyists.

Figure 3.13 Power Module

Power Module:

A power module is an essential component in RC systems, such as drones, that regulates and distributes power from the battery to various electronics. It ensures stable voltage delivery, preventing damage to components from over-voltage or under-voltage. Power modules often include current monitoring to track real-time power consumption and provide telemetry data for the flight controller, helping to manage battery health. Additionally, they protect the system with overvoltage and undervoltage protection, enhancing safety and system longevity. Power modules streamline installation by organizing power distribution and contribute to flight stability by providing consistent power to key components. Advanced modules offer telemetry and performance monitoring, making them ideal for high-performance setups.

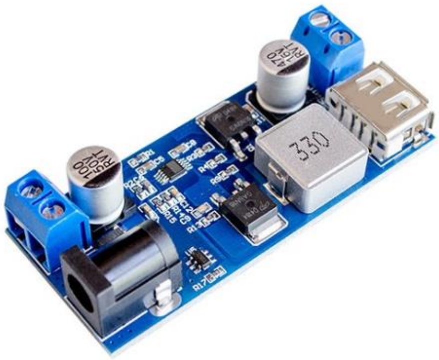

Figure 3.14 Buck Converter

Buck Converter

A Buck Converter is a highly efficient DC-DC power regulator that steps down a higher input voltage to a lower output voltage. It uses fast switching, along with inductors and capacitors, to transfer energy with minimal loss, unlike linear regulators that waste power as heat. Commonly used in drones, microcontrollers, and other electronic systems, it ensures stable voltage supply while preserving battery life. Its compact size, efficiency, and safety features like overcurrent and thermal protection make it ideal for modern electronic applications.

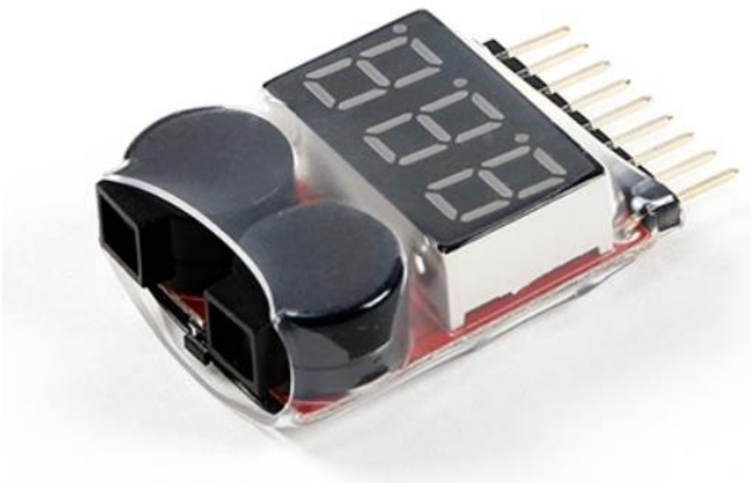

Figure 3.15 Battery Voltage Monitor

Battery Voltage Monitor

A Battery Voltage Monitor is a compact device used to continuously measure and display the voltage of a battery in real time. It plays a key role in protecting batteries—especially LiPo and Li-ion types—from over-discharge or overcharge, which can cause damage or reduce lifespan. Commonly used in drones, RC vehicles, and robotics, it helps users track battery health and ensures safe operation. Many monitors include visual or audible alerts when voltage drops too low, allowing timely action to avoid system failure. Its small size and reliability make it essential for battery-powered systems.

3.3 Challenges Faced

Battery Over-discharge & Damage

During the early testing phases of the drone system, the LiPo (Lithium Polymer) battery was unintentionally discharged beyond its safe operating voltage range. This over-discharge led to overheating and permanent internal damage to the battery cells, rendering the battery unsafe and unusable. This highlights the importance of implementing battery monitoring and cutoff mechanisms to protect the battery from irreversible damage.

Magnetic Interference with GPS Module

The operation of brushless motors generated strong magnetic fields that interfered with the nearby GPS module. This interference caused inconsistent positioning, frequent signal drops, and inaccurate location data. To mitigate this issue, shielding the GPS module or relocating it farther from motor components is necessary to ensure reliable navigation during autonomous flight.

Ultrasonic Sensor Inaccuracy Due to Propeller Wash

The downward airflow (propeller wash) generated by the rotating propellers negatively affected the ultrasonic sensor's ability to measure distance accurately. This disruption led to unstable and fluctuating readings, especially during takeoff and hovering. A possible solution is to mount the sensor in a location shielded from the airflow or use alternative distance-measuring technologies such as LiDAR or optical flow.

Drone Stabilization Issues

Achieving a stable hover and flight trajectory was challenging in the initial tuning phases. The drone experienced drift, oscillations, and uneven thrust distribution, which compromised flight control. These issues were likely due to imperfect PID tuning, frame imbalance, or inconsistent motor performance. Careful calibration of the flight controller and ESCs, along with accurate weight balancing, is essential for improving stability.

Lack of GPS Signal Indoors

During indoor testing, the GPS module failed to acquire satellite signals due to limited or blocked sky visibility. As a result, autonomous waypoint-based missions could not be executed inside lab environments. For indoor navigation, integrating additional localization techniques such as SLAM, visual markers, or indoor positioning systems (IPS) would be required.

Supplying 5A Current to Raspberry Pi 5

The Raspberry Pi 5 demands a stable 5V power supply capable of delivering up to 5A of current, especially when peripherals like cameras, sensors, and wireless modules are connected. During drone operation, voltage fluctuations and limited output from onboard power modules made it difficult to consistently supply this requirement. A dedicated power regulation circuit or high-capacity UBEC (Universal Battery Elimination Circuit) can resolve this issue.

Physical Restraints and Weight Control

Integrating all necessary hardware—such as sensors, battery, Raspberry Pi, camera, and communication modules—into a compact drone frame posed design challenges. The need to keep the frame lightweight yet durable, while ensuring balanced weight distribution, directly impacted flight stability and battery efficiency. Designing with a focus on modularity, efficient layout, and low-weight materials is crucial to overcoming this limitation.

Computer Vision Model Optimization

Although YOLOv5 Nano is designed to be lightweight, running it in real-time for fire detection, smoke recognition, and crowd counting on the Raspberry Pi 5 still demands significant computational resources. Achieving both accuracy and frame rate may require further optimization techniques such as model pruning, quantization, or hardware acceleration via Coral TPU or GPU support to ensure smooth and efficient inference.

Model Generalization

The YOLOv5 Nano model, even when trained on a comprehensive dataset, may not generalize well to unseen scenarios involving different fire types, varying smoke conditions, or diverse crowd environments. This limits its reliability in real-world deployments. To improve generalization, the model should be trained on diverse datasets and possibly fine-tuned with real deployment scenarios using transfer learning.

CHAPTER 4
SYSTEM IMPLEMENTATION

The objective of this project is to design a complete autonomous drone system that incorporates manual operation functionality, accurate GPS-guided navigation, an active obstacle avoidance system, and essential emergency fail-safe features to provide operational safety and versatility. This entails a synergistic integration of well-chosen hardware elements for maintaining flight stability, real-time environmental sensing, and precise navigation, with advanced software modules to facilitate intelligent decision-making for autonomous flight and anomaly detection. The system aims to establish an integrated harmony among these hardware and software components to form a robust platform that can carry out pre-programmed missions, act accordingly to real-time environmental fluctuations, and respond correctly to unexpected critical situations.

To achieve this central goal, some fundamental goals need to be accomplished. The system needs to achieve stable and consistent flight properties. This requires the interference-free integration and careful setup of the drone frame, electronic speed controllers (ESCs), and brushless motors in order to realize consistent and predictable quadcopter flight behavior under manual as well as autonomous operation. In addition, an advanced flight controller system has to be implemented to actively control the desired attitude and altitude of the drone, providing smooth and controlled motion in the air.

The system also needs to include accurate GPS-based autonomous flight. This calls for the efficient integration of the GPS module to receive accurate and dependable location information. In complement, specific software has to be designed to allow definition, uploading, and running of autonomous flight routes defined by a sequence of GPS waypoints. The system needs to provide correct tracking of the pre-defined waypoints in autonomous operations so that the drone can follow complicated routes with less human control.

A reliable real-time obstacle avoidance system is essential for secure autonomous flight. This entails the precise interfacing and calibration of a set of ultrasonic sensors placed at strategic locations around the drone (front, rear, left, right, top, and bottom) to offer holistic awareness of the surrounding environment within a 1.5-meter radius. A smart obstacle avoidance algorithm needs to be designed and integrated to interpret the sensor information in real-time. On sensing an imminent collision danger, this algorithm should initiate corrective maneuvers as applicable, like forcefully bringing forward travel to a standstill, altitude correction if possible, and revising the flying route dynamically so that the observed barrier can be bypassed on route to the targeted destination.

The system must incorporate real-time anomaly detection features, taking advantage of the onboard camera module and the Raspberry Pi 5's processing capabilities. This includes the deployment and tuning of the light-weight deep learning model YOLOv5 Nano for real-time identification of vital anomalies like fire and smoke in the recorded video stream. Also, an advanced crowd counting algorithm, possibly leveraging the density classification available in the DITTO dataset (sparse, moderate, dense), would need to be incorporated to process the camera feed and estimate crowd density in real-time. Upon fire, smoke detection, or detection of certain levels of crowd density, the system should be able to produce and send timely alerts to a specified base station, delivering vital situational awareness.

Implementation of full emergency fail-safe mechanisms is important to ensure the safety and longevity of the drone system. This involves the design and integration of a real-time low battery voltage monitoring system that can automatically sense critical power levels and initiate an immediate emergency landing procedure. In addition, an effective GPS signal loss detection system should be in place to trigger a "Return to Launch" (RTL) fail-safe procedure that brings the drone back to its pre-programmed home base if GPS becomes unavailable. The emergency landing procedure itself also needs to be well designed in order to provide a brief hover at the prevailing altitude to gauge the immediate area of landing, followed by gradual descent with continued monitoring for any obstructions nearby to achieve safe landing in the nearest open ground.

The system should deliver an uninterrupted manual override function, enabling direct human control when required. This entails good integration of the FlySky FS-i6X transmitter and receiver so that a human pilot can instantly take manual control of the drone for purposes such as manual path tracking, system testing, or due to unexpected emergencies. Autonomous to manual control mode switching must be smooth and trustworthy to ensure continuous operation and pilot trust.

The underlying software structure should be modular, efficient, and structured. Python will be the main programming language for the implementation of the core flight logic and the integration of the different hardware components and sensors. OpenCV will be used for efficient processing of the real-time camera feeds needed for computer vision operations. To ensure optimal performance and responsiveness, the Multiprocessing library shall be utilized to parallelize key computational operations, including the ongoing processing of camera streams, logging of GPS and other sensor information, and dynamic flight control parameter adjustments.

Ultimately, systematic testing and validation processes need to be conducted throughout the project cycle. This involves thorough testing of every individual hardware and software component for ensuring proper functioning. In addition, comprehensive integrated system testing will be done in controlled environments for effectively assessing the performance and reliability of autonomous navigation, obstacle avoidance, anomaly detection, and emergency fail-safe mechanisms collectively working in harmony. The final purpose of these testing methods is to provide assurance on the overall operational safety and operational performance of the autonomous drone system across a wide range of anticipated operating conditions.

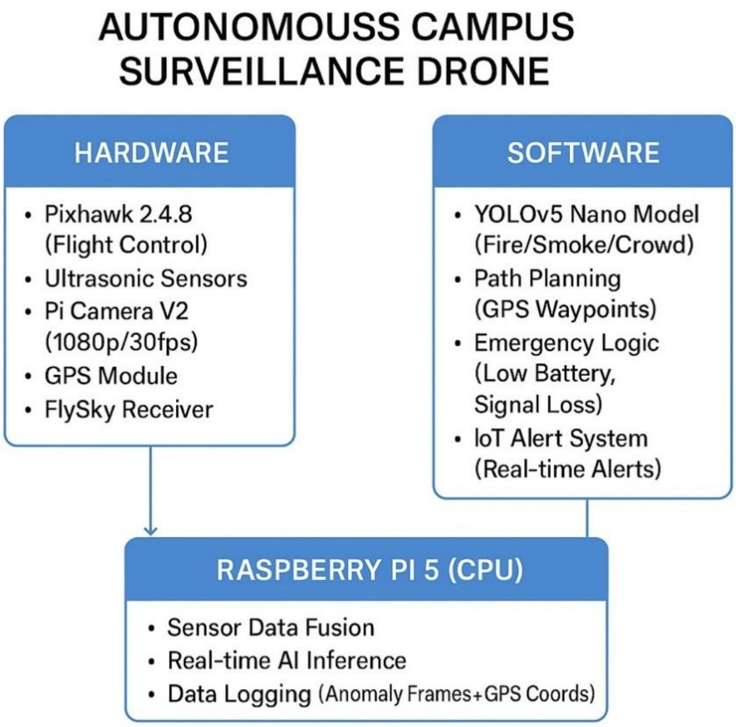

Figure 4.1 System Block

Hardware Components:

- **Drone Frame** – A lightweight, durable quadcopter frame housing all electronics.
- **Raspberry Pi 5** – Onboard computer for processing sensor data, running AI models, and flight control.
- **Camera Module** – Captures real-time video for object detection (supports YOLOv5 Nano).
- **GPS Module** – Provides precise location data for autonomous waypoint navigation.
- **Ultrasonic Sensors** – Detect obstacles for collision avoidance.
- **Electronic Speed Controllers (ESCs) & Brushless Motors** – Control motor speeds for stable flight.
- **FlySky FS-i6X Transmitter/Receiver** – Allows manual override when needed.
- **Battery & Power Distribution Board** – Ensures stable voltage supply to all components.
- **Battery Voltage Monitor** – Detects low battery to trigger emergency landing.

Software Components:

- **Python** – Main programming language for flight logic and sensor integration.
- **YOLOv5 Nano** – Lightweight deep learning model for real-time crowd count, fire and smoke detection.
- **OpenCV** – Processes camera feeds for computer vision tasks.
- **Multiprocessing** – Parallelizes tasks:

  - Camera feed processing

  - GPS & sensor data logging

  - Flight control adjustments
- **Flight Controller** – Maintains stable flight dynamics.
- **Fail-Safe Algorithms** – Trigger emergency landing in critical scenarios.

Dataset Specifications

Two custom datasets were merged and annotated for model training:

- **Fire & Smoke Detection Dataset**

- 15,432 labeled images
- Class distribution:

  1. Fire: 68%
  2. Smoke: 32%

- **Crowd Counting Dataset (DITTO)**

- 2563 images
- Density classification:

  1. Sparse (<5 people): 25%
  2. Moderate (5-20): 55%
  3. Dense (>20): 20%

Flight Modes:

Manual Mode

- Controlled via FlySky FS-i6X transmitter.
- Direct pilot input (throttle, pitch, roll, yaw) for path recording, testing or emergencies.

Autonomous Mode

- Follows predefined or recorded GPS waypoints for navigation.
- Obstacle Avoidance System :
- Ultrasonic sensors scan surroundings (front, back, left, right, top, bottom).
- If an obstacle is detected within 150cm, the drone:
- Stops forward movement
- Adjusts altitude (if possible)
- Recalculates path and keeps moving
- Emergency Landing Feature:
- Triggers automatically in critical conditions:
- Low battery voltage (monitored in real-time)

- GPS signal loss (failsafe RTL – Return to base station)
- Procedure:
- Hovers at current altitude briefly.
- Gradually descends while checking for obstacles.
- Lands safely in the nearest open area.

System Workflow

The working sequence of this autonomous drone system begins with an Initialization stage. In this vital phase, the onboard equipment executes a sequence of vital checks to confirm all hardware elements are operating properly and are ready for use. This entails checking the integrity of connections between the Raspberry Pi 5 and the various sensors and actuators. Next, the system tries to establish a stable GPS lock, which is essential for autonomous operation. After GPS acquisition, the ultrasonic sensors are calibrated to provide precise distance measurements for obstacle detection. After these initialization steps are successfully done, the system moves on to the Mode Selection stage, where the operator can select Manual Mode or Autonomous Mode based on the desired operation.

After a mode is selected, the system enters the Execution stage. In the case of Autonomous Mode being selected, the drone will proceed to execute a predefined or already stored sequence of GPS waypoints and travel autonomously to the destination locations. During autonomous flight, the ultrasonic sensors constantly sweep around the drone for possible obstacles. If an obstacle is perceived within the pre-set threshold of 150cm, then the system will activate its obstacle avoidance procedure, which includes stopping forward motion, trying to adjust altitude if there is a clear vertical line, and dynamically readjusting the flight path to avoid the blockage while continuing to attempt to reach the next waypoint. At the same time, the in-camera camera continually records video streams, which are processed in real-time by the YOLOv5 Nano model to identify anomalies like fire and smoke. In addition, the video stream is also screened by the crowd counting algorithm to predict the density of any crowds it may detect. Once a departure from normal is identified (fire, smoke, or a threshold crowd density), the system will create and send a notice to a configured base station, offering real-time situation awareness. In life-threatening situations, like critically low battery voltage or total loss of GPS signal, the emergency landing mode will be automatically engaged, taking precedence over the ongoing operation. On the other hand, if Manual Mode is chosen when selecting the mode, the flight of the drone will be operated manually by the pilot using the FlySky FS-i6X transmitter, enabling manual track recording, special test maneuvers, or instant intervention in the event of a surprise.

Lastly, the system terminates its operation with the Termination phase, which entails safe landing of the drone at the specified base station. This landing can be done under normal operational conditions upon finishing a mission or as a consequence of an emergency landing procedure being automatically triggered due to mission-critical conditions that are experienced during flight. In either case, the termination phase guarantees the controlled and safe return to the ground of the drone.

4.2 Algorithm / Flowchart

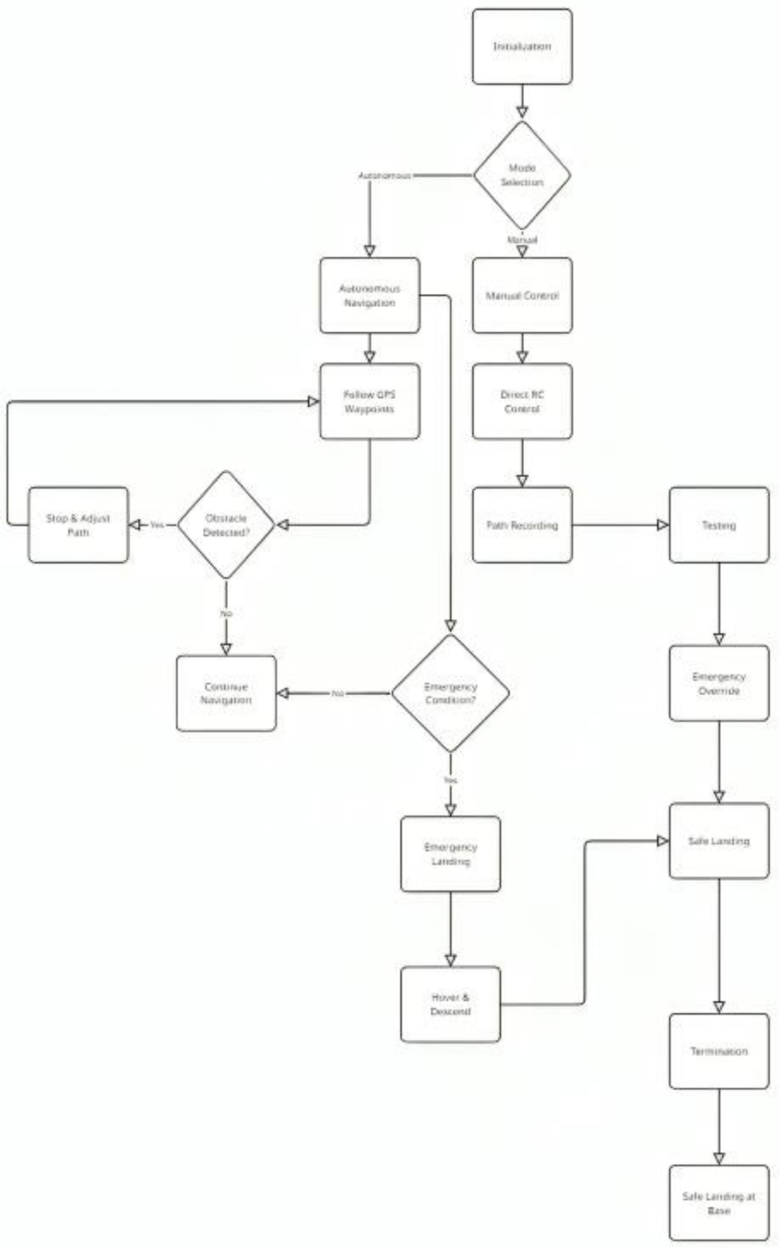

Figure 4.2 Flow Chart

CHAPTER 5
RESULTS AND ANALYSIS

5.1 RESULTS of Implementations

YOLO MODEL OUTPUT

YOLOv5 Nano for Resource-Constrained Deployment: The selection of YOLOv5 Nano was driven by its lightweight design, crucial for deployment on resource-limited hardware like autonomous drones. Its efficiency in terms of computational power and memory footprint makes it highly suitable for real-time object detection on the Raspberry Pi 5. This balance between accuracy and speed is essential for responsive anomaly detection without overburdening the onboard system.

Model Deployment via Optimized Parameters: The trained YOLOv5 Nano model is deployed using the best parameters saved in a ".pt" file. These parameters represent the learned knowledge from the training process, ensuring optimal detection performance. Utilizing these optimized weights and biases maximizes the model's accuracy in identifying fire, smoke, and estimating crowd density during the drone's operation.

Project Results: Output of the Combined Dataset Model: The project's results include the output of the machine learning model trained on the combined dataset. This output demonstrates the model's ability to detect fire and smoke in real-time video feeds and to classify crowd density into categories. Evaluation metrics would quantify the model's effectiveness in these critical detection and classification tasks.

The Foundation: Combined Datasets from Roboflow Universe: The model was trained using a combined dataset sourced from Roboflow Universe, integrating "DITTO.v8.YOLOv5" for crowd/person detection and "FireDetection.v2i.YOLOv5" for fire/smoke detection. Roboflow Universe provided a valuable resource for accessing pre-labeled datasets relevant to the project's objectives. This facilitated efficient data acquisition for training the anomaly detection model.

Roboflow Universe: A Comprehensive Platform for Computer Vision: Roboflow Universe is a platform offering tools for the entire computer vision project lifecycle. It supports dataset creation and management, data annotation, model training, and deployment. This comprehensive suite of features streamlines the development process for computer vision applications, including deploying models on edge devices like the project's drone.

Data Distribution: Emphasis on Critical Anomalies: The combined dataset's distribution shows a higher concentration of data for fire and smoke detection compared to head count and person detection. This emphasis reflects the project's prioritization of addressing critical safety concerns. The focus on fire and smoke detection indicates the importance of early detection of these hazards for the drone's intended applications.

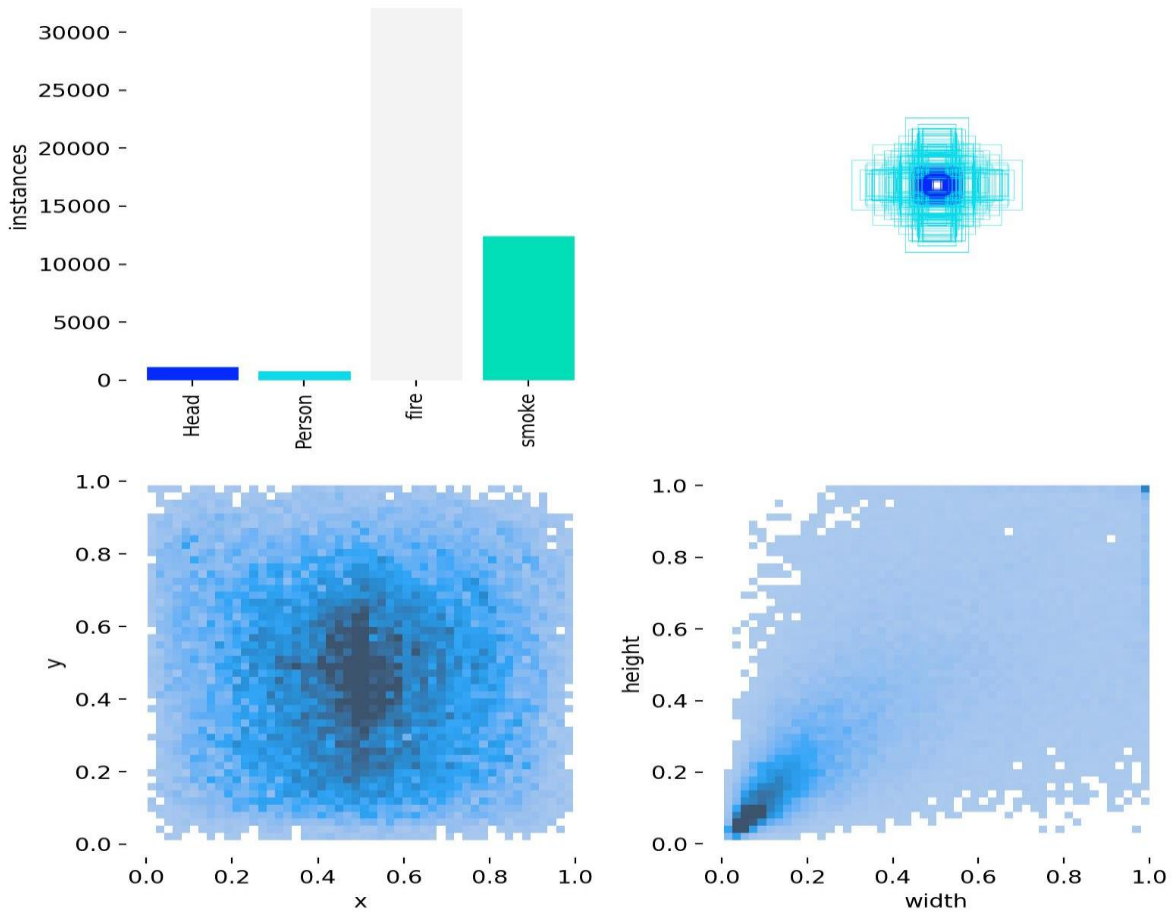

Figure 5.1 Model Labels 1

- The figure below demonstrates the output of the trained model on our dataset and displays a result based on 40 epochs.
- An epoch is one complete pass through the entire training dataset by the learning algorithm.

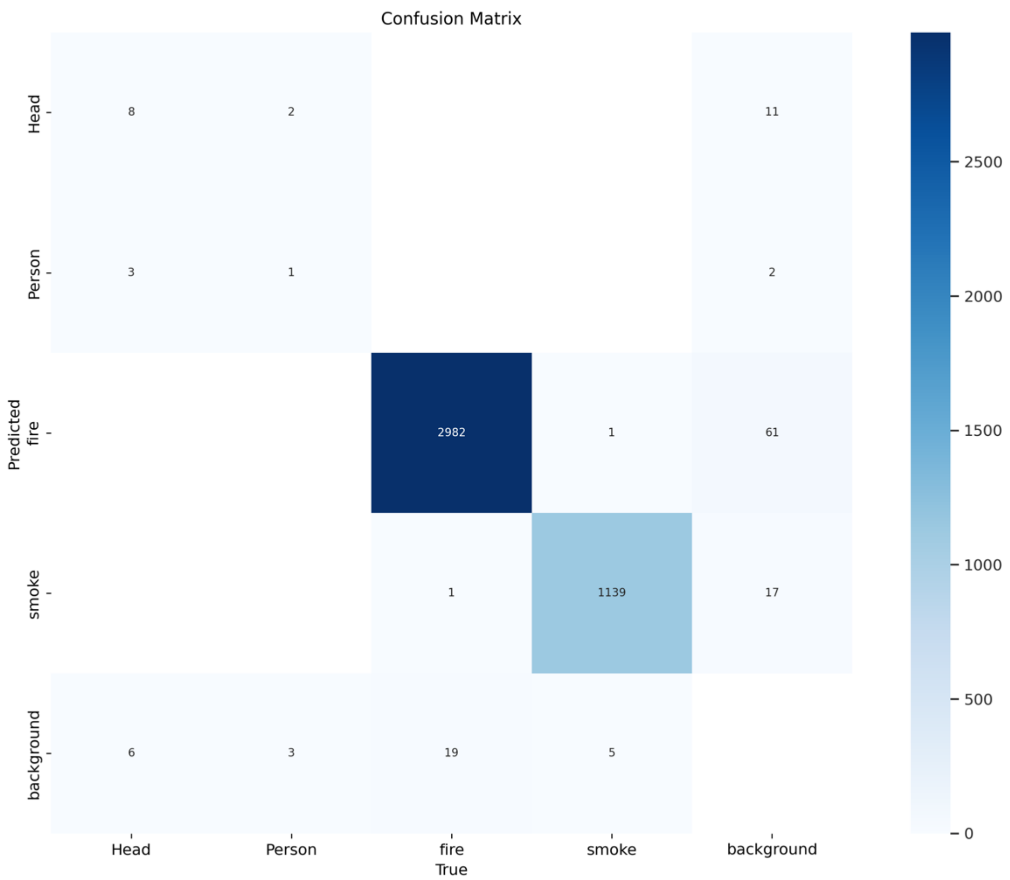

Figure 5.2 Confusion Matrix

IOT ALERT AND LOGGING

- **Real-time Alert and Image Logging:** An alert system has been implemented to provide immediate notification upon the detection of fire or smoke. When such an event is identified in the live video stream from the Raspberry Pi camera, the system captures and stores the corresponding image frame locally within a designated "fire/smoke" folder. Simultaneously, this alert, along with the captured image, can be transmitted over an IoT channel in addition to alerts sent via the Campus Wi-Fi network, ensuring multi-channel dissemination of critical information.
- **Utilizing ThingSpeak for IoT Dashboard:** For the purpose of creating an intuitive and accessible IoT dashboard, the project leverages the ThingSpeak IoT platform. ThingSpeak offers a robust cloud-based infrastructure specifically designed for Internet of Things applications. This platform provides the necessary tools and services to effectively collect, store, and visualize the sensor data and alerts generated by the autonomous drone system in real-time.

- **ThingSpeak** is a cloud-based IoT platform by MathWorks that allows devices like Raspberry Pi or Arduino to collect, store, visualize, and analyze sensor data in real-time using REST APIs and built-in MATLAB tools.

DRONE MANUAL MODE

The drone system features two primary operational modes: manual and autonomous. In manual mode, the FlySky FS-i6X 10-channel transmitter and receiver system provides direct pilot control, enabling complete command over flight operations including throttle, pitch, roll, and yaw. This mode serves multiple critical functions - it allows for real-time testing and calibration of flight parameters, provides a safety override during autonomous operations, and facilitates flight path recording by logging GPS waypoints during manual flights. The system continuously records comprehensive flight data including GPS coordinates, altitude measurements, sensor readings, and timestamped parameters during manual operation. This collected data forms the foundation for creating predefined autonomous flight routes, verifying system performance, training navigation algorithms, and establishing safe operational boundaries.

DRONE AUTONOMOUS MODE

For autonomous operations, the system leverages the ArduPilot open-source autopilot platform with Mission Planner serving as the Windows-based Ground Control Station (GCS) interface. The autonomous mode enables sophisticated mission capabilities through real-time GPS coordinate visualization, comprehensive telemetry monitoring, and system health diagnostics. The Raspberry Pi 5 acts as the mission computer, continuously logging and processing GPS data while storing complete flight telemetry, system events, and sensor readings in structured formats for post-mission analysis. Autonomous features include precise waypoint navigation with <1m accuracy, complex mission planning for various operational patterns (survey grids, perimeter tracking, search routes), and multiple redundancy systems. These include automatic return-to-launch protocols for signal loss situations, low-battery contingency routing, obstacle avoidance overrides, and priority switching to manual control when required. The seamless integration between manual and autonomous modes, combined with robust data recording capabilities, provides a flexible and reliable platform capable of adapting to diverse mission requirements while maintaining the highest safety standards.

Results

- Drone

Figure 5.3 Drone in flight mode

- Crowd Detection:

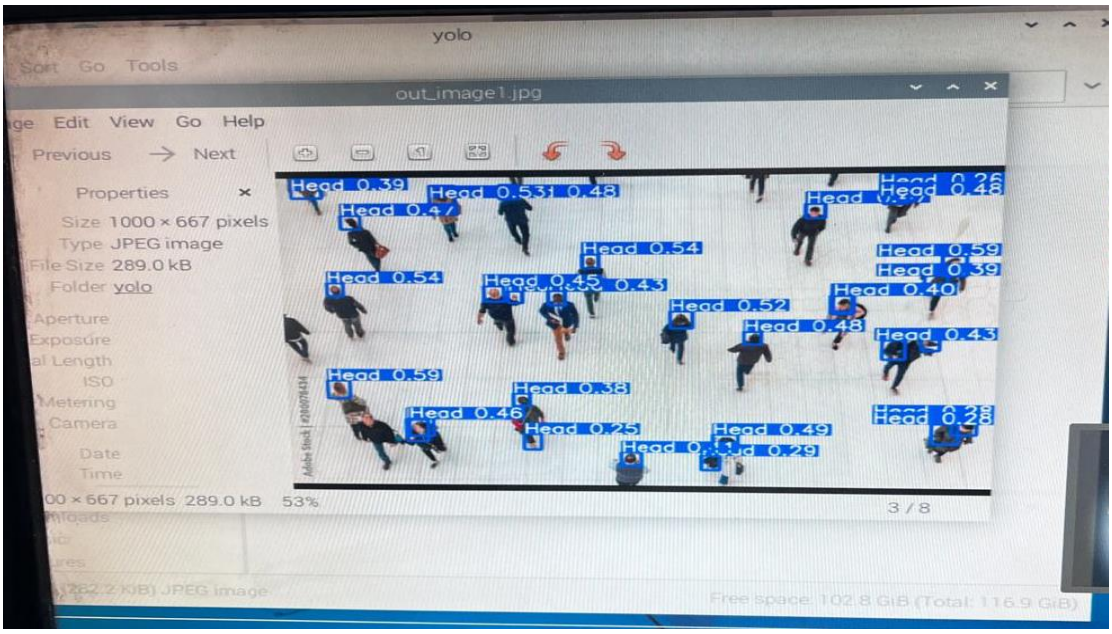

Figure 5.4 Crowd Detection Output

- Fire Detection:

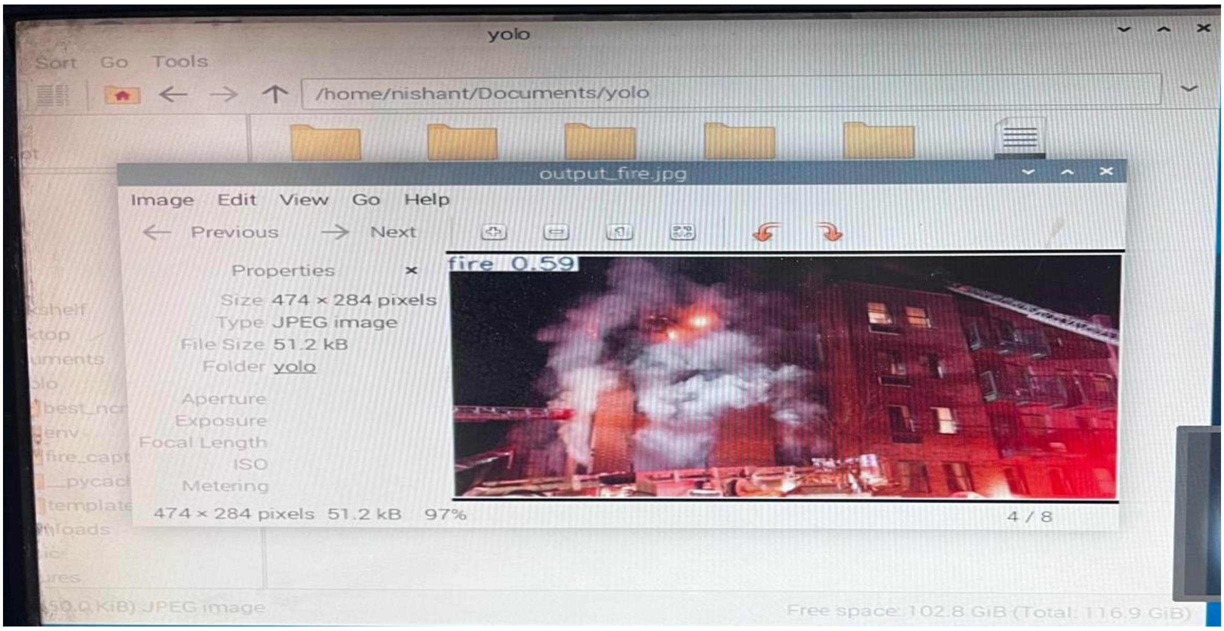

Figure 5.5 Fire/Smoke Detection Output

- Alert Records

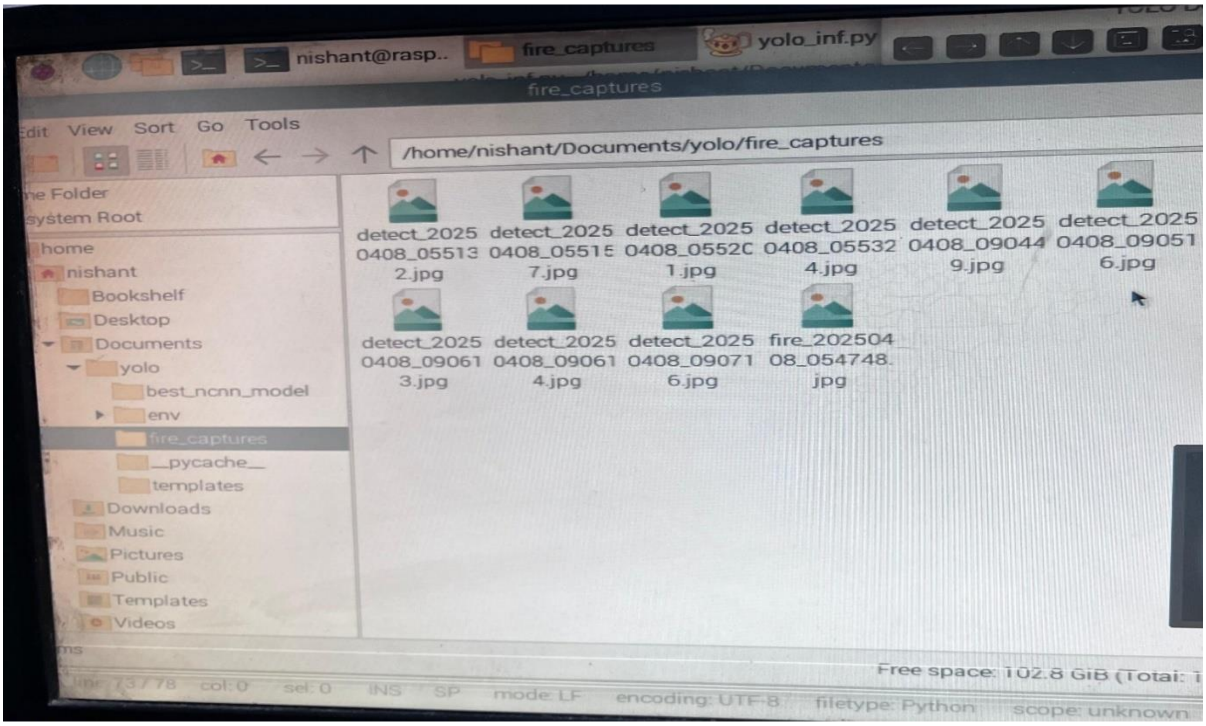

Figure 5.6 Alert Records

CHAPTER 6
DISCUSSION AND CONCLUSION

6.1 DISCUSSION

- A drone with a capability of stable fight control and autonomous surveillance was successfully built.
- We started with the type of drone we must build such as Quad Copter and Hexacopter.
- We decided to build a quadcopter and came up with the application as Campus Surveillance which will restrict the area of access for the drone and thus started with the plan of action.
- We planned in about one week, and came up with using Pixhawk 2.4.8 as the flight controller, F450 drone chassis, and Raspberry Pi for autonomous commands and Camera operations.
- As the hardware was getting assembled, we continued as per the plan and started data search for the anomalies that we were trying to encounter.
- Tested the drone flight in Manual mode.
- This project focuses on the development of a fully autonomous drone system capable of performing complex tasks that typically require manual control. The drone integrates GPS-based navigation, real-time obstacle avoidance, anomaly detection (fire, smoke, and crowd density), and emergency fail-safe mechanisms to ensure safe and reliable operation. The system combines hardware components—such as a Raspberry Pi 5, Pixhawk flight controller, ultrasonic sensors, GPS module, and camera—with software algorithms, including YOLOv5 Nano for AI-based detection, OpenCV for computer vision, and multiprocessing for parallel task execution, to achieve autonomous functionality.
- The primary aim of the project is to create a sturdy, adaptable, and intelligent drone that can operate autonomously while maintaining safety through fail-safe mechanisms. To accomplish this, several objectives were established. First, the drone must achieve stable and reliable flight, requiring proper assembly of the frame, electronic speed controllers (ESCs), and brushless motors, along with a well-tuned flight controller system. Second, precise GPS-based navigation was implemented using a Ublox NEO-M8N GPS module, enabling the drone to follow predefined waypoints autonomously. Third, a real-time obstacle avoidance system was developed using six ultrasonic sensors (HC-SR04) placed around the drone, allowing it to detect and avoid obstacles within a 1.5-meter radius.

- Additionally, the drone was equipped with real-time anomaly detection capabilities. A Raspberry Pi camera captures live video feeds, which are processed using the YOLOv5 Nano deep learning model to detect fire, smoke, and crowd density. If any anomalies are detected, the system generates alerts and logs images for further analysis. To ensure operational safety, emergency fail-safe mechanisms were integrated, including low-battery-triggered emergency landing, GPS signal loss recovery (Return-to-Launch), and a safe landing protocol that scans for obstacles before descending.
- The system also supports manual override via the FlySky FS-i6X transmitter, allowing human intervention when necessary. The software architecture was designed for efficiency, using Python for flight logic, OpenCV for real-time image processing, and multiprocessing to handle multiple tasks simultaneously. Finally, extensive testing and validation were conducted to ensure system reliability under various conditions.
- The hardware setup consists of a 450mm wheelbase quadcopter frame powered by 920KV brushless motors and 30A ESCs. The Raspberry Pi 5 (8GB) serves as the primary onboard computer, handling sensor data processing and AI model execution. A Pixhawk 2.4.8 flight controller manages flight stability, while a Ublox NEO-M8N GPS module provides precise location data. Six ultrasonic sensors (HC-SR04) enable obstacle detection, and a Pi Camera v2 captures real-time video for anomaly detection. Power is supplied by an 11.1V 5200mAh LiPo battery, with a voltage monitoring system to prevent over-discharge.
- The software components include YOLOv5 Nano for real-time fire, smoke, and crowd detection, trained on a custom dataset combining 15,432 fire/smoke images and 2,563 crowd density images. The system processes video feeds using OpenCV, while multiprocessing ensures smooth parallel execution of tasks such as sensor logging, flight control, and AI inference. For remote monitoring, ThingSpeak IoT is used to log alerts and sensor data in real time.
- The anomalies were fixed to fire and smoke detection, head count detection, person detection.
- The datasets were finalized as DITTO people detection and fire detection dataset from Roboflow Universe making the model more fire detection focused.
- We used YOLO v5 nano model for training on the data as it would be feasible to deploy the model on Raspberry Pi.
- Then we designed the IOT alert system based on Campus Wi-Fi providing it a gateway for direct communication with the monitoring station.

6.2 Challenges and solutions

- The Battery used for the drone application is Lipo battery which is Lithium Polymer battery. Since it does not have protective circuit like Lithium-ion battery, it needs to be cared to the fullest. We need to constantly check the battery voltage. Its minimum charge is on 11.4 V and maximum charge is 12.6 V. If the battery Voltage drops below 11 V, then the battery becomes dead permanently.
- The BLDC motors create a magnetic field interference with the ultrasonic sensor creating a need for specific arrangement of the sensor. It needs t be attached at a distance from the motors either below the chassis or preferably above the chassis.
- Ultrasonic sensor gets a disturbance in reading due to the propeller wash generating errors.
- The Raspberry Pi required a supply of 5V-5A, but usual cables provide up to 5V 3A at maximum. So, we need to connect a buck-convertor providing a required current and a suitable cable which supports 5A current. The buck-convertor is connected to the battery terminals
- GPS module does not catch signals in indoors making it capable of flying autonomously only in outdoor environments.

6.3 Conclusion

With our drone-based surveillance system university campuses will be equipped with higher security as compared to the traditional systems. The current technologies are equipped with aerial monitoring that provide a quick response for unwilling situations of fire by detecting the instance of even little fire or smoke. This will enable the authorities to also trace the fire location co-ordinated. It provides a real-time system to prevent crowd mishap leading to stampedes or crimes taking place at a larger rate.

At the expanding college campuses, we feel the need of aerial monitoring. Finally, the integration of drone technology is a step up in campus security and safety, providing a dynamic and holistic approach that augments the capacity of university authorities to safeguard their students, staff, and infrastructure in an ever-complicated environment. This proactive and responsive method not only tackles instant threats but also helps towards a safer and more peaceful learning space for the entire university community, building trust and well-being. Providing an aerial view and rapid response ability, drone surveillance provides an important layer of security that conventional systems are often lacking, making a safer and more secure campus for everyone.

6.4 Individual Contributions

Nishant Soman:

1. Computer Vision and Machine Learning Development:

- Led the deployment of the YOLOv5 Nano model for real-time object detection
- Designed and implemented the merging of datasets that included:
  - Fire/smoke detection dataset (15,432 images)
  - Crowd density dataset (2,563 images)
- Designed a single detection model with four different classes:
  - Person detection (full-body)
  - Head detection (for crowd counting)
  - Fire detection
  - Smoke detection
  - Tuned model performance for deployment on Raspberry Pi 5 hardware

2. Software Architecture:

- Implemented core Python modules for:
- Real-time video processing pipeline
- Integration with flight control and computer vision systems
- Data recording and alert creation
- Implemented multiprocessing for effective utilization of resources

Aditya Kende:

1. Drone Hardware Integration:

- Designed and assembled the entire quadcopter system consisting of:
  - Frame selection and structural integration
  - Motor and ESC setup (920KV brushless motors with 30A ESCs)
  - Power distribution system
- Implemented integration of sensor suite:
  - Ultrasonic sensor array (6-directional obstacle detection)
  - GPS module (Ublox NEO-M8N)

- Raspberry Pi 5 with Pi Camera v2

- Pixhawk 2.4.8 flight controller

2.Autonomous Navigation System:

- Developed obstacle avoidance algorithms:
- Real-time distance measurement processing
- Collision prevention maneuvers (stop, altitude modification, path recalculation)
- Programmed autonomous flight features:
- GPS waypoint navigation
- Fail-safe routines (RTL, emergency landing)

Pranav Rasane:

- Computer Vision and AI Integration:
  - Worked on YOLOv5 model development and dataset preparation
  - Helped in model optimization for edge deployment
  - Created post-processing algorithms for
  - Crowd density classification (sparse/moderate/dense)
  - Fire/smoke confidence thresholding
- IoT System Implementation:
  - Designed and deployed the IoT alert system using ThingSpeak platform
- Designed data transport protocols for:
  - Real-time abnormality notifications
  - Telemetry of sensor data
  - System status updates
  - Designed logging infrastructure
  - Event detection local storage
  - Cloud-based synchronization of alert of record
  - Timestamped picture storage
- Conducted multi-channel alert dissemination:
  - Campus WLAN integration
  - Visualization on IoT dashboard
  - Emergency notification system

REFERENCES

[1] S. S. R. B. S. P. R. College of Engineering, "Raspberry PI-based Surveillance Drone," Undergraduate Project Report, Rajarajeswari College of Engineering, Bengaluru, India, 2021.

[2] A. D. L. Majd and S. Tellex, "Pi Drone: An Autonomous Educational Drone Using Raspberry Pi and Python," Tech. Rep., Brown University, Providence, RI, USA, 2020. [Online]. Available: https://cs.brown.edu

[3] J. Luo, "Research on Unmanned Aerial Vehicle Path Planning," *J. Phys.: Conf. Ser.*, vol. 1132, no. 1, p. 012046, 2018, doi: 10.1088/1742-6596/1132/1/012046.

[4] A. Kumar and V. Kumar, "Raspberry Pi based Autonomous Drone," *Int. J. Comput. Sci. Mobile Comput.*, vol. 8, no. 5, pp. 112–118, May 2019.

[5] N. S. Pannu, S. Muradkar, and A. Chaurasia, "Image Processing Based Fire Detection System," *Int. Res. J. Eng. Technol. (IRJET)*, vol. 6, no. 4, pp. 5563–5566, Apr. 2019.

[6] A. R. Rezaei, M. Ahmadi, and S. H. Rezatofighi, "Real-Time Fire and Smoke Detection Using Deep Learning For Enhanced Safety," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Milan, Italy, 2020, pp. 1234–1241, doi: 10.1109/ICPR48806.2021.9412694.

[7] V. Sindagi and V. M. Patel, "A Survey of Recent Advances in CNN-based Crowd Density Estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4474–4486, Dec. 2020, doi: 10.1109/TCSVT.2019.2929622.

[8] A. Kumar, N. Gupta, and S. Singh, "Realtime Crowd Monitoring—Estimating Count, Speed and Direction of People Using Hybridized YOLOv4," *IEEE Access*, vol. 9, pp. 134625–134638, Sep. 2021, doi: 10.1109/ACCESS.2021.3117067.

[9] M. Sharma and R. Bansal, "Autonomous Drone Navigation with Obstacle Detection using Ultrasonic and GPS Sensors," *Int. J. Sci. Eng. Res.*, vol. 10, no. 4, pp. 1241–1245, Apr. 2019.

[10] F. A. Cheein, J. L. Sanchez, and H. R. Gonzalez, "A Review on Obstacle Detection and Avoidance for Unmanned Aerial Vehicles," *IEEE Access*, vol. 8, pp. 202321–202342, Oct. 2020, doi: 10.1109/ACCESS.2020.3036332.

[11] S. Sharma, R. Patel, and A. Yadav, "IoT and AI-Based Smart Surveillance Drone System for Crowd and Fire Detection," *Int. J. Innov. Technol. Explor. Eng. (IJITEE)*, vol. 9, no. 5, pp. 118–122, Mar. 2020.

[12] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint*, arXiv:1409.1556, 2015. [Online]. Available: https://arxiv.org/abs/1409.1556

[13] A. Rehman, N. Masood, and S. Arif, "Autonomous fire extinguishing system," in *Proc. Int. Conf. Robot. Autom. Inf. (ICRAI)*, 2012, doi: 10.1109/ICRAI.2012.6413387.

[14] D. R. Patrikar and M. R. Parate, "Anomaly detection using edge computing in video surveillance system: Review," *Int. J. Multimedia Inf. Retr.*, vol. 11, no. 2, pp. 85–110, Jun. 2022.

[15] C. Lu, J. Shi, and J. Jia, "Abnormal event detection at 150 FPS in MATLAB," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2720–2727, Dec. 2013.

[16] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 6479–6488, Jun. 2018.

[17] Y. Ito, T. Sasaki, and S. Kondo, "Crowd Anomaly Detection From Drone and Ground," [Online]. Available: https://ieeexplore.ieee.org/document/10879365/references#references

[18] W. Liu, Z. Zhang, and K. Cai, "Bi-level programming based real-time path planning for unmanned aerial vehicles," *Knowl.-Based Syst.*, 2013. doi: 10.1016/j.knosys.2013.01.011

[19] H. Duan and L. Huang, "Imperialist competitive algorithm optimized artificial neural networks for UCAV global path planning," *Neurocomputing*, 2012. doi: 10.1016/j.neucom.2012.09.039

[20] Y. V. Pehlivanoglu, "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV," *Aerosp. Sci. Technol.*, 2011. doi: 10.1016/j.ast.2011.02.006

[21] H. Freitas, B. Faiçal, and A. Silva, "Use of UAVs for an efficient capsule distribution and smart path planning for biological pest control," *Comput. Electron. Agric.*, 2020, doi: 10.1016/j.compag.2020.105387.

[22] Y. Zhao, Z. Zheng, and Y. Liu, "Survey on computational-intelligence-based UAV path planning," *Knowl.-Based Syst.*, 2018, doi: 10.1016/j.knosys.2018.05.033.

[23] K. T. Kamalraj, A. Kumar, R. Jeevan, N. Gowda, and D. B. Dhami, "Raspberry PI-based Surveillance Drone," *Int. J. Mod. Res. Sci. Eng. Technol. (IJMRSET)*, vol. 7, no. 8, 2024. doi: 10.15680/IJMRSET.2024.0708046. [Online]. Available: http://www.ijmrset.com/upload/46_Raspberry.pdf

[24] M. Kumar, N. Kumar, and T. H. Sreenivas, "Autonomous Navigation of Flying Quadcopter," *Int. J. Res. IT Comput. Commun. (IJRITCC)*. [Online]. Available: http://www.ijritcc.org/.

Gmail Nishant Soman <nishantsoman03@gmail.com>

Submission to Engineering Science and Technology, an International Journal - manuscript number

Engineering Science and Technology, an International Journal <em@editorialmanager.com> 22 April 2025 at 18:32
Reply-To: "Engineering Science and Technology, an International Journal" <support@elsevier.com>
To: Nishant Santosh Soman <nishantsoman03@gmail.com>

This is an automated message.

Manuscript Number: JESTCH-D-25-00697
`Autonomous Campus Surveillance Drone

Dear Mr Soman,

Your above referenced submission has been assigned a manuscript number: JESTCH-D-25-00697.

To track the status of your manuscript, please log in as an author at https://www.editorialmanager.com/jestch/, and navigate to the "Submissions Being Processed" folder.

Thank you for submitting your work to this journal.

Kind regards,
Engineering Science and Technology, an International Journal

More information and support

You will find information relevant for you as an author on Elsevier's Author Hub: https://www.elsevier.com/authors

FAQ: How can I reset a forgotten password?
https://service.elsevier.com/app/answers/detail/a_id/28452/supporthub/publishing/kw/editorial+manager/

For further assistance, please visit our customer service site: https://service.elsevier.com/app/home/supporthub/publishing/. Here you can search for solutions on a range of topics, find answers to frequently asked questions, and learn more about Editorial Manager via interactive tutorials. You can also talk 24/7 to our customer support team by phone and 24/7 by live chat and email.

#AU_JESTCH#

To ensure this email reaches the intended recipient, please do not delete the above code

____

In compliance with data protection regulations, you may request that we remove your personal registration details at any time. (Use the following URL: https://www.editorialmanager.com/jestch/login.asp?a=r). Please contact the publication office if you have any questions.