Autonomous Search and Rescue Drone

Zahra A. Elashaal[1], Yousef H. Lamin[1], Mohaned K. Elfandi[1], Adel A. Eluheshi[2]

[1] Faculty of Information Technology, University of Tripoli, Libya.
[2] School of Applied Sciences and Engineering, Libyan Academy for Postgraduate Studies.
z.elashaal@uot.edu.ly

Abstract. One innovative initiative that shows how technology and creativity can save lives in dire circumstances is the creation of a smart autonomous drone system for Search and Rescue Operations in Libya. The Search and Rescue Drone is a ray of hope that is intended to transform rescue operations by offering a quick and effective way to find persons who are in trouble or who may have been lost in Libya's desert or Mediterranean Sea. A Raspberry Pi, a Pixhawk flying controller, the Internet of Things, and a specially created mobile application are the main parts of the study. With the help of the YOLOv4-tiny module and object detection algorithms, the system enables users to operate the drone and quickly and accurately identify those who go missing in challenging environments. By fusing technological innovation with a humanitarian goal, this paper paves the way for future search and rescue operations in Libya and other nations to be safer and more responsive. The work shows how technology can save lives in dire circumstances and serves as an example of how it can benefit humanity at its most vulnerable times.

Keywords: Drone, Unmanned aerial vehicles (UAVs), Search and Rescue Operations, Pixhawk, Raspberry Pi, YOLOv4 detection algorithm.

1 Introduction

Drones, also known as un-manned aerial vehicles (UAVs), have the potential to revolutionize search and rescue operations by providing rapid scanning of wide geographic areas from the air, enabling them to navigate rough terrain that would be dangerous for human rescuers. They also enhance visual capabilities and enable well-informed decision making by providing real-time aerial data. Drones are increasingly used in research and rescue operations due to advancements in technology and affordability. They enhance situational awareness, aid search and rescue efforts, and support disaster management. In Libya, a drone system with object detection capabilities is being developed to improve search and rescue operations. The drone uses a camera and computer vision algorithms to locate missing people in far-off places, and the Flutter app provides real-time control and feedback for users. This system addresses the shortcomings of conventional techniques and incorporates cutting-edge technologies into search and rescue operations. The drone system includes hardware components, a Pixhawk 2.4.8 flight controller, a Raspberry Pi 4 for seamless communication, a Flutter app with an intuitive user interface, an object detection algorithm YOLOv4-tiny module, and autonomous features like GPS-based navigation.

2 Related Work

Due to its capacity to cover large areas quickly and efficiently and to provide real-time situational awareness, unmanned aerial vehicles (UAVs) are becoming more and more common in search and rescue missions. Drones with a variety of sensors, including cameras, thermal imaging devices, and LiDAR, can find and identify people or things in places that are hazardous or challenging for human rescuers to reach. In search and rescue missions, drones are typically used to search for missing persons or survivors in disaster areas and identify dangerous areas or obstacles that may pose a risk to rescue personnel. Drones can also help coordinate rescue efforts and offer real-time situational awareness to aid in decision-making [1, 2]. Drones can also deliver medical personnel, equipment, and supplies to inaccessible or remote locations. Martinez-Alpiste et al. [3] achieved human detection using convolutional neural networks on smartphones and drones. Yang et al. [4] employed reinforcement learning (RL) to accomplish path planning and combined unmanned aerial vehicles and unmanned surface vehicles for maritime search and rescue. To increase the effectiveness and dependability of search and rescue operations, Gotovac et al. [5] employed convolutional neural networks after using drones to pre-acquire aerial images. This method's inability to detect changes in real time is a problem. To increase the accuracy of target localization, YOLOv2 [6] and YOLOv3 [7] both use the Faster R-CNN concept and include an anchor box. As a result of algorithmic improvements, YOLOv4 has performed better with more evenly balanced optimization of speed and accuracy. A drone dataset for human action recognition has been suggested in this work [8] that leads them to suggest a model for human detection and action recognition. When applied it to the standard Okutama dataset, the proposed model outperforms the most advanced detection methods reported in the literature by 7%.

3 The Drone Design and Hardware Setup

This study aims to develop an intelligent, autonomous drone system for search and rescue missions. It details the hardware configuration and architectural layout, integrating software and hardware for reliable operation and smooth communication. The research includes setting software and firmware on flight controllers and Raspberry Pi 4, as well as assembling hardware components like frames, motors, sensors, and communication modules.

3.1 System Architecture

Figure 1, depicts the high-level, or "all overview," architecture of the Drone system, which serves as the basis for its operation and includes the fusion of software and hardware layers. The drone's architecture makes it possible for it to function independently, identify objects, and carry out search and rescue operations efficiently.

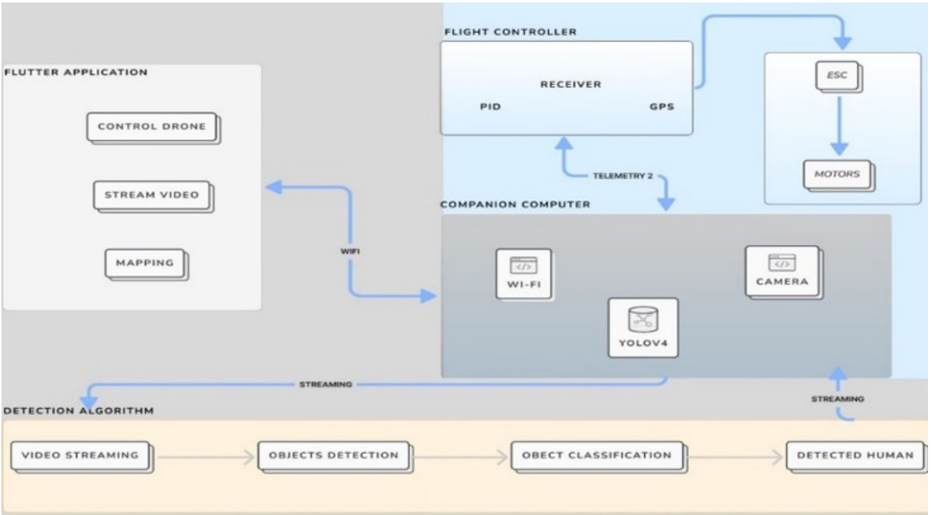

Fig. 1. All overview architecture of Drone system.

3.2 The Drone's final design

A Drone system typically consists of various components, as shows in the Figure 2, that work together to perform search and rescue operations efficiently.

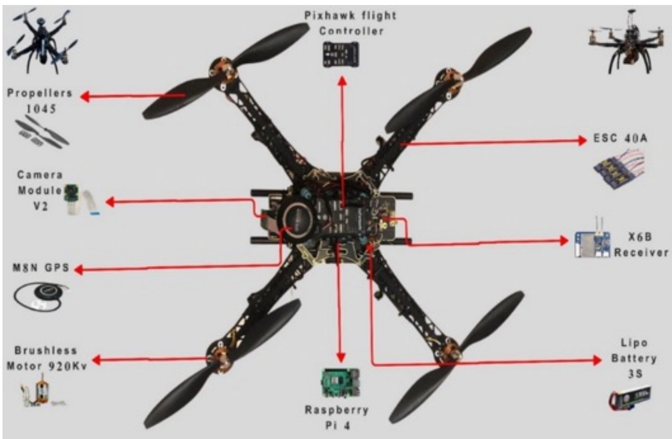

Fig. 2. Main Hardware Components of the Drone.

The Drone architecture consists of several layers that work together to enable the functionality of the drone. The Flight Controller Layer is responsible for controlling the drone's flight operations, utilizing the Pixhawk 2.1 flight controller and interfacing with various sensors to ensure accurate positioning and stabilization. The Communication Layer enables bidirectional data exchange between the drone and external entities using the MAVLink protocol. The Control Application and User Interface, developed using Flutter and Dart, serve as the user interface and control center, allowing users to plan missions, monitor flight status, and interact with the drone's functionalities. The Raspberry Pi 4 Layer acts as the computational hub, performing tasks such as real-time object detection using YOLOv4 and OpenCV algorithms. The object detection outputs are then transmitted back to the Raspberry Pi 4 through the communication layer.

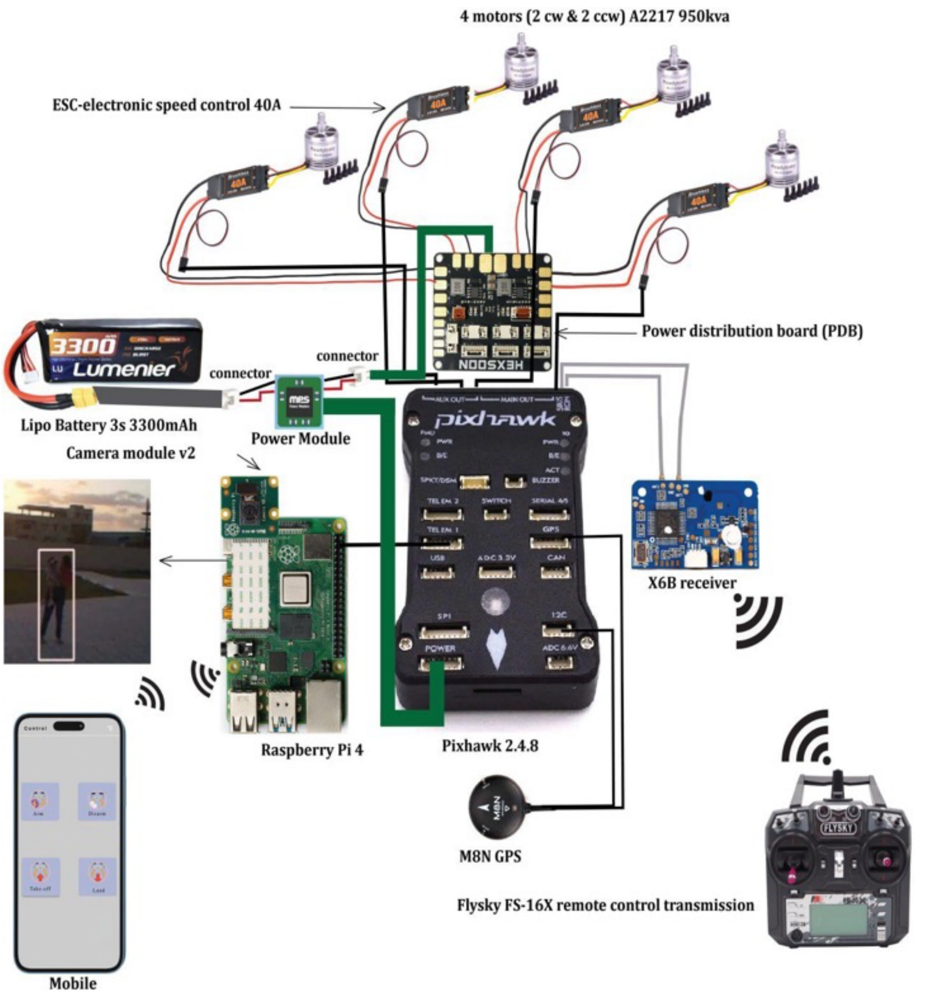

Fig. 3. The Drone System Schematic.

3.3 The Drone Schematic and Design

The autonomous drone is built on the careful selection and integration of various hardware components. The Power Distribution Board PDB distributes power to the battery and brushless motors, while the brushless motors propel the drone. The Flight Controller manages the drone's flight operations and has several connections, including GPS for accurate positioning and navigation, where the flight controller's GPS module enables autonomous navigation and positioning, allowing the drone to follow pre-defined waypoints and return to home. And if internet connectivity drops, it switches to a failsafe mode. a buzzer for sound notifications and alarms, an RC Receiver for manual control or telemetry data, a Raspberry Pi connected to the telemetry2 port for communication, and a camera for streaming video and object detection. The Raspberry Pi also provides Wi-Fi for communication between the Raspberry Pi and the mobile app. The system aims to provide and ensure a seamless blend of reliability, performance, and versatility. Figure 3, illustrates how various components are interconnected to form a cohesive system.

Hardware Components of the Drone System

As shown in the list below, the Drone System requires many electronic cards and hardware parts to perform object detection as shown in Table 1.

Table 1. The Drone's Hardware Components.

Hardware Component	Description
Flight Controller (Pixhawk).	Flight Controller (Pixhawk): The Pixhawk 2.4.8 Flight Controller enhances autonomous drone flight control, enabling navigation, stabilization, communication, and complex mission execution for search and rescue operations. It integrates sensors, processors, and algorithms, making it a valuable choice. its features are: The Pixhawk Flight Controller is a 32-bit ARM Cortex M4 core with FPU, 168 MHz/256 KB RAM/2 MB Flash, and 32-bit failsafe co-processor. It features a MPU6000 accelerometer, ST Micro 16-bit gyroscope, ST Micro 14-bit accelerometer/compass, and MEAS barometer. It is ideal for diode controllers with automatic failover and has 5x UART serial ports, 2 high-power capable ports, Spektrum DSM/DSM2/DSM-X Satellite input, Futaba S.BUS input, PPM sum signal, RSSI input, I2C, SPI, 2x CAN, USB, and ADC inputs.
Raspberry Pi 4 with Camera Module V2.	The Raspberry Pi 4 Model B has improved memory, networking, multimedia capabilities, and CPU speed, making it a versatile and efficient computational core for the drone system. Its high-resolution image capacity, coupled with the Broadcom BCM2711 quad-core processor, enables precise visual data analysis for search and rescue missions. The device also features IEEE 802.11ac wireless, Bluetooth 5.0, BLE, Gigabit Ethernet, 2 USB 3.0 ports, and a micro-SD card slot. It also supports H.265, H264, OpenGL ES 3.1, Vulkan 1.0, and PoE enabled.

S500 Quad-copter Frame.	The S500 quadcopter frame is the foundation of a smart autonomous drone system, supporting flight, navigation, and object detection. Its modular design allows easy hardware integration, serving as the central assembly point for motors, ESCs, flight controllers, and GPS modules.
950kva Brushless Motors.	The A2217 950kva brushless motors are essential for generating thrust in the drone, converting electrical energy into mechanical rotational force. They are securely mounted to the quadcopter frame, collaborating with the flight controller to achieve desired flight behavior and enable various maneuvering directions.
40A 2-6S ESC with 3.5mm Banana Connector.	The 40A 2-6S ESC in a drone regulates power for A2217 brushless motors, ensuring stable flight and responsive maneuvers. It's compatible with 2-6S LiPo batteries and integrates with the motor, interpreting flight controller signals for precise motor speed and thrust adjustment.
1045 Propeller CW&CCW.	The 1045 propellers, designed as CW and CCW variants, convert rotational energy into thrust, allowing stable flight. Compatibility with A2217 motors and quadcopter frame balances lift generation and flight dynamics, with each propeller securely attached to a motor shaft.
LiPo Battery 3S 3300mAh.	The 3S LiPo battery, with a 3300mAh capacity, powers the smart autonomous drone system, providing power to motors, flight controller, communication modules, and computational devices, ensuring optimal flight durations and securely mounting to the drone's frame.
20cm & 30cm LiPo Battery Strap Belt.	The 20cm and 30cm LiPo battery strap belts are essential accessories for securing the LiPo battery to the quadcopter frame, ensuring a secure attachment during flight. They provide a reliable, adjustable method for fastening the battery, enhancing drone safety and stability.
B6AC LiPo Battery Balance Charger.	The B6AC LiPo Battery Balance Charger is a tool for recharging LiPo batteries in smart autonomous drone systems, utilizing advanced charging algorithms and balance charging technology to ensure battery safety and longevity.
Power Distribution Board (PDB).	PDB is a component in distributing electrical power from the LiPo battery to electronic components in a smart autonomous drone system.
Flysky FS-I6X Remote Control Transmitter.	This Remote-Control Transmitter is a component for manual drone control, offering precise navigation, commanding, and parameter adjustment, while maintaining ergonomic design and compatibility.
X6B Receiver.	The X6B Receiver, a component of the Flysky FS-I6X Remote Control Transmitter, enables real-time communication between the drone and the operator, ensuring accurate control and communication.
M8N GPS Module.	The M8N GPS Module is a crucial navigation component for autonomous drone systems, offering precise positioning, orientation, and global satellite-based information, ensuring flight planning and search and rescue operations.

Determining Quadcopter Orientation

The integration of dynamic system controllers and an attitude sensor is crucial for achieving stability in a quadcopter. The attitude sensor helps ascertain the aircraft's orientation with the earth's fixed inertial frame as shown in Figure 4 , Stabilizing the roll and pitch axes is essential for maintaining steady flying. The attitude sensor can also ascertain the aircraft's roll and pitch attitudes. While stability is required for the yaw axis, little drift can be addressed using the radio controller without significant loss of control. Although absolute yaw orientation cannot be measured using only an accelerometer and gyroscope, tracking changes in yaw orientation is sufficient for quadcopter control [9].

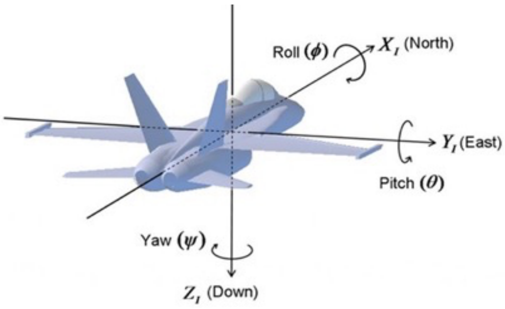

Fig. 4. Inertial Frame of a Free Body [9].

Proportional Integral and Derivative PID Control

To stabilize the quadcopter at the desired attitude, a dynamic system controller, such as a proportional, integral, and derivative (PID) controller, is implemented as shown in the Figure 5.

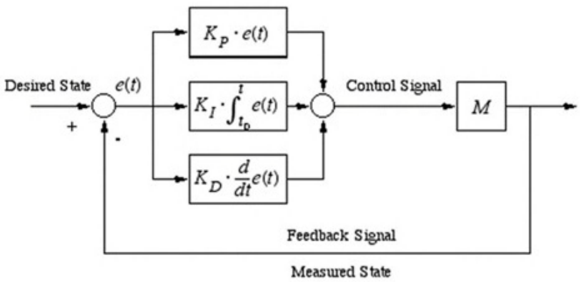

Fig. 5. Standard PID Block Diagram [10].

The PID control consists of several steps: calculating the error between the set point and measured state, determining the proportional term from:

$$P = K_P . e(t) \tag{1}$$

based on the error e(t) multiplied by a proportional gain K_P, calculating the integral term as:

$$I = K_I . \int_{t_0}^{t} e(t) \tag{2}$$

by integrating the error over time multiplied by an integral gain K_I, and calculating the derivative term from:

$$D = K_D . \frac{d}{dt} e(t) \quad (3)$$

by taking the time derivative of the error multiplied by a derivative gain K_D. These terms are then summed as:

$$u(t) = P + I + D \quad (4)$$

The quadcopter uses separate PID controllers for roll, pitch, and yaw axes to produce controller output. The PID controller controls the rotation rate about the yaw axis, as absolute measurement of the yaw axis is not possible using only accelerometer and gyroscope. Proper tuning is essential for achieving desired performance [10].

4 The Drone Software Development

The software development phase is a crucial part of the Drone system, combining advanced algorithms and technologies to make the hardware functional. It involves creating, improving, and integrating software modules, providing insight into the challenges of turning concepts into workable code. This phase covers topics like software architecture, user interface design, and algorithm development, ensuring smooth interaction between the virtual world and physical hardware. The software development phase lays the groundwork for the system's operational excellence through careful coding, rigorous testing, and iterative refinement.

4.1 Software Application Architecture

The Drone system's Software Architecture is a blueprint for its organization and interaction, defining data flow, control, and communication among modules. It uses Activity Diagrams and Use Case Diagrams for visual clarity, in Figure 6, illustrating interactions between external actors and the system's various use cases, providing a detailed view of how components collaborate to fulfill specific functionalities. The user interacts with the drone application, while the Raspberry Pi handles image processing and communication, and the Flight Controller manages navigation and flight.

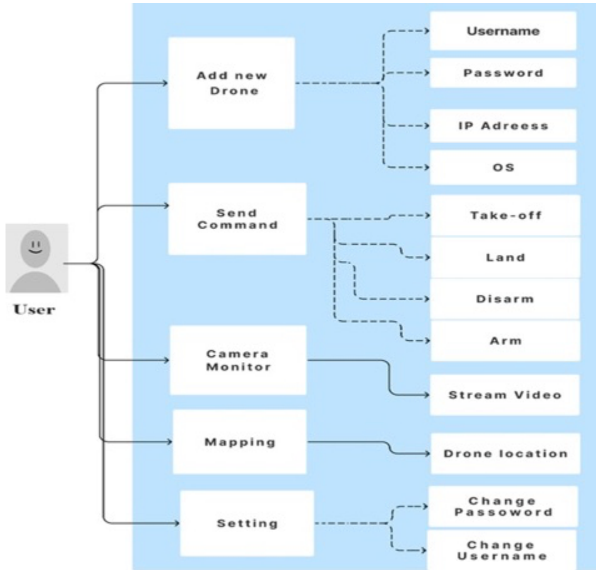

Fig. 6. The Drone's Use Case Diagram

The Raspberry Pi serves as a communication intermediary between the user and the flight controller, facilitating communication and object detection. as it is indicated in Figure 7.

It receives and relays user commands, captures, and streams live video for monitoring, and analyzes images for detection. The flight controller maintains stable flight using PID control algorithms and is connected to a GPS module for navigation and location determination. It also allows flight mode switching, enabling transitions between autonomous and manual control for testing purposes. The Raspberry Pi also has Wi-Fi communication capabilities, allowing data exchange with other components via commands and coordinates.

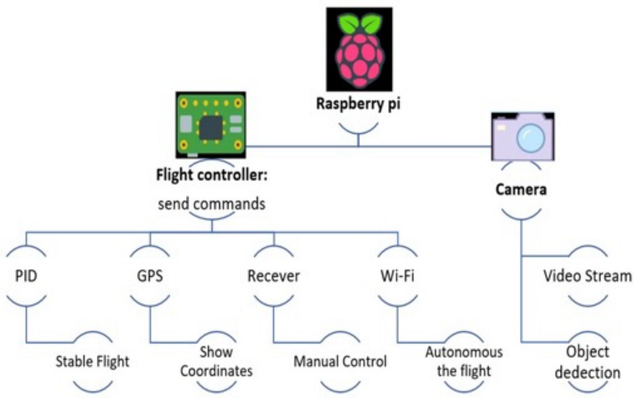

Fig. 7. Raspberry pi and Flight controller Use Case Diagram

Activity Diagrams for the Drone App

Activity diagram models system, shown in Figure 8, flow from user interactions to application responses, highlighting execution order and decision points, providing a clear path from user interactions to application responses.

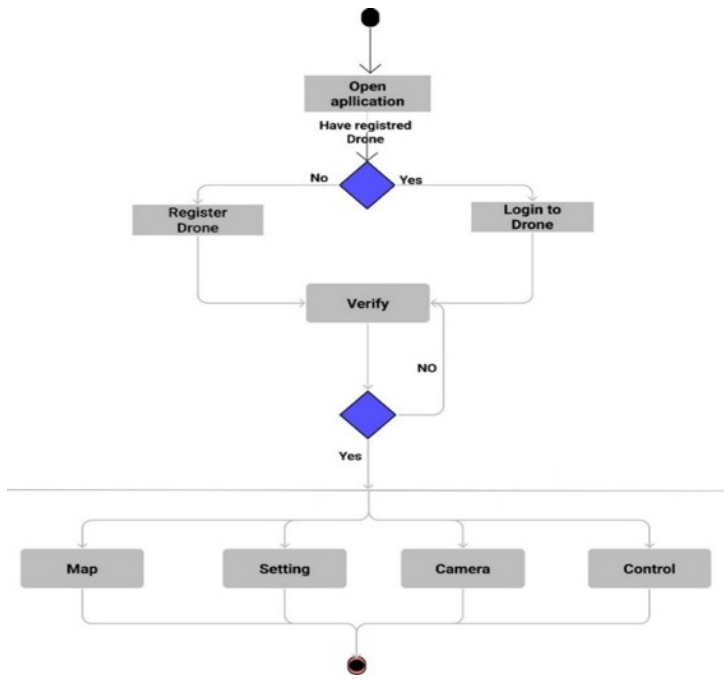

Fig. 8. Application Activity Diagram

Data and Control Plane for the Drone System

The Drone system outlines the communication and control flow between the Raspberry Pi and the flight controller in the system as shown in Figure 9. The main process sends commands and instructions to the Raspberry Pi, while the flight controller receives commands and performs tasks accordingly. The flight controller can change its mode to "guide mode" for autonomous flight operations. A pre-arm check ensures drone safety and readiness, then arming and initiating flight.

Once the drone is armed and in flight, the script running on the Raspberry Pi executes its intended tasks, such as object detection using YOLO and tracking. The flight controller reverts to its default mode after completing its tasks, ensuring the drone returns to normal operation. The flight controller then disarms the drone to end the flight. The Raspberry Pi initiates a process to request video streaming from the drone using the Real-Time Transport Protocol for Streaming (RTPS) protocol, allowing real-time video feed to be displayed on the Raspberry Pi. The YOLO module is used for object detection in video processing, detecting missing persons in the video stream. Relevant information is displayed on the screen for monitoring.

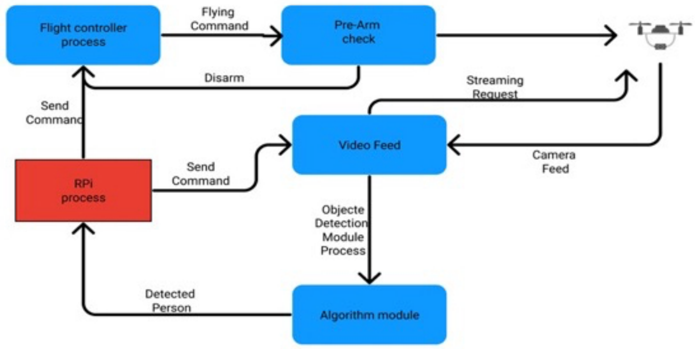

Fig. 9. Data and control plane.

User's Application

A new drone can be added to the user's list by entering its IP address, username, password, and Raspberry Pi operating system. The drone can be instructed to take off, land, arm, and disarm by them. The program can stream live video. GPS data can also be used by the user to locate the drone. They can change the username and password for the drone, these steps show Figure 10.

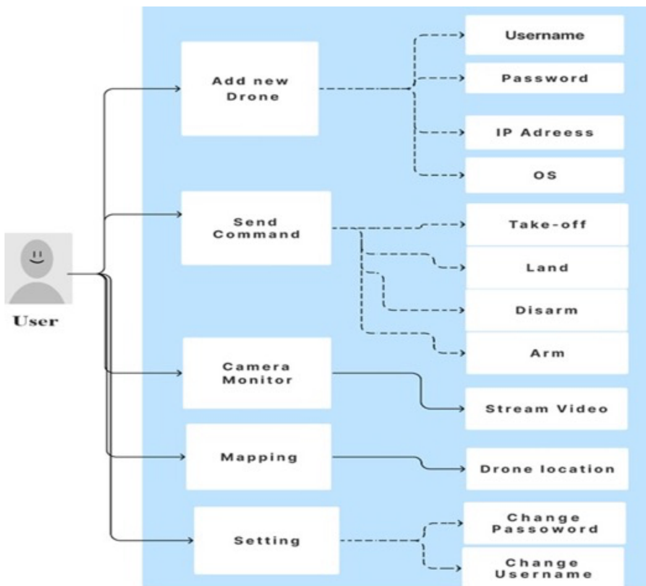

Fig. 10. User's application Use case Diagram.

4.2 Onboarding User Interfaces Application

The onboarding screens guide users through creating an intelligent drone app. The menu panel, featuring Control, Monitor, Mapping, and Settings, offers easy access to important features. Control allows users to control the drone, monitor monitors it, Mapping allows routes planning, and Settings allows users to personalize their experience. The menu screen is designed for easy navigation. Figure 11 displays some of the screens for Drone's application.

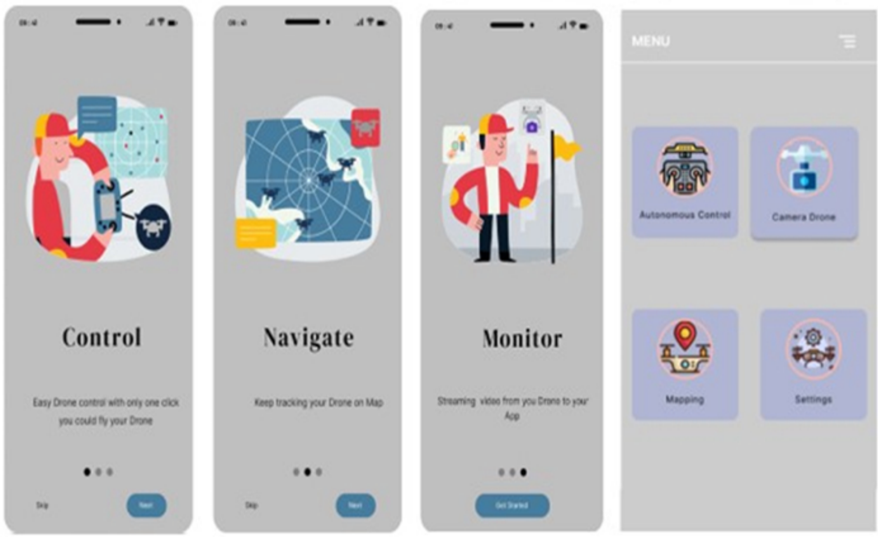

Fig. 11. The Drone user interface.

4.3 Object Detection Algorithm

The drone system uses a Raspberry Pi 4 board model to train a model in aquatic or desert environments. YOLOv4, a cutting-edge object detection algorithm, is used, improving on YOLOv3 with new techniques [11]. Known for its speed, accuracy, and Single-Pass Detection, YOLOv4 is ideal for real-time applications in smart autonomous drone systems. It can detect multiple object classes simultaneously in a single pass.

Using YOLO models

YOLO (You Only Look Once) models are highly efficient object detection systems, particularly in surveillance, robotics, and search and rescue operations. They process entire images or frames in a single pass, reducing computation time and memory requirements. YOLO's design optimizes object detection, making them suitable for real-time performance on embedded systems like Raspberry Pi. These models are compact and have fewer parameters than full-scale models, making them ideal for resource-constrained devices. The primary objective is to detect human victims, maintaining algorithm simplicity. The model's memory footprint must be under 4GB, aligning with embedded GPUs suitable for drone usage [12].

Darknet is an open-source C and CUDA framework for implementing DNN models, supporting both CPU and GPU computing. It is used in an autonomous drone system to implement YOLOv4, a neural network, with its modular architecture ensuring smooth integration with YOLO models. Darknet optimizes Raspberry Pi 4 platform performance and ensures seamless operation.

Average Precision and Mean Average Precision

Average Precision (AP) is a crucial metric in object detection and recognition tasks, assessing a model's ability to accurately identify and localize objects within an image or frame. It provides a balanced assessment of precision and recall, essential for evaluating bounding box predictions. AP is calculated by plotting a precision-recall curve based on varying confidence thresholds, dividing True Positives (TP) by False Positives (FP), and False Negatives (FN) to calculate the ratio of true positive detections (Recall) to the total number of actual positive objects being calculated (Precision). [14].

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

$$AP = \int (Recall) d(Precision) \tag{7}$$

In tasks involving object detection and recognition, Mean Average Precision (mAP) is a statistic that takes precision and recall into consideration across several object classes. It requires to calculate precision and recall values for various confidence score thresholds or Intersection over Union (IoU) thresholds. Then, to obtain mAP, the mean of these AP values is calculated.

Intersection over Union

Intersection over Union is a crucial metric in object detection and image segmentation, assessing the overlap between predicted and ground truth bounding boxes [15]. It determines the accuracy of detections and is calculated by:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{8}$$

IoU values range from 0 (no overlap) to 1 (perfect overlap), with thresholds like 0.5 and 0.75 used for classification. Figure 12 shows Mean Average Precision for the paper's work.

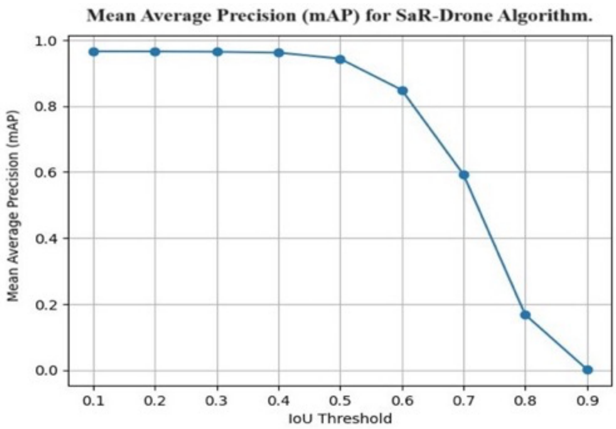

Fig. 12. Mean Average Precision for the Drone system.

Frames Per Second (FPS) in Real-Time Object Detection

FPS is a performance indicator in real-time object detection applications like autonomous vehicles and surveillance systems. It enables vehicles to quickly identify obstacles, pedestrians, and other vehicles, ensuring road safety and facilitating timely decision-making. In surveillance systems, high FPS is essential for real-time monitoring, enabling faster response times and increased situational awareness.

4.4 Object Detector Using YOLOv4

The drone system uses YOLOv4, an optimized model for Raspberry Pi 4, to train an object detector. The dataset used is the "Multi-Person Re-Identification and Tracking Dataset in Top View," ideal for search and rescue missions. The dataset provides top-view video footage and nearly 4000 manually labeled images in YOLO format, providing substantial diversity for the drone's mission scenarios. [13].

The YOLOv4-based object detection model undergoes a meticulous training process, including data loading, preprocessing, initialization, hyperparameter tuning, loss function, training loop, validation, testing, checkpoint saving, and model optimization. Preprocessing steps include resizing images, augmentation, and formatting annotations in YOLO format.

Training Process Results

After a lengthy training phase, the object detection model showed promising performance. At various intersection over union thresholds, the model's average precision for the "Person" class was high, hitting 96.54% at IoU 0.50, 96.49% at IoU 0.60, 96.40% at IoU 0.70, and 96.14% at IoU 0.80. The model consistently demonstrated great recall and precision across a range of scenarios, as evidenced by its impressive mean average precision of roughly 96.44% over varied IoU thresholds. With a total detection time of about 13 seconds, real-time or nearly real-time performance was shown.

The YOLOv4 model is initialized with weights pretrained on a large object detection dataset, which allows the model to leverage prior knowledge. Hyperparameters, such as learning rate, batch size, and number of training epochs, are fine-tuned to optimize training performance. A suitable loss function, often a combination of classification and localization losses, is employed to guide the model's learning. Figure 13 shows multiple object detection used in the Drone.

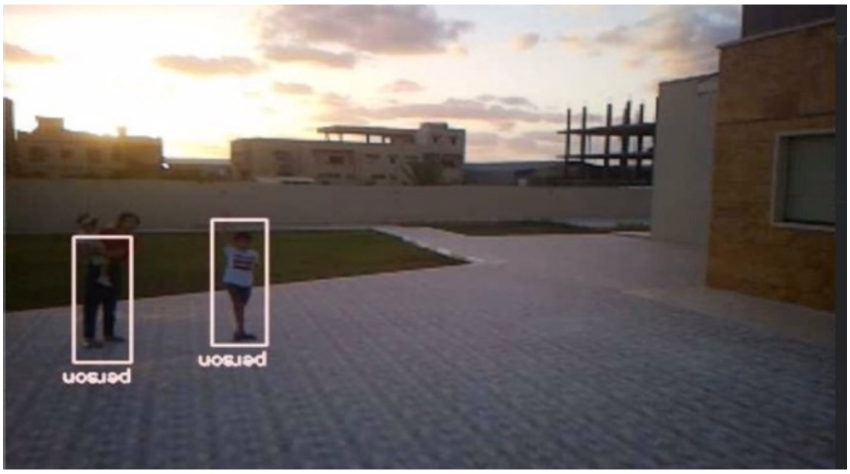

Fig. 13. Person Detection by the Drone system.

The model undergoes multiple training epochs, with periodic evaluation on the validation dataset to monitor progress and prevent over-fitting. Validation and testing metrics, such as mean average precision, are calculated to assess the model's performance. Checkpoints are saved at regular intervals during training, allowing for model restoration and continued training if needed.

5 Comparing the Drone to other studies

There is a slight distinction between this study and the other two study papers as shown in Table 2. In the original study paper, communication is established via telemetry radio, but in this project, the flutter program and the Raspberry Pi 4 communicate via built-in Wi-Fi, that, simplifying design and reducing hardware requirements. Wi-Fi offers a flexible platform for experimentation, allowing rapid prototyping and testing of various technologies. The choice to use Wi-Fi was driven by budget, Raspberry Pi's built-in capabilities, and its affordability, making it a cost-effective, accessible, and robust solution.

The second difference was, although research paper ref. [4] employed YOLOv7 Algorithms for Human-Detection, this study used a drone to detect objects using a Raspberry Pi 4 with Tiny-YOLO, which introduces certain processing capabilities in the Raspberry Pi with the Tiny-YOLO neural network.

Table 2. Comparison between the Drone system study and similar papers.

Function	Paper ref. [3]	Paper ref. [4]	The Drone sys.
Flight Controller	N/A	Pixhawk	Pixhawk 4
Companion Computer	Nvidia Jetson	N/A	Raspberry Pi
Communication	RF Telemetry	RF Telemetry	Wi-Fi
Algorithm	N/A	YOLOv7	Tiny-YOLO
Autonomy	YES	YES	YES
Real-Time	YES	YES	YES
GPS	N/A	YES	YES
GCS	YES	YES	Flutter App.

6 Cost of Hardware and Sensor

The Drone system operates under specific budget limitations for various components and expenses. The drone frame, motors, ESC, and other accessories have an estimated combined cost of approximately $300. Similarly, the Raspberry Pi 4, camera module, and related accessories are restricted to a total cost of $110. Additionally, the flight controller, Pixhawk, and its accompanying accessories do not exceed $140. The overall hardware project budget is capped at $750, equivalent to 3,750LDY, encompassing all expenses associated with prototype components, testing, and labor.

7 Conclusion

The paper presents the design and development of an autonomous search and rescue drone system for Libya, aiming to transform rescue operations by quickly and effectively locating missing persons. The system consists of a Raspberry Pi 4, a Pixhawk flight controller, the Internet of Things, and a custom mobile application developed using Flutter. The drone uses computer vision algorithms, specifically the YOLOv4-tiny object detection model, to identify and locate missing persons in challenging environments like the desert or sea. The system architecture includes the flight controller layer, communication layer, control application, and user interface. The hardware setup includes power distribution board, brushless motors, sensors, and communication modules. The paper demonstrates how technology can be integrated to enhance search and rescue operations and save lives in dire circumstances. The total budget for the hardware project is limited to $750, or 3,750LDY, and includes all costs related to labor, testing, and prototype componentry.

8 Future Work

By employing RF Telemetry Communication, the drone system seeks to increase its autonomy and range while concentrating on flight endurance, precise navigation, and flying skills. Additionally, it will make real-time data analysis and decision-making possible by fusing machine learning and artificial intelligence to react quickly to items that are spotted.

9 References

1. Valsan, Abhijith, B. Parvathy, Vismaya Dev GH, R. S. Unnikrishnan, Praveen Kumar Reddy, and A. Vivek. "Unmanned aerial vehicle for search and rescue mission." In 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184), pp. 684-687. IEEE, 2020.
2. Kulkarni, Shriyanti, Vedashree Chaphekar, Md Moin Uddin Chowdhury, Fatih Erden, and Ismail Guvenc. "UAV aided search and rescue operation using reinforcement learning." In 2020 SoutheastCon, vol. 2, pp. 1-8. IEEE, 2020.
3. B. Mishra and D. Garg, "Drone-surveillance for search and rescue in natural disaster," JournalPre-proof, p. 19, 2020.
4. Q. Li and X. Yang, "Vision ICE: Air–Ground Integrated Intelligent Cognition Visual," MDPI, p. 17, 2023.
5. Gotovac, Sven, Danijel Zelenika, Željko Marušić, and Dunja Božić-Štulić. "Visual-based person detection for search-and-rescue with uas: Humans vs. machine learning algorithm." Remote Sensing 12, no. 20 (2020): 3295.
6. Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263-7271. 2017.
7. Redmon, Joseph, and Ali Farhadi. "Yolov3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).
8. Mishra, Balmukund, Deepak Garg, Pratik Narang, and Vipul Mishra. "Drone-surveillance for search and rescue in natural disaster." Computer Communications 156 (2020): 1-10.
9. Kou, Kit Ian, and Yong-Hui Xia. "Linear quaternion differential equations: basic theory and fundamental results." Studies in Applied Mathematics 141, no. 1 (2018): 3-45.
10. Kumar, Mahendra, P. Vandana, and V. Patel. "Two degree of freedom PID controller for speed control of dc motor." American International Journal of Research in Science, Technology, Engineering & Mathematics 39 (2015): 94-97.
11. Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934 (2020).
12. Sowa, Piotr, and Jacek Izydorczyk. "Darknet on OpenCL: A multiplatform tool for object detection and classification." Concurrency and Computation: Practice and Experience 34, no. 15 (2022): e6936.
13. Prodaiko, I. "Person re-identification in a top-view multi-camera environment." PhD diss., Master Thesis, 2020.
14. Everingham, Mark, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. "The pascal visual object classes (voc) challenge." International journal of computer vision 88 (2010): 303-338.
15. Rezatofighi, Hamid, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. "Generalized intersection over union: A metric and a loss for bounding box regression." In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 658-666. 2019.