Forest Fire and Animal Detection Using Drone

R. Elavarasi[1], S. Arjun[2], P. Karthik[3], B. Manjunath[4], N. Naveen Reddy[5]
[1]Associate Professor, D*epartment of Electronics and Communication Engineering, Er. Perumal Manimekalai College of Engineering, Hosur, India*
[2,3,4,5]UG Students, D*epartment of Electronics and Communication Engineering, Er. Perumal Manimekalai College of Engineering, Hosur, India*

***Abstract*— The design and development of an autonomous drone aim to detect and alert forest fires and identify animals within forest areas, particularly when they encroach upon human settlements, thereby notifying the appropriate authorities. The drone is built using a plastic structure designed in a hexagonal shape for aerodynamic efficiency. Its control system is divided into two main components: the flight controller and the intruder detection camera control unit. The flight controller integrates a telemetry module, a receiver module, and an ESC (Electronic Speed Controller) module, all of which are managed through the Mission Planner control software. For image-based detection of forest fires and animals, the camera system operates on a Python platform. The drone is capable of autonomous operation using Auto mode, navigating via waypoints to perform fire and animal detection tasks. During testing, both manual and automatic flight modes were evaluated. In manual mode, the drone successfully identified fire and animals within the surveillance zone, provided the speed and camera range were suitably configured for effective detection. In autonomous mode, particularly at very low altitudes, the system demonstrated its highest accuracy in detecting and alerting potential threats. The effectiveness of the detection depends on factors such as the active range of the camera and maintaining an appropriate flight speed during manual operation.**

***Keywords*— Compact Drone, Camera, GPS, RF Communication, Python, OpenCV, YOLO, TensorFlow, GSM, Convolutional Neural Network (CNN) Model, Object Detection, Fire Detection, Real-time Monitoring.**

I.INTRODUCTION

Currently, search and rescue missions in forest fire scenarios rely heavily on human vision to detect fires and identify animals. However, during large-scale search operations, this method proves inefficient and often inaccurate, primarily due to the small size of animals in forest environments. Furthermore, the emotional and physical condition of search personnel can impact their performance, making it challenging to rapidly locate animal targets in fire-affected areas. With advancements in computer vision, object detection technology has garnered significant attention and remains a prominent area of research. Integrating this technology into search and rescue missions can greatly enhance search efficiency and accuracy, offering a more reliable alternative to human-dependent methods.

Object detection techniques are generally categorized into two main types: traditional object detection methods and those based on deep learning. Traditional methods often rely on hand-crafted features and include well-known algorithms such as Haar combined with SVM [1], HOG with SVM [2], and Shapelet with AdaBoost [3]. These approaches require manual feature design, which imposes strict demands on the expertise of the developers. In contrast, deep learning-based object detection methods utilize convolutional neural networks (CNNs) to automatically learn and extract features. These features tend to carry richer semantic information, resulting in improved robustness and accuracy, especially in complex environments. Researchers such as Liu Yu from the Microelectronics Institute at Tianjin University, Rufei Zhang from the AI Research and Development Center at the Beijing Institute of Control and Electronic Technology, and Niu Fu from the Institute of Systems Engineering at the Academy of Military Sciences in Beijing (niufu@vip.sina.com) have contributed to this field.

Deep learning-based object detection methods are generally classified into two categories: two-stage detection algorithms and one-stage detection algorithms. Two-stage algorithms, such as R-CNN [4], Fast R-CNN [5], and Faster R-CNN [6], begin by roughly generating candidate regions where a target might exist, followed by a refined extraction of these regions to determine the exact location and classification of objects. While this approach offers

high accuracy, it tends to suffer from slower detection speeds. On the other hand, one-stage detection algorithms—represented by models like YOLO and SSD—bypass the region proposal step and directly predict the object's location and category from the input image. These algorithms are significantly faster but generally have lower accuracy compared to their two-stage counterparts. Many recent advancements in object detection are built upon enhancements and optimizations of these foundational models.

II. RELATED WORK

1. Recent progress in autonomous drone technology has significantly improved real-time environmental surveillance. Drones equipped with sensors and cameras are now widely used to monitor natural phenomena such as forest fires, flooding, and illegal logging. For instance, thermal imaging systems integrated with UAVs can detect fires even in smoke-covered areas. However, traditional systems often rely on cloud-based processing, which introduces latency. To mitigate this, our proposed system adopts edge computing, enabling faster local data processing and quicker response times.
2. Earlier surveillance systems utilized traditional object detection methods like Haar Cascades and HOG combined with SVM, which depended on manually designed features and were limited in performance under complex conditions. The development of deep learning has led to the adoption of CNN-based models such as YOLO (You Only Look Once) and Faster R-CNN, which provide improved accuracy and speed. Among these, YOLO stands out for its balance of real-time detection capabilities and efficiency, making it highly suitable for aerial surveillance, wildlife tracking, and autonomous navigation applications.
3. Previous research has explored the use of drones for wildlife monitoring, where both thermal and RGB cameras are employed to detect and count animals. Deep learning models have been used to classify animals, such as elephants and zebras, from aerial images. However, many of these systems operated in a semi-autonomous manner, often requiring manual control. Additionally, existing wildlife detection systems lacked integrated alert mechanisms to notify authorities of animals encroaching into human-inhabited areas. Our proposed system addresses these limitations by incorporating real-time classification and GSM-based alert notifications for prompt responses.
4. Recent studies in edge computing for drones emphasize its potential to minimize data transmission delays. Unlike cloud-based platforms, edge computing enables image analysis directly on the drone or through a nearby embedded system. Research has demonstrated that this method enhances response times and reduces bandwidth usage, particularly in remote forest areas with limited connectivity. In our project, Python-based CNN models are implemented on Raspberry Pi devices for on-device detection of fires and animals. This approach improves the speed, autonomy, and reliability of the system, making it more effective in mission-critical scenarios.

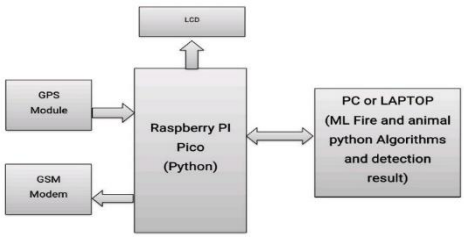
III. EXISTING WORK

Most current forest monitoring systems rely on manual surveillance, watch towers, and stationary cameras, which offer limited coverage and slow response times. These methods frequently fail to deliver real-time alerts, particularly in remote or expansive forest areas. While satellite imagery is sometimes utilized, it often lacks the necessary resolution and timeliness for early fire detection. Additionally, wildlife movement monitoring is not typically incorporated into these systems, leaving them incapable of preventing wild animal intrusions or quickly addressing rapidly spreading fires.

Cloud-based surveillance platforms leveraging IoT and sensor networks have emerged, but they come with several limitations. Key challenges include network latency, high power consumption, and the need for constant internet connectivity. In dense forests or rural areas, this connectivity is often unstable. Furthermore, most cloud systems lack the ability to process and respond to data locally, resulting in delayed critical alerts. These issues hinder the system's efficiency in real-time operations.

Animal detection systems, when implemented, typically rely on motion detectors or thermal cameras placed in fixed positions. These systems are static, costly, and have limited coverage. They lack mobility, autonomous tracking, and the ability to classify multiple species. Additionally, the absence of integrated alert mechanisms and mobile response features makes these solutions inadequate for

safeguarding both wildlife and human settlements. Therefore, there is a need for a more dynamic, autonomous, and intelligent system to address these limitations.



III. PROPOSED WORK

The proposed system features an autonomous drone equipped with edge computing and AI-powered image recognition to detect forest fires and wild animal intrusions. Using a Python-based CNN model, the system identifies threats in real-time, independent of continuous internet connectivity. The drone is fitted with a wireless camera, GPS, and GSM modules to capture, process, and send alerts instantly. It can operate autonomously through waypoints and also allows for manual control to enhance testing and improve accuracy. This combination ensures improved safety, rapid response, and effective surveillance of expansive forest areas.

Unlike traditional cloud-based systems, this drone performs on-device inference, significantly reducing both latency and power consumption. It utilizes a YOLO/CNN-based model to classify animals such as tigers, elephants, and lions, and to detect fires. Upon detecting a threat, the system promptly sends alerts to relevant authorities via GSM messaging, along with location tracking provided by the GPS module. This solution offers a low-cost, high-efficiency approach to real-time forest surveillance and animal tracking. Additionally, the drone's flexible design and modular setup make it scalable and adaptable to a variety of environments.

A. Key Features and Innovations

1. Edge Computing Integration

Onboard processing using Raspberry Pi ensures real-time detection without internet dependence, improving speed and efficiency.

2. AI-Based Detection

A CNN model trained in Python is used to identify animals and fire, enabling smart classification with high accuracy.

3. Autonomous Navigation

The drone follows pre-defined waypoints using Mission Planner, covering wide areas without manual control.

4. GSM and GPS Modules

Alerts and location details are sent immediately to authorities or local residents via SMS and map coordinates.

5. Multi-Class Object Detection

The system detects multiple animal types (e.g., tiger, elephant) along with forest fire, enhancing its environmental protection capabilities.

6. Compact and Modular Design

The drone uses lightweight components like Raspberry Pi Pico and wireless cameras, making it portable and easy to maintain.

7. Real-Time Image Transmission

With images and OpenCV, live footage is transmitted and analyzed instantly, providing visual confirmation and accuracy validation.

B. System Workflow

8. Drone Initialization and Calibration

The drone powers up, calibrates its sensors, and establishes a connection with the controller or autopilot module to initiate waypoint execution.

9. Live Video Capture

The wireless camera streams real-time video, which is then processed using a pre-trained CNN model on the Raspberry Pi.

10. Image Classification

The captured frames are analyzed to classify objects, such as fire or specific animals, using deep learning techniques like YOLO/CNN.

11. Threat Detection and Alert Trigger

Upon detecting a fire or a dangerous animal, an alert is triggered and sent to the microcontroller for appropriate action.

12. GSM Alert Transmission

The system sends an emergency SMS via the GSM module, including details about the threat and the GPS coordinates of the drone's location.

13. Live Monitoring Interface

The drone streams detection results visually through a connected display or live monitoring software, utilizing tools like imagezmq and OpenCV.

14. Waypoint Navigation Continuation

After sending alerts, the drone continues its patrol along the pre-set waypoints or can be redirected for detailed area scanning.

IV. IMPLEMENTATION DETAILS

A. Hardware Components and Design

15. Drone Frame (Hexagon-Shaped)
A lightweight plastic hexagonal frame supports the motors, electronics, and payload. It provides aerodynamic stability for both manual and autonomous flight, with easy mounting of sensors and modules.
16. Flight Controller
This is the central unit of the drone, managing motor speeds, GPS, and telemetry modules. It operates using Mission Planner software for auto-mode navigation, ensuring stable flight and waypoint execution.
17. Wireless Camera
Captures real-time video for fire and animal detection. It is connected to the Raspberry Pi for image classification using OpenCV and TensorFlow, enabling continuous aerial surveillance and detection.
18. Raspberry Pi Pico
Serves as the edge-computing platform for running Python-based detection models, handling image classification tasks with lightweight CNNs. It enables on-device processing, reducing latency and dependency on the cloud.
19. GPS Module
Provides the drone's real-time geographic coordinates. It is essential for waypoint navigation, sending location-based alert messages, and accurately mapping areas for threat localization.
20. GSM Module
Sends SMS alerts when a fire or animal is detected. It communicates with the controller using AT commands, transmitting details such as threat type and the drone's GPS location.
21. Power Supply (7805 Voltage Regulator)
Converts and regulates voltage for stable drone operation, providing a +5V DC output to microcontrollers and sensors. It also includes protection against overvoltage and overheating.
22. Motors with ESCs (Electronic Speed Controllers)
Four or more brushless motors provide lift and movement. ESCs control motor speed based on signals from the flight controller, enabling smooth, balanced, and agile drone operation.
23. 16x2 LCD Display
Displays detection status and drone activity feedback on a simple alphanumeric screen, interfacing with the Raspberry Pi or microcontroller. It is useful for manual testing or setup mode.

B. Software Components

24. Python Programming Language
Python is used to develop the entire image classification and alerting logic. It handles camera input and runs the CNN model for processing.
25. TensorFlow & Keras
These frameworks are used to create and train the CNN model for detecting fire and animals. The model is saved as a .h5 file and loaded during runtime on the drone, allowing efficient and accurate on-device classification.
26. OpenCV
OpenCV is used for capturing, processing, and displaying images from the drone's camera. It draws detection results on images and shows real-time footage. It works with imagezmq to stream video frames between devices.
27. imagezmq
A lightweight messaging library used to send image frames from the drone to a server or monitor. It enables real-time video transmission over TCP, providing visual confirmation and live monitoring.
28. Mission Planner
Mission Planner software is used to define waypoints and manage the drone's autopilot missions. It communicates with the flight controller for automated flight paths and provides telemetry data and map-based navigation.
29. Serial Communication (PySerial)
Python communicates with microcontrollers like Arduino or GSM through serial ports using PySerial. It reads sensor data and writes detection signals, ensuring proper coordination between subsystems like GPS, GSM, and the display.

V. RESULTS

The autonomous drone system was tested in both manual and automated waypoint modes to assess its real-time detection performance. During the trials, the drone accurately identified fires and different animal species, such as tigers, elephants, and lions, using the trained CNN model. The system demonstrated high accuracy in classification with few false positives, even under varying lighting and movement conditions. Live video streams displayed the detection results in real-time, providing assurance in the accuracy of the classifications during surveillance operations.

The system achieved an average detection response time of under 2 seconds from image capture to alert transmission. This quick turnaround was enabled by the edge computing setup, where image processing

was performed directly on the device using Raspberry Pi. The GSM module quickly transmitted threat alerts, including GPS coordinates, to relevant authorities. Such real-time response is crucial in mitigating forest fire spread or wildlife intrusion. Additionally, the model demonstrated reliable performance during short-range aerial operations.

The drone demonstrated efficient power consumption during multiple test cycles, operating without overheating or significant battery drain. Sleep modes and optimized sensor polling helped extend the drone's operational duration. Its compact hardware setup ensured a good balance and speed, which is crucial for forest operations. The integration of components such as GPS, GSM, camera, and Raspberry Pi Pico proved stable and well-matched, with the Raspberry Pi Pico handling GSM/GPS communication effectively.

The user interface, featuring live display, label overlays, and image logging, played a crucial role in validating the predictions. The detection model labeled objects with their corresponding names and confidence levels during live feed monitoring, enabling observers to track results in real-time and quickly identify any errors. Overall, the system demonstrated high accuracy, fast response times, and energy efficiency, reinforcing its potential for practical applications in forest monitoring and early warning systems.

VI. CONCLUSION AND FUTURE POSSIBLITY

This project demonstrates an autonomous drone system that can detect forest fires and animals in real-time using AI-powered image processing. By leveraging YOLO and CNN models along with edge computing, the system ensures fast detection with minimal delay. The integration of GSM and GPS modules allows for precise alert transmission and location tracking. Designed to be both efficient and low-cost, this system can be deployed across vast forest areas, providing early warning and surveillance capabilities. Its modular hardware and open-source software framework allow for customization and further development, making it adaptable to various needs.

The drone's capability to detect various animal species and fire events makes it a versatile tool for monitoring. Operating in Auto Mode with predefined waypoints, it can cover extensive areas autonomously, eliminating the need for manual control. The real-time video feed further aids authorities by providing enhanced visibility during emergency situations. This system overcomes key limitations of traditional surveillance methods, such as limited coverage, slow response times, and lack of mobility. Ultimately, it plays a significant role in environmental protection, wildlife management, and disaster response efforts.

Looking forward, the system can be enhanced by incorporating thermal and gas sensors to detect heat signatures and smoke more precisely. The inclusion of reinforcement learning could allow drones to make adaptive flight decisions based on threat zones. Swarm intelligence could also be introduced, where multiple drones collaborate for faster area coverage. Integration with cloud dashboards could offer long-term data logging, analytics, and centralized monitoring. These features will make the solution more powerful and suitable for national-scale forest monitoring.

REFERENCES

[1] B. Pareek, P. Gupta, G. Singal and R. Kushwaha, "Person Identification using Autonomous Drone through Resource Constraint Devices," 2019 Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), Granada, Spain, 2019, pp. 124- 129, doi: 10.1109/IOTSMS48152.2019.8939254.

[2] Papageorgiou C, Poggio T.A Trainable System for Object Detection[J]. International Journal of Computer Vision,2000,38(1):15-33.

[3] N. Dalal, B. Triggs. Histograms of oriented gradients for human detection. in: Computer Vision and Pattern Recognition, San Diego, CA, June 2025, 2005.

[4] P. Sabzmeydani, G. Mori. Detecting Pedestrians by Learning Shapelet Features[C]. IEEE Computer Vision and Pattern Recognition, Minneapolis, 2007, 1-8.

[5] J. Redmon, S. Divvala, R. Girshick, et al. You Only Look Once: Unified, Real-Time Object Detection[C]. IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas,2015,779-788.