IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ꕤ Peer-reviewed & Refereed journal ꕤ Vol. 15, Issue 3, March 2026

DOI: 10.17148/IJARCCE.2026.153124

AI AND IOT-ENABLED AUTONOMOUS ANTI-DRONE DEFENSE SYSTEM

Aien Joel Paul[1], Akhil Ginoy[2], Albin Babu[3], Mahesh T.S[4], Ann Josy John[5]

Electrical and Electronics Engineering, College of Engineering Kidangoor Kottayam, India[1-4]

Assistant Professor, Electrical and Electronics Engineering, College of Engineering Kidangoor Kottayam, India[5]

Abstract: The rapid growth of unmanned aerial vehicles (UAVs), commonly known as drones, has created new opportunities across industries such as surveillance, agriculture, and delivery systems. However, unauthorized or malicious drone usage poses serious security risks to sensitive areas including airports, military installations, and public infrastructure. This paper presents the design and implementation of an autonomous drone detection, tracking, and countermeasure system using an edge computing platform. The proposed system utilizes a deep learning-based object detection model implemented through YOLOv8 to identify drones from real-time video captured by a camera module. The system is deployed on a compact embedded platform based on Raspberry Pi 5 and optimized using the NCNN inference framework to enable efficient edge AI processing. Once a drone is detected, a pan–tilt servo mechanism automatically tracks the object using real-time positional feedback. The system also integrates a directional countermeasure module based on ESP32 and NRF24L01 to demonstrate a 2.4 GHz signal interference mechanism. A web-based monitoring interface provides live video streaming through HTTP Live Streaming generated using FFmpeg, enabling remote system monitoring and event logging. Experimental evaluation demonstrates the feasibility of implementing an intelligent drone monitoring system using low- cost embedded hardware and open-source software technologies.

Keywords: Drone Detection, UAV Monitoring, Edge AI, YOLOv8, NCNN, Raspberry Pi, Computer Vision, Drone Countermeasure, HLS Streaming, Embedded Systems

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) have become increasingly popular due to their versatility and accessibility. While drones are widely used for legitimate purposes such as aerial photography, mapping, surveillance, and delivery services, their misuse has raised significant security concerns. Unauthorized drones can be used for espionage, smuggling, or disruption of critical infrastructure.

Traditional drone detection methods often rely on radar systems or radio frequency detection equipment. However, these solutions are typically expensive and require specialized infrastructure. In recent years, computer vision and deep learning technologies have emerged as effective alternatives for detecting aerial objects using camera-based systems.

This research proposes an autonomous drone detection and tracking system that integrates computer vision algorithms with embedded hardware. The system uses a deep learning model to detect drones in real time and automatically track them using a mechanical turret mechanism. In addition to detection and tracking, the system also demonstrates a basic countermeasure module capable of generating directional signal interference in the 2.4 GHz communication band commonly used by consumer drones.

The primary objective of this project is to develop a low- cost and scalable drone monitoring system that can operate independently using edge computing technology.

II. LITERATURE REVIEW

Recent advancements in computer vision and deep learning have significantly improved the capability of detecting unmanned aerial vehicles (UAVs) in surveillance systems. Several studies have applied deep learning-based object detection algorithms, particularly the YOLOv8 architecture, for identifying drones in aerial images and real-time video streams. These models employ multi-scale detection mechanisms that allow the recognition of small drone objects appearing at different distances within a camera frame. Research has shown that deep learning-based detection systems provide high accuracy and reliable performance when trained with diverse datasets containing various environmental conditions, lighting variations, and drone orientations [1], [2].

Another research direction focuses on integrating object detection with tracking mechanisms to maintain continuous monitoring of moving aerial targets. Such systems are useful in applications such as border surveillance, restricted airspace monitoring, and infrastructure security. Studies comparing traditional image processing techniques with convolutional neural network-based approaches have demonstrated that deep learning models significantly outperform classical methods in detecting drones in dynamic environments [3].

However, deploying deep learning models on embedded sys- tems presents several challenges due to limited computational resources and power constraints. To address this problem, researchers have proposed lightweight detection architectures optimized for edge computing platforms. These models reduce computational complexity while maintaining acceptable detec- tion accuracy, enabling deployment on embedded hardware platforms such as the Raspberry Pi 5 [4]. Despite these advancements, many existing research efforts focus primarily on improving detection algorithms rather than developing fully integrated surveillance systems. Most studies evaluate detection accuracy using datasets or simulated envi- ronments without addressing practical integration challenges such as camera tracking, remote monitoring, and real-time data streaming [5].

The proposed Autonomous Drone Detection, Tracking, and Countermeasure System addresses these limitations by integrating deep learning-based detection using YOLOv8 with optimized inference using NCNN on an embedded edge computing platform. In addition to detection, the system incorporates a mechanical pan–tilt tracking mechanism and a web-based monitoring interface with live video streaming using FFmpeg and the HTTP Live Streaming protocol. This integrated architecture provides a practical framework for real- time drone monitoring applications.

III. SYSTEM ARCHITECTURE

The proposed Autonomous Drone Detection, Tracking, and Countermeasure System is designed as an edge-computing platform that integrates computer vision, embedded hardware control, and web-based monitoring. The system architecture consists of four major modules: the drone detection module, the tracking mechanism, the countermeasure module, and the monitoring interface.

The drone detection module performs real-time analysis of video frames captured by the camera connected to the Raspberry Pi 5. A deep learning model based on YOLOv8 is used to detect drone objects in the incoming video stream. The trained model is optimized using NCNN to enable efficient inference on the embedded platform. When a drone is detected, the system calculates the position of the detected object within the frame and generates positional data used for tracking.

The tracking module consists of a pan–tilt mechanism driven by servo motors controlled through the PCA9685 PWM Servo Driver. Based on the positional information obtained from the detection module, the system adjusts the orientation of the camera to keep the detected drone near the center of the frame. This enables continuous monitoring of the target as it moves within the surveillance area.

The countermeasure module is designed to demonstrate directional signal interference against drones operating in the 2.4 GHz communication band. This subsystem utilizes an ESP32 connected to NRF24L01 transceiver modules to generate interference signals. The activation of this module is controlled through a transistor-based switching circuit triggered by the main processing unit when a confirmed drone detection occurs.

To enable remote monitoring, the system includes a web- based interface that provides real-time video streaming and telemetry information. Video streaming is implemented using FFmpeg, which converts the camera feed into HTTP Live Streaming (HLS) format. The web application displays the live stream along with system status and detection logs, allowing users to monitor drone activity remotely.

Overall, the system architecture integrates artificial intelligence-based detection, mechanical tracking, wireless countermeasure demonstration, and remote monitoring capabilities into a unified embedded surveillance platform.

IV. SYSTEM DESIGN

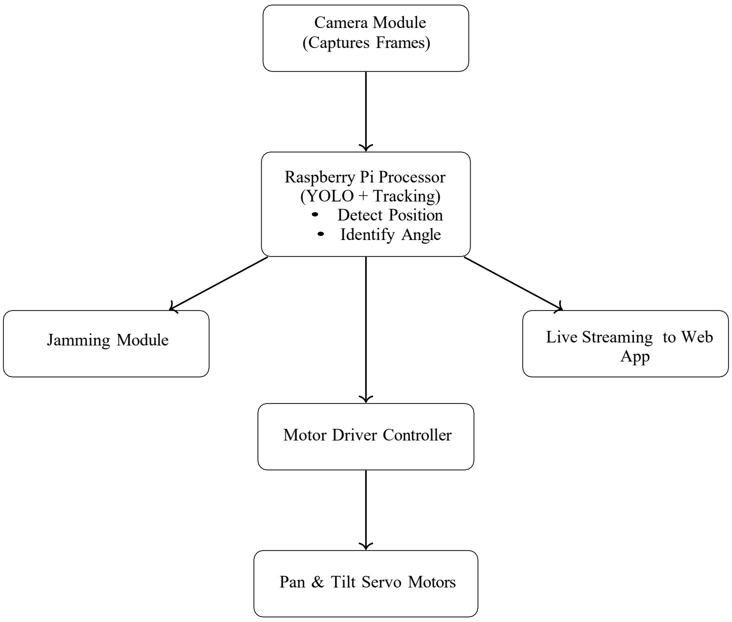


Fig. 1. Block Diagram of the Proposed System

A. Overview

The proposed Autonomous Drone Detection, Tracking, and Countermeasure System is designed as a comprehensive embedded surveillance platform that integrates artificial intelligence, real-time control mechanisms, and wireless communication technologies. The system is capable of detecting unmanned aerial vehicles (UAVs), tracking their movement dynamically, and activating a countermeasure mechanism while simultaneously providing remote monitoring capabilities.

The architecture follows a modular pipeline consisting of image acquisition, preprocessing, object detection, tracking control, countermeasure activation, and web-based monitoring. Each module operates independently while maintaining synchronized data flow through the central processing unit, the Raspberry Pi 5. This modular approach ensures scalability, ease of debugging, and efficient system performance.

B. System Workflow

The overall workflow of the system begins with continuous video acquisition from the camera module. The captured frames are preprocessed and passed to the detection module, where drone objects are identified using a deep learning model. Once a drone is detected, its position is used to calculate tracking error, which drives the pan–tilt servo mechanism.

Simultaneously, confirmed detections trigger the counter- measure module. The system also streams video data and system parameters to a web interface for real-time monitoring. This parallel execution of tasks is achieved using multithreading, ensuring minimal latency and efficient resource utilization.

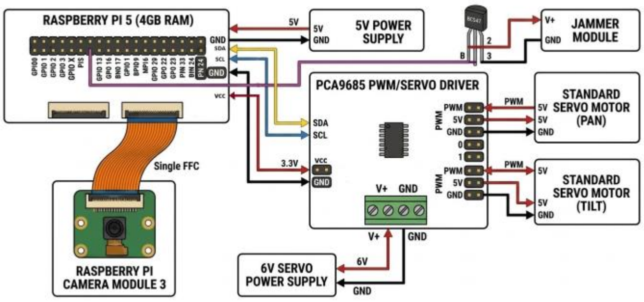


Fig. 2. System Workflow

C. Image Acquisition and Preprocessing

frames using a camera interfaced with the Raspberry Pi 5. The captured frames are processed at a fixed frame rate to maintain consistent system performance.

Preprocessing operations include resizing the image to the required input dimensions of the detection model, normalization of pixel intensities, and optional noise reduction. These steps improve detection accuracy and reduce computational overhead. Efficient preprocessing is essential to ensure that the system meets real-time processing requirements on an embedded platform.

D. Drone Detection Module

The detection module employs a deep learning-based object detection model using the YOLOv8 architecture. The model processes each frame and generates outputs in the form of bounding boxes, class labels, and confidence scores for detected objects.

To enable efficient deployment on embedded hardware, the trained model is optimized using the NCNN inference framework. This optimization reduces memory usage and improves inference speed, allowing real-time performance on the Raspberry Pi 5.

A confidence threshold is applied to filter weak detections, and a temporal validation mechanism ensures that only consistent detections across multiple frames are considered valid. This enhances system robustness and reduces false positives.

E. Tracking Mechanism

The tracking mechanism ensures continuous monitoring of detected drones by adjusting the camera orientation using a pan–tilt system.

1) Error Estimation: The tracking algorithm computes the positional error between the detected drone and the center of the image frame. Let (x_d, y_d) denote the detected object center and (x_c, y_c) denote the frame center. The error is calculated as:

$$e_x = x_d - x_c \qquad (1)$$

$$e_y = y_d - y_c \qquad (2)$$

These error values determine the required correction in pan and tilt angles.

2) Control Strategy: A proportional control mechanism is implemented to update the servo angles:

$$\theta^{new}_{pan} = \theta^{old}_{pan} + K_p \cdot e_x \qquad (3)$$

$$\theta^{new}_{tilt} = \theta^{old}_{tilt} + K_p \cdot e_y \qquad (4)$$

where K_p is the proportional gain constant. This approach ensures smooth tracking while maintaining system stability.

3) Normalization for Robustness: To maintain consistent tracking performance under varying resolutions, the error is normalized:

The image acquisition module continuously captures video

$$e'_x = \frac{xd - xc}{W}, \qquad e'_y = \frac{y_d - y_c}{H'} \qquad (5)$$

where W and H represent frame width and height.

F. Servo Control System

The pan–tilt mechanism is driven by servo motors controlled using the PCA9685 PWM driver. This driver generates high-precision PWM signals required for accurate positioning of servo motors. Communication between the Raspberry Pi and the driver is established via the I^2C protocol.

This design reduces the processing burden on the Raspberry Pi and enables smooth and precise control of the mechanical tracking system.

G. Countermeasure Module

The countermeasure subsystem is designed to demonstrate a response mechanism against detected drones. It operates in the 2.4 GHz communication band commonly used by UAVs. The module is implemented using an ESP32 microcontroller integrated with NRF24L01 transceiver modules. Upon con- firmed detection, the Raspberry Pi sends a trigger signal to activate the module.

A BC547 transistor-based switching circuit is used to control the activation safely. This ensures electrical isolation and protects the main processing unit from high-current components.

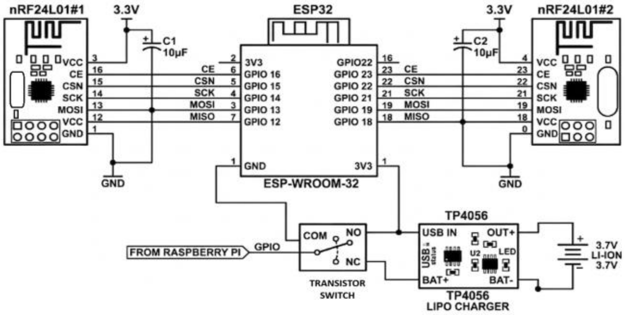


Fig. 3. Countermeasure Module

H. Web Monitoring System

The system includes a web-based interface for remote monitoring and control. Video streaming is implemented using FFmpeg, which encodes the video into a suitable format for transmission.

The encoded stream is delivered using HTTP Live Streaming (HLS), enabling users to access live video through a web browser. The web dashboard provides real-time system information, including detection status, servo orientation, and event logs.

Detection events are recorded and stored in spreadsheet format, allowing performance analysis and validation over time.

I. Multithreading Architecture

To achieve real-time performance, the system employs a multithreading architecture. Independent threads are used for object detection, tracking control, web communication, and video streaming.
This parallel processing approach minimizes latency and ensures that critical operations such as detection and tracking are executed without interruption.

J. Summary

The system design integrates deep learning-based detection, real-time tracking, countermeasure activation, and remote monitoring into a unified embedded platform. The modular and scalable architecture ensures efficient operation on resource- constrained hardware while providing flexibility for future enhancements such as multi-sensor integration and advanced countermeasure techniques.

V. RESULTS AND DISCUSSION

A. Overview

The proposed Autonomous Drone Detection, Tracking, and Countermeasure System was tested under various real-world conditions to evaluate its performance in terms of detection accuracy, tracking efficiency, and system responsiveness. The system was deployed on a Raspberry Pi 5 and tested using live video input from a camera module.

DOI: 10.17148/IJARCCE.2026.153124

B. Drone Detection Performance

The YOLOv8-based detection model demonstrated reliable performance in identifying drones in real-time video streams. The model was able to detect drones at different distances, orientations, and lighting conditions.

The performance of the detection model was evaluated using standard metrics such as Intersection over Union (IoU) and mean Average Precision (mAP). The results are summarized in Table I.

TABLE I
DRONE DETECTION PERFORMANCE METRICS

Metric	Value
Mean Average Precision (mAP)	0.89
Intersection over Union (IoU)	0.76
Detection Confidence Threshold	0.5
Frame Rate (FPS)	18–22

The results indicate that the model achieves high detection accuracy while maintaining real-time performance on an embedded platform.

C. Tracking Performance

The pan–tilt tracking system successfully followed the detected drone by continuously adjusting servo angles based on positional error. The proportional control algorithm ensured smooth tracking with minimal oscillations.

The system was able to maintain the drone near the center of the frame under moderate movement conditions. However, slight delays were observed when tracking fast-moving drones due to processing limitations.

D. Countermeasure System Performance

The countermeasure module was successfully triggered upon confirmed drone detection. The ESP32 and NRF24L01-based system generated interference signals in the 2.4 GHz band.

The activation delay between detection and countermeasure triggering was minimal, demonstrating effective integration between the detection and control subsystems. The system verified the feasibility of implementing a response mechanism in real-time drone surveillance.

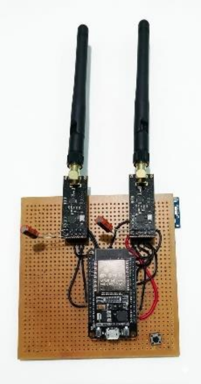
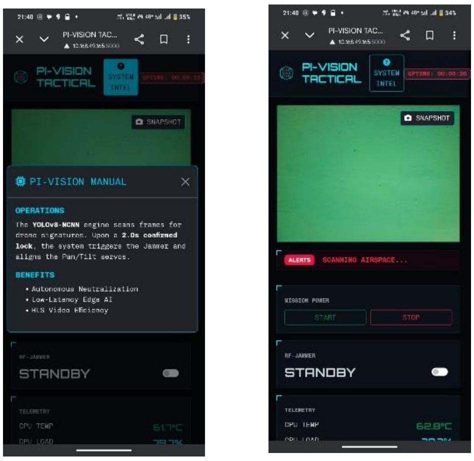


Fig. 4. Countermeasure System

E. Web Application Performance

The web-based monitoring system provided real-time video streaming and system telemetry. The video stream, implemented using FFmpeg and HLS, showed stable performance with minimal latency.

The system also logged detection events into a structured spreadsheet format, including timestamp, turret orientation angles, and system status. This log enabled performance analysis and validation over time.



PI-VISION MANUAL

OPERATIONS

The **YOLOv8-NCNN** engine scans frames for drone signatures. Upon a **2.0s confirmed lock**, the system triggers the Jammer and aligns the Pan/Tilt servos.

BENEFITS

- Autonomous Neutralization
- Low-Latency Edge AI
- HLS Video Efficiency

Fig. 5. (a) Webapp Intel (b) In Idle Time

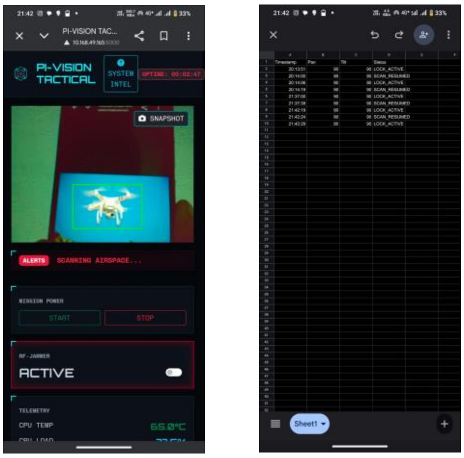


Fig. 6. (a) When Drone Detected (b) log Sheet

F. overall system performance

The overall system performance demonstrates that real-time drone detection and tracking can be effectively implemented on an embedded platform. The use of NCNN optimization and multithreading allowed efficient utilization of system resources.

However, certain limitations were observed:

- Reduced detection accuracy for very small or distant drones
- Slight tracking delay for high-speed targets
- Limited range of the countermeasure module

G. Discussion

The results confirm that the proposed system provides a complete and functional solution for drone detection, tracking, and monitoring. Compared to traditional systems that focus only on detection, this system integrates multiple functionalities into a single platform.

The combination of deep learning, embedded computing, and real-time control makes the system suitable for applications such as surveillance, restricted area monitoring, and research in drone defense technologies.

H. Summary

The experimental results demonstrate that the proposed system achieves reliable detection accuracy, effective tracking performance, and successful system integration. The system operates in real time and provides a scalable platform for future improvements.

DOI: 10.17148/IJARCCE.2026.153124

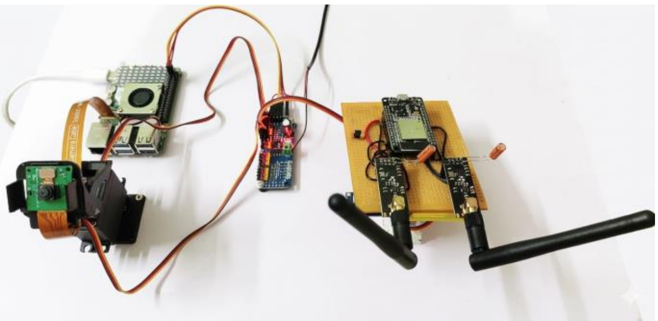


Fig. 7. Final Hardware Implementation of the Proposed System

REFERENCES

[1]. T. Hong, Q. Yang, and P. Wang, "Multitarget Real-Time Tracking Algorithm for UAV IoT," *Wireless Communications and Mobile Computing*, 2021.

[2]. Author unknown, "Building an ai-based surveillance drone cloud plat- form," *IEEE*, 2024.

[3]. E. Unlu et al., "Deep learning-based strategies for the detection and tracking of drones using several cameras," *Ipsj Transactions on Computer Vision and Applications*, 2019.

[4]. K. P. Phung, T. H. Lu, and T. T. Nguyen, "Multi-model Deep Learning Drone Detection and Tracking in Complex Background Conditions," *ATC*, 2021.

[5]. R. Ferreira et al., "Effective GPS Jamming Techniques for UAVs Using Low-Cost SDR Platforms," *GWS*, 2018.

[6]. R. Hakani and A. Rawat, "Edge Computing-Driven Real-Time Drone Detection Using YOLOv9 and NVIDIA Jetson Nano," *Drones*, 2024.

[7]. A. D. B. A. Rahman et al., "Unmanned Aerial Vehicle (UAV) GPS Jamming Test by using Software Defined Radio (SDR) platform," *Journal of Physics: Conference Series*, 2021.

[8]. Author unknown, "Cloud platform for AI-based surveillance with detection, classification, and alerts," *Drone Security Insights*, 2026.