Islamic University of Lebanon

Faculty of Engineering

Department of Computer and Communication

Embedded Systems Project
Drone Detection And Protection

Mohamed Samhat – Ali Sleiman – Sleiman Shaito – Mira Daher

ELBC803 Embedded Systems Design

Professor Oussama Mostafa

May 15, 2024

Table of Contents

1.Acknowledgement:

I would like to express my deepest gratitude to Professor Oussama Mostafa for his invaluable guidance, support, and encouragement throughout this project. His expertise and insights were instrumental in shaping the direction and execution of this work. Professor Mostafa's dedication to fostering a rigorous academic environment and his unwavering commitment to student success have been a constant source of inspiration. This project would not have been possible without his mentorship and the resources he provided. Thank you, Professor Mostafa, for your significant contribution to my academic and professional development.

2.Introduction:

This project aims to develop an embedded system that detects drones using the YOLOv8 model and responds by aiming a laser pointer at the detected drone. The system processes real-time video feeds to accurately detect drones, transmits detection data to an Arduino, and controls a servo motor to target the drone with a laser pointer. The project demonstrates the integration of AI with embedded systems, offering an accessible solution for drone detection and mitigation.

3.Literature Review:

3.1. Drone Detection Technologies:

Traditional drone detection methods, including radar, radio frequency (RF) analysis, and acoustic detection, each have their strengths and limitations. Radar can detect drones at long ranges but struggles with small, low-flying drones and can be expensive. RF analysis detects communication signals between drones and their controllers but is ineffective against autonomous drones. Acoustic detection identifies drone motor sounds but is less reliable in noisy environments.

3.2. Computer Vision and AI for Drone Detection

Advances in computer vision and artificial intelligence have led to the development of efficient and accurate real-time drone detection systems. Convolutional Neural Networks (CNNs), particularly the YOLO (You Only Look Once) models, are widely used for their speed and accuracy. YOLO models process images in a single pass, making them suitable for real-time applications. The latest version, YOLOv8, features improvements in architecture and training techniques, enhancing its detection performance. The model features a deeper network with more convolutional layers, allowing it to capture finer details and improve accuracy. Additionally, it uses advanced training strategies such as data augmentation and transfer learning to generalize better across diverse datasets. Recent studies have demonstrated the effectiveness of YOLOv8 in various object detection tasks, including drone detection, where it has shown superior performance in terms of both precision and recall.

3.3. Embedded Systems for Real-Time Applications

Embedded systems are critical for implementing real-time drone detection and response mechanisms due to their low power consumption and ability to interface with various sensors and actuators. Arduino microcontrollers, in particular, are widely used for their simplicity, versatility, and strong community support. Combining an Arduino with a servo motor provides an effective way to create responsive systems that can track and target moving objects. Prior research has demonstrated the feasibility of using Arduino-controlled servo motors for various applications, including automated targeting systems and robotic arms.

4.Problematic:

The rapid increase in drone usage for various applications has raised significant security and privacy concerns, with unauthorized drones posing threats to sensitive areas. Traditional detection methods like radar, RF analysis and acoustic detection are often costly and complex, particularly for small, low-flying drones. Advances in artificial intelligence and computer vision, such as the YOLO (You Only Look Once) models, provide real-time detection capabilities suitable for embedded systems. There is an urgent need for a cost-effective, efficient, and reliable system capable of detecting drones in real-time and responding promptly to mitigate potential threats. This project addresses these challenges by integrating advanced AI-driven detection with responsive

embedded systems, ensuring both accuracy and affordability while adhering to safety and ethical standards

5.Methodology:

In this project, the work will be divided into two parts:

1. Software
2. Hardware

For the software part, we will train an AI model to detect drones and then send data from this model to the hardware to aim the protection system at the center of the drone. We will use YOLOv8 for the detection and Python to communicate with the hardware.

For the hardware part, we will receive the data sent from the software and convert it into motion to aim at the center of the drone. We will use an Arduino for this purpose along with two servo motors (one for horizontal movement and one for vertical movement). Additionally, we will need a laser to target the drones accurately.

6.Implementation:

First, we prepared a virtual environment using Anaconda3 to train the model and run the project. We created a virtual environment and installed all necessary dependencies.

Next, we gathered drone images from the internet, ensuring a variety of drone types, angles, and backgrounds.

After collecting the images, we labeled them using a tool called "LabelImg," which we installed in the Anaconda environment. In the "LabelImg" tool, we opened each photo, created a class called "drone," drew a box around each drone, and assigned it to the "drone" class. We chose the YOLO format for labeling to ensure compatibility, preparing the dataset for training.

We then trained the YOLOv8 model using the labeled custom dataset. YOLOv8 has various versions, but we used the nano version for compatibility with our laptop. We started by installing Ultralytics into the virtual environment, which installs all necessary libraries for YOLO. Ultralytics installation includes Torch, a fully featured framework for building deep learning models, commonly used in applications like image recognition and language processing. The installation included the Torch CPU version, which is slow for training, so we installed the GPU version by installing the CUDA toolkit from NVIDIA, providing the tools needed for GPU-accelerated applications. We then ran the YOLO training command, specifying the task, number of epochs (one complete pass of the training dataset through the algorithm), the custom dataset, and the YOLO version (nano).

Next, we wrote a Python code to control the detection and send data to the Arduino. This code, available in the GitHub repository linked in the references, launches the model, puts a box around the detected objects, gets the box's dimensions and center, converts the center's coordinates into horizontal and vertical angles, and sends them to the Arduino.

We then wrote the Arduino code to control two servo motors and the laser beam, receiving serial data from the Python code to move the servos according to the received angles and activate the laser beam.

For the hardware, we used an Arduino Uno to control the servos and laser beam. We connected the VCC input of the servos and laser to the 5V pin on the Arduino, their grounds to the GND pin, provided the horizontal angle through pin 9, the vertical angle through pin 10, and the laser signal via pin 11.

7.Results And Discussion:

After running the Python script, the system initiates drone detection and begins transmitting data to the Arduino. The detection algorithm identifies drones within the camera's field of view, and the Python script calculates the necessary angles to aim the protection system accurately.

Initially, the angles calculated by the Python script are precise and effective for drones that are relatively close to the camera. In these scenarios, the laser beam successfully targets and hits the drone, providing accurate results for short-range detection.

However, as the drone moves farther away from the camera, the accuracy of the angles diminishes. The laser beam, which was initially hitting the drone, starts to miss the target. This reduction in accuracy is due to several factors, including the limitations of the detection algorithm in estimating the precise

location of distant objects and potential errors in the angle calculation process over longer distances.

To address this issue, further calibration and adjustments to the detection algorithm and angle calculations may be necessary. This could involve refining the algorithm to account for distance and perspective changes or implementing additional sensors to improve the system's accuracy over a broader range. Additionally, integrating advanced techniques such as machine learning models that adapt to varying distances could enhance the overall performance and reliability of the system, ensuring the laser beam consistently hits the drone regardless of its distance from the camera.

8.Conclusion And Perspective:

We can conclude that the AI model is functioning correctly and accurately sends the appropriate angles to the hardware components. However, we have identified a significant issue with long-distance detection. The current system's accuracy diminishes as the distance between the drone and the camera increases, leading to the laser beam missing the target at greater ranges.

For future work, our primary focus will be on improving the detection accuracy over longer distances. This will involve refining the detection algorithm, enhancing the angle calculation process, and potentially integrating additional sensors to maintain precision regardless of distance.

Once we achieve better long-distance detection results, we plan to transition this project into a fully embedded system using a Raspberry Pi. This will enhance the portability and integration of the system, making it more suitable for real-world applications. The Raspberry Pi will serve as the central processing unit, handling both the AI model and hardware control, thus simplifying the overall setup.

Additionally, we aim to incorporate a radar system into the project to enable 360-degree drone detection. The radar system will continuously scan the surroundings, identifying drones in any direction. Upon detecting a drone, the system will send commands to a 360-degree rotating motor that holds the camera and servos, aligning the detection system with the drone's location. This will provide comprehensive coverage and protection, ensuring that drones are detected and targeted from any angle.

For the protection mechanism, we propose adding an air gun equipped with web bullets. Once the system detects a drone, the air gun will fire a web bullet to immobilize the drone. This non-lethal approach will effectively neutralize the threat while minimizing damage. Integrating the air gun with the existing system will involve coordinating the detection, aiming, and firing processes to ensure precise and timely responses to detected drones.

By addressing the long-distance detection issue, transitioning to an embedded system, incorporating 360-degree detection, and

enhancing the protection mechanism, we can significantly improve the overall effectiveness and reliability of our drone detection and neutralization system.

9.References:

Drone Detection And Protection System Github Repositories