DroneEye

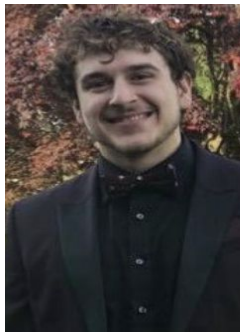
By: Ryan Cortes, Tony DeMark, Maxwell Morrison, and Charles Farrell

University of Delaware

The team:

Ryan Cortes - Machine Learning / Model

Maxwell Morrison - Machine Learning



Tony Demark - Hardware/Software

Charles Farrell - Hardware

Overall Project Idea

Drone Eye is a device designed to identify and track the locations of drones in a set area. Following the presence of unidentified drones in New Jersey in the winter of 2024, public concern increased sharply. Using this equipment, users will be able to identify and track drones in a short range in order to prevent safety and security issues in the future.

Accomplished this Fall 2025

- Remote portable device
- Devices that communicate with each other
- Detection of a drone
- Shows the X and Y of the drone

Accomplished for Spring 2026

- Automatic tracking
- User Friendly GUI
- Upgraded to 1080p 60 FPS camera

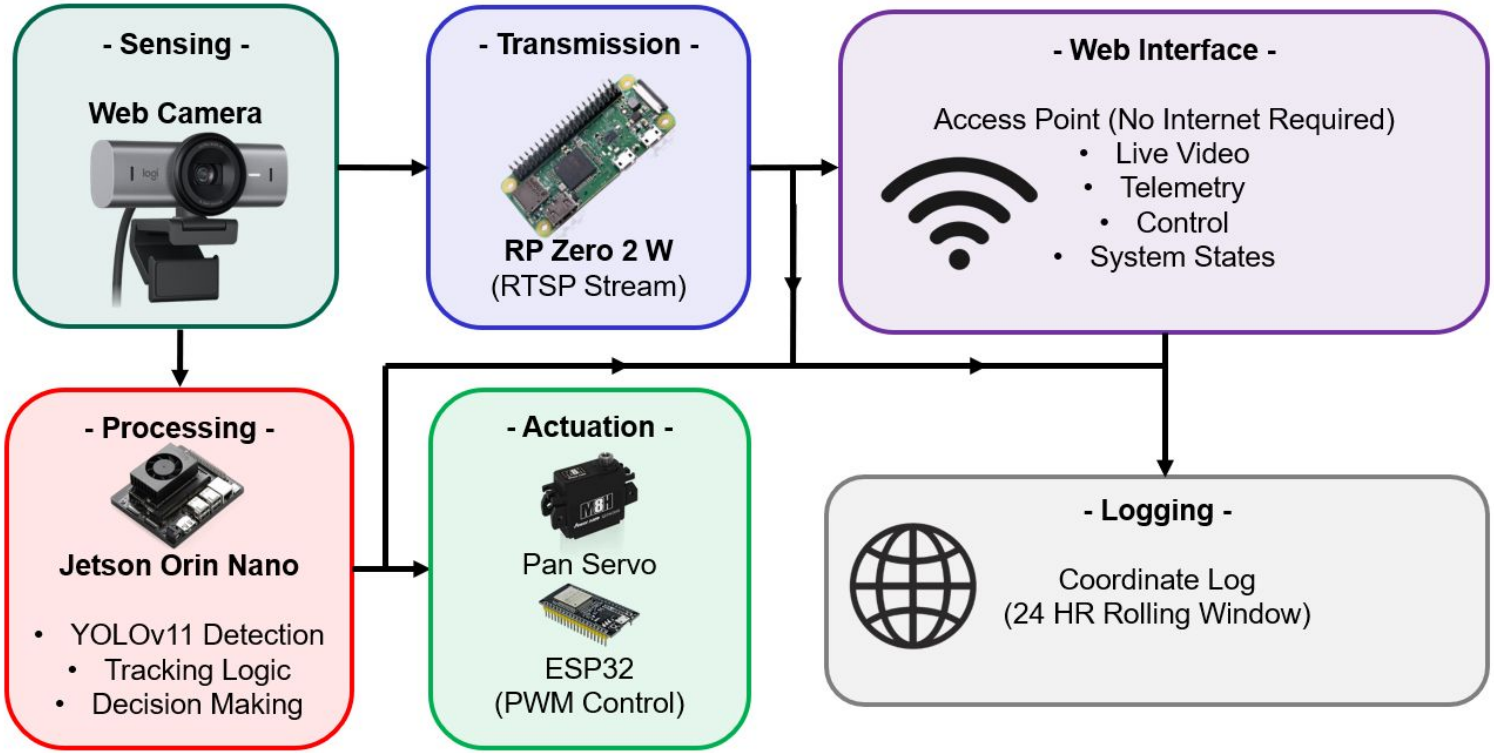
Future Improvement

- Implement Lidar ranging (budgetary issues)
- Show the z-position of drone
- Drone Differentiation

Market

- Military
- Police
- Security
- Personal Use

DroneEye Network



Hardware:

- Jetson Orin Nano
- ESP32
- Raspberry PI Zero 2 W
- MX brio (Web Camera)
- Slip ring
- Feedback 360 parallax servo
- Lithium 12 volt battery
- LiFePo4 6 volt battery
- ADC

Hardware challenges:

- Supply voltages
- Cost limitations
- data and data transfer limitations
- Heating issues
- Precision

Software:

- **Jetson:**
  - Python
  - JavaScript
  - CSS
  - HTML
  - Flask (lightweight, "micro" web framework)
- **ESP32:**
  - C++
- **Raspberry Pi Zero 2 W:**
  - RTSP

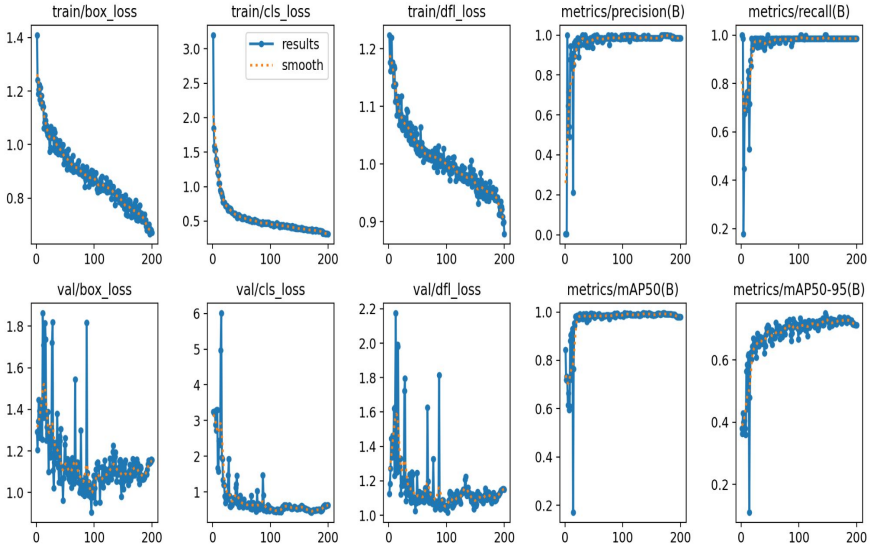
Software Challenges:

- Getting all the languages connected
- Complex code (Seperated code)
- AI challenges that affect the software (frames not detected)
- Hardware challenges that affect the software (Noise and precision)

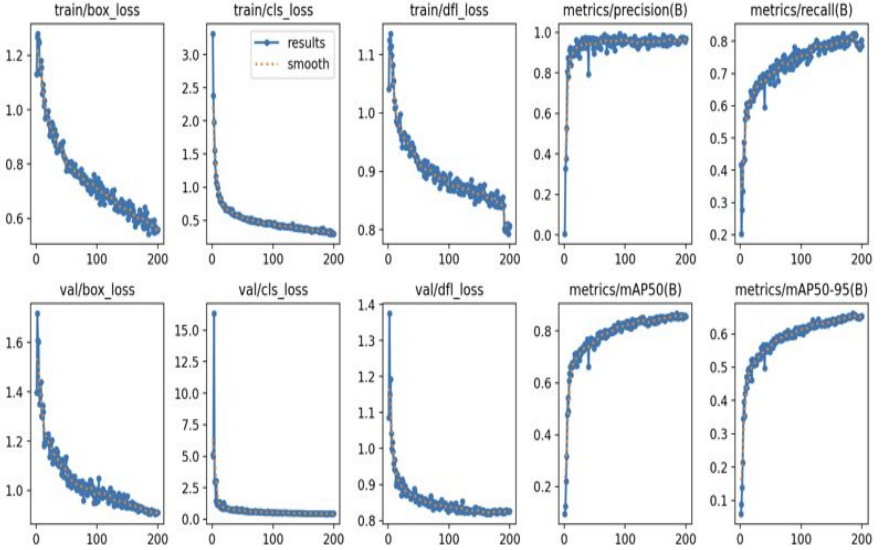
ML Logic Improved

- Ultralytics Library using Yolo v11 for Training and Fine Tuning
  - Base Training for Drone Detection and Weights
- Increased Dataset Content and Differentiation
  - Tackled False Positives with Increased Picture Background Variability
  - Drastically Increased Confidence of Model and Tracking

Old Model Graphs vs New

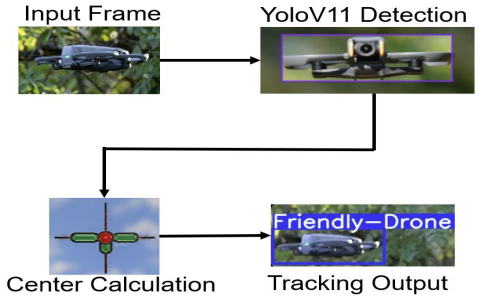


Old Model Graphs: More sporadic results and tons of loss and false positives



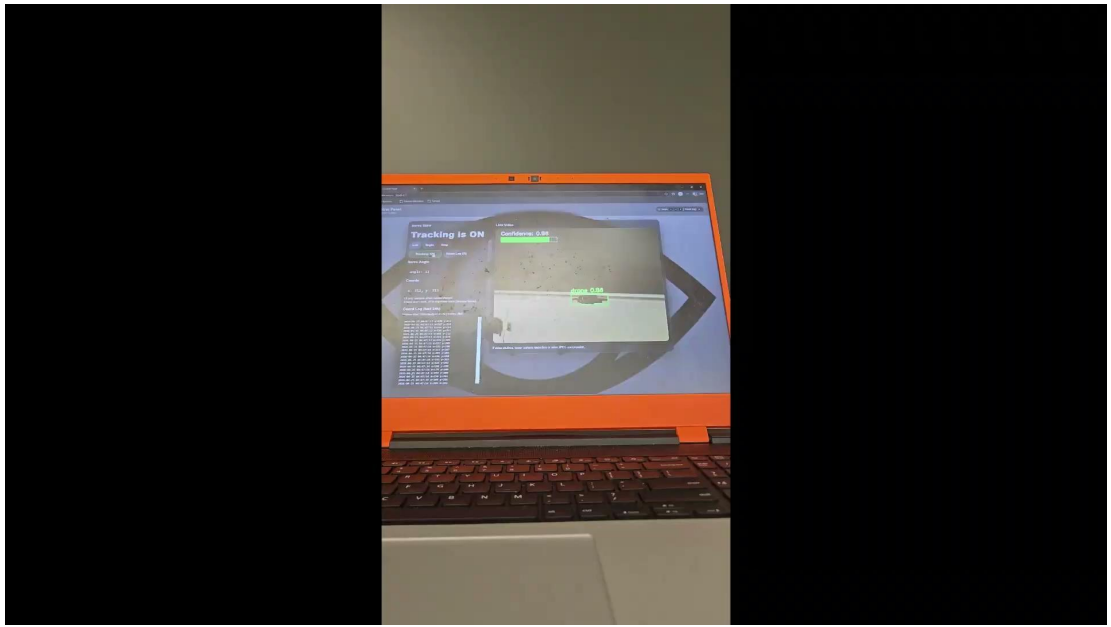
New Model Graphs: Much smoother results, loss is minimized and much less false positives.

ML Challenges



- False Positives (Minimal)
  - The model is extremely sharp at detecting the drone and tracking it properly. Sometimes it struggles same color backgrounds or 30+ meter tracking.
- Automated Rotational Tracking
  - During automated rotation, sometimes the drone can get lost during rotation.
  - Possible fix is to add more “half photos” of the drone into the model

Drone Detection with Auto Tracking (Inside)



Drone Detection Tracking Device View