Table of Contents

Introduction

- Developing a real-time drone detection system using YOLOv8.
- **Objective**: Enhance security by identifying drones in critical areas.
- **Applications**: Airports, military zones, public events, and wildlife monitoring.
- **Why Visual Detection?**:
  - Low-cost and high-accuracy solution.
  - Effective for small drones in cluttered environments.
- **Motivation**:
  - Address limitations of radar, acoustic, and RF-based methods.
  - Ensure safety against UAV threats in real-world conditions.

Literature Review: Non-ML Methods

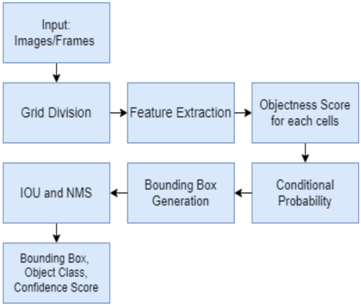
- **Radar-Based Methods:**
  - Ding et al. (2022): Used Doppler shift for tracking drones in cluttered environments. Achieved 85% accuracy but suffered from high false positives in urban areas.[4]
- **RF-Based Methods:**
  - Gupta et al. (2023): Used demodulation and pattern recognition. Achieved 80% accuracy with low energy consumption but ineffective for autonomous drones.[6]
- **Acoustic-Based Methods:**
  - Brungart et al. (2023): Used frequency-domain analysis for detection. Achieved 70% accuracy but was highly sensitive to background noise.[2]
- **Hybrid Radar-Optical System:**
  - Wang et al. (2022): Combined radar and optical features. Achieved 88% accuracy but faced challenges with overlapping targets.[16]
- **LIDAR-Based Methods:**
  - Ahmed et al. (2023): Used 3D mapping for small drones. Achieved 85% accuracy but limited for high-speed drones.[1]

Literature Review: ML and DL Methods

- **Traditional ML Approaches:**
  - **SVM (Smith et al., 2022):** Used HOG features for drone detection. Achieved 85% accuracy with 60ms latency but unsuitable for real-time applications.[12]
  - **Random Forest (Lee et al., 2022):** Achieved 80% accuracy using pixel-based features but had high computational complexity, limiting real-time usage.[8]
  - **k-NN (Taylor et al., 2023):** Achieved 75% accuracy with 90ms latency but was computationally expensive and ineffective for dynamic environments.[13]
- **Deep Learning Approaches:**
  - **YOLOv3 (Ullah et al., 2023):** Real-time detection with 93% accuracy and 25ms latency. Reliable in cluttered environments.[14]
  - **YOLOv8 (Martinez et al., 2024):** Achieved 95% accuracy with low latency (20ms). Optimized for complex, real-time scenarios.[9]
  - **GANs (Miller et al., 2023):** Enhanced training data with synthetic images, achieving 92% accuracy. Limited by high computational costs.[10]
  - **Hybrid Networks (Jones et al., 2024):** Combined CNN and RNN for time-series data, achieving 94% accuracy. Effective for fast-moving scenarios but required optimization for real-time applications.[7]

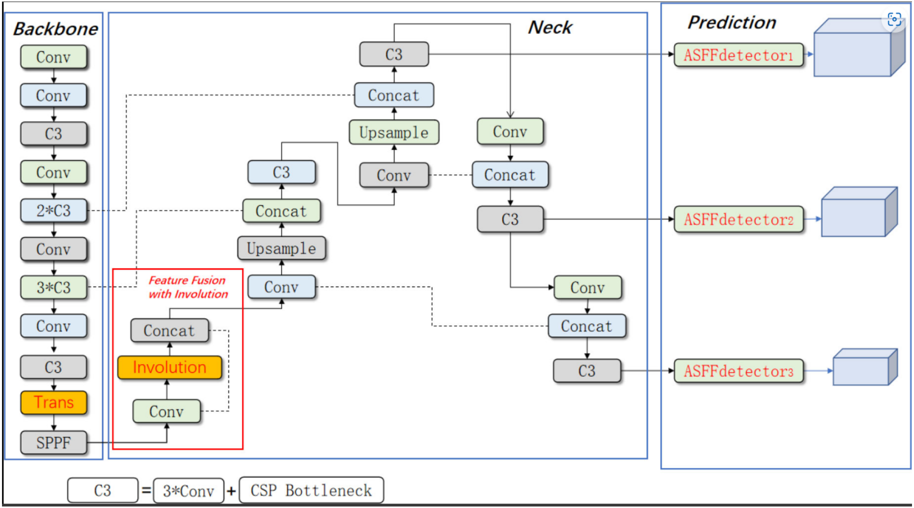
Methods

- **Input**: Images/frames.
- **Grid Division**: Divides input into cells.
- **Feature Extraction**: Uses pretrained CNN network.
- **Objectness Score**: Calculated for each cell.
- **Bounding Box**: Center, height, width of the object.
- **Output**: Bounding boxes with object class and confidence score.
- **Techniques**: IOU (Intersection of Union), Non-Maximum Suppression.



YOLO Object Detection Algorithm

YOLOv8s Architecture



YOLOv8s Architecture

- **Anchor-Free Model**: Predicts bounding boxes based on object centers.
- Divided into:
  - **Backbone**: Feature extraction (ConvNet, C2f, SPPF modules).
  - **Neck**: Combines upper and lower feature layers (PAN-FPN).
  - **Head**: Predicts object location and classes with decoupled branches.
- Loss Functions:
  - **Classification**: Binary cross-entropy.
  - **Detection**: DFL, CIoU.

YOLOv8s Architecture

Backbone

- Feature extraction by series of ConvNet, C2f, and SPPF modules

ConvNet Module:

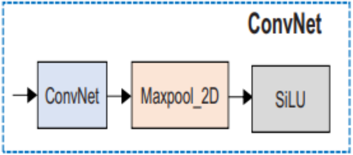
- Convolution operation
- Maxpooling layer
- Activation using SiLU

Spacial Pyramid Pooling Feature (SPPF):

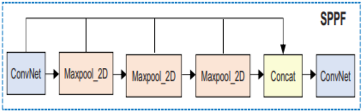
- Pools input feature map to fixed map for adaptive-sized output
- Maxpooling layers connected in series reduce computational complexity and latency

Augmentation Techniques Used:

- Mosaic
- Mixup
- Cutmix



ConvNet Module Block

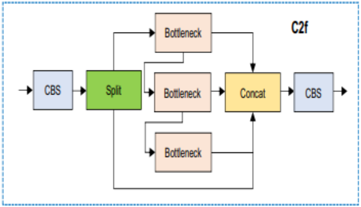


SPPF Module Block

YOLOv8s Architecture

Neck

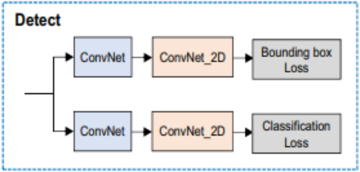
- Modified version of Path Aggregation Network-Feature Pyramid Network (PAN-FPN) used to combine features.
- Upper layer captures more info, lower layer retains location details.



C2f Module Example

Head (Detect)

- Predicts object's location and classes.
- Decoupled head, with separate classification and localization branches.
- **Loss Functions:**
  - Classification: Binary cross-entropy.
  - Detection: DFL, CIoU.



Detection Head Example

Database Description

Drone Detection Dataset

- **Dataset Size**: 1400 images.
- **Split**: 1012 training, 348 validation images.
- **Capture Conditions**:
  - Illumination: Daylight, twilight, night.
  - Weather: Sunny, cloudy, rainy.
  - Altitudes: Low, medium, high.
- **Source**: Kaggle dataset by Muki (2023)[11].
- **Strength**: Ensures diversity in scenarios for generalization and evaluation.

Drone Detection Dataset Example

Training Images	Validation Images
1012	348

Performance Metrics

- **Accuracy**: Proportion of correctly classified instances.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- **Normalized Confusion Matrix**: Proportions of TP, TN, FP, FN for better visualization.

$$C_{ij} = \frac{N_{ij}}{\sum_k N_{ik}} \quad \text{where} \quad C_{ij} \text{ is the proportion of class } i \text{ classified as } j, \quad N$$

- **F1 Score**: Harmonic mean of precision and recall.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

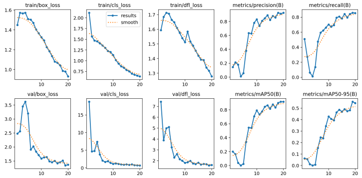
- **Loss Curve**: Tracks training and validation loss across epochs.

Training Results

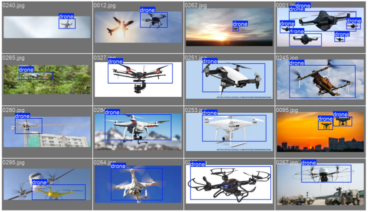
Results

YOLOv8s Model was trained with Drone Detection Dataset with 20 epochs each with 32 batch size.
Model achieved performance of:

- **Precision**: 92%
- **Recall rate**: 85%
- **F-1 Score**: 88%
- **mAP50**: 91%
- **Inference Time**: 333ms
- **Model Size**: 21.4 MB



Loss Curve



Sample Result

Result Comparison

Model	Precision (%)	Recall (%)	mAP@0.5 (%)	CT (ms)	Size (MB)
SVM (Smith et al., 2022)[12]	85	-	-	60	50
Random Forest (Lee et al., 2022)[8]	80	-	-	75	70
k-NN (Taylor et al., 2023)[13]	75	-	-	90	-
YOLOv3 (Ullah et al., 2023)[14]	93	-	93	25	236
YOLOv8 (Martinez et al., 2024)[9]	95	-	95	20	126
GANs (Miller et al., 2023)[10]	92	-	92	-	-
Hybrid Networks (Jones et al., 2024)[7]	94	-	94	40	-
YOLOv8s, Our Method[5]	92	85	91	-	21.5

Table: Performance Comparison of Various Drone Detection Models

Testing

Datasets Used:

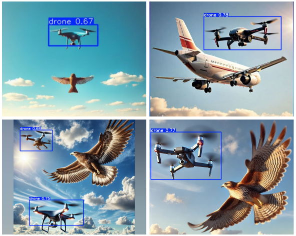
- Birds vs Drone Dataset (Kaggle) [15] and Mendeley Drone Dataset [3].

Performance:

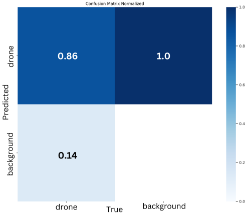
- **Category:** Drone (with different backgrounds)
- **Number of Images:** 400
- **Correct Detection:** 89%
- **Misclassified:** 11%

Confusion Matrix Values:

- **True Positive Rate (0.86):** Proportion of correctly classified drone images.
- **True Negative Rate (1.0):** Proportion of correctly classified non-drone images.
- **False Positive Rate (0.14):** Proportion of non-drone images misclassified as drones.
- **False Negative Rate (0.0):** Proportion of drone images misclassified as non-drones.



Drone Detection with Different Objects

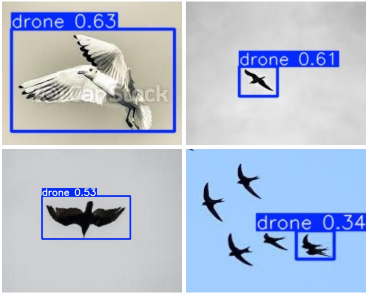


Confusion Matrix

Testing

Detecting Bird as a Drone

- **Problem:** Birds misclassified as drones due to similar shapes and motion patterns.
- **Observation:** Misclassification occurs under specific lighting or camera conditions.
- **Proposed Solutions:**
  - Enhance dataset with more bird images for better distinction.
  - Incorporate motion/flight pattern analysis: Drones (steady) vs. Birds (erratic).
  - Add auxiliary classifier to post-process detections.



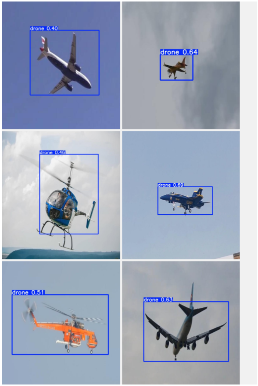
Misclassification of Bird as Drone

Category	Total Images	Correct Detections	Misclassified
Birds	100	95	5

Testing

Detecting Aircraft as a Drone

- **Problem:** Large flying objects like aircraft are detected as drones, particularly at a distance.
- **Observation:** Misclassifications occur due to similar structural features like wings or tails, especially when viewed from a distance.
- **Proposed Solutions:**
  - Use transfer learning with aircraft image datasets to improve classification.
  - Incorporate altitude/size estimation to filter out objects larger than typical drones.
  - Increase confidence threshold for large-scale objects to avoid misclassifying aircraft.



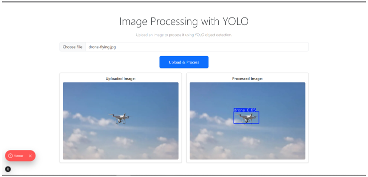
Misclassification of Aircraft as Drone

Category	Total Images	Correct Detections	Misclassified
Aircraft	100	90	10

Graphical User Interface

GUI

- **GUI Purpose:** Assess real-time drone detection capability.
- **Built with:** Python (Flask) for the interface, Next.Js for frontend.
- **Features:**
  - Image Upload: Upload image for analysis.
  - Real-Time Processing: Process image with YOLOv8 model.
  - Visual Results: Displays detected objects with bounding boxes.
  - Accuracy Feedback: Shows confidence of detections.
- **Real-World Use:** Demonstrates interactive real-time system capabilities.



Graphical User Interface in operation

Post-Retraining Performance

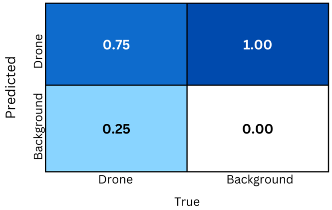
Result

Hyperparameters:

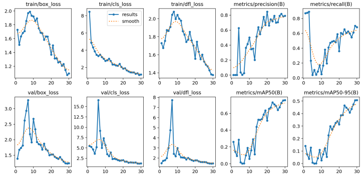
- Optimizer: Adam
- Learning Rate: 0.001
- Batch Size: 32
- Epochs: 30

Performance:

- Precision: 80%
- Recall: 76%
- F1 Score: 78%
- False Positives reduced on birds/planes



Updated Confusion Matrix



Updated Loss Curve

Visual Results - Post Retraining

Figure: Drone detection with improved accuracy after retraining

Robustness Testing: Noise and Filtering

Testing with Noise and Filtering

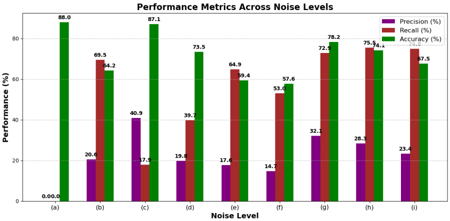
Noise Types:

- **Gaussian Noise**: Reduced accuracy to 79.7%
- **Salt & Pepper Noise**: Most disruptive, accuracy dropped to 63.5%
- **Speckle Noise**: Accuracy at 73.3%

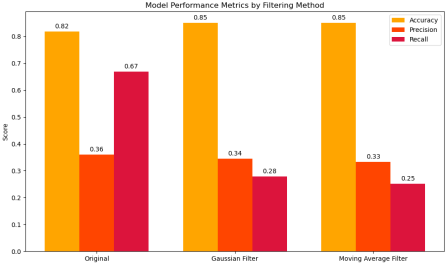
Filtering Techniques:

- **Gaussian Filter**: Smoothed pixel intensities, accuracy improved to 85%
- **Moving Average Filter**: Reduced high-frequency noise, accuracy improved to 85%

Conclusion: Filtering significantly improved detection on noisy inputs.



Noise Impact on Accuracy



Effect of Filtering Techniques

Conclusion

Project Overview

This project developed a robust drone detection system using the YOLOv8 architecture, achieving reliable drone identification in complex aerial environments. The system was evaluated under different noise conditions (Gaussian, salt-and-pepper, and speckle noise) to ensure robustness.

Performance Summary

- F1-score achieved: **0.78**
- Accuracy in clean conditions: **94%**
- Performance under noise improved using:
  - **Gaussian filtering**
  - **Moving average filtering**
- System shows strong potential in applications like:
  - Aerial surveillance
  - Drone regulation enforcement

Future Work

Enhanced Data Augmentation

To improve robustness and generalizability, advanced augmentation strategies will be applied:

1. **Median filtering** on original images to reduce high-frequency noise.
2. **Add Gaussian noise** followed by median filtering to simulate realistic conditions.
3. **Add Salt-and-Pepper noise** followed by median filtering for impulse noise handling.

These transformations will triple the dataset size and support improved retraining.

Architectural Improvements

- Replace dropout layers with **comprehensive sensing mechanisms**.
- Optimize YOLOv8 internal blocks for better feature extraction.
- Enhance real-time performance on **embedded platforms**.

References I

M. Ahmed and Others.
Lidar-based drone detection system.
Elsevier Journal of Robotics, 2023.

J. Brungart and Others.
Acoustic-based detection of drones.
Springer Acoustics Journal, 2023.

Mendeley Data.
Drone dataset for detection.
https://data.mendeley.com/public-files/datasets/cd5z895tr2/files/fb36418a-ca4d-4628-8e42-c6e5b2575ac2/file_downloaded, 2024.
Accessed: 2024-10-06.

References II

A. Ding and Others.
Radar-based tracking of drones.
IEEE Transactions on Signal Processing, 2022.

Doguilmak.
Drone detection yolov8x.
https://github.com/doguilmak/Drone-Detection-YOLOv8x, 2024.
Accessed: 2024-10-06.

R. Gupta and Others.
Rf signal analysis for drone detection.
Elsevier Journal of Communication Systems, 2023.

P. Jones and Others.
Hybrid networks for drone detection.
Elsevier Deep Learning, 2024.

References III

- K. Lee and Others.
  Random forest for drone detection.
  Springer Machine Learning Journal, 2022.
- L. Martinez and Others.
  Yolov8 for enhanced drone detection.
  IEEE Transactions on AI, 2024.
- S. Miller and Others.
  Gan-based drone detection enhancement.
  Springer Journal of AI, 2023.
- Muki.
  Yolo drone detection dataset.
  `https://www.kaggle.com/datasets/muki2003/yolo-drone-detection-dataset`, 2023.
  Accessed: 2024-10-06.

References IV

- J. Smith and Others.
  Svm for drone classification.
  IEEE Transactions on Machine Learning, 2022.
- R. Taylor and Others.
  k-nn-based drone classification.
  Elsevier Journal of AI, 2023.
- F. Ullah and Others.
  Yolov3 for real-time drone detection.
  Springer Computer Vision, 2023.
- Harsh Walia.
  Birds vs drone dataset.
  `https://www.kaggle.com/datasets/harshwalia/birds-vs-drone-dataset`, 2023.
  Accessed: 2024-10-06.

References V

X. Wang and Others.
Hybrid radar-optical system for drone detection.
IEEE Access, 2022.