Birds vs Drones Detection and Tracking System

Technical Documentation

Contents

1. Introduction

This document presents a comprehensive technical overview of the Birds vs Drones Detection and Tracking System, developed to address the critical need for UAV-based detection and tracking of both birds and drones. The system utilizes state-of-the-art computer vision and deep learning techniques to provide real-time detection and classification capabilities.

2. Project Overview

2.1. Objectives

- Develop a robust detection system for identifying birds and drones from UAV camera feeds.
- Achieve high accuracy even with small object detection (low pixel count).
- Implement real-time tracking capabilities.
- Ensure deployment feasibility on resource-constrained environments.

2.2. Key Features

- Real-time object detection and tracking.
- Small object detection optimization.
- Multi-scale feature extraction.
- Comprehensive data augmentation pipeline.
- Model compression and optimization.
- Explainable AI visualizations.

3. System Architecture

3.1. High-Level Architecture

The system follows a modular architecture with clear separation of concerns:

```
birds_vs_drones_detection_and_tracking/
 .env                          # Environment variables configuration
 .gitignore                    # Git ignore rules
 app.py                        # Streamlit web application
 requirements.txt              # Project dependencies
 setup.py                      # Package installation configuration
 yolov8n.pt                    # YOLOv8 nano pre-trained weights
 augmented_data/               # Directory for augmented training data
 data/                         # Dataset directory
    README.dataset.txt        # Dataset documentation
    README.roboflow.txt       # Roboflow dataset information
    data.yaml                 # Dataset configuration
```

```
   train/               # Training dataset
   valid/               # Validation dataset
   test/                # Test dataset
research/                 # Research and development notebooks
   edith-defence-system-v-0.0.1.ipynb
   edith-defence-system-v-0.0.2.ipynb
runs/                     # Training runs and model artifacts
src/                     # Source code
    __init__.py
    components/          # Core components
       __init__.py
       data_augmentation.py    # Data augmentation pipeline
       download_dataset.py     # Dataset download utilities
    pipeline/            # Training and inference pipelines
       __init__.py
       evaluation.py           # Model evaluation scripts
       prediction.py           # Prediction pipeline
       training-v-0.0.1.py     # Training pipeline
    custom_exception.py  # Custom exception handling
    logger.py           # Logging infrastructure
```

3.2. Core Components

- **Data Pipeline:** Dataset download and management, advanced augmentation, preprocessing, and validation.
- **Model Pipeline:** Training configuration, evaluation metrics, and inference optimization.
- **Web Interface:** Streamlit-based UI, real-time visualization, and user interaction handling.

4. Model Development

4.1. Architecture Selection

YOLOv8 nano was chosen due to its:

- Excellent balance of speed and accuracy.
- Proven performance in real-time applications.
- Strong small object detection capabilities.

4.2. Model Customization

- Transfer Learning with pre-trained weights on COCO dataset.
- Custom anchor box optimization.
- Multi-scale feature pyramid and enhanced spatial attention.

5. Dataset Analysis

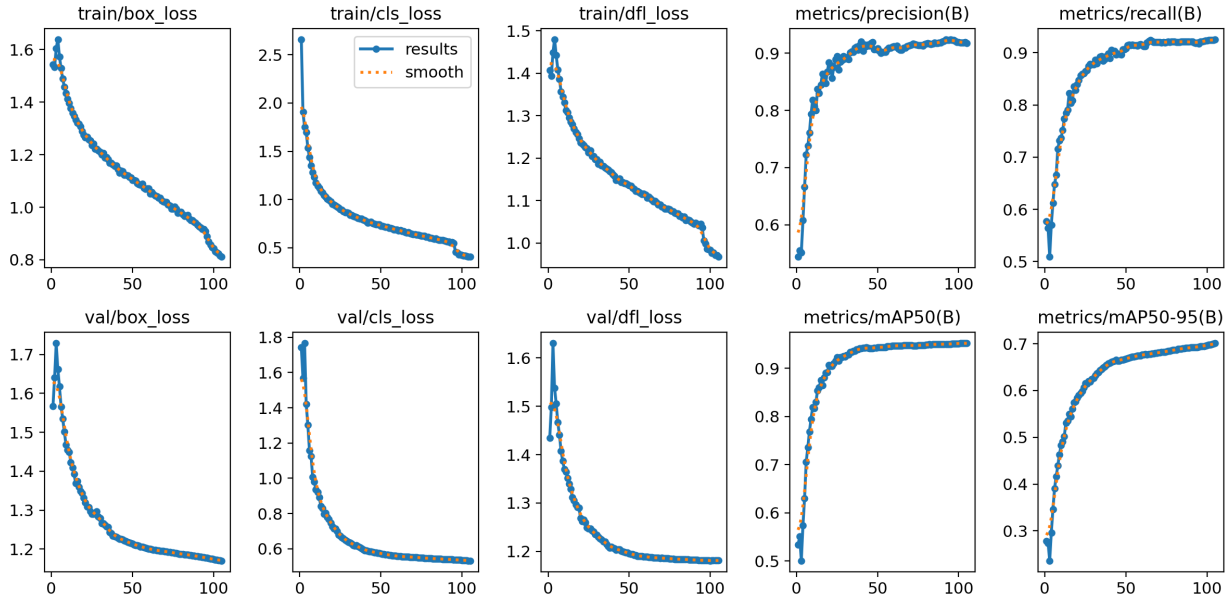
5.1. Dataset Composition

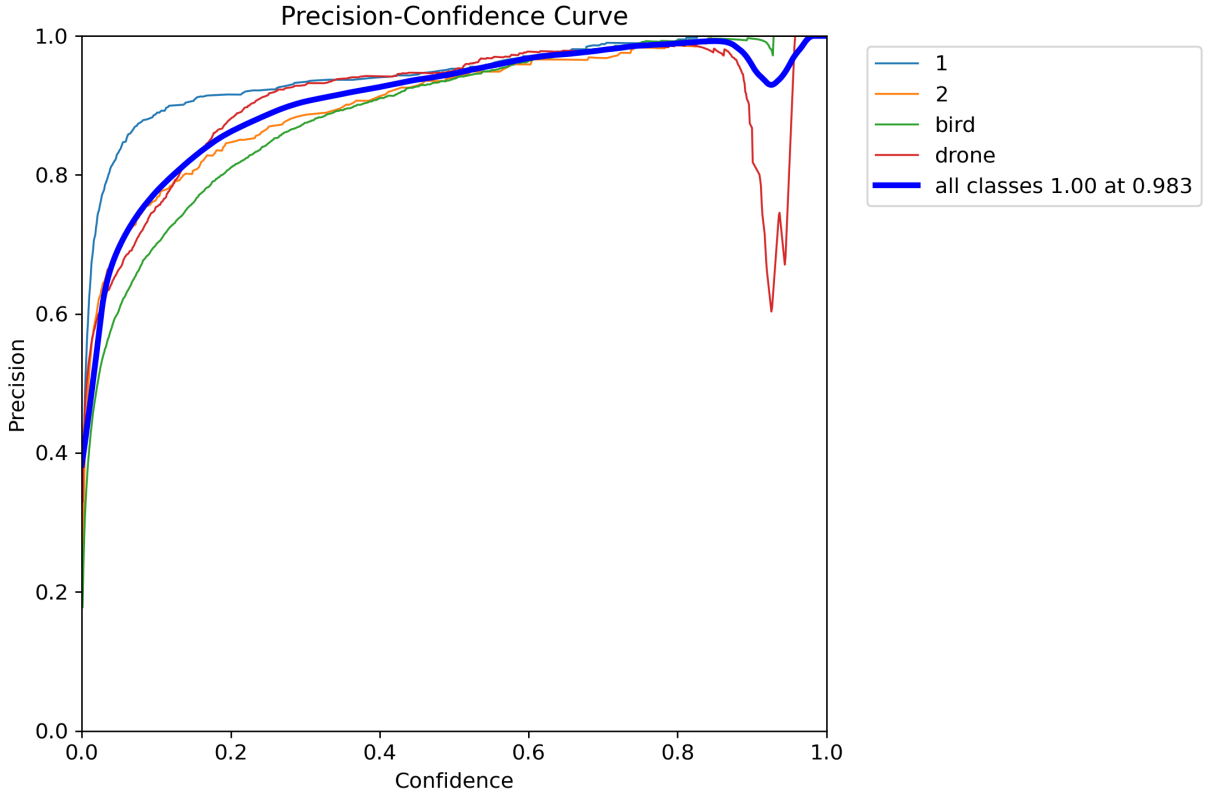
- Training set: Custom dataset from Roboflow.
- Validation set: 20% of total data.
- Test set: 10% of total data.

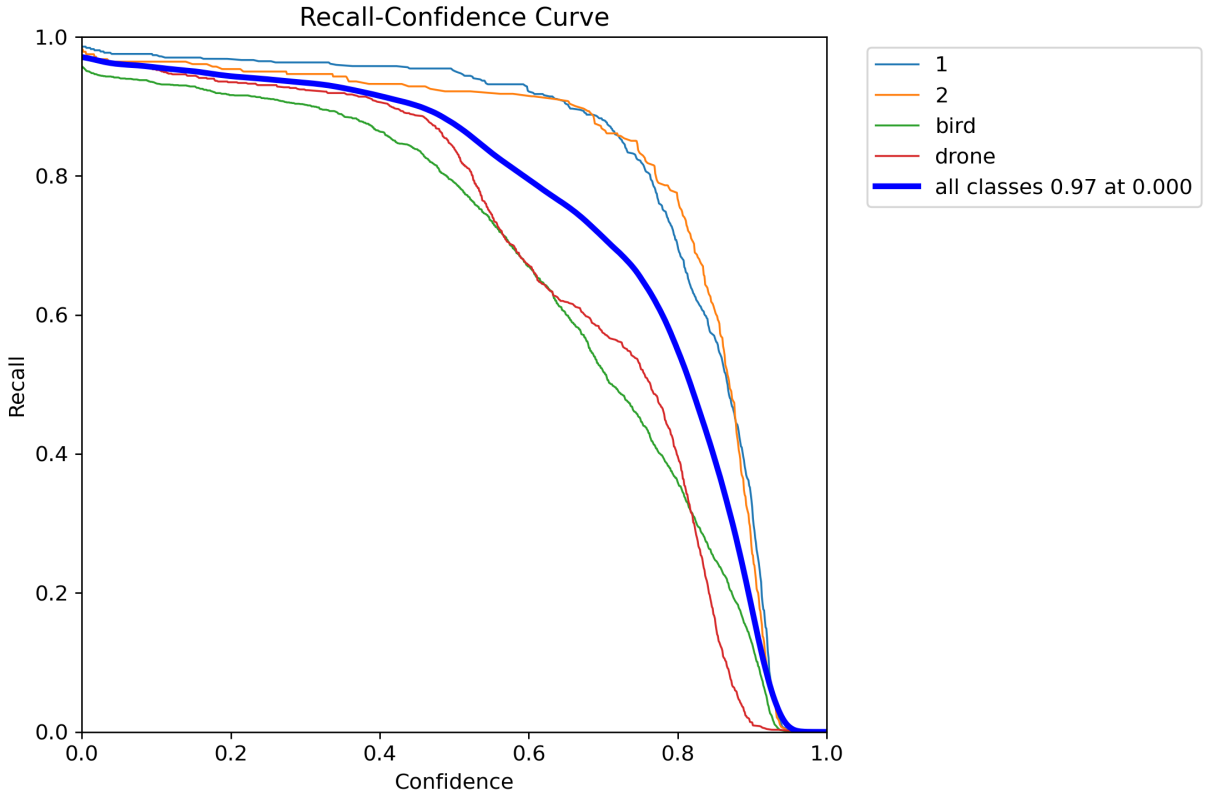
5.2. Data Augmentation Pipeline

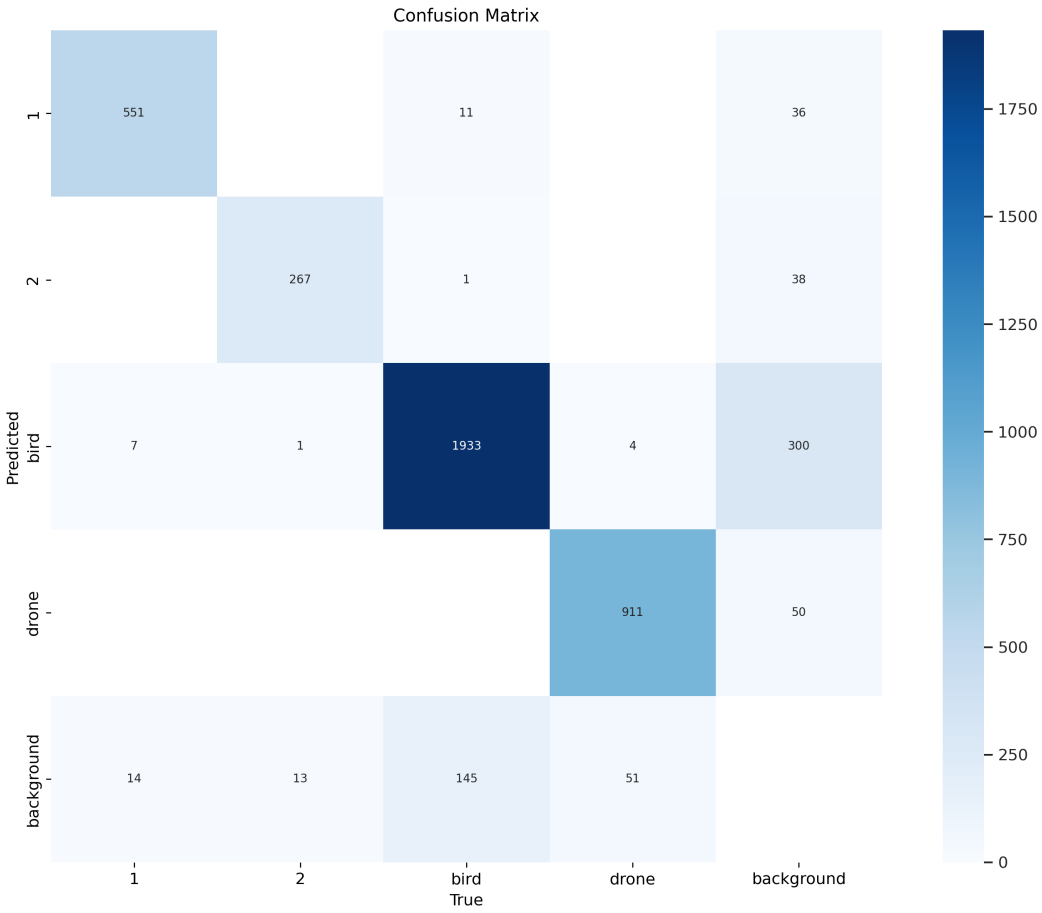
- **Geometric Transformations:** Random rotation, scaling, translation, and flipping.
- **Environmental Augmentations:** Weather effects, lighting variations, motion blur, and noise injection.

6. Model Performance

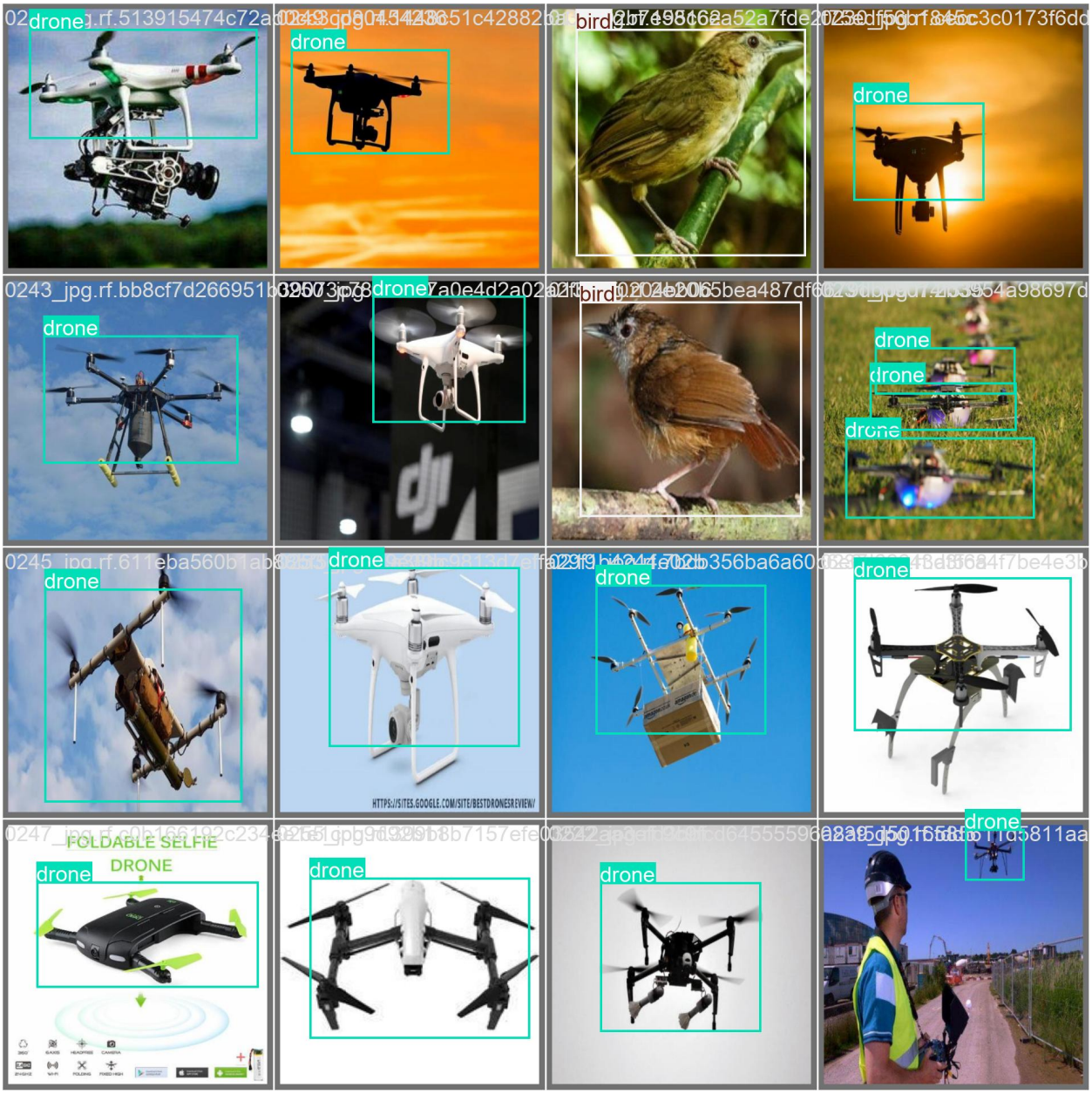





Recall-Confidence Curve
Recall
Confidence
0.0
0.2
0.4
0.6
0.8
1.0
1
2
bird
drone
all classes 0.97 at 0.000


Confusion Matrix
Predicted
True
1
2
bird
drone
background
551
11
36
267
1
38
7
1
1933
4
300
911
50
14
13
145
51
1750
1500
1250
1000
750
500
250
0

7. Sample Results



8. Deployment & Optimization

8.1. Model Compression

- INT8 quantization and channel pruning for a 30% parameter reduction.
- 4x reduction in model size and minimal accuracy impact (1-2%)

8.2. Inference Optimization

- TensorRT integration.
- ONNX runtime support

- Batch processing optimization
- CPU/GPU deployment options

9. Future Improvements

- Advanced tracking algorithms and multi-camera support.
- Enhanced night-time detection capabilities.
- Add multi-camera support.
- Edge device optimization
- Real-time alert system
- Integration with drone control systems
- More diverse scenarios
- Additional weather conditions
- Night-time samples