VIT-AP UNIVERSITY

Dept. of Mathematics, School of Advanced Sciences

Master of Science in Data Science

Conference Paper Presentation

by

Kunal Kaustav Nath

(23MSD7027)

Under the guidance of

Dr Soumen Kundu

Submission Date:

CONTENTS:

CONFERENCE PRESENTATION CERTIFICATE:

CERTIFICATE OF ATTENDANCE

This certificate is awarded to

Kunal Kaustav Nath, VIT-AP University, Andhra Pradesh, India

In recognition of their active participation/ presentation and outstanding presentation titled

Bird Vs Drone Detection using CNNs

at the **International Conference on Applied Mathematics and Data Science (ICAMDS24)**, held in Hybrid mode from October 28 to 29, 2024. The conference was jointly organized by the Department of Mathematics, Little Flower College, Kerala, India, and Advanced Mathematical Innovation Center for Artificial Intelligence and Scientific Computing, AMICAISC Center of Excellence Private Limited, Kerala, India.

Directors of AMICASIC
(Convener)

Sr.BINCY
Little flower college
(Chief Patron)

PUBLICATION STATUS:

a) Full Paper Publication
b) Extended Abstract
c) Short Abstract
d) None ✓

Bird Vs Drone detection using CNNs

Kunal Kaustav Nath

Dept. of Mathematics, School of Advanced Sciences, VIT - AP University, Andhra Pradesh - 522241, India.

Email: kunalnathkaustav01@gmail.com

Abstract:

The aim of this work is to address the problem of detecting the presence of one or more drones in scenes where birds may suddenly appear, considering key factors like background and foreground motion. Here the goal is to train and compare several newer versions of YOLO, on some benchmark dataset, which can effectively detect drones even in the presence of birds and finally compare which model is performing the best. The proposed solutions should accurately detect and localize the target objects in the scene only when they are present, avoiding confusion caused by other elements and the dynamic nature of the environment.

Keywords*: Deep Learning, Convolutional Neural Networks, Image localization and detection, YOLO, Airborne Object Detection.*

---

1. Introduction:

The escalation of drones and their increased accessibility have significantly raised concerns about their potential misuse, especially as drone technology becomes more affordable and easier to operate. Once limited to military and professional applications, drones are now widely available to the public, fueling their adoption in various sectors like delivery, photography, and agriculture. However, this standardization of drone technology also comes with the risk of exploitation for harmful purposes. Drones, equipped with high-resolution cameras and advanced surveillance tools, can easily invade personal spaces, capturing images and videos without consent and enabling unlawful monitoring. Also, drones can be used by terrorists or smugglers to strike public locations or to smuggle drugs. Additionally, drones have been leveraged in spying and unauthorized data collection, posing severe threats to privacy and national security. The increasing presence of drones in restricted or sensitive airspaces adds another layer of complexity, requiring stronger regulatory measures and advanced detection systems to mitigate these risks. The challenge is compounded by the difficulty of monitoring the rapidly growing number of drones, with law enforcement agencies often finding it hard to differentiate between legitimate and illicit activities. For example, thirteen tiny drones attacked Russian soldiers in Syria, causing substantial damage. Considering the possibility of drones being used as lethal weapons, authorities shut down the London Gatwick airport for 18 hours due to serious drone intrusion, causing 760 flights with over 120,000 people to be delayed.

Unlike conventional approaches that often rely on pre-defined features and rule-based systems such as radar, acoustic sensors, or visual tracking. Deep learning methods have proven to be significantly more effective than traditional techniques in drone detection due to their ability to analyze complex patterns and process large amounts of data with high accuracy. This capability enables them to handle diverse environments, distinguish drones from other airborne objects, and adapt to variations in size, speed, and other characteristics. Moreover, deep learning algorithms, such as convolutional neural networks (CNNs) can leverage temporal and spatial data to improve detection reliability, even in challenging conditions like low visibility or cluttered backgrounds. These models excel at analyzing images, videos, and even sound signatures with superior precision, minimizing false positives and enhancing real-time detection performance. As a result, deep learning-powered systems are better choice and highly suitable for modern drone surveillance applications, where quick and accurate identification is crucial for effective response and threat mitigation.

Object detection is a computer vision task that involves identifying and localizing objects within an image or video. It combines classification (to recognize the object) and localization (to specify its position using bounding boxes). YOLO (You Only Look Once) is a state-of-art algorithm used for tasks like object detection, image segmentation. It is quite faster than most of the detection algorithms and it is so fast that it can run in real time on a video. It has revolutionized object detection by offering real-time performance with high accuracy. Unlike traditional models that run the image through multiple classifiers at different regions, YOLO processes the entire image in a single pass using a single neural network, making it extremely fast. It divides the image into a grid and predicts bounding boxes and class probabilities simultaneously, which enhances efficiency. YOLO's end-to-end approach allows for quicker detection without compromising precision, making it suitable for applications requiring rapid and reliable object recognition.

This paper mainly focuses on comparing the newer YOLO models, which includes the v8, v9, v10 and the latest v11, for detecting and localizing drones and birds accurately. It involves fine tuning the pretrained YOLO models on the benchmark Airborne Object Detection datasets and measure their performance. The models are trained using a bunch of hyperparameters and several evaluation metrics were used to determine their performance.

---

2. Review of Literature:

For the detection of drones and birds [1] YOLO algorithm was used. You Only Look Once (YOLO) is a groundbreaking real-time object detection system that revolutionized detection by framing the task as a single regression problem, rather than a classification problem followed by a bounding box prediction. This approach allows YOLO to process images at high speeds while maintaining impressive accuracy, making it suitable for applications such as surveillance, autonomous driving, and robotics. YOLO's key advantage lies in its ability to process images in a single pass, making it exceptionally fast and efficient. It divides an image into a grid of cells and predicts bounding boxes and class probabilities for objects within each cell simultaneously. Over the years YOLO algorithm has undergone several enhancements in its architecture, from YOLOv1 to the latest YOLOv11, where each newer version introduced better feature extraction, improved detection accuracy and reduced inference time.

The YOLO algorithm has undergone several improvements from its initial version, YOLOv1, to YOLOv7, each addressing critical challenges and enhancing performance. [1] YOLOv1 introduced the innovative concept of framing object detection as a single regression problem, achieving real-time performance but struggling with small object detection and localization errors. [2] YOLOv2 (or YOLO9000) improved on this with batch normalization, high-resolution classifiers, and anchor boxes, increasing detection

accuracy and enabling real-time detection on more complex datasets. [3] YOLOv3 introduced a multi-scale detection system using a feature pyramid network (FPN), boosting performance on objects of various sizes and improving the detection of smaller objects. [4] YOLOv4 continued this progression with advancements like CSPDarknet53 as the backbone, Mish activation, and the use of a complete set of bag-of-freebies and bag-of-specials for better speed-accuracy trade-off. [5] YOLOv5 further refined the architecture, introducing cutting-edge techniques like cross-stage partial connections (CSP), spatial attention modules (SAM), and data augmentation strategies to boost performance, offering a streamlined PyTorch implementation, along with auto-learning bounding box anchors and model scaling. [6] YOLOv6 focused on optimizing performance for industrial applications, achieving faster inference. [7] YOLOv7 pushed further by introducing extended efficient layer aggregation networks (E-ELAN) and model scaling, decoupling head mechanisms for classification and regression, and refining architecture to deliver state-of-the-art object detection results while keeping speed a priority. A new backbone network called C2f was introduced in [8] YOLOv8. It utilized a new detection head called Decoupled Head v2 and implemented a new training strategy called Label Smoothing Regularization v2. [9] YOLOv9 prevented data loss and ensured accurate gradient updates by introducing Programmable Gradient Information (PGI). Utilized Generalized Efficient Layer Aggregation Network (GELAN) for optimizing lightweight models with gradient path planning. [10] YOLOv10 Introduces a novel training strategy that allows for end-to-end training without the need for NMS post-processing, leading to faster inference speeds. YOLOv10 outperforms YOLOv9 and YOLOv8 in terms of both speed and accuracy, making it a state-of-the-art real-time object detection model. YOLOv11 builds upon YOLOv8 with few changes in the model architecture. It achieves higher performance with fewer parameters and produces state-of-art results for different tasks.

Previously In [11], the authors utilized YOLOv2 to detect drones and birds, achieving precision and recall scores exceeding 90. Meanwhile, in [12], a drone detection pipeline was proposed featuring three models: Faster R-CNN with ResNet-101, Faster R-CNN with Inceptionv2, and SSD. After 60,000 iterations, the models attained mAP values of 0.49, 0.35, and 0.15, respectively. In [13], within a proposed drone detection model authors employed MobileNetV2 as a classifier. The methodology involved a moving object detector alongside a drone-bird-background classifier. The researchers trained the drone-vs-bird challenge dataset on an NVIDIA GeForce GT 1030 2GB GPU, using a learning rate of 0.05. When evaluated at an IoU of 0.5, the model achieved peak precision, recall, and F1 scores of 0.786, 0.910, and 0.801, respectively, based on tests conducted on three videos.

The authors in reference [14] built a dataset by collecting various types of drone images from the internet and videos. They annotated the images in the YOLO format to train a YOLOv3 model. The training was conducted on an NVIDIA GeForce GTX 1050 Ti GPU, with selected parameters that included a learning rate of 0.0001, a batch size of 64, and 150 total epochs. The model achieved a best mAP value of 0.74. PyTorch, an open-source machine learning framework, was used for both training and testing the YOLOv3 model.

As reported in [15], the authors utilized YOLOv4 to automate drone detection, aiming to integrate a trained model into a CCTV camera system, thereby minimizing the need for manual monitoring. They compiled their dataset from public sources, including Google Images and various open-source websites. To prepare the images for the YOLO format, they used a combination of free and paid annotation tools. The YOLOv4 architecture was fine-tuned by adjusting parameters such as filters, max batches, subdivisions, and batch sizes. After training for 1300 iterations, the model achieved an impressive mAP of 0.99. However, despite the high mAP, the dataset comprised only 53 images, and no measures were taken to address model overfitting, indicating room for further improvement.

The study in [16] proposed a YOLOv5-based approach, utilizing a dataset of 1,359 drone images sourced from Kaggle. They fine-tuned the model on a local system equipped with an 8 GB NVIDIA RTX 2070 GPU, 16 GB of RAM, and a 1.9 GHz CPU. The dataset was split into 60% for training, 20% for testing, and 20% for validation. By training the model using COCO pre-trained weights, they achieved a precision of 94.70%, a recall of 92.50%, and a mAP of 94.1%.

The work in [17] by the authors compares YOLOv5 with their previously proposed drone detection methodology using YOLOv4 [18], utilizing the same dataset and computing configurations for a fair assessment. They fine-tuned YOLOv5 on a customized dataset containing two classes—bird and drone—optimizing hyperparameters like learning rate, momentum, and decay. To enhance training efficiency, they applied transfer learning with pre-trained weights from the [19] MS COCO dataset and employed data augmentation through the Roboflow API, along with preprocessing techniques to mitigate overfitting. The model was evaluated using precision, recall, F-1 score, and mAP, achieving values of 0.918, 0.875, 0.896, and 0.904, respectively, with a 21.57% improvement in mAP over the previous model. Additionally, they tested detection speed on videos of DJI Phantom III and DJI Mavic Pro drones at altitudes of 20 ft, 40 ft, and 60 ft, recording maximum FPS values of 23.9 and 31.2.

---

3. Methodology:

3.1 Dataset Description:

The [20] Roboflow API was used to efficiently preprocess and load the Airborne Object Detection dataset for training. The API facilitated seamless data augmentation, enhancing the model's performance by generating diverse samples and reducing overfitting. This integration streamlined the data pipeline, allowing for effective training of our YOLO models on the bird vs drone detection task, while also saving time and computational resources.

The Airborne Object Detection dataset is designed to train models for detecting objects in the sky, particularly those that may pose risks to airspace safety. The dataset typically includes a variety of aerial objects such as aircraft, drones, birds and other flying entities. It is often used for tasks like surveillance, monitoring, and safety applications, where detecting these objects in diverse conditions, lighting, and backgrounds (e.g., clear skies, cloudy weather, urban environments) is critical.

Key characteristics of the dataset include:

- Object types: Drones, birds, planes, or any airborne objects.
- Image variety: High-resolution images with different altitudes, angles, and environmental conditions.
- Annotations: Bounding boxes around the objects to facilitate supervised learning for object detection.
- Applications: Drone detection, airspace monitoring, bird strike prevention etc.

The dataset contains 14725 training images, 4203 validation images, 2097 test images belonging to classes aeroplanes, drones, birds & helicopters.

Dataset link: https://universe.roboflow.com/airborne-object-detection/airborne-object-detection-4-aod4/dataset/3

3.2 Training Platform:

The models were trained using Google Colab notebook, an open-source platform for python programming. The Google L4 high RAM GPU was used to maximize computational efficiency. The robust GPU support provided accelerated model training, enabling faster iterations and better handling of large dataset. This setup allowed to fine-tune the YOLO models effectively, taking full advantage of Colab's high-performance hardware to achieve optimal detection results.

3.3 Model Training:

For the task, the [21] Ultralytics library was used to load the pretrained YOLOv8s, v9s, v10s, v11s models, which facilitated efficient object detection. Ultralytics HUB provides an intuitive AI platform where users can create, train, and deploy machine learning models using a no-code interface. With support for deep learning frameworks, it simplifies the workflow and offers an end-to-end solution, enabling seamless deployment to the cloud, edge, or browser. The library's ease of use and robust features allow seamless customization and transfer learning, enhancing its utility in various computer vision tasks.

For this work YOLOv8s, YOLOv9s, YOLOv10s and YOLOv11s models pretrained on MS COCO datasets were loaded from the Ultralytics library.

The YOLO algorithm has mainly 3 blocks:

- **Backbone**: The backbone extracts crucial features from the input image, identifying basic patterns like edges and textures in the early layers. As it progresses, it captures features at multiple scales, representing different levels of abstraction. This results in a comprehensive, hierarchical understanding of the input.
- **Neck**: The neck serves as a link between the backbone and the head, gathering feature maps from various stages of the backbone. It combines features from different scales, enabling the network to detect objects of various sizes effectively. By incorporating contextual information, it enhances detection accuracy by taking into account the overall scene context. Additionally, it reduces spatial resolution and dimensionality to optimize computational efficiency.
- **Head**: The head is the concluding component of the network, tasked with producing outputs like bounding boxes and confidence scores for object detection. It assigns a confidence score to each bounding box, reflecting the probability of an object's presence. Additionally, it classifies the detected objects in the bounding boxes based on their categories.

YOLOv8s: It has **11.2M params & 8.7B FLOPs**. The backbone of YOLOv8 incorporates standard convolutional layers, bottleneck layers, and Cross-Stage Partial Networks (CSPNet), which divides the feature map for efficient processing while maintaining accuracy. The neck of the model creates a multi-scale feature pyramid by combining feature maps from various backbone layers, enhancing the detection of objects of different sizes. It utilizes a Feature Pyramid Network (FPN) to merge high-level semantic features with lower-level details, and a Path Aggregation Network (PAN) to facilitate information flow through multiple paths for improved feature fusion. Finally, the head of YOLOv8 features a decoupled design that separates object classification from bounding box regression, leading to enhanced detection accuracy.

YOLOv9s: It has **7.2M params & 26.7B FLOPS.** YOLOv9 is an advanced object detection model that enhances the YOLOv8 architecture through several key components. Its backbone features the Efficient Large-scale Attention Network (ELAN) block, which integrates dilated convolutions and an innovative attention mechanism to effectively capture both local and global information. Additionally, the RepBlock, a lightweight module, improves feature representation by repeatedly applying convolutional layers. The neck of the model utilizes the Cross-Stage Partial Network (CSPPAN), which enhances feature fusion by splitting feature maps into two parts and concatenating them at various stages. Finally, the YOLO Head retains a familiar structure from previous models, comprising anchor boxes and prediction layers for determining objectness, class, and bounding box coordinates.

YOLOv10s: It has **7.2M params & 21.6B FLOPS.** The YOLOv10 architecture features an enhanced CSPNet backbone for efficient feature extraction, improving gradient flow and reducing redundancy. Its neck uses PAN layers to merge multiscale features effectively, optimizing the data sent to the head. The model employs a dual-head approach: a "One-to-Many" head generates multiple predictions per object during training to enhance learning, while a "One-to-One" head produces a single optimal prediction per object during inference,

eliminating the need for Non-Maximum Suppression (NMS) and reducing latency. Additionally, large-kernel convolutions and partial self-attention enhance the model's accuracy with minimal computational impact.

YOLOv11s: It has **9.4M params & 21.5B FLOPs.** YOLOv11s introduces an upgraded backbone and neck, boosting object detection accuracy and enabling efficient handling of complex tasks at faster speeds. It achieves high accuracy with fewer parameters and is versatile, supporting a range of computer vision applications, such as object detection, instance segmentation, image classification, pose estimation, and oriented object detection (OBB).

All the models were trained using the same set of hyperparameters setup listed in Table 1.

Sl.no	Hyperparameters	Values
1	Epochs	40
2	Warmup epochs	3
3	Warmup momentum	0.8
4	Batch size	16
5	Image size	640
6	Initial learning rate	0.01
7	Final learning rate	0.01
8	Patience	100
9	Optimizer	AdamW
10	Momentum	0.937
11	Mask ratio	4
12	Weight Decay	0.0005

Table 1. Hyperparameters setup

The albumentations used for all the models were Blur (p= 0.01, blur_limit= (3, 7)), MedianBlur (p= 0.01, blur_limit= (3, 7)), ToGray (p= 0.01, num_output_channels= 3, method= 'weighted_average'), CLAHE (p= 0.01, clip_limit= (1, 4.0), tile_grid_size= (8, 8)).

---

4. Results:

The metrics used to evaluate the model included precision, recall, and mean average precision (mAP). In addition to these metrics, classification loss, box loss, and distribution focal loss were tracked for each epoch. Precision-recall curves were plotted for each model to assess performance. Although the dataset contains four classes, the model's performance on the "bird" and "drone" classes was analyzed first, followed by a comparison based on performance across all classes.

- **Box Loss**: Measures the error between the predicted bounding box coordinates and the ground truth box coordinates
- **Classification Loss**: Measures the error in the predicted class labels for detected objects. It is typically a form of cross-entropy loss or focal loss used to improve the accuracy of object classification.
- **Distribution Focal Loss**: This is a loss function used in object detection to improve the precision of the bounding box regression. It helps the model refine the predicted bounding box by focusing more on difficult-to-localize objects.

- **Mean Average Precision (mAP)**: It is a metric used to evaluate object detection models by averaging the precision across all classes and different Intersection over Union (IoU) thresholds. It measures how well the model detects objects and how accurately the predicted bounding boxes match the ground truth.
    - **mAP50 –** Calculates the mAP for instances having IoU more than 0.5.
    - **mAP50-90 -** The mAP is computed for different IoU thresholds (0.50, 0.55, 0.60, …, 0.95), and the final metric is the mean of all these mAPs.
- **Intersection Over Union (IoU):** It is the metric which is the ratio of the area of intersection of the actual the predicted bounding boxes to the union of the area of the actual and predicted bounding boxes
- **Precision**: Precision can be thought of as the fraction of positive predictions that *actually* belong to the positive class.

$$\text{precision} = \frac{TP}{TP+}$$

- **Recall**: Recall can be thought of as the fraction of positive predictions out of all positive instances in the data set.

$$\text{recall} = \frac{TP}{TP+}$$

- **Precision – Recall curve**: A precision-recall curve is a graphical plot used to visualize the trade-off between precision and recall for a classification model. It's particularly useful when dealing with imbalanced datasets, where the number of instances in each class is significantly different.

Table 2 summarizes the precision, recall, mAP50, and mAP50-90 values for only bird and drone classes achieved by the models on the test set.

Models	Classes	Precision	Recall	mAP50	mAP50-90
YOLOv8s	Bird	0.952	0.957	0.976	0.725
	Drone	0.931	0.924	0.957	0.629
YOLOv9s	Bird	0.955	0.946	0.974	0.727
	Drone	0.932	0.921	0.956	0.63
YOLOv10s	Bird	0.942	0.936	0.97	0.724
	Drone	0.934	0.915	0.959	0.632
YOLOv11s	Bird	0.947	0.958	0.976	0.736
	Drone	0.939	0.92	0.955	0.627

Table 2. Evaluation metrics results for bird and drone class

Table 3 summarizes the precision, recall, mAP50, and mAP50-90 values for all the classes achieved by the models on the test set.

Model	Precision	Recall	mAP50	mAP50-90
YOLOv8s	0.948	0.943	0.969	0.682
YOLOv9s	0.947	0.939	0.966	0.68
YOLOv10s	0.939	0.927	0.965	0.676
YOLOv11s	0.947	0.94	0.967	0.68

Table 3. Evaluation metrics results for all the classes

The following figures show the precision-recall curves and overall training summaries for each model, providing a visual overview of model performance and training progress.

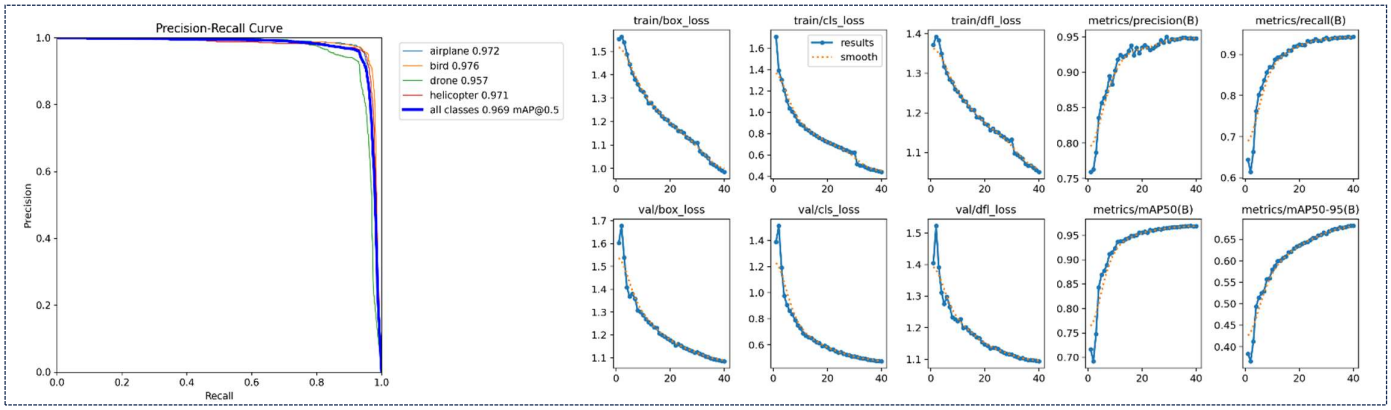

Figure 1. Precision-recall curve and overall training summary of YOLOv8s

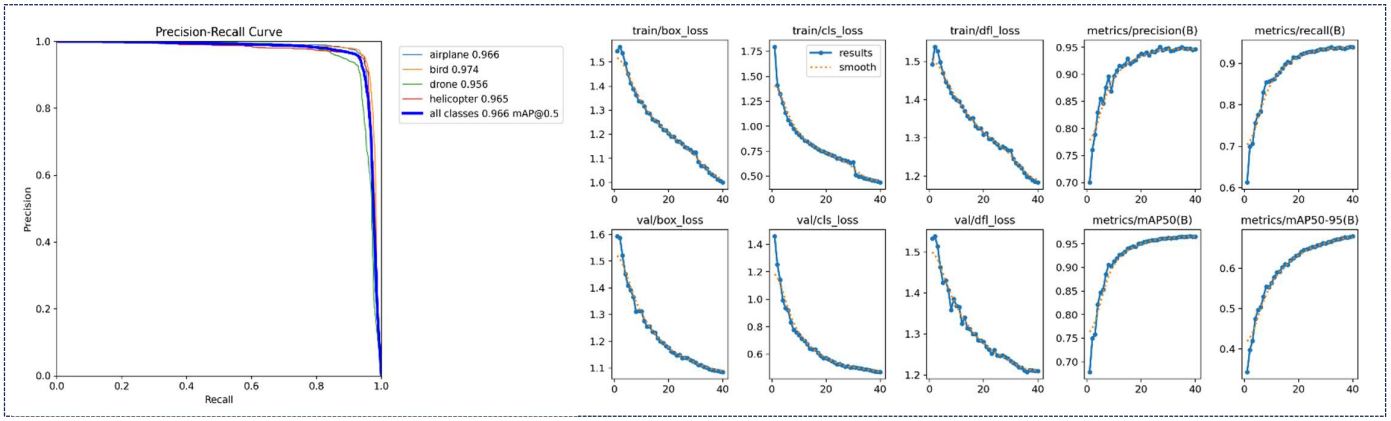

Figure 2. Precision-recall curve and overall training summary of YOLOv9s

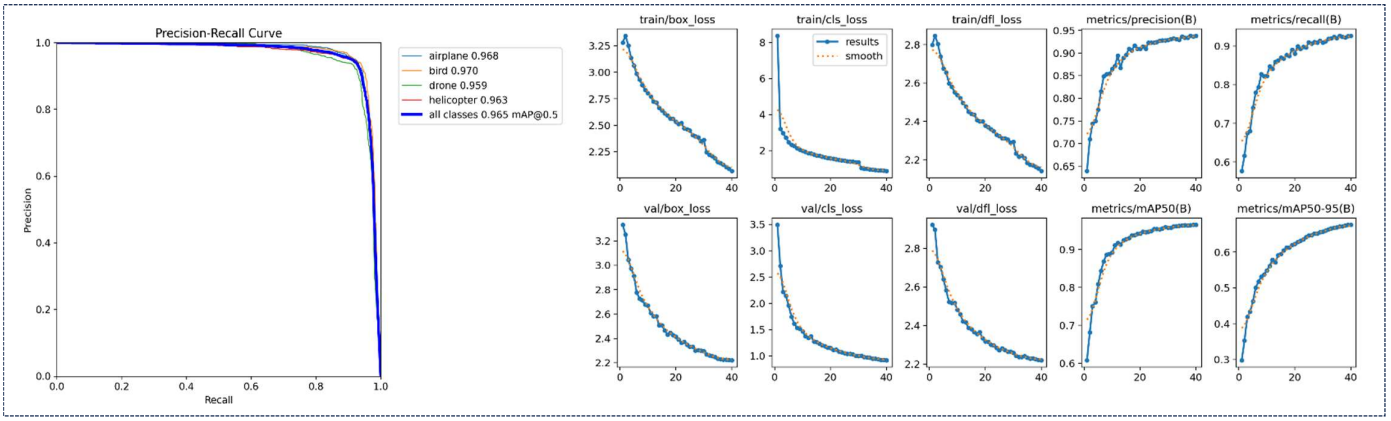

Figure 3. Precision-recall curve and overall training summary of YOLOv10s

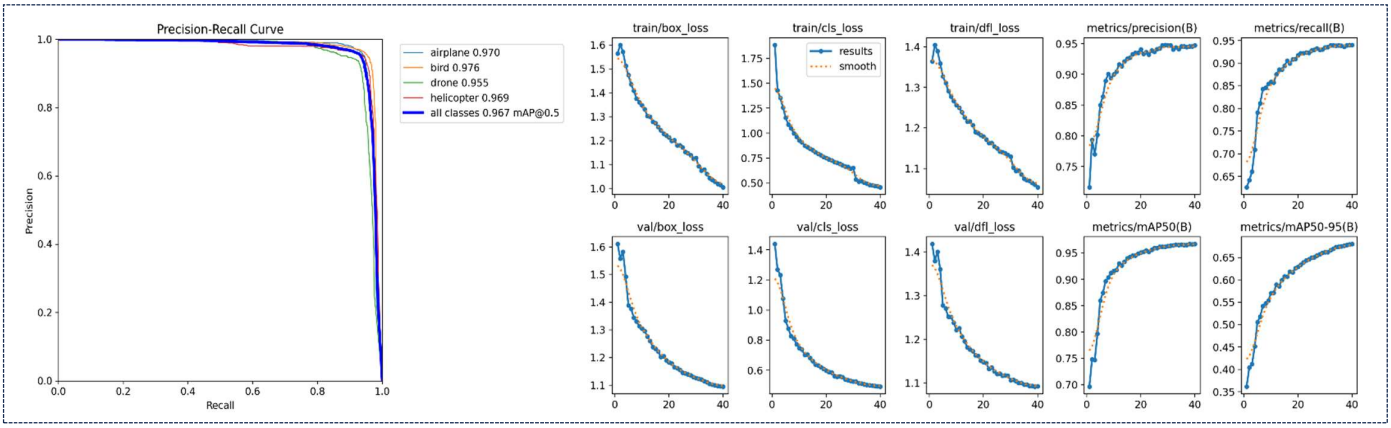

Figure 4. Precision-recall curve and overall training summary of YOLOv11s

---

5. Discussions:

For only the bird and drone classes to evaluate the performance of each model, we considered precision, recall, mAP50, and mAP50-90 metrics. Among these, YOLOv9 achieves the highest precision for the bird class, reaching 0.955, which suggests its superior ability to avoid false positives for birds. For the drone class, YOLOv11 shows the highest precision at 0.939, indicating its robustness in accurately identifying drones with minimal false detections.

In terms of recall, which measures the ability to capture true positives, YOLOv11 leads for the bird class with a recall of 0.958, signifying its effectiveness in detecting most instances of birds. YOLOv8, on the other hand, achieves the highest recall for the drone class at 0.924, indicating its strong detection capability for drones.

When evaluating mAP50 (mean Average Precision at an IoU threshold of 0.50), YOLOv8 and YOLOv11 both attain a high score of 0.976 for the bird class, demonstrating their precision-recall balance at this threshold. However, for the drone class, YOLOv10 slightly outperforms the others, achieving a mAP50 of 0.959, indicating its top performance in capturing and localizing drones.

Finally, for mAP50-90 (mean Average Precision across IoU thresholds from 0.50 to 0.90), which evaluates model robustness across varying IoU thresholds, YOLOv11 excels in the bird class with a score of 0.736, highlighting its consistency in precise localization for birds across stricter criteria. YOLOv10 performs best for the drone class with a score of 0.632, showing its adaptability in accurately localizing drones over a range of IoU values.

This analysis reveals each model's specific strengths across the bird and drone classes, with YOLOv11 showing overall balanced performance, especially for the bird class, while YOLOv10 and YOLOv8 also demonstrate key advantages in certain metrics. This comparison helps highlight the trade-offs between the models, particularly in detection accuracy and localization across different IoU thresholds.

While for all the classes, from the comparative analysis conducted between YOLOv8, YOLOv9, YOLOv10, and YOLOv11, it is evident that YOLOv8 demonstrates an impressive overall performance. Specifically, YOLOv8 outperforms the other versions across various metrics when considering the detection of all the classes in the dataset achieving 0.948, 0.943, 0.969, 0.682 values for precision, recall, mAP50 and mAP50-90 respectively. This suggests that despite being the earliest among the four models, YOLOv8 still

proves to be a reliable and highly efficient model choice for general object detection scenarios within this domain.

However, when focusing on the performance of specific classes, namely Birds and Drones, YOLOv11 stands out in several key aspects. The model achieves the highest value for at least one of the performance metrics for either Bird or Drone detection compared to the other versions. This implies that YOLOv11 is showing a balanced performance and possesses specialized strengths that make it more effective for targeted detection tasks, especially in scenarios where precise identification of airborne objects like birds or drones is crucial.

The following figures shows the detection of drones by the trained YOLOv8s, YOLOv9s, YOLOv10s and YOLOv11s models.

Figure 5. Drone detection by trained YOLOv8s

Figure 6. Drone detection by trained YOLOv9s

Figure 7. Drone detection by trained YOLOv10s

Figure 8. Drone detection by trained YOLOv11s

Figures 5–8 demonstrate that the trained models successfully detect drones of various sizes and levels of visibility.

The following Figures 9 – 12 shows the detection of airborne objects belonging to different classes present in the dataset by the trained models.

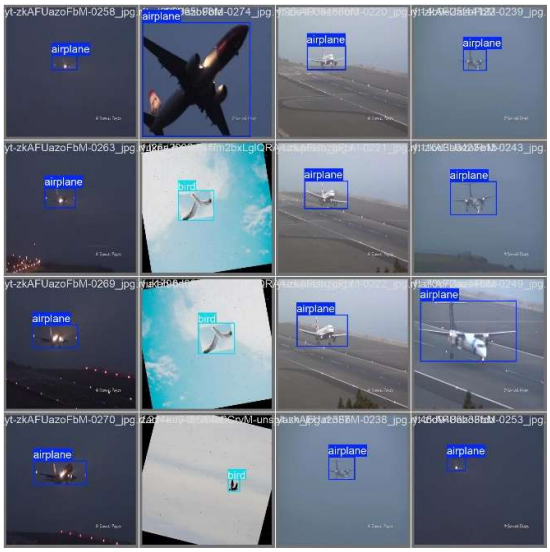

Figure 9. Airborne object detection by trained YOLOv8s

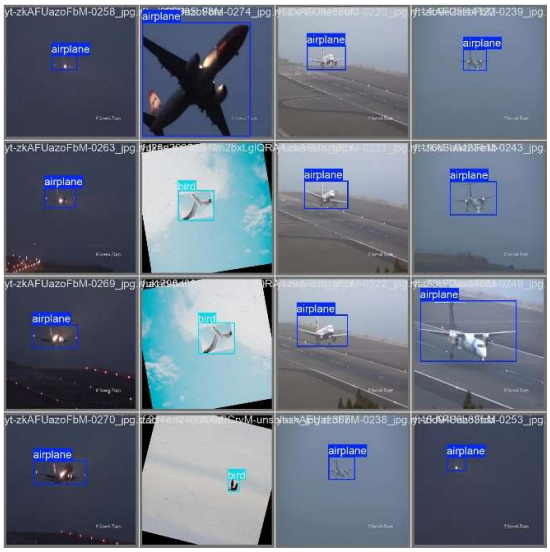

Figure 10. Airborne object detection by trained YOLOv9s

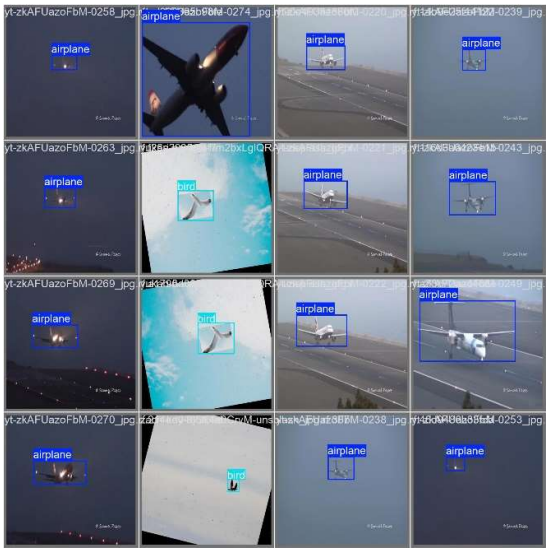

Figure 9. Airborne object detection by trained YOLOv10s

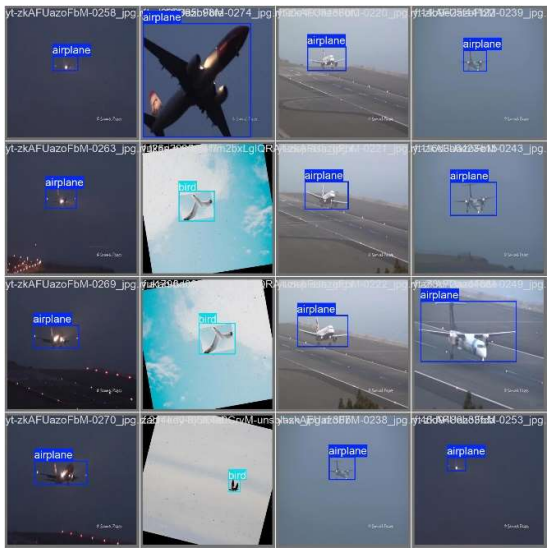

Figure 10. Airborne object detection by trained YOLOv11s

Figures 9 – 12 shows that the trained models can also successfully detect and localize other airborne objects in addition to drones and birds.

6. Conclusion:

This study provides a comprehensive comparative analysis of YOLO models (YOLOv8s, YOLOv9s, YOLOv10s, and YOLOv11s) for airborne object detection, specifically targeting birds and drones. Google Colab notebook was used for training the models that were loaded from the Ultralytics library. The dataset used in this task was loaded using the Roboflow API and the required data augmentation were applied to avoid overfitting. Each model demonstrated unique strengths across performance metrics, revealing specialized capabilities and trade-offs. YOLOv9 excelled in precision for birds, YOLOv11 led in recall for birds, while YOLOv8 achieved the highest recall for drones. YOLOv10 showed an edge in drone localization at mAP50, and YOLOv11 proved most robust across varying IoU thresholds for birds. Despite being the earliest model, YOLOv8 displayed consistently high performance across all dataset classes, underscoring its reliability for general object detection tasks.

Notably, YOLOv11 emerged as the top performer for specific tasks, indicating its suitability for targeted airborne object detection, where precise identification of birds and drones is essential. Furthermore, this project demonstrated a 5% improvement over previous [18] YOLOv5 results, highlighting advancements in YOLO architectures for this domain. This analysis underscores the value of selecting models based on specific detection needs and contextual performance requirements, providing a refined approach to airborne object detection tasks.

References:

[1] Redmon, J. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[2] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

[3] Redmon, Joseph. "Yolov3: An incremental improvement." *arXiv preprint arXiv:1804.02767* (2018).

[4] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." *arXiv preprint arXiv:2004.10934* (2020).

[5] Zhu, Xingkui, et al. "TPH-YOLOv5: Improved YOLOv5 based on transformer prediction head for object detection on drone-captured scenarios." *Proceedings of the IEEE/CVF international conference on computer vision*. 2021.

[6] Li, Chuyi, et al. "YOLOv6: A single-stage object detection framework for industrial applications." *arXiv preprint arXiv:2209.02976* (2022).

[7] Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023.

[8] Reis, Dillon, et al. "Real-time flying object detection with YOLOv8." *arXiv preprint arXiv:2305.09972* (2023).

[9] Wang, Chien-Yao, I-Hau Yeh, and Hong-Yuan Mark Liao. "Yolov9: Learning what you want to learn using programmable gradient information." *European Conference on Computer Vision*. Springer, Cham, 2025.

[10] Wang, Ao, et al. "Yolov10: Real-time end-to-end object detection." *arXiv preprint arXiv:2405.14458* (2024).

[11] Aker, C.; Kalkan, S. Using deep networks for drone detection. In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–6.

[12] Nalamati, M.; Kapoor, A.; Saqib, M.; Sharma, N.; Blumenstein, M. Drone Detection in Long-Range Surveillance Videos. In Proceedings of the 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; pp. 1–6.

[13] Seidaliyeva, U.; Akhmetov, D.; Ilipbayeva, L.; Matson, E.T. Real-Time and Accurate Drone Detection in a Video with a Static Background. Sensors 2020, 20, 3856.

[14] Behera, D.K.; Raj, A.B. Drone Detection and Classification using Deep Learning. In Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 13–15 May 2020; pp. 1012–1016.

[15] Mishra, A.; Panda, S. Drone Detection using YOLOV4 on Images and Videos. In Proceedings of the 2022 IEEE 7th International conference for Convergence in Technology (I2CT), Mumbai, India, 7–9 April 2022; pp. 1–4.

[16] Al-Qubaydhi, N.; Alenezi, A.; Alanazi, T.; Senyor, A.; Alanezi, N.; Alotaibi, B.; Alotaibi, M.; Razaque, A.; Abdelhamid, A.A.; Alotaibi, A. Detection of Unauthorized Unmanned Aerial Vehicles Using YOLOv5 and Transfer Learning. Electronics 2022, 11, 2669.

[17] Aydin, Burchan, and Subroto Singha. "Drone detection using yolov5." *Eng* 4.1 (2023): 416-433.

[18] Singha, Subroto, and Burchan Aydin. "Automated drone detection using YOLOv4." *Drones* 5.3 (2021): 95.

[19] Lin, Tsung-Yi, et al. "Microsoft coco: Common objects in context." *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer International Publishing, 2014. Link: https://cocodataset.org/#download

[20] Dwyer, B., Nelson, J., Hansen, T., et. al. (2024). Roboflow (Version 1.0) [Software]. Available from https://roboflow.com

[21] Jocher, Glenn, Jing Qiu, and Ayush Chaurasia. Ultralytics YOLO. Version 8.0.0, Ultralytics, 10 Jan. 2023, https://github.com/ultralytics/ultralytics