Article

YOLOv8 Based Drone Detection: Performance Analysis and Optimization

Betul Yilmaz * and Ugurhan Kutbay

Posted Date: 26 July 2024

doi: 10.20944/preprints202407.2127.v1

Keywords: YOLOv8; drone detection; deep learning

Article

YOLOv8 Based Drone Detection: Performance Analysis and Optimization

Betul Yilmaz [1,2,*], Ugurhan Kutbay [3]

[1] Graduate School of Natural and Applied Science, Gazi University, Ankara, Turkey; 22830601533@gazi.edu.tr
[2] Aselsan Inc., Ankara, Turkey; betulyilmaz@aselsan.com
[3] Department of Electrical and Electronics Engineering, Gazi University, Ankara, Turkey; ukutbay@gazi.edu.tr
[*] Correspondence: 22830601533@gazi.edu.tr

Abstract: The extensive utilization of drones has led to numerous scenarios that encompass both advantageous and perilous outcomes. By using deep learning techniques, it is aimed to reduce the dangerous effects of drone use through early detection of drones.The state of the art of the study is evaluation of deep learning approaches such as pre trained YOLOv8 drone detection for security issues. In this study, dataset collected by Mehdi Özel for a UAV competition are sourced from GitHub. These images are labeled using Roboflow, and the model is trained on Google Colab. To enhance model performance, dataset augmentation techniques including rotation and blurring are implemented. Following these steps, a significant precision value of 0.946, a notable recall value of 0.9605 and a considerable precision-recall curve value of 0.978 are achieved, surpassing many popular models such as Mask CNN, CNN, and YOLOv5.

Keywords: YOLOv8; drone detection; deep learning

1. Introduction

In recent years, drones have emerged as a noteworthy technological advancement with vast potential across numerous industries [1]. Their far-reaching selection of usages has led to innovative solutions in fields ranging from package delivery to agricultural practices and security surveillance [2]. However, the increasing use of drones has also brought new challenges to public safety, particularly regarding misuse of these devices for illegal or harmful purposes. Especially with the inclusion of the cameras to drones the problem has spread to the concerns on various fields such as privacy and terrorism matters [3–5]. Therefore, security authorities accelerated to create systems that are effective to detect and recognize drones. This concluded in a new research area addressing to limitations of traditional detection methods and more effective solutions [6]. Eventually, employing deep learning techniques has become a crucial part of the studies in the area. Deep learning, a subset of artificial intelligence, facilitates the automated recognition of complex data patterns, offering more precise and adaptable detection systems through extensive data analysis to find safer solutions. Current unmanned aerial vehicle detection methods are sound signal-based, radar-based, radio frequency-based, and image and video-based approaches [7,8]. This article examines techniques centered on images as those techniques offer numerous advantages contributing to enhanced performance in deep learning. One of the various contributions is that image-based approaches enable one to gather rich information with the help of images, objects, patterns and colours. Furthermore, thanks to high dimensionality, they manage the complexity of deep networks. This approach would lead wide range of applications in various fields such as defense industry. And also image-based approach offers data augmentation capability and shows resilience in noisy environments [9].

The outputs gathered from this study can be applied to enhance security and provide preservation against various threats in both military and civilian areas. The results of this study can also be used for the following purposes. The first one is threat detection and identification. Possible threats (drones) can be detected and identified using images acquired through cameras. It can also be useful for warning systems. When a threat is detected, audible, visual, or other warnings can be provided to alert personnel or users. This ensures timely responses to notify the presence of danger and take necessary

doi:10.20944/preprints202407.2127.v1

precautions. It can be used to protect civilian or military facilities, areas, or vehicles. Early detection and warning of potential threats help prevent attacks or at least mitigate their impact. Finally, it can be evaluated for tactical use in military operations. It can be used to determine the approach of the enemy and initiate defense preparations by alerting military units.

This study continues as follows: Section 2 outlines the present research, Section 3 elucidates experimental particulars including setup, evaluation, parameter tuning, results, and comparisons, while Section 4 offers a summary of the study's findings. 5 gives a brief suggestion for future works.

2. Literature Review

Over the years, many methods for drone detection have been tried; suggested and related studies have been carried out. Many different studies have been executed to detect drones via using radars and alternative methods [10], and it has been evaluated that the electromagnetic waves emitted by drones are not sufficient for detection by radars [11]. Through studies on relevant methods, it has been evaluated that the price-performance ratio of sound-based and radio frequency-based methods studies are low [12] The most prominent method used for drone detection is image processing.

Angelo Coluccia and colleagues studied on using radar sensor networks for drone detection and the main challenges that occurs in this context [13]. An investigation of the existing literature about the most promising approaches adopted to solve different challenges such as detecting the possible presence of drones, target verification and classification has been executed. This content was also effective in concluding that the most efficient method that can be used for drone detection is image processing.

Image processing is vital for drone detection. The study by B. Srinivasa S. P. Kumar and Irshad Ahmad Wani used fish-eye cameras to detect flying drones [14]. Fish-eye cameras are cameras with a 180-degree or wider angle of view designed to capture a wide area in a single shot. In the study carried out with these cameras, drone detection was made using drone, convolutional neural network (CNN), support vector machine (SVM) and nearest neighbor classification methods. Among these three methods, the accuracy of the convolutional neural network classifier was found to be satisfactory when compared with other classifiers with the same experimental conditions [14].

In another study, a deep learning-based object detection algorithm "you only look once (YOLOv5)" was used to defend restricted areas or special regions from illegal drone interference [15]. One of the most challenging issues in drone detection inside the context of surveillance videos is to distinguish the apparent similarity of drones on different backgrounds. Therefore, it is aimed to improve the performance of the model. Since there are not enough examples in the dataset, transfer learning is used to pre-train the model. Thus, very high results were obtained for experiments, loss value, drone location detection, precision and recall.

Haoyu Wang and colleagues [16] focused on 3 main issues in their study and in this way aimed to improve the performance of the original YOLOV8 algorithm for the DOTA V1.0 dataset. Firstly, to extract more information about small targets in the images, an additional detection in the base network layer to detect targets of different sizes in images. To detect targets of different sizes in the images, they emphasized that it would be beneficial to use an Efficient Multiscale Attention (EMA) module based C2f-E structure. Lastly, they used Wise-IoU as a replacement for the CIoU loss function in the original algorithm to increase the robustness of the model. It has been proven that the improved algorithm has 1.3% better performance for DOTA V1.0 data and it can effectively increase target detection accuracy in remote sensing images. This study presents a remote sensing image target detection algorithm based on YOLOv8 that focuses on the complexity of remote sensing images background, targeting a large number of small targets and a variety of target scales.

In another study, YOLOv8 have been used for real-time drone detection by B. Srinivasa and colleagues [17]. Real-time detection is a mandatory prerequisite for timely identification and response service. To achieve this goal, the YOLOv8 model is integrated with TensorFlow.JS, thus enabling system installation and integration into web-based applications without the need for server-side processing is provided. The proposed real-time drone detection system has exceptional accuracy rates and established a robust criterion for reliability. This research utilized the MS COCO dataset, containing images of drones, aircraft, and birds. The model yielded highly successful results in terms of recall, precision, F-1 score, and mAP values.

2.1. History of YOLO

In the realm of deep learning, there exists a multitude of algorithms; however, YOLO (You Only Look Once) is favored predominantly due to its rapid processing speed and its more accurate results. The fundamental reason of its speed is the process-developing working mechanism. YOLO family has evolved through multiple iterations, each building upon the previous versions to address limitations and enhance performance.

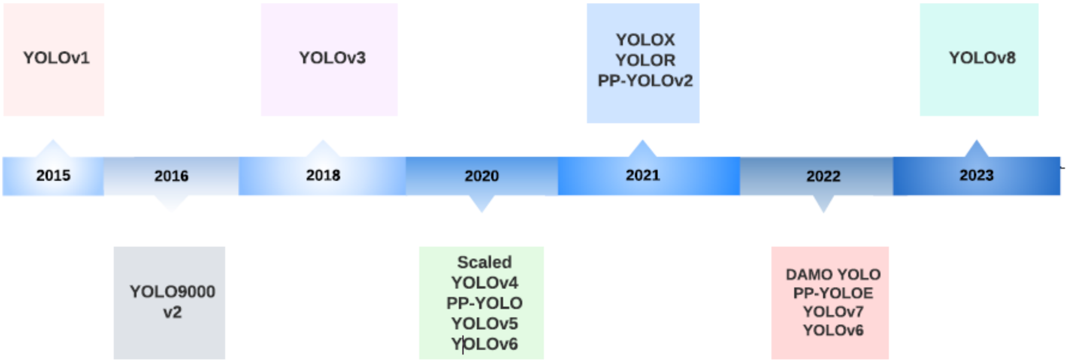

Figure 1. YOLO Family Timeline

YOLO was developed by Joseph Redmon and Ali Farhadi at the University of Washington. Launched in 2015, YOLO quickly gained popularity for its high speed and accuracy. There are several key patterns to be able to benchmark the development of the YOLO versions, anchors, backbone, framework and performance [18].

YOLOv2 released (in 2016) to solve the problem of YOLOv1 that didn't detect the different size objects. To be able to solve this, batch normalization, anchor boxes and dimension clusters were improved.

YOLOv3, which was launched in 2018, enhance the model's performance. To be able to do this backbone network, multiple anchors and spatial pyramid pooling were enhanced. YOLOv4, which was launched in 2020, shows minimal deviation from YOLOv3 in terms of significant changes. There is more CNN layer in YOLOv4 than YOLOv3.

YOLOv5 stands out as the most distinct version compared to its predecessors, primarily because it adopts PyTorch instead of Darknet. This change has led to an enhancement in the model's performance. Additionally, the introduction of hyper-parameter optimization, integrated experiment tracking, and automatic export features further contribute to its advancements.

YOLOv6 was has been improved to have better performance, have faster training and better detection. To be able to success this model architecture has been improved.

YOLOv7 has expanded its capabilities by incorporating new tasks like pose estimation using the COCO key point's dataset.

YOLOv8 is the latest iteration of the YOLO series of real-time object detectors, offering the ultimate in performance in terms of accuracy and speed. Building on the developments of previous versions, it offers new features and optimizations that make it an ideal choice for a variety of object detection tasks in a wide range of applications.

Exploring the advancements of YOLO provides a thorough understanding of the framework's progression. This progression can be outlined in various areas such as network design, adjustments to the loss function, modifications to anchor boxes, and changes in input resolution. Additionally, each new version of YOLO demonstrates improvements in both speed and accuracy. When choosing a YOLO version, it is crucial to consider the specific application's context and requirements.

3. Material and Method

This chapter discusses the materials used for evaluation of YOLOv8 and the evaluation criteria after experiment conducted.

3.1. Experimental Environment and Dataset

In this paper, the YOLOv8 model was implemented using the free cloud-based Jupyter notebook, Google Colab. The biggest advantage of Google Colab is that it provides a Tesla T4 NVIDIA GPU with 15110MiB of memory while training the model. Additionally, Google Colab can store projects and can be easily accessed since it is integrated with Google Drive.

In this study, we utilized the freely available dataset collected by Mehdi Özel for a UAV competition was used [19].This dataset includes both ".txt" and ".xml" files, facilitating training on Darknet (YOLO), TensorFlow, and PyTorch models. The dataset comprises 1359 images, all of which have been meticulously labeled and annotated for the purpose of detecting and recognizing drone objects. The dataset contains multiple categories of drones, and the images were captured at different distances from the drones. The dataset includes various angles, altitudes, and backgrounds to ensure diversity. In this study, all categories of drones are treated as a single class. The dataset was split in an 70:20:10 ratio for training, validation and testing the YOLOv8 model by using Roboflow.

The samples of the dataset, which include various categories and sub-categories, can be seen in Figure 2. The dataset features various drone orientations and sizes, as illustrated in Figure 3. Figures 3-a and 3-b demonstrate that each instance contains only one drone, and the entire dataset is classified as drones. Figure 3-c illustrates the distribution of drone object locations within the dataset, while Figure 3-d depicts the size distribution of the drone objects.

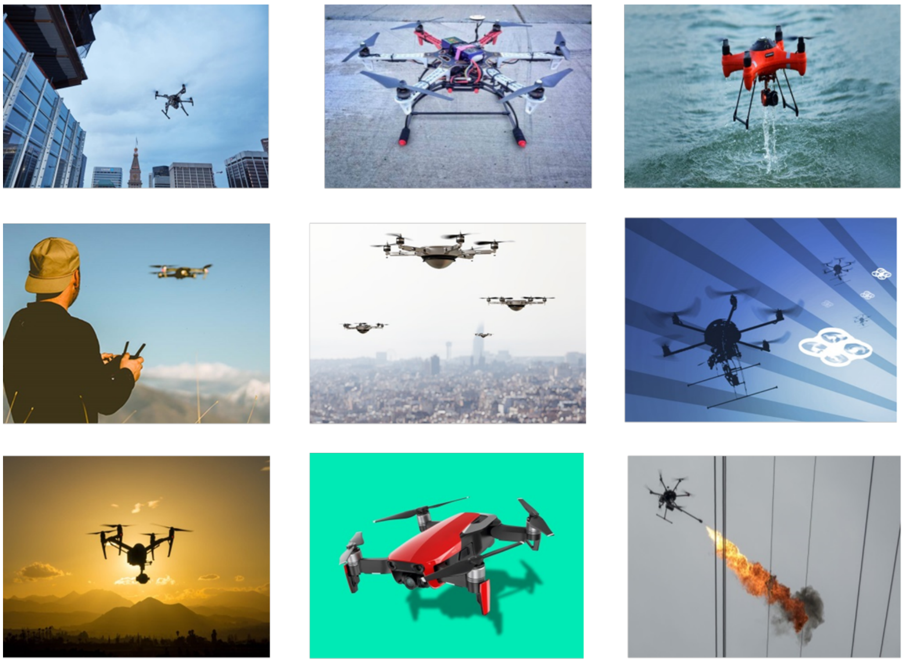

Figure 2. Drone Images Samples From The Dataset [19]

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 26 July 2024 doi:10.20944/preprints202407.2127.v1

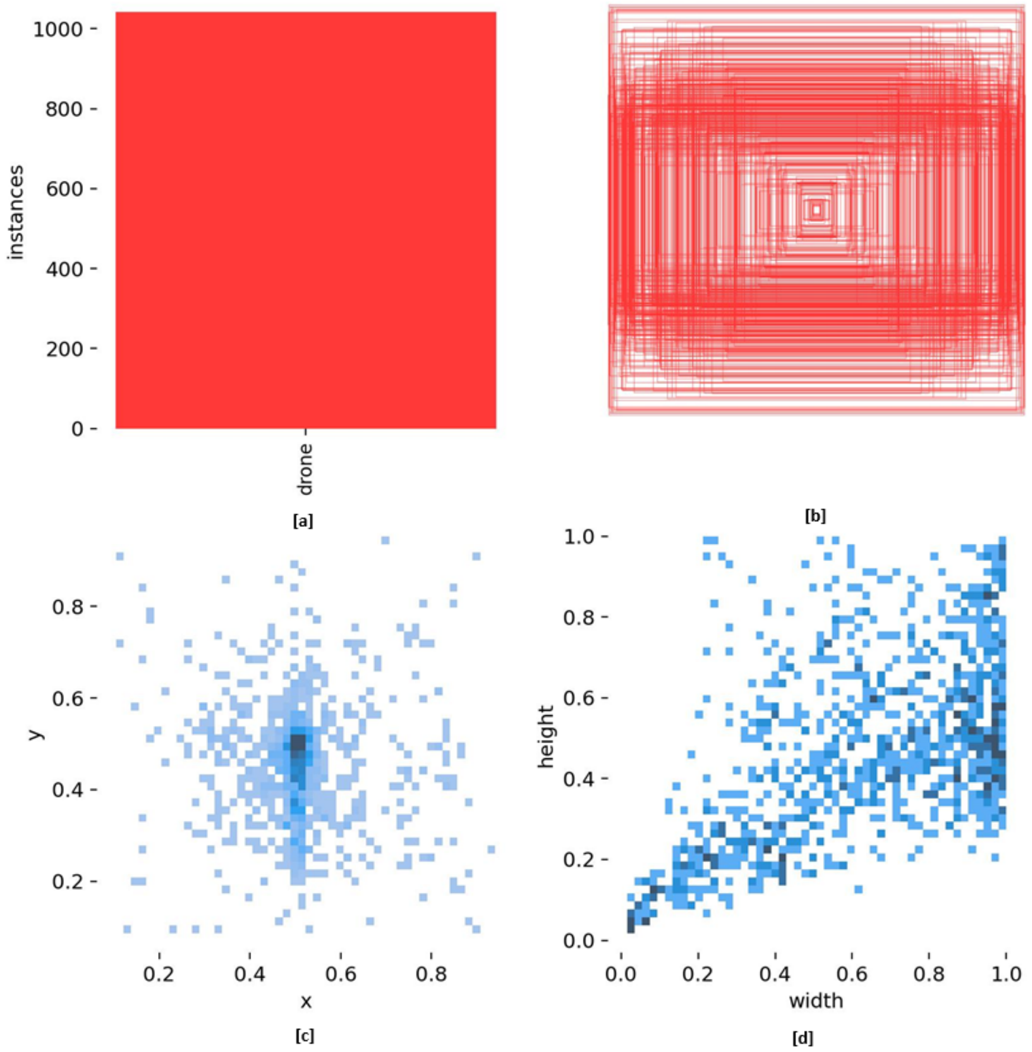

Figure 3. [a] The drone classification distribution of instances [b] The drone classification distribution of instances [c] The location distribution of drone object within dataset [d] The size distribution of the drone objects within dataset

The variation in the sizes of the drone objects can lead to an excessive number of computational parameters. Therefore, before training, the original image size is uniformly reduced to 640 pixels × 640 pixels by pixel transformation. In addition to resizing, auto-orientation is applied to the dataset as a preprocessing step. To enable comparison of the effects of various augmentation methods on the dataset, techniques such as rotation, flipping, blurring, cropping, and gray scale conversion were applied. The results of the augmentation methods will be elaborated upon in Section 4.2 Dataset Augmentation.

3.2. Evaluation Indicators

To facilitate the assessment of detected samples, a confusion matrix acts as a valuable tool. It delineates four distinct outcomes: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). Real parameters correspond to columns, while predicted parameters correspond to rows, enabling the display of these four results as illustrated below:

Table 1. Confusion Matrix

Confussion Matrix		Real Parameters	
		POSITIVE	NEGATIVE
Predicted Parameter	POSITIVE	TP	FP
	NEGATIVE	FN	TN

To assess the efficacy of the trained model in drone detection, three key metrics are utilized: recall (R), precision (P), and average precision (AP). The recall parameter is determined by dividing the count of correctly identified positive targets by the total number of positive targets, as demonstrated below:

$$Recall(R) = \frac{TP}{(TP + FN)} \tag{1}$$

Where TP is the number of correctly identified positive samples by the algorithm, FN is the positive samples incorrectly detected as negative samples by the algorithm. Precision parameter is determined by dividing the count of correctly identified positive targets by the total number of positive detections, as demonstrated below:

$$Precision(P) = \frac{TP}{(TP + FP)} \tag{2}$$

Where TP is the number of correctly identified positive samples by the algorithm, FP is the negative samples incorrectly detected as positive samples by the algorithm.

Once Precision (P) and Recall (R) parameters are computed, the Average Precision (AP) can be visualized by plotting precision on the y-axis against recall on the x-axis. The Mean Average Precision parameter represents the average of AP values across multiple classes. The formulas for calculating Average Precision (AP) and Mean Average Precision (mAP) are provided as follows:

$$AP = \int_0^1 PRdR \tag{3}$$

$$mAP = \frac{\sum_{k=1}^{n} APk}{n} \tag{4}$$

Each parameter listed above holds significant importance in evaluating the model, and all results will be utilized for assessment purposes.

4. Result Discussion

4.1. Hyperparameter Settings

The hyper-parameter settings in YOLO play a crucial role in determining the performance, speed, and accuracy of the model. These configurations significantly influence the behavior of the YOLO model across different stages of its development: training, validation, and testing. The study involved altering the hyper-parameters to assess the system's behavior, the changes made are here:

Epoch represents a complete pass over the entire dataset. Adjusting this value may affect training time and model performance. In this study, the best results were consistently achieved with 150 epochs for all trials. When the number of epochs was set to 150, it was noticed that the trained model became steady and stable. When non-default values were assigned to other hyper-parameters, 150 epochs emerged as the optimal choice for achieving the maximum value. Figure 4 represents some of the results obtained in this study.

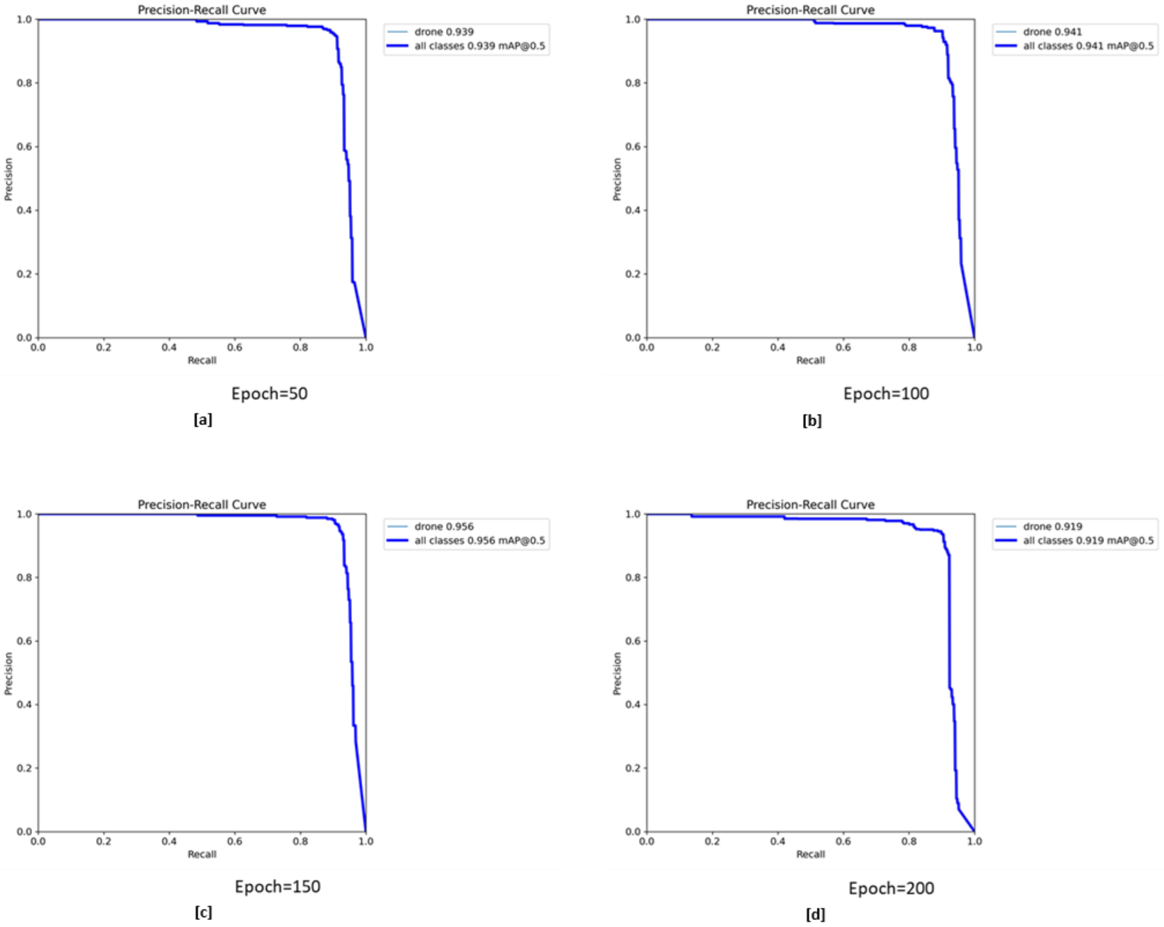

Figure 4. [a] Precision -Recall Curve for Specific Dataset at 50 Epoch [b] Precision -Recall Curve for Specific Dataset at 100 Epoch [c] Precision -Recall Curve for Specific Dataset at 150 Epoch [d] Precision -Recall Curve for Specific Dataset at 200 Epoch

Figure 4-a shows a precision-recall value of 0.939 at 50 epochs, while Figure 4-b indicates a slightly higher precision-recall value of 0.941 at a higher epoch. Figure 4-c demonstrates the maximum precision-recall performance value of 0.956. In contrast, Figure 4-d illustrates a dramatic decrease in the precision-recall value to 0.919 at 150 epochs.

Imgsz hyperparameter is employed to adjust the dimensions of input images, aiming to enhance both the accuracy and computational efficiency of the YOLOv8 model. During this study, it was observed that configuring the imgsz parameter to 640, as determined during preprocessing, enhanced the model's accuracy in contrast to alternative values. Experimenting with different imgsz values revealed that alternatives resulted in inferior performance compared to 640. Figure 5 represents some of the results obtained in this study.

Figure 5-a demonstrates that an imgsz value of 640 results in a higher precision-recall value of 0.956. Conversely, Figure 5-b shows that using a different imgsz value from the preprocessing size decreases the precision-recall value to 0.944.

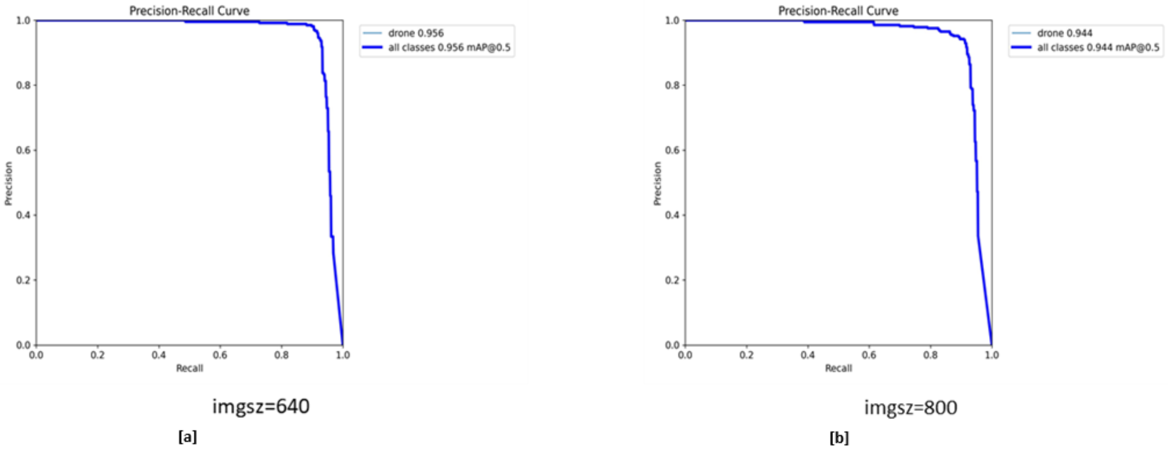

Figure 5. [a] Precision -Recall Curve for Specific Dataset at 640 Image Size [b] Precision -Recall Curve for Specific Dataset at 800 Image Size

Batch parameter is utilized to regulate the quantity of images processed simultaneously. When set to -1, it activates auto-batching, dynamically adjusting the number of images based on GPU memory availability. In our research, setting batch to -1 led to a significant decrease in precision. Hence, the default value of 16 was retained.

Warmup epoch parameter is employed to gradually increase the learning rate during the initial epochs, aiming to stabilize the training process. By doing so, the model operates more consistently from the outset, resulting in more accurate outcomes. In this research, the value was adjusted to 5 from the default of 3, leading to an improvement in the precision-recall curve values.

Numerous hyper-parameters are utilized in YOLOv8, such as momentum, warmup_momentum, weight_decay and others. All remaining hyper-parameters have been retained at their default values.

4.2. Dataset Augmentation

In our study, we employed the YOLOv8 model configured specifically for the Kaggle drone dataset [19]. Pre-processing and augmentation of the dataset were conducted using Roboflow. The efficacy of augmentation techniques varies depending on the dataset, which may consist of real-world images, close-distance shots, long-distance captures, noisy visuals, angled perspectives, and so forth. Various augmentation techniques were applied during our study and their results were observed, the impacts of which are summarized as follows:

Flip augmentation entails augmenting the dataset by symmetrically reflecting the sample images. This technique is aimed at training the model to recognize various aspects of the sample images. In our study, we found that this method did not positively impact the results; therefore, it was not utilized to achieve better outcomes.

Rotation augmentation involves generating new image samples by rotating the original ones by a few degrees. This approach aims to enhance model performance by presenting images to the model from various perspectives. In our research, implementing rotation augmentation alone yielded notably positive outcomes, resulting in an increase in the precision-recall function value. It appears that rotation augmentation is indeed an effective technique for this particular dataset.

Crop augmentation involves augmenting the sample image count by cropping certain sections of the sample images. This approach aims to diversify the aspects of the target that the model learns. However, in our research, employing this augmentation method did not change the results, suggesting the presence of critical components necessary for drone detection within this particular dataset.

Blur augmentation entails creating additional image samples by introducing blur. The purpose is to reduce the level of detail in the images, thereby training the model on more challenging samples. In our study, exclusively employing blur augmentation led to an increase in precision-recall values, indicating compatibility between this augmentation method and the model combination for this dataset.

Gray-scale augmentation involves generating extra images through the conversion of color images into gray-scale versions. This adjustment encourages the model to prioritize fundamental properties of the images. However, in our study, gray-scale augmentation did not change the model performance, suggesting that color variability may not be a critical factor for this dataset.

The evaluation criteria for YOLOv8 performance on various dataset augmentations, precision-recall curves, are shown on Figure 6:

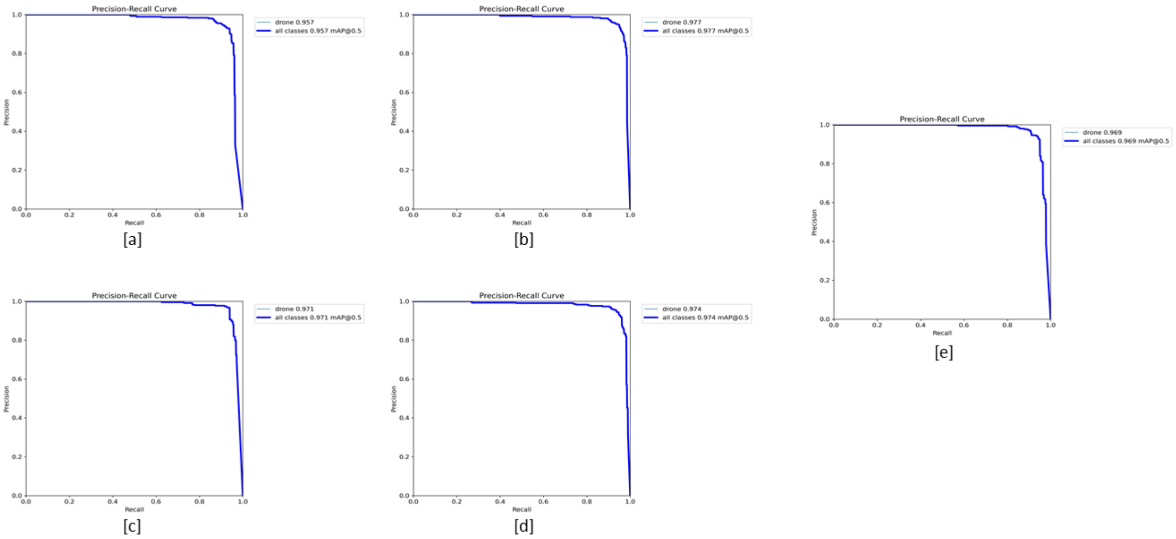

Figure 6. [a] Precision -Recall Curve for Flip Augmented Specific Dataset [b] YOLOv8 Model Precision -Recall Curve for Rotation Augmented Specific Dataset [c] YOLOv8 Model Precision -Recall Curve for Crop Augmented Specific Dataset [d] YOLOv8 Model Precision -Recall Curve for Blurring Augmented Specific Dataset [e] YOLOv8 Model Precision -Recall Curve for Gray-scale Augmented Specific Dataset

Figure 6-a displays the YOLOv8 model's performance on a dataset with flip augmentation, yielding a precision-recall value of 0.957. Figure 6-b presents the model's performance on a dataset with rotation augmentation, achieving a precision-recall value of 0.977. Figure 6-c shows the results for the model on a dataset with crop augmentation, with a precision-recall value of 0.971. Figure 6-d illustrates the model's performance on a dataset with blurring augmentation, also achieving a precision-recall value of 0.971. Finally, Figure 6-e depicts the model's performance on a dataset with gray-scale augmentation, with a precision-recall value of 0.969. Based on these experiments, it was concluded that applying both rotation and blurring augmentation methods to the dataset can enhance model performance.

4.3. Model Performance

To achieve successful outcomes, two primary approaches were employed on both the model and dataset, as detailed in preceding sections: adjusting hyper-parameters and augmenting the specific dataset. These two methods were applied sequentially. Initially, hyper-parameters were adjusted to optimize the results, followed by the application of augmentation techniques to further enhance performance. When hyper-parameter tuning was conducted to achieve the best possible outcomes, the results can be summarized as follows:

Figure 7 is presented to summarize the performance of the YOLOV8 model throughout the entire training process. Based on the results, it can be concluded that selecting 150 epochs as the optimal value for this dataset and model is justified for several reasons. Upon examining the first row, which represents the training performance results, it's evident that the loss values consistently decrease between epochs 100 and 150. Moreover, both precision and recall values stabilize around epoch 150. Similarly, in the second row representing validation performance, mAP values also stabilize around epoch 150. Overall, the results exhibit minor oscillations, which are not critical and can be attributed to the weights used in the training process.

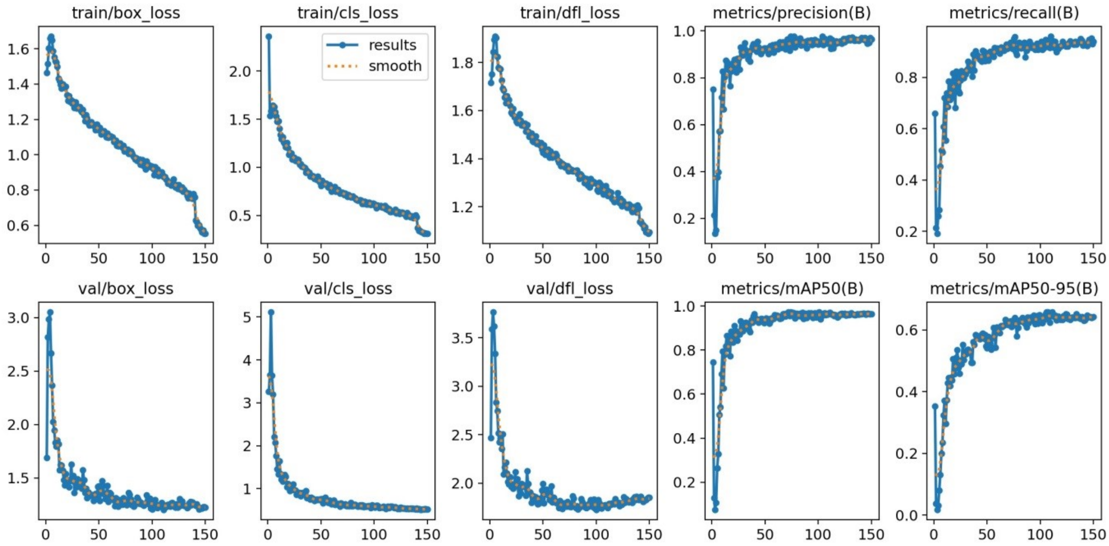

Figure 7. YOLOv8 Model Results For Specific Dataset

For the model with hyper-parameter tuning, additional results such as the Precision curve, Recall curve, F1 score, and PR curve are available and can be displayed below:

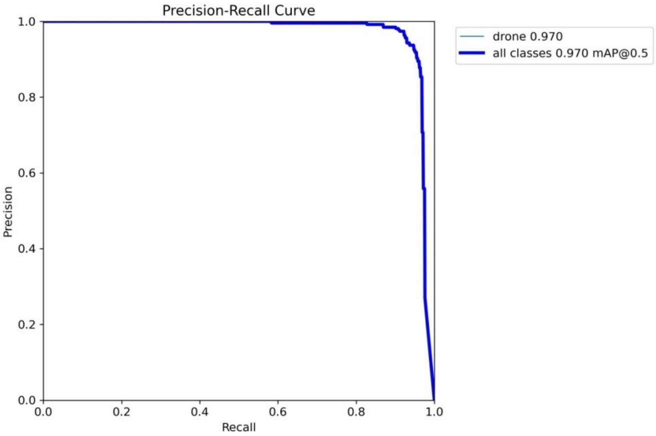

Figure 8. Precision-Recall Curve For Specific Dataset

The Precision-Recall curve value significantly improves with hyper-parameter tuning, reaching a notably high value of 0.970.

Figure 9-a indicates that the F-1 curve reaches its peak value of 0.588 at a specific point. In Figure 9-b, the recall value is indicated as 0.97, whereas in Figure 9-c, the precision value is shown to be 0.832.

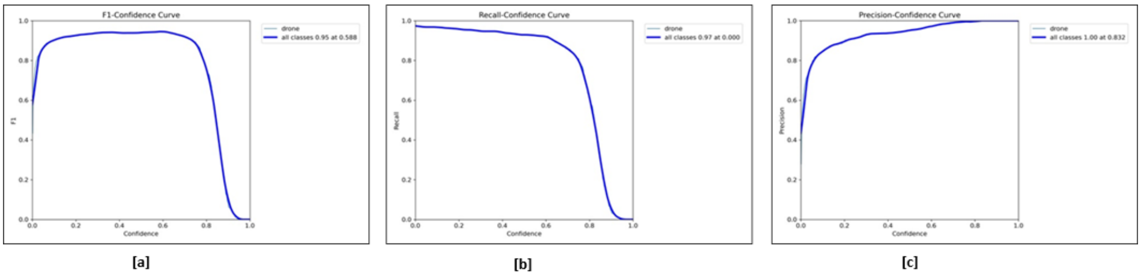

Figure 9. [a] F1-Confidence Curve for Specific Dataset [b] Recall-Confidence Curve for Specific Dataset [c] Precision-Confidence Curve for Specific Dataset

Figure 10 depicts the test outcomes of this method, showcasing the model's ability to identify drones of different sizes and orientations within the test images, with precision values ranging from 0.6 to 0.9.

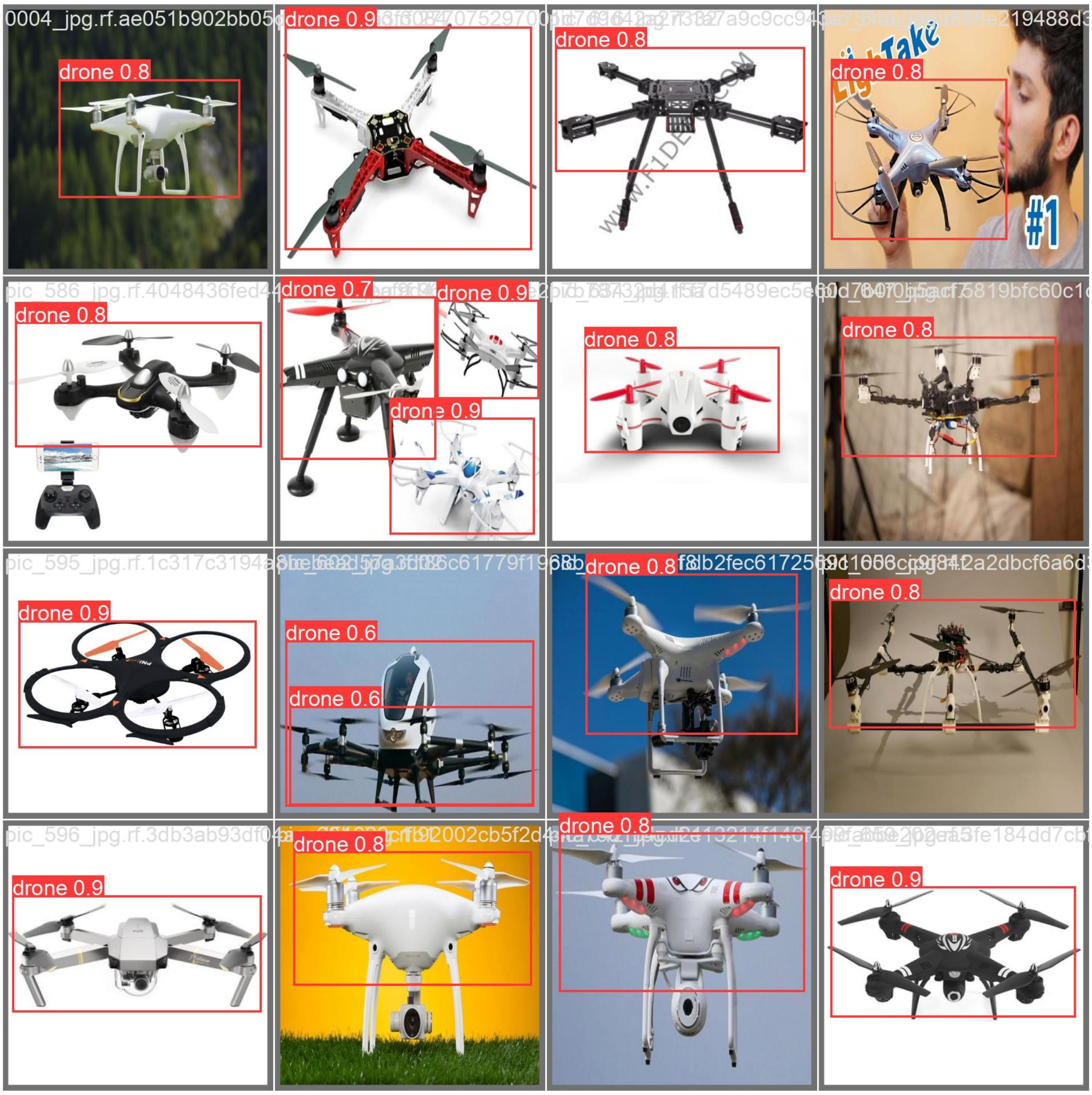

Figure 10. Test Results For Specific Dataset

After hyper-parameter tuning, diverse augmentation methods were applied, and the best among them was selected, resulting in the following (Figure 11) obtained results.

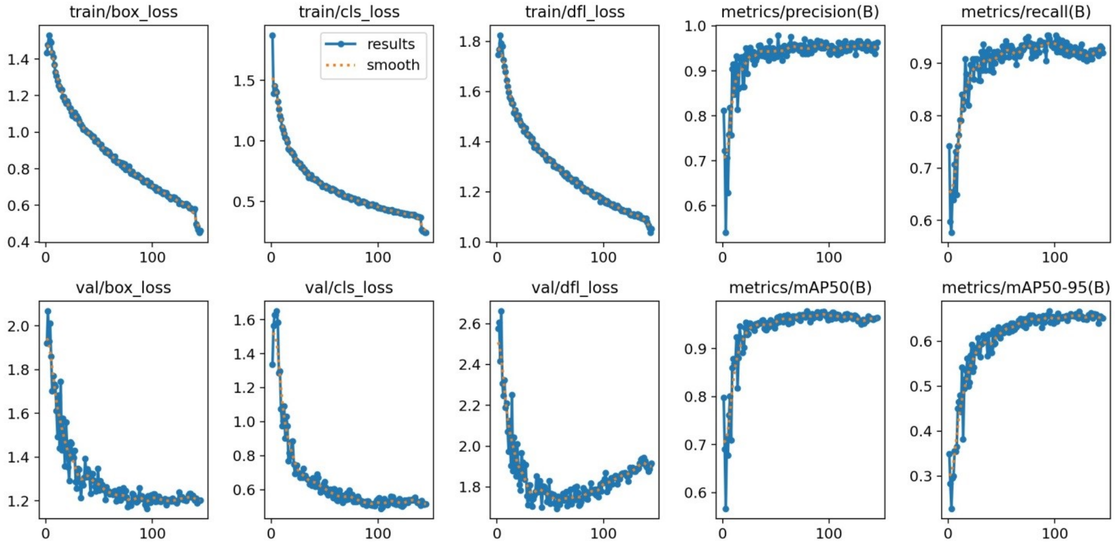

Figure 11. YOLOv8 Model Results For Augmented Dataset

The notable distinction between Figure 7 and Figure 11 lies in the range of the loss values, with Figure 7 exhibiting higher maximum and minimum values. It is evident that larger datasets lead to increased loss values. Another significant impact of augmentation is observed in the precision-recall curve. This metric was enhanced by carefully selecting appropriate augmentation methods tailored to the specific dataset, namely blurring and rotation. These methods were determined through iterative experimentation, as detailed in earlier sections.

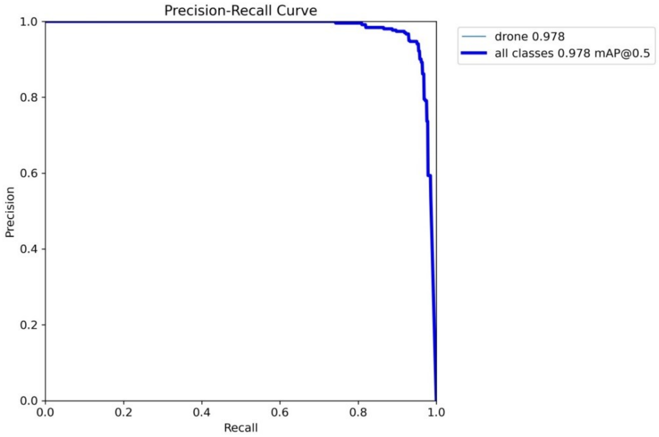

Figure 12. Precision-Recall Curve For Augmented Dataset

The Precision-Recall curve value significantly improves with dataset augmentation reaching a high value of 0.97.

Figure 13-a demonstrates that the F-1 curve achieves its maximum value of 0.236 at a particular point. In Figure 13-b, the recall value is shown to be 0.98, while Figure 13-c displays the precision value as 0.840.

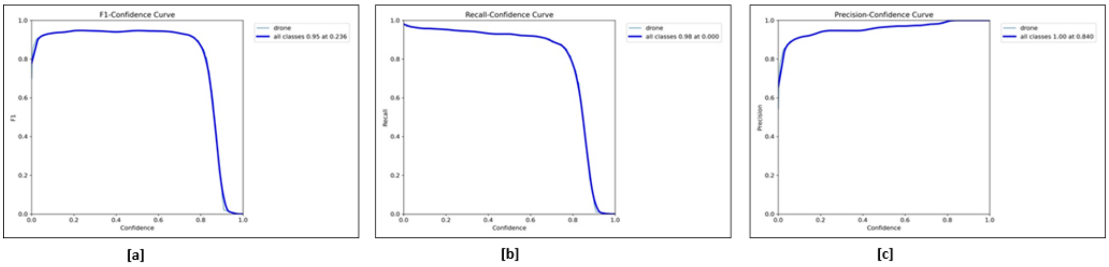

Figure 13. [a] F1-Confidence Curve for Augmented Specific Dataset [b] Recall-Confidence Curve for Augmented Specific Dataset [c] Precision-Confidence Curve for Augmented Specific Dataset

Figure 14 illustrates the test results of this approach, demonstrating the model's capability to detect drones of various sizes and orientations in the test images, with precision values ranging from 0.8 to 0.9.

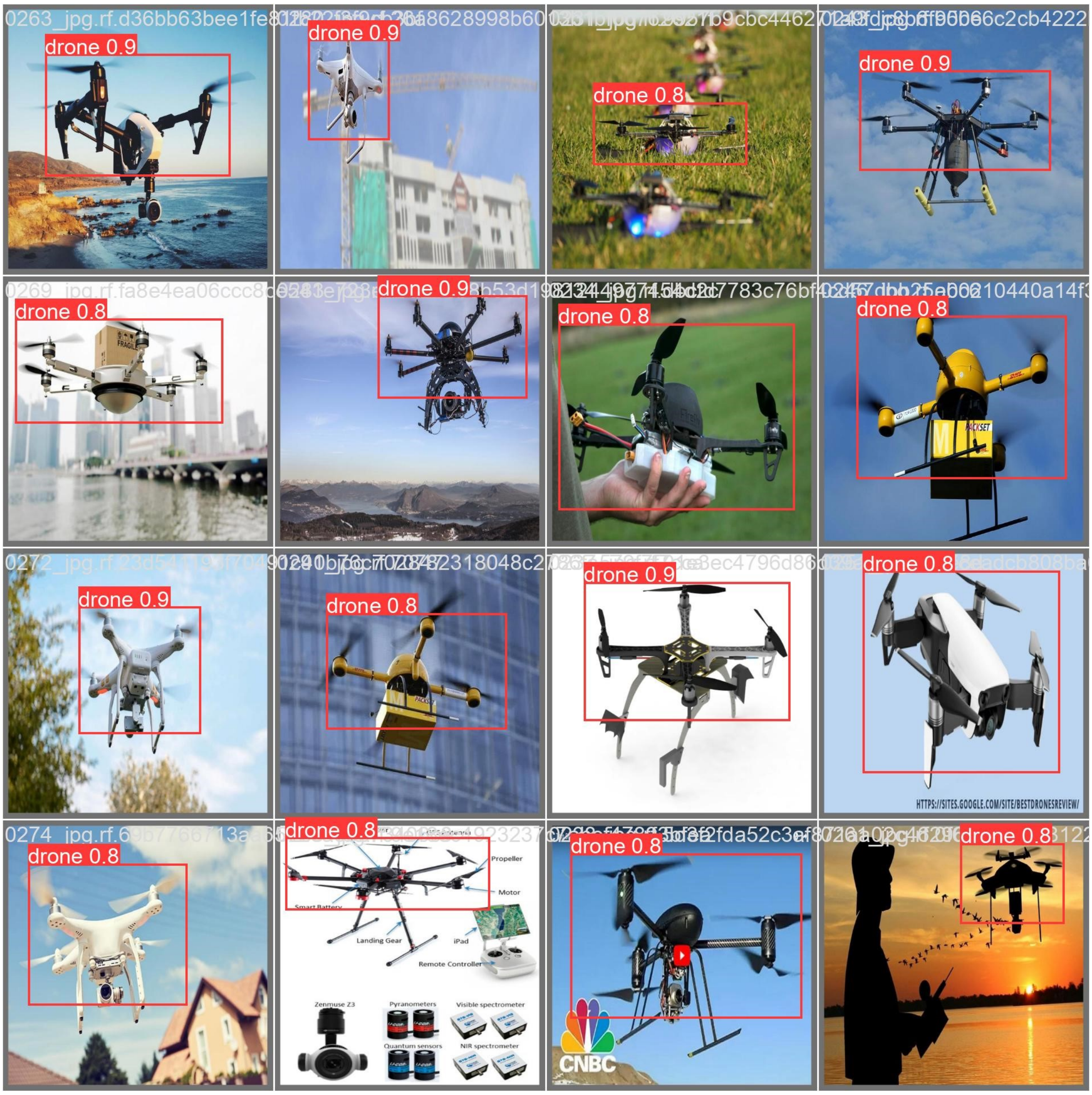

Figure 14. Augmented Dataset Result

4.4. Comparison With Other Models

In this research, the YOLOv8 model was trained on a specific dataset, yielding significant success. To assess the position of YOLOv8 among other models, results from YOLOv4, YOLOv3, tranfer learning applied YOLOv5 and the Mask R-CNN model for the same specific dataset were used [15].

Table 2. Comparison Table

Method	Dataset Size	Precision	Recall	mAP
CNN [14]	712	96%	94%	95%
SVM [14]	712	82%	91%	88%
KNN [14]	712	74%	94%	80%
MaskRCNN [20]	1359	93.6%	89.4%	92.5%
YOLOv3 [15]	1359	92%	70%	78.5%
YOLOv4 [15]	1359	91%	89%	93.8%
YOLOv5 [15] (Transfer Learning)	1359	94.7%	92.5%	94.1%
YOLOv8	1359	95.4%	93.4%	97%
Proposed Model	**3212**	**94.6%**	**96.05%**	**97.8%**

In this study, the performance criteria were determined based on multiple parameters, including the precision-recall curve, precision, recall, and true positive values, which directly influence precision and recall, as mentioned in previous sections. The term "True Positive values" is used for comparision in the title because the performances of other models are compared using this parameter. When examining the True Positive (TP) values for YOLOv8, notably successful results were obtained even without using transfer learning. Additionally, PR curve values for YOLOv8, both with augmented and non-augmented datasets, were determined to be 0.978 and 0.970, respectively, both of which surpass the values obtained by other models.

5. Future Works

In the future, YOLOv8's performance will undergo assessment with a more extensive dataset. This dataset can be broadened by integrating supplementary data categories like land vehicles and marine vessels. Furthermore, the dataset will be enriched by introducing ambiguous images featuring drones, land vehicles, and marine vessels. Additionally, the diversity within these vehicle types will be enhanced to bolster the effectiveness of YOLOv8.

6. Conclusion

This study concludes that YOLOv8 demonstrates high performance and achieves successful results on the specific dataset used. Various augmentation techniques were applied, along with adjustments to hyperparameters, with specific recommendations proposed to enhance the model's performance. It was demonstrated that rotation and blurring augmentation methods were suitable for this dataset, leading to improved model performance. Optimal values were proposed and their effects were showcased. These adjustments were compared with other popular and similar deep-learning models such as YOLOv5, YOLOv4, and YOLOv3, revealing that YOLOv8 outperforms them on the same specific dataset. For future endeavors, exploring the performance of YOLOv8 in drone detection could involve utilizing diverse datasets. Expanding the scope of the dataset to encompass a wider range of scenarios and complexities would allow for more comprehensive training of the model.

Author Contributions: Conceptualization, B.Y. and U.K.; methodology, B.Y. and U.K.; software, B.Y.; validation, B.Y.; formal analysis, B.Y.; investigation, B.Y.; resources, B.Y.; data curation, B.Y.; writing-original draft preparation, B.Y.; writing and editing, B.Y.; reviewing, B.Y. and U.K.; visualization, B.Y.; supervision, U.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset utilized in this research study is publicly available on Kaggle at https://www.kaggle.com/dasmehdixtr/drone-dataset-uav, accessed on 25 April 2024.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

YOLO	You Only Look Once
CNN	Convolutional Neural Networks
UAV	Unmanned Aerial Vehicle
SVM	Support Vector Machine
EMA	Efficient Multiscale Attention

References

1. Mohsan, S.A.H.; Khan, M.A.; Noor, F.; Ullah, I.; Alsharif, M.H. Towards the unmanned aerial vehicles (UAVs): A comprehensive review. *Drones* **2022**, *6*, 147.
2. Niu, R.; Zhi, X.; Jiang, S.; Gong, J.; Zhang, W.; Yu, L. Aircraft Target Detection in Low Signal-to-Noise Ratio Visible Remote Sensing Images. *Remote Sensing* **2023**, *15*, 1971.
3. Sivakumar, M.; Tyj, N.M. A literature survey of unmanned aerial vehicle usage for civil applications. *Journal of Aerospace Technology and Management* **2021**, *13*, e4021.
4. Udeanu, G.; Dobrescu, A.; Oltean, M. Unmanned aerial vehicle in military operations. *Sci. Res. Educ. Air Force* **2016**, *18*, 199–206. doi:10.19062/2247-3173.2016.18.1.26.
5. Pedrozo, S. Swiss military drones and the border space: a critical study of the surveillance exercised by border guards. *Geographica Helvetica* **2017**, *72*, 97–107.
6. Zheng, Z.; Lei, L.; Sun, H.; Kuang, G. A review of remote sensing image object detection algorithms based on deep learning. 2020 IEEE 5th International Conference on Image, Vision and Computing (ICIVC). IEEE, 2020, pp. 34–43.
7. Elsayed, M.; Reda, M.; Mashaly, A.S.; Amein, A.S. Review on real-time drone detection based on visual band electro-optical (EO) sensor. 2021 Tenth International Conference on Intelligent Computing and Information Systems (ICICIS). IEEE, 2021, pp. 57–65.
8. Basak, S.; Rajendran, S.; Pollin, S.; Scheers, B. Combined RF-based drone detection and classification. *IEEE Transactions on Cognitive Communications and Networking* **2021**, *8*, 111–120. doi:10.1109/TCCN.2021.3099114.
9. Liu, B.; Luo, H. An improved Yolov5 for multi-rotor UAV detection. *Electronics* **2022**, *11*, 2330.
10. Coluccia, A.; Fascista, A.; Schumann, A.; Sommer, L.; Fraunhofer.; Ghenescu, M.; Piatrik, T.; Cubber, G.D.; Nalamati, M.; Kapoor, A.; Saqib, M.; Sharma, N.; Blumenstein, M.; Magoulianitis, V.; Ataloglou, D.; Dimou, A.; Zarpalas, D.; Daras, P.; Craye, C.; Ardjoune, S.; Iglesia, D.D.L.; Méndez, M.; Dosil, R.; Gonzalez, I. Drone-vs-Bird Detection Challenge at IEEE AVSS2019. IEEE, 2019.
11. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2017**, *39*, 1137–1149. doi:10.1109/TPAMI.2016.2577031.
12. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556* **2014**.
13. Coluccia, A.; Parisi, G.; Fascista, A. Detection and classification of multirotor drones in radar sensor networks: A review. *Sensors (Switzerland)* **2020**, *20*, 1–22. doi:10.3390/s20154172.
14. Mahdavi, F.; Rajabi, R. Drone Detection Using Convolutional Neural Networks. 6th Iranian Conference on Signal Processing and Intelligent Systems, ICSPIS 2020, Institute of Electrical and Electronics Engineers Inc., 2020. doi:10.1109/ICSPIS51611.2020.9349620.
15. 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 (Switzerland)* **2022**, *11*, 2669. doi:10.3390/electronics11172669.
16. Wang, H.; Yang, H.; Chen, H.; Wang, J.; Zhou, X.; Xu, Y. A Remote Sensing Image Target Detection Algorithm Based on Improved YOLOv8. *Applied Sciences* **2024**, *14*, 1557. doi:10.3390/app14041557.

17. Kumar, B.S.S.; Wani, I.A. Realtime Drone Detection Using YOLOv8 and TensorFlow. JS. *Journal of Engineering Sciences* **2024**, *15*.
18. Terven, J.; Córdova-Esparza, D.M.; Romero-González, J.A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction* **2023**, *5*, 1680–1716.
19. Ozel, M. Drone Dataset (UAV) (available online: https://www.kaggle.com/dasmehdixtr/drone-dataset-uav). WebPage, 12. Accessed on 25 April 2024.
20. Wu, Q.; Feng, D.; Cao, C.; Zeng, X.; Feng, Z.; Wu, J.; Huang, Z. Improved mask r-cnn for aircraft detection in remote sensing images. *Sensors* **2021**, *21*. doi:10.3390/s21082618.
21. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement **2018**.
22. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection **2020**.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.