IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.17, No.3, July 2023, pp. 237~248
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.84785

The Effect of Data Augmentation in Deep Learning with Drone Object Detection

Ariel Yonatan Alin[1], Kusrini, Kusrini*[2], Kumara Ari Yuana[3]
[1]Magister of Informatics Engineering, Universitas Amikom Yogyakarta, Indonesia
[2,3]Dept. of Information Technology, Universitas Amikom Yogyakarta, Indonesia
e-mail: [1]ariel.s2@students.amikom.ac.id, ***[2]kusrini@amikom.ac.id**, [3]kumara.a@amikom.ac.id

Abstrak

Deteksi objek drone merupakan salah satu penerapan utama dari teknologi pemrosesan gambar dan pengenalan pola menggunakan deep learning. Namun, terbatasnya data citra drone yang dapat diakses untuk melatih algoritma pendeteksian menjadi tantangan tersendiri dalam pengembangan teknologi pendeteksian objek drone. Oleh karena itu, banyak penelitian yang telah dilakukan untuk meningkatkan jumlah data citra drone dengan menggunakan teknik augmentasi data. Penelitian ini bertujuan untuk mengevaluasi pengaruh augmentasi data terhadap akurasi deep learning pada deteksi objek drone dengan menggunakan algoritma YOLOv5. Metode yang digunakan dalam penelitian ini meliputi pengumpulan data citra drone, augmentasi data dengan rotate, crop, dan cutout, pelatihan algoritma YOLOv5 dengan dan tanpa augmentasi data, serta pengujian dan analisis hasil pelatihan.

Hasil dari penelitian tersebut menunjukkan bahwa augmentasi data tidak dapat meningkatkan akurasi algoritma YOLOv5 dalam pendeteksian objek drone. Terbukti dengan menurunnya nilai precision dan mAP@0.5 serta nilai recall dan F-1 score yang relatif konstan. Hal ini disebabkan karena augmentasi yang terlalu banyak dapat menyebabkan hilangnya informasi penting pada data dan augmentasi yang tidak tepat dapat menyebabkan noise atau distorsi pada data.

Kata kunci*—Augmentasi data, YOLOv5, Deteksi objek, Deteksi drone.*

Abstract

Drone object detection is one of the main applications of image processing technology and pattern recognition using deep learning. However, the limited drone image data that can be accessed for training detection algorithms is a challenge developing drone object detection technology. Therefore, many studies have been conducted to increase the amount of drone image data using data augmentation techniques. This study aims to evaluate the effect of data augmentation on deep learning accuracy in drone object detection using the YOLOv5 algorithm. The methods used in this research include collecting drone image data, augmenting data with rotate, crop, and cutout, training the YOLOv5 algorithm with and without data augmentation, as well as testing and analyzing training results.

The results of the study show that data augmentation can't improve the accuracy of the YOLOv5 algorithm in drone object detection. Evidenced by the decreasing value of precision and mAP@0.5 and the relatively constant value of recall and F-1 score. This is caused by too much augmentation, which can cause a loss of important information in the data and cause noise or distortion.

Keywords—*Data Augmentation, YOLOv5, Object detection, Drone detection.*

Received May 25th,2023; Revised July 26th, 2023; Accepted July 29th, 2023

1. INTRODUCTION

The drone market is spreading rapidly. Unit sales will grow from 828K in 2021 to nearly 1.4 million in 2026 at a CAGR of 10.6 [1]. The large drone market carries greater risks. Drones can be misused for dangerous purposes, such as carrying bombs, buying and selling drugs, and entering restricted areas. According to the Ministry of Transportation of the Republic of Indonesia, the prohibited areas for drone flights include areas designated for government or civil flights, areas within a fifteen-kilometer radius of aircraft runways, flights exceeding an altitude of 150 meters, controlled airspace, and areas where air traffic control is in effect [2].

The primary issue with drone operations close to airports and other air-controlled areas is the collision risk posed by crewed aircraft and drones, which increases the possibility of material and human losses. According to UK government tests, a 400 g drone can shatter a helicopter's windscreen, while a 2 kg drone can seriously damage the windscreen of a passenger jet [3]. While less expensive UAV models barely have enough power to fly for half an hour, more expensive models can maintain flight for several hours. Therefore, the entire airport may be shut down for safety whenever an unauthorized drone is discovered near an airport and its facilities, a runway, or even the security perimeter [4]. It is important to control illegal drones, one way of which is by detecting drones that enter restricted areas. However, the obstacle experienced is the difficulty of detecting small drones. Small drones emit very weak electromagnetic waves, making them hard to detect by radar. One detection method that can be used is deep learning object detection [5].

The accuracy of drone object detection relies heavily on the availability of high-quality training data for deep learning algorithms. However, collecting sufficient drone image data for training can present challenges in terms of cost, time, and complexity. Researchers have explored various techniques, including data augmentation, to address this issue to expand the available training data. Data augmentation involves applying diverse transformations, such as rotation, translation, scaling, and flipping, to the original data, thereby generating additional data [6].

Existing domain-specific image object detectors typically fall into one of two categories, the most prevalent of which is the Faster R-CNN two-stage detector [7]. The other is a one-stage detector, like the SSD or YOLO [8]. One-stage detectors achieve high inference speed, while two-stage detectors achieve high localization and object recognition accuracy [9].

In this study, we investigate the effect of data augmentation on deep learning in drone object detection using the YOLOv5 algorithm. YOLOv5 is a state-of-the-art deep learning algorithm that has demonstrated excellent performance in object detection tasks. We aim to evaluate the impact of data augmentation on the accuracy of YOLOv5 in detecting drone objects and identify the most effective data augmentation techniques for improving detection performance.

Based on previous research, drone detection is a research topic that has been developed by many researchers from around the world. The dataset used also varies from images to videos. Research conducted by Coluccia et al., entitled Drone vs. Birds: Deep Learning Algorithms and Results from Big Challenges, the goal is to detect one or more drones appearing at multiple points in time in the video sequence where birds and other distractor objects may also be present, along with movement in the background or foreground [10]. The challenge in drone object detection is that the ability to detect objects is difficult to recognize drone objects if the size of the drone is small or far away.

Research conducted by Subroto Singha et al. Detecting drones automatically using YOLOv4, the reason for using the YOLOv4 algorithm is because it can detect in real-time and has a high level of precision when compared to Region-Based Convolutional Neural Network (R-CNN) and Single-Shot Multi-box Detector (SSD). Even though the detection process is fast, the mean average precision (mAP) of YOLOv4 is still at 74.36% [5]. At the time the author will conduct this research, the latest release of YOLO has been released, which is YOLOv5, so we are try to conduct drone detection research using YOLOv5 and performing various data augmentation processes to see how the performance of YOLOv5 compares to each augmentation process.

Research conducted by Shorten, C., & Khoshgoftaar, entitled "A survey on image data augmentation for deep learning". The author of this article identified that the problem of unbalanced data and a limited amount of training data is often a challenge in training deep learning models. Therefore, data augmentation techniques can help improve the quality and quantity of training data, thereby improving model performance. This article discusses various data augmentation techniques that can be used in deep learning, especially for image classification. The author presents various data augmentation techniques, such as flip, rotation, brightness adjustment, and others, and provides an evaluation of the performance of each technique [6]. In addition, data augmentation techniques are used to improve the quality and quantity of training data. For example, the CutMix technique, in which two different images are combined by cutting and joining parts of the images. This article also discusses the use of mosaic augmentation technology and MixUp augmentation technology to increase the variation of training data and reduce overfitting [9]. The conclusions from previous studies are: To improve object detection abilities, it is necessary to develop a method, one of which is data augmentation.

One comparison study about augmentation data type was carried out by [11] on the CIFAR-10 and ImageNet datasets, comparing GANs, WGANs, cropping, rotation, flipping, translation, jittering (PCA and color), and adding noise. They discovered that WGAN, flipping, cropping, and rotation outperformed the competition. Another comparison by [12] compared six different data augmentation techniques, including skew, shear, random erasing, gaussian distortion, and random distortion, using the ResNet model over a portion of the CIFAR-10 dataset. They discovered that methodical augmentation could raise accuracy to 95.85 percent, an increase of +2.83 percent. Additionally, their findings indicated that applying standard augmentation is less effective than injecting it after the initial learning phase. Additionally, the performance of various data augmentation techniques can be evaluated concerning various design choices.

2. METHODS

Data augmentation increases the available training data, enabling machine learning models to make more accurate predictions. It also helps mitigate the risk of overfitting by introducing variation during training. Another advantage of data augmentation is its ability to generate additional data at a low cost.

2.1 Data Augmentation

Data augmentation is a strategy that allows practitioners to substantially enhance the diversity of available training data for models without the need to collect new data. Techniques such as cropping, rotating, and cutout are frequently employed in data augmentation to train large neural networks effectively.

- Cropping

Cropping may only sometimes be successful in enhancing the data. In some cases, cropped images may need more details to contribute significantly to data augmentation or may divert attention from the original problem. Therefore, it is important to consider alternative approaches to effectively combine the augmented data with the original data [13]. To address this concern, we can formulate a cropping technique that ensures the inclusion of important details in the augmented data. One approach is to generate cropped images using a range of zoom levels, such as a minimum of 0% zoom and a maximum of 20% zoom. By including a zoom-in capability, we can capture a diverse range of image content and minimize the risk of losing crucial details during the data augmentation process.

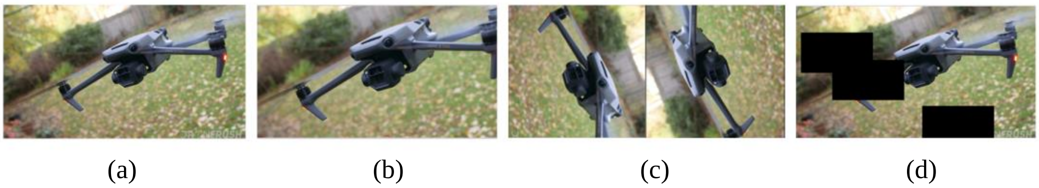

Figure 1 Data Augmentation Type (a) Original (b) Crop (c) Rotate 90° (d) Cutout

- Rotate

In this step, the augmented image can be further enhanced by applying random transposition and rotation. This can involve randomly flipping or mirroring the image, as well as rotating it by 90 degrees in any direction. By incorporating these transformations, the augmentation pipeline becomes more intricate and allows for greater flexibility in testing various augmentation strategies. The implementation of this detailed augmentation pipeline can be expressed concisely in a code snippet [14].

- Cutout

The cutout regularization method is an effective technique for convolutional neural networks (CNNs) that introduces new samples to the dataset by partially obscuring input images. It achieves this by removing contiguous portions of the images. Cutout can be considered as an extension of dropout in the input space, with the incorporation of a spatial prior. This approach bears resemblance to how CNNs utilize spatial before surpassing feed-forward networks in performance when handling image data [15].

However, many approaches employed in training neural networks tend to utilize only basic types of augmentation. Meanwhile, extensive research has been conducted on exploring and optimizing neural network architectures [16]. An example of image data augmentation can be observed in **Error! Reference source not found.**

2.2 Convolutional Neural Network

A Convolutional Neural Network (CNN) is an artificial neural network that can be visualized as a collection of neurons arranged in a graph without any loops. A distinctive characteristic of CNNs is the presence of hidden layers that connect with subsets of neurons from the previous layer. This connectivity enables CNNs to learn features implicitly. A typical CNN consists of four layers. The first layer focuses on identifying edges or color variations, allowing the network to detect basic visual patterns. The second layer builds upon the initial layer's findings to recognize shapes, enabling the network to discern more complex structures. The third layer specializes in studying localized parts of objects, enabling the network to understand different components of an object. Finally, the fourth layer is responsible for identifying complete objects, utilizing the knowledge gained from the previous layers to make accurate object classifications. By progressively analyzing visual information through these layers, CNNs can effectively extract and learn hierarchical representations of features, leading to robust and accurate object recognition.

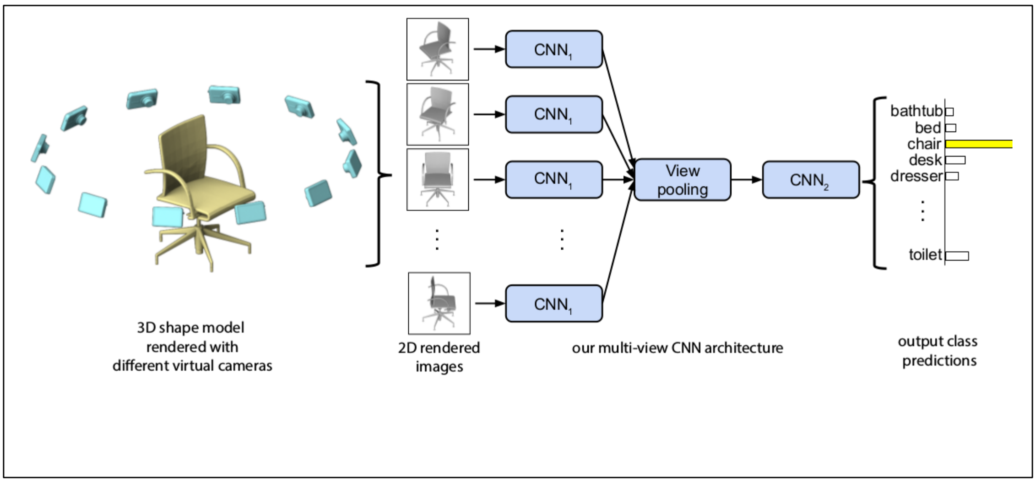

Figure 2. Convolutional Neural Network Architecture

2.3 *YOLOv5 Architecture*

In this study, the authors will incorporate data augmentation into the YOLOv5 algorithm, also known as "You Only Look Once Version 5". YOLOv5 is a target recognition algorithm introduced by Glenn Jocher in 2020. It is a single-stage approach that focuses on efficient and accurate object detection tasks [17]. Based on variations in network depth and width, the YOLOv5 algorithm can be categorized into four different model versions: YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x. Each version offers distinct trade-offs between calculation speed and average precision. The YOLOv5s network exhibits the fastest calculation speed among the four versions but has a relatively lower average precision. On the other hand, the YOLOv5x network demonstrates the opposite characteristic, with the highest average precision but a slower calculation speed. The general architecture of YOLOv5 can be summarized as follows [18] :

- Backbone: YOLOv5 employs a convolutional neural network (CNN) as its backbone, responsible for feature extraction from input images. The chosen backbone architecture in YOLOv5 is a variant of the CSPNet (Cross-Stage Partial Network). This architecture enhances the model's efficiency and accuracy by leveraging cross-stage partial connections.
- Neck: YOLOv5 incorporates a neck component, which includes additional convolutional layers inserted between the backbone and the detection head. This neck component plays a crucial role in refining the extracted features from the backbone network and capturing additional context information. By introducing these intermediate layers, YOLOv5 can further enhance the representation of features and improve the model's ability to detect objects accurately.
- Detection Head: The detection head serves as the final component of the YOLOv5 architecture and is responsible for generating predictions for bounding boxes and class probabilities. YOLOv5 utilizes a feature pyramid network (FPN) as its detection head. This FPN integrates features from multiple scales to effectively detect objects of varying sizes. By leveraging the combined information from different scales, YOLOv5 improves its capability to accurately identify and classify objects in the input image.

The YOLOv5 architecture consisting of the Backbone, Neck, and Head is shown in **Error! Reference source not found.**.

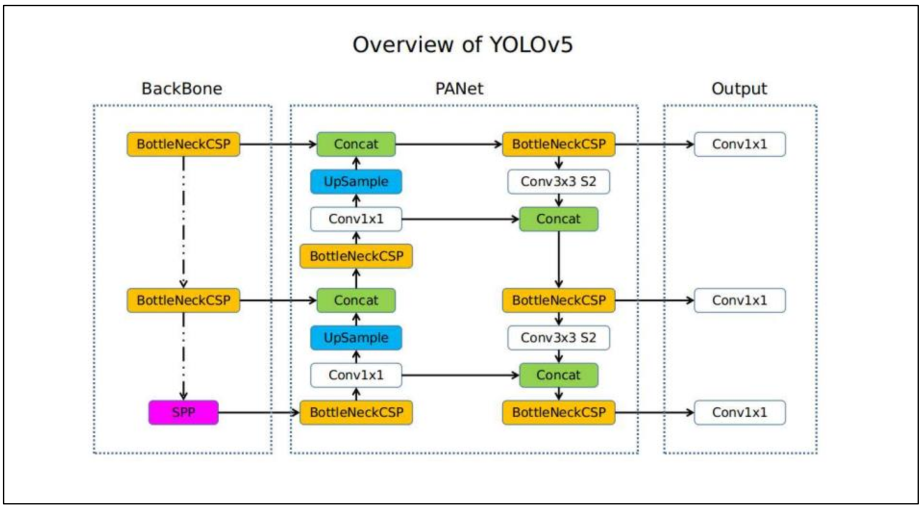

Figure 3. YOLOv5 Architecture

2.4 *TensorFlow 2.0*

TensorFlow 2.0, a significant update of the widely-used deep learning framework TensorFlow, was introduced by Google in 2019 [19]. This update brings forth numerous important features and enhancements when compared to its predecessor, TensorFlow 1, aiming to simplify and streamline the process of developing deep learning models. TensorFlow 2.0 is constructed on top of the Python programming language and employs data flow diagrams to perform numerical computations and generate models.

2.5 *Roboflow*

Roboflow is a tool designed to simplify computer vision tasks within the deep learning domain. It aids developers in constructing computer vision applications by offering various capabilities. These include providing annotations for datasets, pre-processing datasets, merging projects or datasets, verifying dataset conditions, exporting datasets, and training models. With these functionalities, Roboflow streamlines the workflow of developers working on computer vision projects [20].

2.6 *Research Approach*

The research conducted in this study can be categorized as quantitative research. It involves testing various data augmentation scenarios and comparing the accuracy results of each augmentation process. The main objective of this research is verification, aiming to test the validity of existing hypotheses or theories. By conducting experiments and testing each augmentation process, the study investigates whether they support or refute the hypothesis. The research follows a deductive approach, starting from an existing hypothesis or theory and then empirically testing its validity using data. The study aims to determine the extent to which data augmentation can impact the accuracy of results.

2.7 *Dataset*

The data utilized in this study consists of primary data collected indirectly. It comprises images of drones obtained from the Kaggle platform. These images were originally collected by Mehdi Zel for the UAV competition with the intention of training UAVs for navigation and evasion of other UAVs. The dataset consists of 1359 labeled photos of drones. This dataset is compatible with training models such as Darknet (YOLO), TensorFlow, and PyTorch. It can be accessed through the following link: https://github.com/dasmehdix/drone-dataset. An example image from the drone dataset is depicted in Figure3.

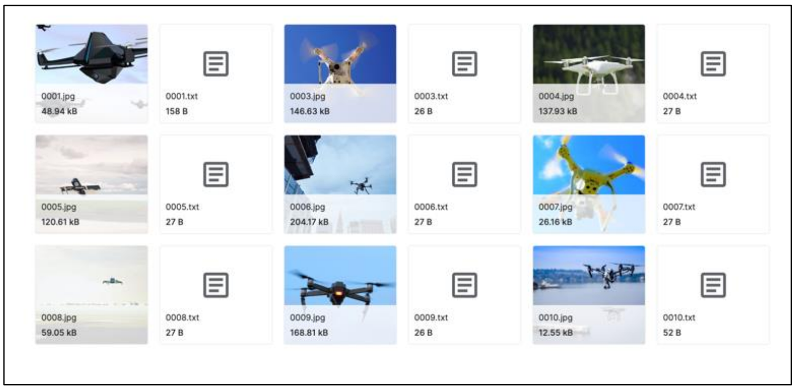

Figure 3. Dataset

2.8 Research Flow

The research process begins with the collection of the dataset. Subsequently, the dataset goes through the labeling or image annotation process, where bounding boxes are created around the objects of interest, in this case, drones. Once all the datasets have been labeled, the pre-processing stage follows, involving Auto-Orient and Resize operations. During the resizing process, the images are adjusted to a size of 640 x 640 pixels. After pre-processing, the data undergo augmentation, which enhances the diversity of the dataset and promotes better generalization of the model's performance.

The next step involves dividing the dataset into three subsets: 70% for training, 20% for validation, and 10% for testing. This partitioning ensures separate sets of data for training, fine-tuning, and evaluating the model's performance. Labeling, pre-processing, and dataset export stages are carried out using the Roboflow website. Once labeling and pre-processing is complete, the dataset is exported to YOLOv5 PyTorch and integrated into the code within Google Colab for further analysis and model training. See Figure 4.

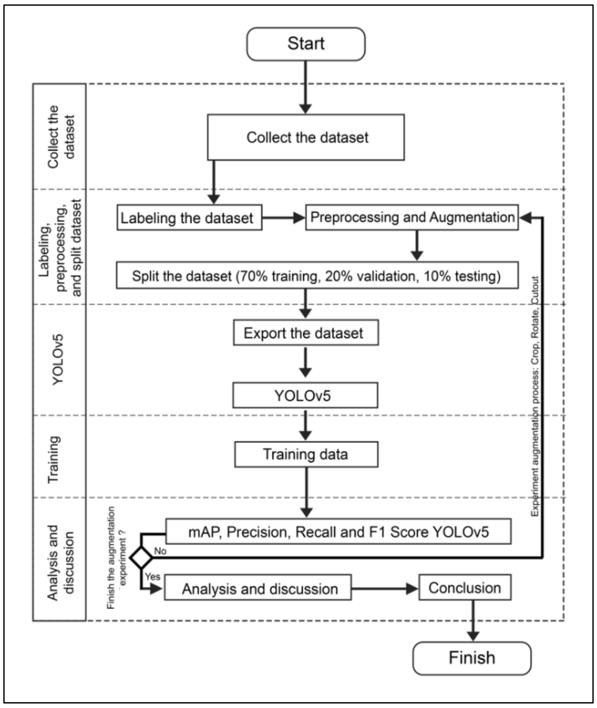
Start
Collect the dataset
Collect the dataset
Labeling, preprocessing, and split dataset
Labeling the dataset
Preprocessing and Augmentation
Split the dataset (70% training, 20% validation, 10% testing)
YOLOv5
Export the dataset
YOLOv5
Training
Training data
Experiment augmentation process: Crop, Rotate, Cutout
Analysis and discussion
mAP, Precision, Recall and F1 Score YOLOv5
Finish the augmentation experiment ?
No
Yes
Analysis and discussion
Conclusion
Finish

Figure 4. Research Flow

2.9 Roboflow and YOLOv5

The YOLOv5 workflow on Google Colab starts by duplicating the YOLOv5 program into a personal Google Colab environment. Then, the previously exported dataset is imported and incorporated into the program's code for loading and extraction. After importing the dataset, the training process commences using YOLOv5s. The training runs for a specified number of epochs, typically 100, to optimize the model's performance. The program also displays validation data during the training process to assess the model's progress and performance. Once the training is complete, the program can be used for object detection. It takes an input image or video and applies the trained YOLOv5 model to detect objects within the given data. The provided information outlines a concise process flow in Roboflow and YOLOv5, demonstrating the steps involved in training and utilizing the YOLOv5 model for object detection. See **Error! Reference source not found.**.

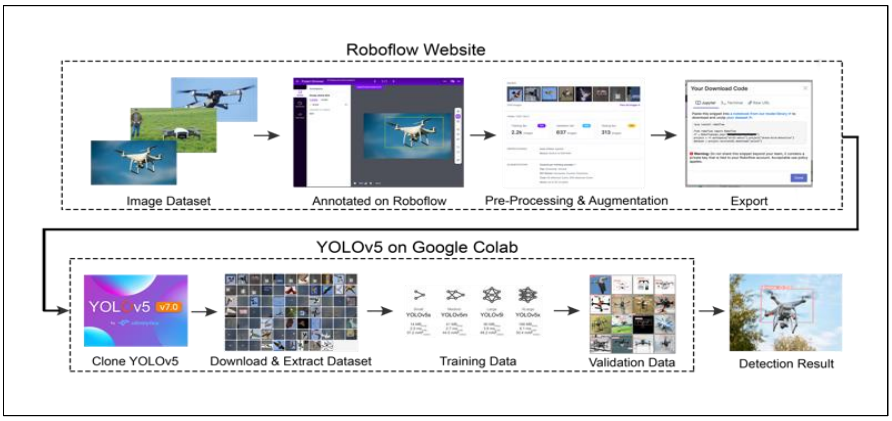

Figure 6. Research Flow on Roboflow and YOLOv5 on Google Colab

2.10. Evaluation

Several model evaluation metrics are commonly used in object detection, such as:

- Mean Average Precision (mAP)

  mAP (mean Average Precision) is one of the most commonly used metrics for evaluating object detection models. It measures the model's ability to accurately recognize objects in images or videos by considering both precision and recall. The mAP score ranges from 0 to 1, with higher values indicating better model performance.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{1}$$

- Precision

  Precision measures how many objects the model correctly detects. Precision has a value between 0 and 1, where the higher the precision value, the more accurate the model is in detecting objects.

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

- Recall

  Recall measures how many objects the model has successfully detected from all the objects that should have been detected. Recall also has a value between 0 and 1, where the higher the recall value, the more objects the model has successfully detected.

$$Recall = \frac{TP}{(TP+FN)} \tag{3}$$

- F1 Score

  The F1 score is a model evaluation metric that combines precision and recall. The F1 score has a value between 0 and 1, where the higher the F1 score, the better the model performance.

$$F1\ Score = \frac{2*(Recall * Precision)}{(Recall + Precision)} \tag{4}$$

3. RESULTS AND DISCUSSION

3.1 Experimental

To test the effectiveness of various data augmentation methods, we ran 15 experiments on YOLOv5. Since the original images are too large to fit in a GPU memory entirely, we resized them to 640 × 640 pixels during training. Experiments were run with 25 epochs, 50 epochs, and 100 epochs, each containing 5 types of data augmentation:

- No augmentation: No changes to the image were made.
- Rotate: image rotated 90° left and right.
- Crop: the image was cropped minimum of 0% zoom and a maximum of 20% zoom.
- Cutout: image cutout 3 boxes with 20% each size.
- Combined augmentation: this kind of augmentation is combined from rotate, crop, and cutout.

3.2 Visualization

Image visualization refers to the process of creating a graphical representation of an image or set of images. The goal of image visualization is to help users better understand and analyze the content of images. The following is a visualization of the data augmentation rotate 90°, crop, and cutout. See Figure 5.

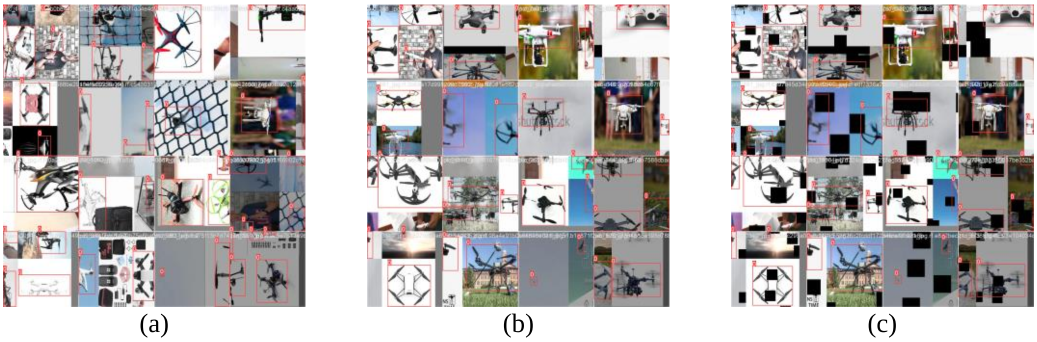
(a) (b) (c)

Figure 5. Visualization Image (a) Rotate 90° (b) Crop (c) Cutout

The output is Precision, Recall, F1 Score, and mAP values. The experimental results are tabulated in the following table.

Table 1. Result in 25 epochs

Augmentation type	**Score**			
	Precision	***Recall***	***F1 Score***	***mAP@0.5***
No Augmentation	0.91	0.96	0.93	0.948
Rotate	0.886	0.97	0.95	0.954
Crop	0.952	0.96	0.94	0.953
Cutout	0.951	0.96	0.93	0.950
Combined	0.882	0.96	0.92	0.943

Table 2. Result in 50 epochs

Augmentation type	**Score**			
	Precision	***Recall***	***F1 Score***	***mAP@ 0.5***
No Augmentation	0.956	0.96	0.94	0.949
Rotate	0.948	0.96	0.94	0.952
Crop	0.957	0.94	0.93	0.949
Cutout	0.957	0.94	0.93	0.949
Combined	0.897	0.96	0.93	0.936

Table 3. Result in 100 epochs

Augmentation type	**Score**			
	Precision	***Recall***	***F1 Score***	***mAP@ 0.5***
No Augmentation	0.986	0.95	0.94	0.953
Rotate	0.954	0.96	0.94	0.956
Crop	0.983	0.93	0.92	0.933
Cutout	0.955	0.94	0.93	0.939
Combined	0.959	0.94	0.95	0.943

In the table, it can be seen that combining the types of augmentation has an impact on reducing accuracy. In the 25 epoch table, it can be seen that the precision value decreased by 0.028, the recall remained the same, the F1-Score decreased by 0.01, and mAP@0.5 decreased by 0.005 compared to without augmentation.

Research conducted by [12] discovered that adding multiple single augmentations to the original dataset is the most effective augmentation strategy, increasing the accuracy by +2,36% to 95,85%. Random and Gaussian distortion are the worst types of augmentation tested, causing changes in the accuracy of -0.15% and +0.05%, respectively. This highlights the significance of the augmentation choice and is caused by the augmented images not accurately representing the original class.

4. CONCLUSIONS

We are currently investigating an issue related to small object detection. Our findings suggest that the inadequate representation of small objects in the training data contributes to the low average precision for these objects. To address this issue, we propose using data augmentation techniques. However, it is important to note that data augmentation techniques may not always be successful in overcoming this problem. This is supported by the observed decrease in precision and mAP@0.5 values, while the recall and F-1 scores remain relatively constant.

This is caused by too much augmentation, which can result in the loss of important information in the data. The more variations that are added to the data, the less likely the original object appears in the data variations. If a model is trained on excessively augmented data, its ability to detect the original object in the non-augmented data may suffer. Improper augmentation can also introduce noise or distortion in the data. If the augmentation used is inappropriate or too extreme, it can produce noise or distortion in the data, which can confuse the model. The model may find it difficult to distinguish the original object from the data with noise or distortion. Additionally, augmentation can lead to overfitting of the training data. Suppose the model relies too heavily on data augmentation during training. In that case, it will primarily learn to recognize objects based on the variety of augmented data provided and may struggle to learn the features of the original object. Consequently, the model can become overly specific to the variations in the training data and may need help to recognize objects in unaugmented data.

By choosing the right data augmentation, it is expected to increase accuracy. Object detection with good accuracy can replace human work, in which object detection can be placed in restricted areas for drone flights. The prohibited areas for drone flights include areas designated for government or civil flights, areas within a fifteen-kilometer radius of aircraft runways, flights exceeding 150 meters, controlled airspace, and areas where air traffic control is in effect.

REFERENCES

[1] L. Kapustina, N. Izakova, E. Makovkina, and M. Khmelkov, “The global drone market: main development trends,” *SHS Web of Conferences*, vol. 129, p. 11004, 2021, doi: 10.1051/shsconf/202112911004.

[2] Volunteers, “Drone Laws in Indonesia [Updated March 29, 2022],” 2022. https://drone-laws.com/drone-laws-in-indonesia/#Agencies_Responsible_for_regulating_drones_in_the_Republic_of_Indonesia (accessed May 30, 2022).

[3] D. Jones, “Small Remotely Piloted Aircraft Systems (drones),” 2017, Accessed: Jul. 28, 2023. [Online]. Available: http://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/

[4] G. Lykou, D. Moustakas, and D. Gritzalis, “Defending airports from UAS: A survey on cyber-attacks and counter-drone sensing technologies,” *Sensors (Switzerland)*, vol. 20, no. 12. MDPI AG, pp. 1–35, Jun. 01, 2020. doi:10.3390/s20123537.

[5] S. Singha and B. Aydin, “Automated drone detection using YOLOv4,” *Drones*, vol. 5, no. 3, 2021, doi:10.3390/drones5030095.

[6] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J Big Data*, vol. 6, no. 1, Dec. 2019, doi:10.1186/s40537-019-0197-0.

[7] Y. Zhou, S. Wen, D. Wang, J. Mu, and I. Richard, “Object detection in autonomous driving scenarios based on an improved faster-RCNN,” *Applied Sciences (Switzerland)*, vol. 11, no. 24, Dec. 2021, doi: 10.3390/app112411630.

[8] W. Wu *et al.*, “Application of local fully Convolutional Neural Network combined with YOLO v5 algorithm in small target detection of remote sensing image,” *PLoS One*, vol. 16, no. 10 October, Oct. 2021, doi: 10.1371/journal.pone.0259283.

[9] L. Jiao *et al.*, “A survey of deep learning-based object detection,” *IEEE Access*, vol. 7, pp. 128837–128868, 2019, doi:10.1109/ACCESS.2019.2939201.

[10] A. Coluccia *et al.*, “Drone vs. Bird detection: Deep learning algorithms and results from a grand challenge,” *Sensors*, vol. 21, no. 8, pp. 1–27, 2021, doi:10.3390/s21082824.

[11] J. Shijie, W. Ping, J. Peiyi, and H. Siping, “Research on Data Augmentation for Image Classification Based on Convolution Neural Networks.”

[12] S. O’gara and K. Mcguinness, “Comparing Data Augmentation Strategies for Deep Image Comparing Data Augmentation Strategies for Deep Image Classification Classification Comparing Data Augmentation Strategies for Deep Image Classification”, doi: 10.21427/148b-ar75.

[13] S. Park, S. Baek Lee, and J. Park, “Data augmentation method for improving the accuracy of human pose estimation with cropped images,” *Pattern Recognit Lett*, vol. 136, pp. 244–250, Aug. 2020, doi: 10.1016/j.patrec.2020.06.015.

[14] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, “Albumentations: Fast and flexible image augmentations,” *Information (Switzerland)*, vol. 11, no. 2, Feb. 2020, doi: 10.3390/info11020125.

[15] T. DeVries and G. W. Taylor, “Improved Regularization of Convolutional Neural Networks with Cutout,” Aug. 2017, [Online]. Available: http://arxiv.org/abs/1708.04552

[16] D. Ho, E. Liang, and R. Liaw, “1000x Faster Data Augmentation – The Berkeley Artificial Intelligence Research Blog,” Jun. 07, 2019. https://bair.berkeley.edu/blog/2019/06/07/data_aug/ (accessed Apr. 24, 2023).

[17] G. Jocher *et al.*, “ultralytics/yolov5: v4.0 - nn.SiLU() activations, Weights & Biases logging, PyTorch Hub integration,” Jan. 2021, doi: 10.5281/ZENODO.4418161.

[18] Z. Chen *et al.*, “Plant Disease Recognition Model Based on Improved YOLOv5,” *Agronomy*, vol. 12, no. 2, 2022, doi: 10.3390/agronomy12020365.

[19] A. Biswal, “What is TensorFlow 2.0 [The Best Guide to Understand TensorFlow 2.0],” *Simplilearn*, Mar. 28, 2022. https://www.simplilearn.com/tutorials/deep-learning-tutorial/tensorflow-2 (accessed Dec. 05, 2022).

[20] S. Theophilus, “Roboflow: Converting Annotations for Object Detection | by Samuel Theophilus | Analytics Vidhya | Medium,” *Medium*, Sep. 03, 2021. https://medium.com/analytics-vidhya/converting-annotations-for-object-detection-using-roboflow-5d0760bd5871 (accessed Dec. 05, 2022).