International Journal of Aviation Science and Engineering
e-ISSN: 2715-6958 p-ISSN: 2721-5342

AVIA

Original Article

Development of Anti-UAV System Using Visual Artificial Intelligence

Javensius Sembiring [1,*], Dimas Prianggoro [1], Rizal Adi Saputra [2], Tarkono [2], Khairul Ummah [1]

[1] Faculty of Mechanical and Aerospace Engineering, Institut Teknologi Bandung. Labtek II, Jl. Ganesa 10, Bandung, 40132 Indonesia
[2] Department of Mechanical Engineering, Faculty Engineering, Universitas Lampung, Jl. Prof. Soemantri Brojonegoro no. 1, Bandar Lampung 35143 Indonesia
* Correspondence: javen.sembiring@itb.ac.id

Received: 10 March 2023; Accepted: 30 May 2023; Published: 12 June 2023

Unmanned Aerial Vehicles (UAV) was first developed as a tool for military purposes. Due to the rapid growth in technology, UAVs are now used in various applications including civil needs. Of course, there are consequences for this where UAVs can be misused by irresponsible parties. One example is the use of UAVs in airport fields which can disrupt the airport operations and possibly become a serious threat towards security and safety of flights in the airport. This paper will discuss the artificial intelligence (AI) modeling to detect UAVs. This AI modeling is the first step in designing counter unmanned aerial system (C-UAS). UAV detection will use deep learning using YOLOv4 (single-stage detection) for optimal detection speed and accuracy. There are a total of 500 image data processed and used in two AI modeling experiments in this study. Gaussian blur filter is used to create dataset variations so that the training can be processed more efficiently and the model can detect better. The results shows that the training dataset that has been processed with gaussian blur (filtered dataset) increases the AI model's detection performance in rainy and clear conditions. Therefore, the model trained using filtered datasets is more suitable for use in detecting UAV objects in anti-UAV systems.

Keywords: anti-UAV system, artificial intelligence, deep learning, YOLOv4, counter unmanned aerial system

---

1. Introduction

In December 2018, an unauthorized drone entered Gatwick Airport's Airspace which prompt to the shutdown of the airport. Similar incident occurred in Frankfurt Airport on May 2019 [1]. According to Indonesian Department of Transportation, Indonesia also experiences several unauthorized drone operations in Indonesian Airports. Such incidents can disturb flight operation and be detrimental to the airport. Undetected UAV have high risk of mid-air collision with other airspace user, such as commercial aircraft, private jets, and military aircraft [2]. Research by Crashworthiness for Aerospace Structure and Hybrids (CRASH) Lab of Virginia Polytechnic Institute stated that mid-air collision could lead to damage of radome, cockpit exterior, leading edge, flaps, stabilizer, and propulsion system [3].

Modern UAV can easily bypass traditional security system such as high fence [4]. One solution to the problem is with the modeling of automated UAV detection system. This system will help security personnel to detect and identify unauthorized UAV operation around airport. With this system, airspace monitoring will be more efficient and the risk of safety and security hazard can be reduced. This research aims to assess the performance of the visual AI model to detect UAV in fair condition and the the performance of the visual AI model to detect UAV in rain condition, and determine the most suitable visual AI model for anti-UAV system.

2. Methodology

2.1. Data Gathering

Dataset that will be used in this research is gathered from a Kaggle site owns by dasmehdixtr [5] and another open-source dataset available. The dataset includes images of multi-rotor UAV with various shape, size, and background. There are images of both quad-copter and octa-copter UAV in various distance. Images variations are in accordance with the scope and limitation of the research.

A total of 500 data are used for this research. The data will be grouped into two categories, the first group is for training the model and the second group is for model validation. Training dataset has 350 data or 70% out of the total, while validation dataset has 150 data or 30% out of the total [6].

2.2. Data Preprocessing

There are two steps of data preprocessing that this research use, data labelling and data filtering. Data labeling is being done to define the ground truth for model training. In YOLO format, bounding box around the object is translated into several parameters in txt file format. Those parameters define the class of the object and its coordinate [7]. Data that was gathered from Kaggle [5] has already being labelled. Manual annotation is only required for other data with web-based application, Roboflow [8].

Data filtering is being done to add new image variation. This research use gaussian blur filter. This filter creates a blurry effect on an image, so that the model can detect object in rain condition since rain can make blurry effect on an image [9]. Raindrops also produce noise, which can be reduced by using this filter. OpenCV is used to apply the filter.

2.3. Train the Model

Transfer learning method is used in this research. Transfer learning is using pretrained model to achieve better accuracy, even with limited dataset [10]. This research use YOLOv4.conv.137 pretrained model from AlexeyAB's Github Repository [11]. Model training process uses Google Colab platfrom, which has free virtual machine and graphics processing unit (GPU). The output of this process is a file in weight format for YOLOv4. The author modified the notebook to evaluate the model every 100 iteration.

Training configuration used in this research refers to previous study [6] and other reference [12]. Table 1 below describe the parameters for this research.

Table 1. Parameter setup

Parameter	Value
Batch size	64
Subdivision	16
Width	416
Height	416
Channel	3
Class	1

2.4. Model Evaluation and Deployment

Evaluation is carried out every 100 iterations. Each evaluation result is stored automatically by the system. The model will be evaluated by several metrics, such as true positive (TP), false positive (FP), false negative (FN), precision, recall, F1-score, and mean average precision (mAP).

The model is deployed using local machine equipped with GPU. Several software is used for the model deployment, such as Anaconda, Git Bash, and Visual Studio Code. Anaconda contains virtual environment to install the required packages, such as OpenCV, TensorFlow, CUDA toolkit, cudnn, and Python. Visual Studio Code is used to edit the code and Git Bash is used to run the command. The author refers to Adrian Rosebrock [13] and The AI Guy's Github Repository [12] for the model deployment.

3. Results and Discussions

3.1. Experiment Results

3.1.1. First Experiment

In the first experiment, model is trained using prepared dataset. The dataset is grouped into train dataset (70%) and validation dataset (30%). The dataset is directly used for model training without any filter applied beforehand. Table 2 shows evaluation metrics on first experiment.

Table 2. Metric Evaluation on First Experiment

Metric	Value
TP	151
FP	8
FN	12
Precision	0.95
Recall	0.93
F1-Score	0.94
mAP	97.28%

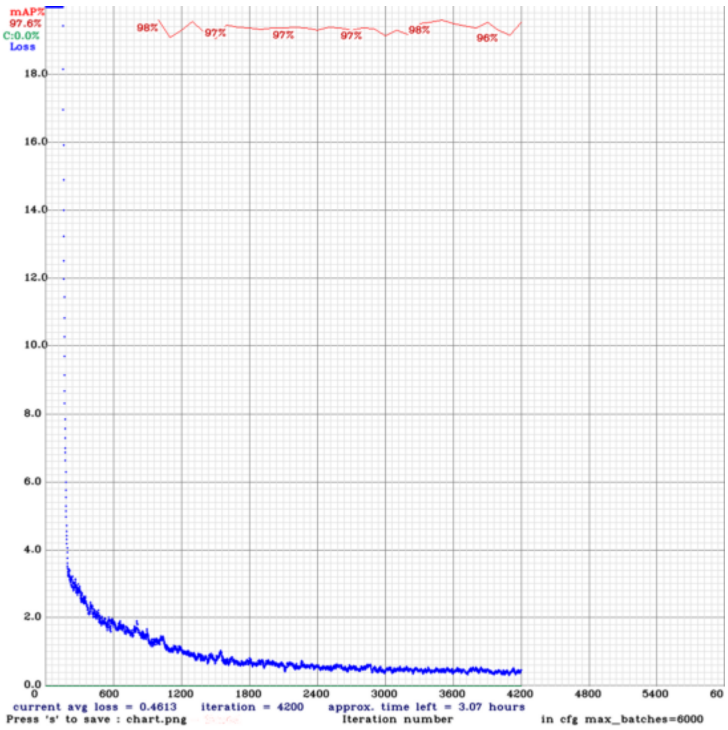


Figure 1. mAP Graph of First Experiment

The model is trained with 4000 iterations as shown in Fig.1. Table 2 shows that the first experiment produces 159 predictions, of which 151 are accurate while 8 are not accurate. The model fails to detect 12 objects in the validation dataset. Overall, first experiment has a F1-score of 0.94 and mAP of 97.28%.

3.1.2. Second Experiment

In the second experiment, model is trained using prepared dataset. The dataset's labelling and grouping is the same as the first experiment. The dataset is filtered with gaussian blur to train the model to detect objects in lower image quality. The filter application creates some changes in the evaluation metrics. Table 3 shows evaluation metrics on second experiment.

TABLE 3. Metric Evaluation on Second Experiment

Metric	Value
TP	159
FP	5
FN	4
Precision	0.97
Recall	0.98
F1-Score	0.97
mAP	98.37%

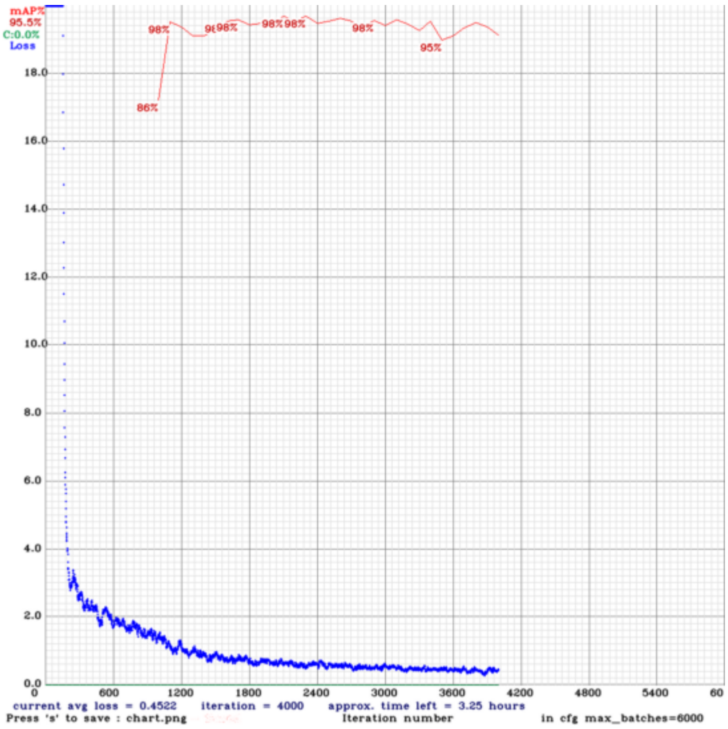


FIGURE 2. mAP Graph of Second Experiment

Table 3 and Fig.2 above shows that second experiment is able to detect 159 objects accurately, 5 objects inaccurately, and the model fails to detect 4 objects in the validation dataset. The model has a F1-score of 0.97 and mAP of 98.37%.

e-ISSN: 2715-6958 p-ISSN: 2721-5342

3.1.3. Third Experiment

The third experiment uses model from the first experiment. Image sharpening is used in the data testing to improve the performance of the model in rain condition. Image sharpening enhances the quality of the image in rain condition. Image sharpening is done using OpenCV function, cv2.filter2D.

4. Analysis

4.1. Evaluation Metrics Comparison

The result of first experiment is defined as model 1, which was trained without applying any filter, and the result of the second experiment is defined as model 2, which was trained using gaussian blur filter. Table 4 shows the comparison of those models.

Table 4. Comparison of Evaluation Metrics

Metric	Model 1	Model 2
TP	151	159
FP	8	5
FN	12	4
Precision	0.95	0.97
Recall	0.93	0.98
F1-Score	0.94	0.97
mAP	97.28%	98.37%

As shown in the Tab.4, model 2 achieves better performance compared to model 1. Every evaluation metrics of model 2 is higher and better than model 1. This comparison proves that the application of gaussian blur filter on the dataset can reduce noises and object's edges. Hence, the model can extract features more efficiently. Better detection capability produces better evaluation metrics' value.

4.2. Visual Comparison

Visual comparison is done using resulting images of the model deployment. In accordance with the scope and limitation of the research, images inputs are in both fair and rain condition. There are at least one UAV in the image. In input 1 of fair condition, there are several UAV in various size and distance, while in input 2 there is a UAV and a bird flying together. As for both inputs for rain condition, UAVs are obstructed by raindrops which makes the shape of UAVs appear blur. Raindrops can disrupt the detection process. Figure 3 and Fig.4 show visual comparison for fair condition.

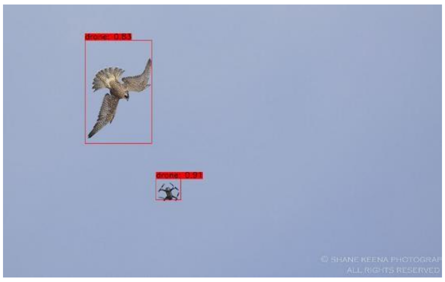
(a) (b) (c)

FIGURE 3. (a) Input 1 of Fair Condition and Output for (b) First Experiment and (c) Second Experiment

 doi: 10.47355/AVIA.V5I1.83

(a)

(b)



(c)

FIGURE 4. (a) Input 2 of Fair Condition and Output for (b) First Experiment and (c) Second Experiment

Third experiment is not needed for fair condition since the quality of the test image is acceptable. As for rain condition, the first, second, and third experiment is compared visually. Figure 5 and Fig.6 are those comparisons.

(a)

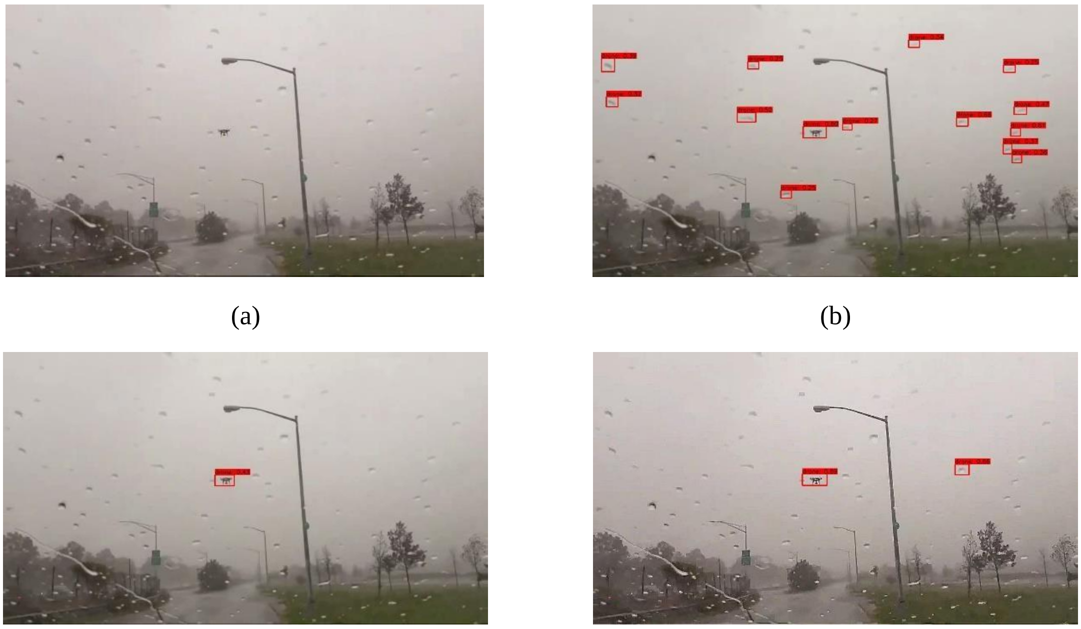
(b)

(c)

(d)

FIGURE 5. (a) Input 1 of Rain Condition and Output for (b) First Experiment, (c) Second Experiment, and (d) Third Experiment

e-ISSN: 2715-6958 p-ISSN: 2721-5342



(a) (b)

(c) (d)

FIGURE 6. (a) Input 2 of Rain Condition and Output for (b) First Experiment, (c) Second Experiment, and (d) Third Experiment

Visual comparison shows similar evaluation with the metrics comparison. As shown in figures above, second experiment is able to detect UAVs better than both first and third experiment. In fair condition, first experiment can detect every UAV accurately, while second experiment shows a false positive on a bird. However, false positive is preferable than false negative for this research.

In rain condition, first and third experiment show false negative, while second experiment can detect every UAV accurately. First and third experiment also show false positive on some of the raindrops, while second experiment shows no false positive at all. Visually, second experiment is better than both first and third experiment.

Anti-UAV system is demanded to flag potential UAV sightings as a preventive measure to mitigate the risk of UAV breachment. The model is preferred to be able to detect objects in lower precision (shows false positive) rather than being not able to detect UAV (false negative). This research proves that second experiment is able to detect every UAV in all images, although it still has false positive on a bird. Therefore, the most suitable model for anti-UAV system is the second experiment.

5. Conclusions

It can be concluded that the performance of visual AI model that was trained using earlier dataset, first experiment is able to detect every UAV accurately in fair condition. Performance of visual AI model that was trained using gaussian blur filter, second experiment is able to detect every UAV, but shows a false positive on a bird. Therefore, the application of gaussian blur filter has no effect for UAV detection on fair condition. Also, the performance of visual AI model that was trained using earlier dataset, first and third experiment is not able to detect UAV accurately and show few false negatives in rain condition. Performance of visual AI model that was trained using gaussian blur filter, second experiment is able to detect every UAV accurately. Performance of visual AI model is not affected by image sharpening on test data. Meanwhile, the application of gaussian blur filter on training dataset increases the performance of visual AI model to detect UAV in rain condition.

e-ISSN: 2715-6958 p-ISSN: 2721-5342

The most suitable visual AI model to detect UAV in anti-UAV system is the one that has been trained using gaussian blur filter, the second experiment. Application of gaussian blur filter on training dataset makes the model able to detect every UAV, both in fair and rain condition.

References

1. S. Park, H. T. Kim, S. Lee, H. Joo, H. Him, IEEE Access 9, 42635-42659 (2021).
2. D. Hodgkinson and R. Johnston, Aviation Law and Drones (Routledge, London, 2018).
3. J. Pyrgies, Journal of Airline and Airport Management 9, 63-96 (2019).
4. J. Knight, “Countering Unmanned Aircraft System” Master’s Thesis, Naval Postgraduate School, 2019.
5. Dasmehdixtr, Drone Dataset (UAV). https://www.kaggle.com/dasmehdixtr/drone-dataset-uav (Accessed 20 May 2021).
6. M. T. Hidayat. “Designing an Aerial Flood Victims Detection using Deep Learning” Undergraduate Thesis, Insitut Teknologi Bandung, 2021.
7. R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, E. A. B. da Silva, Electronics 10 (2021).
8. Roboflow, Inc. Roboflow. https://roboflow.com (Accessed 22 May 2021).
9. X. Fu, J. Huang, X. Ding, Y. Liao, J. Paisley, IEEE Transactions on Image Processing 26, 2944-2956 (2021).
10. C. H. Lee and C. W. Lin, Appl. Sci. 11 (2021).
11. A. Bochkovskiy, Github Repository (2021).
12. TheAIGuys, Github Repository (2021).
13. A. Rosebrock, Deep Learning for Computer Vision with Python: Starter Bundle (PyImageSearch, 2017).