YOLOv911: An Improvement to YOLOv7 for Airborne Object Detection Task

Dion Andreas Solang, Reza Fuad Rachmadi, I Ketut Eddy Purnama
Department of Computer Engineering
Faculty of Intelligent Electrical and Information Technology
Institut Teknologi Sepuluh Nopember (ITS)

***Abstract*—In this research, we present some approach to improve the detection capability of YOLOv7 for airborne objects. Airborne objects appear very small in camera images when they are located at a considerable distance from the camera. However, due to their high speed of movement, it is crucial to detect them while they are still far away. Therefore, to effectively detect these objects with YOLOv7, its small object detection capability must be enhanced. To address this challenge, we proposed several modifications that include changes in the architecture (adding an extra detection head, modifying the feature-map source, and replacing the detection head with a detached anchor-free head), application of bag-of-freebies techniques (anchor recalculation and mosaic augmentation), and change in the inference process (partitioning the image and performing inference on each partition). Through comprehensive experimentation, we have discovered that the combination of replacing the detection head with a detached anchor-free head, and performing inference on partitions yields the most promising results, with a significant increase in mean average precision (mAP) of 46.18% while still maintaining real-time inference speed (greater than 10 FPS). This improvement is notably higher compared to the unmodified plain YOLOv7, which achieved an mAP score of 0%.**

***Index Terms*—Small Object Detection, YOLOv7, Architecture Modification, Bag-of-Freebies Modification, Airborne Object**

Fig. 1: Example Image from AOT Dataset (From [3])

I. Introduction

Autonomous Aerial Vehicle (AAV) has the potential to significantly impact industries, particularly in commercial delivery [1]. To realize this potential, a reliable and efficient Sense and Avoid (SAA) system is needed. While the airspace in which AAVs operate may be relatively sparse, there are still risks of encountering static obstacles or airborne objects such as birds or other drones. With an effective SAA system, we can mitigate these risks and safeguard both the AAV and the valuable cargo it carries during commercial delivery.

As most AAV uses camera as its sensor due to its smaller weight and lower price, there is a need for camera based object detection system for SAA purpose. One problem is that, airborne objects can appear unexpectedly and approach rapidly from long distances. For this reason, airborne objects must be detected as early as possible, which means that they must be detected when their image on the camera were still very small. Thus, the object detector for SAA system must be able to detect small objects.

In this research, we attempt to optimize YOLOv7 to be able to perform airborne object detection. YOLOv7 was chosen due to its ability to perform real-time object detection accurately even under complex outdoor environment. At the time of execution of this research, YOLOv7 had the highest mAP score amongst all published real-time object detector [2]. This research was not about finding the ultimate solution for airborne object detection, but rather an exploration of methods that can be used to enhance YOLOv7 on detecting small objects, which extends to airborne objects.

To optimize YOLOv7, we defined a set of modification to be applied to YOLOv7 that includes modification to its neural network architecture, some applications of bag-of-freebies, and modification on the way the neural network perform inference. To find the best probably suboptimal solution, we will try combinations of the modification within the set, and choose the one with the highest mAP score.

We will be using The Airborne Object Tracking (AOT) Dataset [3] to train and test the modification combinations. AOT Dataset consist of aerial vehicle flights footage. In this dataset, the resolution of each image are 2048×2448 px, meanwhile the size of the airborne object bounding box can be as small as 4 px (0.00008% of resolution). An example of the dataset can be seen on Fig. 1.

II. Related Works

Several attempts have been made in the past to increase YOLO architecture ability to detect small objects. Here we present some of them.

A. YOLO-Z

YOLO-Z is a modification of YOLOv5 to optimize its ability to detect cone for autonomous racing purpose [4]. YOLO-Z modify the backbone of YOLOv5r5.0 to down-scaled DenseNet, while the neck was changed to PANet. These modification results in an increase of accuracy to detect cone that are far away while still being able to do detection in real-time.

B. exYOLO

exYOLO used YOLOv3 as the basis for modification [5]. exYOLO modified the neck of YOLOv3 by adding Receptive Field Block before combining the feature maps. These modifications produce a higher mAP score than plain YOLOv3 on PASCAL VOC2007 dataset.

C. Barunastra ITS' Object Detection System 2022

asdfasdfasdf

III. Experimental Setup

A. Instruments

To conduct the experiment, we use Nvidia RTX 2080 Ti GPU which has 11 GB of VRAM. We performed a pilot test using this hardware and found that with this limited amount of memory, training large YOLOv7 model such as W6, E6, and E2E is infeasible.

Furthermore, the dataset that will be used to train the model will be limited to 400 images. The sampling strategy for the dataset will be explained in section III-B. In a pilot test of the experiment, we found that to train a model for 300 epochs with 400 images with this setup, it will take around 20 hours. This is the reason why we limit the amount of images to 400 and epoch to 300.

B. Dataset

The AOT dataset, consist of more than 11 TB images of drone camera footage. Amongst these 11 TB of data, there are images taken from planned and unplanned encounters with airborne objects. In this research, we sample the data from planned encounters. There are millions of image in the planned encounters. To obtain 400 images for training as explained in section III-A and some more for validation and testing, we use the sampling strategy described in Table I.

TABLE I: Dataset Sampling Strategy

	Total Images	Percentage				
		Airplane	Helicopter	Bird	Drone	Negative
Training	400	23.75%	23.75%	23.75%	23.75%	5%
Validation	100	20%	20%	20%	20%	20%
Test	200	20%	20%	20%	20%	20%

C. Modifications

We proposed some modifications that can be applied to YOLOv7

1) Mosaic Augmentation: Mosaic Augmentation was reported to increase the mAP score of the model in [6]yolov5. This augmentation is simple to implement. Therefore, we included mosaic augmentation in our set of modifications combine.

2) Anchor Recalculation: Anchor points that were available on the implementation code of YOLOv7 are optimized for COCO2017 dataset. As the AOT dataset distribution are heavily skewed, the anchors need to be adjusted to match the data distribution. For this reason, anchor recalculation is included in our set of modifications.

3) EIoU localization loss: The localization loss used in YOLOv7 is CIoU. Both EIoU and CIoU are designed to solve the vanishing gradient problem of the standard IoU. The advantage EIoU has over CIoU is that when the bounding boxes intersect, EIoU behaves like the standard IoU while CIoU doesn't. The metrics used to evaluate the models like mAP are based on the standard IoU, thus it's better for the loss function to mimic the metric [7]. [7] reported that EIoU performs better on Faster-RCNN than CIoU, making this modification a good candidate to be tested on YOLOv7.

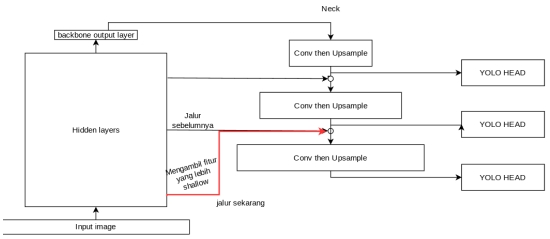

Fig. 2: Moving Neck Feature-map Source

4) Modify Neck Feature-map Source: The source feature map that was fed on the feature pyramid can be moved to a shallower layer of the backbone (look at Fig. 2). The shallower layer (layers that are closer to the input) has more information about the image, albeit have less abstraction. By doing this, we avoid the loss of information.

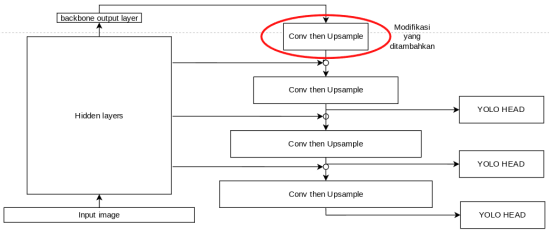

Fig. 3: Adding More Detection Layer

5) Additional Detection Layer: An additional detection layer enable YOLOv7 to detect at more scales. With more scales, the detection layers can be more specialized to specific

cluster of data. This approach has been tried in [8] by increasing the number of detection layer from 2 to 3 in YOLOv4-tiny. In this experiment, we will increase the number of detection layer from 3 to 4.

6) Replacing Detection Layer to Decoupled Anchor-Free Head: One of the advantage of anchor-free model to anchor-based model is that it decreases the amount of heuristic tuning parameter as we don't have to define the anchors. [9] reported that using anchor-free head increases the accuracy and decreases the number of parameters in the model, resulting in a faster and more accurate model.

IV. Result

A. Initial Performance

At first, we evaluate the performance of a plain YOLOv7 without all the modifications proposed in section III-C. With 300 epochs and 400 data sample, we find the model was unable to detect anything in the test set (mAP = 0). For the purpose of comparison with other modification combination, we will call this model as `YOLOv7-plain`.

B. Mosaic Augmentation and Anchor Recalculation

In this section, we will compare 3 modification combination

- YOLOv7-plain + Mosaic
- YOLOv7-plain + Anchor Recalculation
- YOLOv7-plain + Mosaic + Anchor Recalculation

We calculated the anchors using k-means clustering algorithm on the training dataset. The result of the recalculation can be seen on Fig. 4. As can be seen on the figure, 8 out of 9 of the old anchor points are placed in the first quadrant of the median line. This means that those 8 anchors responsible only for 25% of the dataset, which is very ineffective. Compare that to the recalculated anchor. Every quadrant has at least one anchor point responsible to it. Table II shows that the model was

TABLE II: Mosaic Augmentation and Anchor Recalculation Performance

No	Modification	mAP@50
0	**YOLOv7-plain**	0%
1	**YOLOv7-plain + mosaic**	0%
2	**YOLOv7-plain + anchor recalculation**	0%
3	**YOLOv7-plain + mosaic + anchor recalculation**	**11,2%**
	Improvement	**+11,2%**

only able to detect something on the test dataset after being applied mosaic augmentation and anchor recalculation.

This model will be used as the basis for further modification, as it was the only model that could detect objects in test dataset. For this reason, this combination of modification shall be called `YOLOv7-base` from this point on.

C. EIoU Localization Loss

In addition to just using EIoU, we also tested EIoU with its convexication technique mentioned in [7]. The result can be seen on Table III It turns out that EIoU only worsen the performance of YOLOv7 in AOT dataset.

TABLE III: EIoU Localization Loss Performance

No	Modifikasi	mAP@50
0	**yolov7-base + CIoU (original)**	**11,2%**
1	**yolov7-base + EIoU**	0%
2	**yolov7-base + EIoU + Convexication**	4.92%
	Improvement	**-6.28%**

D. Modify Neck Feature-map Source

We move the connection of the first pyramid from scale 8 to scale 4 of the backbone. That is from layer 24 to layer 11 of the configuration file. The moving of this connection is illustrated in Fig. 5. The performance of this modification can be seen on Table IV. This modification managed to increase

TABLE IV: Moving Feature-map Source Performance

No	Modification	mAP@50
0	**yolov7-base**	11.2%
1	**yolov7-base + modifikasi neck-backbone**	**14.09%**
	Improvement	**+2.98%**

the mAP score by 2.98%. For the purpose of comparison with other modification combinations, we shall call this model `YOLOv7-moveconnection` from this point on.

E. Adding Extra Detection Layer

We add an extra feature pyramid stage connected to scale 4 of the backend, and put a detection head on it. This modification is illustrated in Fig. 6. We find that this modification perform worse than YOLOv7-moveconnection as seen on Table V.

TABLE V: Performance of Adding Extra Detection Layer

No	Modifikasi	mAP@50
0	**yolov7-base**	**11.2%**
1	**yolov7-base + additional head**	5.19%
	Improvement	**-6%**

F. Decoupled Anchor-free Head

We changed the head of the model to anchor-free head with Task Aligned Labelling. It resulted in 0 mAP.

V. Conclusion

From the result of the experiment, we can conclude that, from the set of modification candidates proposed in this research, we found that the combination of mosaic augmentation, anchor recalculation, and modifying the connection of neck and backbone produces a model with the greatest mAP score which is 14.09%.

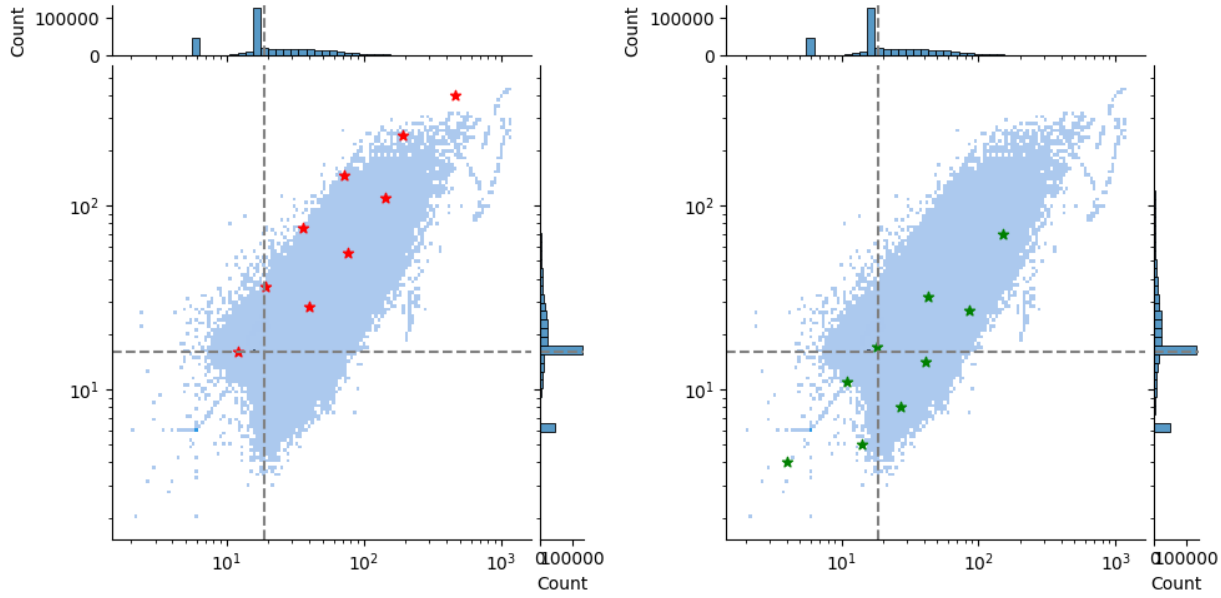

Fig. 4: Anchor Points in Dataset Distribution. Left: Original Anchors. Right: Recalculated Anchors

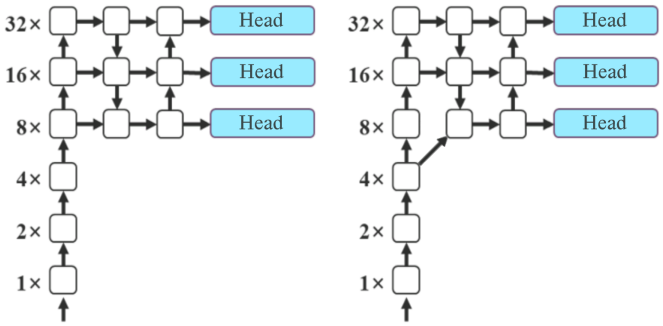

Fig. 5: Moving Feature-map Source

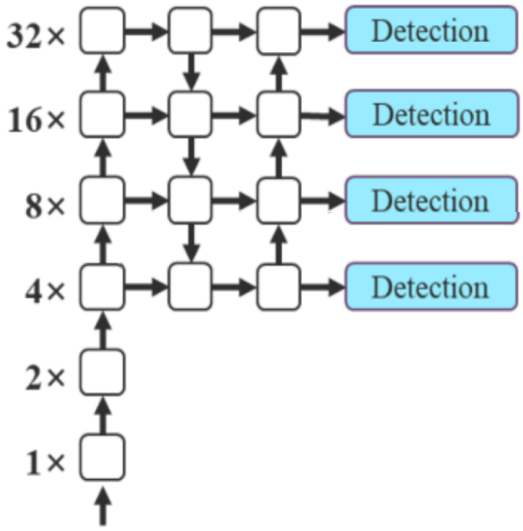

Fig. 6: Adding More Detection Layer

VI. Discussion

As of today, the modification proposed in this research only includes the modification that doesn't significantly impact the latency of the model. Some modification like partitioning the image and perform detection on each of partition could produce a great increase in mAP as the objects are magnified (number of partition) times. The latency also multiplied by the number of partition. But we can make the model faster by down-scaling the model. Thus, it is needed to find the optimal combination between down-scaling and number of partition that can produce a model that have high accuracy, but still have a latency that can be categorized as real-time.

References

[1] Amazon. (2022) Amazon prime air prepares for drone deliveries. [Online]. Available: https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries

[2] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022.

[3] (2021) Airborne object tracking dataset. [Online]. Available: https://registry.opendata.aws/airborne-object-tracking

[4] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "Yolo-z: Improving small object detection in yolov5 for autonomous vehicles," 2021.

[5] J. Xiao, "exYOLO: A small object detector based on YOLOv3 object detector," *Procedia Computer Science*, vol. 188, pp. 18–25, 2021. [Online]. Available: https://doi.org/10.1016/j.procs.2021.05.048

[6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[7] H. Peng and S. Yu, "A systematic iou-related method: Beyond simplified regression for better localization," *CoRR*, vol. abs/2112.01793, 2021. [Online]. Available: https://arxiv.org/abs/2112.01793

[8] N. Aditya, A. Indaryo, D. Solang, M. Santiung, H. Atmaja, A. Azis, F. Januar, F. Javanica, G. Kautaman, E. Kazaksti, D. Nugraha, R. Permadani, R. Ramadhan, R. Ramadhan, Z. Damayanti, M. Valentia, P. Sundana, F. Shodiq, R. Waisnawa, A. Farhan, A. Adifatama, and R. Dikairono, "Roboboat 2022: Technical design report Barunastra ITS roboboat team," https://robonation.org/app/uploads/sites/3/2022/05/RB22_Institut-Teknologi-Sepuluh-Nopember_TDR.pdf, 2022.

[9] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," 2021.