Exploitation of Epoch Parameter in Low Computing Cost Model-Based EfficientDet for UAV Object Detection

1st Iga Narendra Pramawijaya
School of Electrical Engineering
Telkom University
Bandung, Indonesia
iganarendra@student.telkomuniversity.ac.id

2nd Suryo Adhi Wibowo
School of Electrical Engineering
Telkom University
Bandung, Indonesia
suryoadhiwibowo@telkomuniversity.ac.id

***Abstract*—Unmanned Aerial Vehicles are equipped with high-resolution cameras. Zhu, et al. [1] noveled object detection models for UAV images, but available models took high computational cost. Models that not trained using the UAV image will result in poor performance. When object detection models retrained using UAV images, it will take up a high computational cost. EfficientDet object detection model is significantly lower in computing cost [2]. Therefore, the author wants to examine the performance of the low computing cost model-based EfficientDet on UAV images. An object detection system on UAV imagery is trained using conventional computers will be designed to detect 10 classes object in Visdrone dataset [3] using the D0 version of the EfficientDet model. After the data is obtained then preprocessing will be carried out in the form of annotation conversion. Next, model training process will be carried out several times. In each training, the model will be tested so as to produce a validation value. The last validation value will be analysed as a benchmark for performance, measured in six values, AP50, AP75, APl, ARmax1, ARmax10, and ARmax100. As a result, EfficientDet d0 model surpassed TridentNet model according to latest research with ARmax1 value of 2.33%.**

***Index Terms*—EfficientDet, UAV Images, Low Computing Cost.**

I. Introduction

Unmanned Aerial Vehicles (UAV) are equipped with high-resolution camera. Data capture using UAV has become more common since the demand for data in remote and rural areas is increasing. Since UAV cameras are able to take high resolution, we can use images and videos from this kind of camera for many usecase. Remote sensing and surveillance [4] [5], mitigation, and post-disaster mapping are some usage examples of UAV images [6] [7]. UAV images are having unique characteristics such as (i) ultra-high spatial resolution [8], (ii) affected by weather, and (iii) affected by altitude.

Artificial Neural Network Model can help implement the use UAV image [9]. Arificial neural network model can be shaped to recognise objects, known as object detection model. The available object detection models are not trained using the UAV image, and will result in poor performance when detecting objects from UAV images. When existing object detection models are retrained using UAV images, it will take up a high computational cost due to high resolution images from UAV. Hence, we usually use high computing hardware such as high performance computing (HPC) device if we want to train the model.

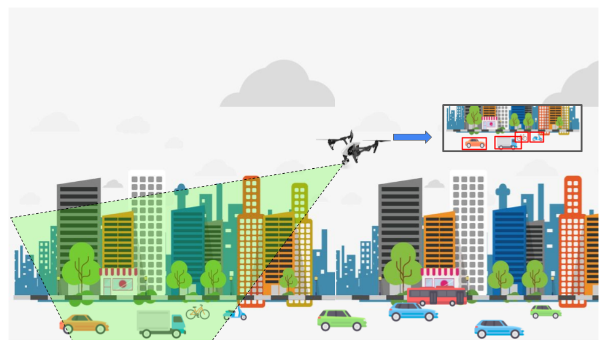

Fig. 1. Illustration of Object Detection-Based UAV in urban scenario.

Since HPC is not available for everyone, there is a demand to tackle this high computational cost problem. To fullfil this demand, Tan, et al developed EfficientDet, an efficient object detection model capable to deliver state-of-the-art capability [2]. This research will utilise the smallest version of EfficientDet (D0) and exploit its performance when faced with UAV dataset to highlight the most of its low computing capability under strict limited resources. As an addition to highlight its capability as low computing cost, we will train the EfficientDet D0 for maximum 50 epochs only to simulate the capability of personal-grade laptop.

II. Theory

A. *Object Detection-Based UAV*

In area of computer vision research, object detection is a branch branch that recognise, then pinpoint objects relative position in an image, as illustrated in figure 1. Deep learning approach shows incredible result solving this problem [10]. Szegedy, et al [11] proofed that deep learning network not only learning image's features for classification, but also geometric information that will make the model easier to detect objects.

Mittal, et al discussed about several deep learning algorithms in UAV images object detection model, such as RCNN and its variants like Cascade RCNN and Faster RCNN, YOLO and its modifications, RetinaNet, RetinaNet, SSD, and more [12]. Mittal, et al. discussed mean average precision result from various network on VisDrone dataset, with the novel of PENet structure has the highest mean average precision value (41.1 mAP) [13]. In the previous research [12], outside of high computing cost model that even trained around 2000 epochs such as in [14], it is also discussed about others low computing cost model. With tiny-YOLO used as criterion, [15] tested several low computing cost model such as TinyYoloVoc, TinyYoloNet, SmallYoloV3 and DroNet. Amongs other models, DroNet shows best accuracy around 5 to 18 fps. In all literatures, low computing cost based EfficientDet performance result is not available.

B. *EfficientDet*

EfficientDet is a object detection model based on EfficientNet [16] developed by Tan, et al [2]. This object detection model focuses on delivering top-performing object detection quality and accuracy even limited by less computational cost. The largest version of EfficientDet (D7) achieved 52.2% AP on COCO test-dev. EfficientDet reached this level with only 325 million FLOPs, 4-9 times smaller in size and 13-42 times fewer FLOPs between its variations than its predecessor state-of-the-art model, such as AmoebaNet-based NAS- FPN detector [2]. This model uses a two-way pyramidal network as in figure 2 to feature extraction tool. This pyramid model consists of 8 levels, where the Bi-directional pyramid network (BiFPN) adapted from Chen, et al [17] applied from third to seventh level. This pyramid network is intended for making smaller feature maps, but with more information in each features. One tier pyramid to another is connected using the convolution operation, so that when moving levels, the feature map matrix becomes smaller in dimension, but increasingly dense information. In addition, when moving from one level to another is given a weight as width difference, as in equation 1 :

$$Width_{BiFPN} = 64 \times (1.35^n) \tag{1}$$

where n is the EfficientDet level. This represents during the learning process, EfficientDet network will learn which levels have high and low weights. Each level has its own BiFPN layer's depth formulated in equation 2 :

$$Depth_{BiFPN} = 3 + n \tag{2}$$

.

By using this adaptability, it can increase precision by assigning a higher weight to the level which produces a feature map that is representative of the predicted results. In this research, we exploited the performance of lightest EfficientDet with $n = 0$, or the least level available in the detection model.

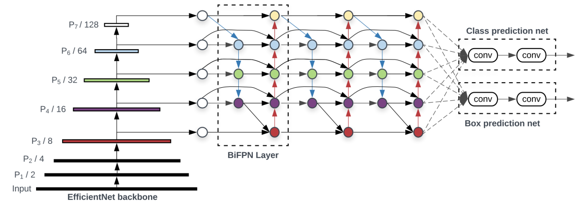

Fig. 2. EfficientDet Architechture as in Tan, et al. [2]

III. Method

A. *VisDrone Dataset*

This research is using VisDrone dataset. Visdrone dataset is a collection of UAV photos from AISKYEYE team. This team created in the laboratory of Machine Learning and Data Mining in Tianjin University. Tianjin University is located in mainland China, making the dataset consisting of Chinese urban city setup [1]. The dataset created consist of many scenarios. Object detection scenario part of the dataset containing 10,209 UAV areal photos. These photos splitted into three parts. 548 of the photos meant for validation; 6,471 train images; and leaving the rest 3,190 photos as model tester. The UAV areal photos was captured using real drone, not a dolly-crane platform. The ASKYEYE team gave heights and places variations across the dataset. All ground-truth annotations are created using human aid. This dataset including train, valid, and test sets are publicly accessible. VisDrone dataset has 10 different classes to detect (such as tricycle, a not moving person, van, car, truck, passenger bus, motorcycle, awning-tricycle,bicycle, and pedestrian [moving person]), and rare or uncommon-purposed vehicles (for example like forklift, tanker, fire-engine) are ignored.

B. *Training Process and Data Preparation*

Before being trained, we transformed the annotation format from PASCAL VOC to tfrecord, in order to accommodate the EfficientDet D0 model trained using automl framework. In each epoch, we randomly picked 5696 from 6471 images for training, and the random picking process will be repeated each epoch. We used EfficientDet D0 from ImageNet checkpoint as backbone. Training process will be limited to maximum 50 epochs. Training will be analysed every 10 epochs. We trained the model not using any high performance computing device GPUs to represent lacking of access to high performance computing devices. Instead we used a laptop-grade GPU. We used a laptop with NVIDIA GeForce GPU of GTX 1660 Ti mobile edition or Max-Q Design with 6GB of GDDR6 dedicated GPU memory, 9th generation 2.6 GHz (up to 4.5 GHz performance turbo) Intel Core i7-9750H CPU and 16 GB, DDR4 2933 MHz dual channel RAM. EfficientDet D0 trained on two different optimiser function, stochastic gradient descend [18], and Adam [19]. As in previous research [1], we measured the model performance with six values, AP50, AP75, APl, ARmax1, ARmax10, and ARmax100.

TABLE I
EFFICIENTDET D0 RESULT OPTIMISED WITH STOCHASTIC GRADIENT DESCEND

Epochs	AP50	AP75	APl	ARmax1	ARmax10	ARmax100
10	0.015	4.26×10^{-3}	0.03285	8.36×10^{-3}	0.0257	0.0334
20	0.0184	5.63×10^{-3}	0.0388	9.18×10^{-3}	0.0298	0.0386
30	0.0346	0.0161	0.0702	**0.0135**	0.0414	0.0522
40	0.0366	0.0149	0.0689	**0.0151**	0.0463	0.0580
50	0.0414	0.0193	0.0841	**0.0169**	0.0492	0.0618

TABLE II
EFFICIENTDET D0 RESULT OPTIMISED WITH ADAM

Epochs	AP50	AP75	APl	ARmax1	ARmax10	ARmax100
10	0.0173	5.97×10^{-3}	0.0395	8.08×10^{-3}	0.0284	0.0339
20	0.0382	0.0169	0.0789	**0.0158**	0.0463	0.0572
30	0.0480	0.0239	0.0901	**0.0185**	0.0529	0.0647
40	0.0524	0.0245	0.1022	**0.0208**	0.0575	0.0698
50	0.0621	0.0303	0.1080	**0.0233**	0.0633	0.0767

IV. EXPERIMENT RESULT

A. Using Stochastic Gradient Descend as Optimiser Function

The EfficientDet D0 model using stochastic gradient descend optimiser function training result available at Table I. As expected, the average precision values are small. This is due to average precision needs hundreds or even thousands of epoch to get its optimal value. Especially when we are using smaller level of the EfficientDet object detection model. Beside of average precision, average recall actually revealed impressive values, especially in ARmax1. Bold values are the training results that exceeded previous top-3 in the category [1]. By only 30 epochs, EfficientDet d0 succeeded to be the second best ARmax1 model, beating other vigorously trained object detection model. Adding up to 50 epochs, this result is getting better, with 0.0169 ARmax1 score worth 50 epochs of training.

B. Using Adam as Optimiser Function

The EfficientDet D0 model using Adam optimiser function training result available at Table II. Unlike the average precision value, average recall value shows a promising result. By only 20 training iteration, the EfficientDet D0 able to position its performance as the second best ARmax1 value [1]. Only adding 10 epochs to the training process, the EfficientDet D0 able to surpass TridentNet, the current top performing model for ARmax1. TridentNet only able to reach 1.17% ARmax1 value, and worth to mention that this model typically trained in hundreds of epochs [1]. EfficientDet D0 able to reach 1.185% ARmax1 value by only 30 epochs. This result making EfficientDet easily reproducible, and deployed across systems. This result is getting better the more epochs added. Finally after 50 epochs, EfficientDet D0 scored 2.33% ARmax1 value, almost twice as much than TridentNet [1].

C. Model Implementation as Object Detection-Based UAV

As seen on both of the results, average precision score value of EfficientDet D0 is far below another state-of-the-art model [1]. This result is tolerable because the importance of average recall value in object detection-based UAV implementation scenario. With better average recall value, the model will detect more object and focuses on localisation accuracy [20]. When an object detection model has high recall but low precision value, model will classify more object, but some of them will be classified as a different object, for example a truck got detected as a car [21]. In object detection-based UAV implementation scenario, a higher recall value is needed for live inference when the UAV is on mobile. Recall is being prioritised over precision because detecting there is an object is more important than correctly detect the class of an object. After detecting object presence, detection speed is also important in object detection-based UAV implementation scenario, especially when the UAV is on mobile. In many scenario, decreases of the detection quality to prioritise detection speed is common and tolerable [20].

V. CONCLUSION AND FUTURE RESEARCH

Looking up to the simulation and model performance analysis, EfficientDet d0 model able outperform in terms of ARmax1 score of the previous research model in 30 iterations or more, even though it trained under limited hardware computing capabilities. The EfficientDet d0 model is suitable as a low-cost object detection model for UAV implementations because it has a high value of recall. Due to the lower computational costs required, it will simplify the training process, increase the detection speed of the UAV and the scalability of object detection in the future. Suggestions that can be applied to further research related to the topic of the research is that there is potential to increase the detection quality of EfficientDet models. Not only by improving recall values, but also to increase precision. Future research can improve the overall model reliability by applying more pre-processing to simplify the input image, as well as modifying the pooling layer on the model to increase the model's capabilities when faced with high-resolution images from UAV.

REFERENCES

[1] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," 2021.

[2] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 781–10 790.

[3] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, Q. Nie, H. Cheng, C. Liu, X. Liu *et al.*, "Visdrone-det2018: The vision meets drone object detection in image challenge results," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[4] J. Wu, G. Yang, X. Yang, B. Xu, L. Han, and Y. Zhu, "Automatic counting of in situ rice seedlings from uav images based on a deep fully convolutional neural network," *Remote Sensing*, vol. 11, no. 6, 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/6/691

[5] S. Oleire-Oltmanns, I. Marzolff, K. D. Peter, and J. B. Ries, "Unmanned aerial vehicle (uav) for monitoring soil erosion in morocco," *Remote Sensing*, vol. 4, no. 11, pp. 3390–3416, 2012. [Online]. Available: https://www.mdpi.com/2072-4292/4/11/3390

[6] K. G. Panda, S. Das, D. Sen, and W. Arif, "Design and deployment of uav-aided post-disaster emergency network," *IEEE Access*, vol. 7, pp. 102 985–102 999, 2019.

[7] J. M. M. Neto, R. A. da Paixão, L. R. L. Rodrigues, E. M. Moreira, J. C. J. dos Santos, and P. F. F. Rosa, "A surveillance task for a uav in a natural disaster scenario," in *2012 IEEE International Symposium on Industrial Electronics*, 2012, pp. 1516–1522.

[8] G.-H. Kwak and N.-W. Park, "Impact of texture information on crop classification with machine learning and uav images," *Applied Sciences*, vol. 9, no. 4, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/4/643
[9] C. Kyrkou, G. Plastiras, T. Theocharides, S. I. Venieris, and C.-S. Bouganis, "Dronet: Efficient convolutional neural network detector for real-time uav applications," in *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2018, pp. 967–972.
[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.
[11] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," 2013.
[12] P. Mittal, R. Singh, and A. Sharma, "Deep learning-based object detection in low-altitude uav datasets: A survey," *Image and Vision Computing*, vol. 104, p. 104046, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885620301785
[13] Z. Tang, X. Liu, G. Shen, and B. Yang, "Penet: Object detection using points estimation in aerial images," *CoRR*, vol. abs/2001.08247, 2020. [Online]. Available: https://arxiv.org/abs/2001.08247
[14] C. Iuga, P. Drăgan, and L. Busoniu, "Fall monitoring and detection for at-risk persons using a uav," *IFAC-PapersOnLine*, vol. 51, pp. 199–204, 01 2018.
[15] C. Kyrkou, G. Plastiras, T. Theocharides, S. I. Venieris, and C.-S. Bouganis, "Dronet: Efficient convolutional neural network detector for real-time uav applications," in *2018 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2018, pp. 967–972.
[16] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *CoRR*, vol. abs/1905.11946, 2019. [Online]. Available: http://arxiv.org/abs/1905.11946
[17] J. Chen, H. Mai, L. Luo, X. Chen, and K. Wu, "Effective feature fusion network in bifpn for small object detection," in *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2021, pp. 699–703.
[18] C. Lemaréchal, "Cauchy and the gradient method," 2012.
[19] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[20] J. H. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" *CoRR*, vol. abs/1502.05082, 2015. [Online]. Available: http://arxiv.org/abs/1502.05082
[21] A. Gad, "Evaluating object detection models using mean average precision," Mar 2022. [Online]. Available: https://www.kdnuggets.com/2021/03/evaluating-object-detection-models-using-mean-average-precision.html