

Partial half fine-tuning for object detection with unmanned aerial vehicles

Wahyu Pebrianto, Panca Mudjirahardjo, Sholeh Hadi Pramono

Department of Electrical Engineering, Faculty of Engineering, University of Brawijaya, Malang, Indonesia

Article Info

Article history:

Received Oct 29, 2022

Revised Jan 16, 2023

Accepted Mar 10, 2023

Keywords:

Deep learning

Fine-tuning

Object detection

Unmanned aerial vehicles

VisDrone dataset

ABSTRACT

Deep learning has shown outstanding performance in object detection tasks with unmanned aerial vehicles (UAVs), which involve the fine-tuning technique to improve performance by transferring features from pre-trained models to specific tasks. However, despite the immense popularity of fine-tuning, no works focused on to study of the precise fine-tuning effects of object detection tasks with UAVs. In this research, we conduct an experimental analysis of each existing fine-tuning strategy to answer which is the best procedure for transferring features with fine-tuning techniques. We also proposed a partial half fine-tuning strategy which we divided into two techniques: first half fine-tuning (First half F-T) and final half fine-tuning (Final half F-T). We use the VisDrone dataset for the training and validation process. Here we show that the partial half fine-tuning: Final half F-T can outperform other fine-tuning techniques and are also better than one of the state-of-the-art methods by a difference of 19.7% from the best results of previous studies.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Wahyu Pebrianto

Department of Electrical Engineering, Universitas Brawijaya

St. M.T. Haryono 167, Ketawanggede, Lowokwaru, Malang, East Java 65145

Email: wahyu.pebrianto1@gmail.com

1. INTRODUCTION

Object detection on intelligent machines integrated with drones and cameras or unmanned aerial vehicles (UAVs) has been many applied to help various real-life domains, such as military [1], forest fire detection [2], agriculture [3], security [4], [5], and urban surveillance [6]. That demands researchers in the field of object detection to be able to analyze images obtained from UAVs. Currently, numerous research has focused on deep learning [7] to overcome various object detection challenges, particularly in UAVs [8].

Deep learning advances are due to the availability of large-scale data, computing power such as graphics processing unit (GPU), and continuous research. Since deep convolutional neural network (deep CNN) has proposed by [9], deep learning methods have outperformed traditional machine learning in the imageNet large scale visual recognition challenge (ILSVRC) 2010 challenge [10]. These results also encourage many proposed other architecture such as visual geometry group (VGG) [11], GoogLeNet [12], and residual networks (ResNets) [13], [14], cross stage partial network (CSPNet) [15], efficientNet [16] which are widely used as backbone for feature extraction in classification and object detection tasks. In the object detection task, region-based convolutional neural network (R-CNN) is the first deep learning-based object detection method [17] that has outperformed other traditional detectors such as deformable parts model (DPM) [18] and SegDPM [19] in the challenge PASCAL visual object classes (PASCAL VOC) 2010 [20]. Other popular methods, such as the two-stage detector: Fast R-CNN [21], faster R-CNN [22], and the one-stage detector: you only look once (YOLO) [23]–[28], retinaNet [29], single shot multibox detector (SSD) [30] which have demonstrated strong

detection performance on the general object, such as Microsoft common objects in context (COCO) [31], PASCAL VOC, and imageNet. But those methods average it has poor performance when detecting small objects, which is the characteristic of the data captured by the UAVs. Therefore requires the right adjustment strategy in the model architecture or the right training strategy for object detection tasks in the UAVs. Much of the research recently has been proposed to overcome the challenge of object detection in UAVs. Such [32] used a deformable convolutional layer [33] in the last three stages of ResNet50, which also adopted a cascade architecture and augmentation data to improve model performance when detecting small and dense objects. [34] proposed RRNet with adaptive resampling as an augmentation technique followed by a regression module to improve bounding box prediction accuracy. The current work also involves transformer architecture which many use in natural language processing tasks [35]. Such as [36] that replaced the YOLOv5 detection layer with the detection of transformers, and [37] proposed a ViT-YOLO that combined multi-head self-attention [38] in the original CSP-Darknet backbone YOLOv4-P7 [27], bidirectional feature pyramid network (BiFPN) [39], and YOLOv3 as head detection layers. ViT-YOLO obtained an mean average precision (*mAP*) of 39.41% and was one of the top results in the dataset VisDrone2021-Det 2021 challenge [40]. The results not be separated by the availability of annotated data, which has become the key to the training process and benchmark. However, problems arise if the amount of annotated data is insufficient for the training process. It can easily be overfitting during the training process. Therefore also have an impact on the performance of the object detection method. While in practical situations, availability annotated data generally is difficult to obtain.

The current popular solution is to use a fine-tuning technique [41]. Fine-tuning is a common technique that takes advantage of the pre-trained model on large-scale data. Then transferred its features to new tasks with fewer data. This technique has been evident to increase the generalization of the model and avoid overfitting [42]–[47]. Such as has been done [17], [21]–[23], [29], [30] in general object detection tasks and [34], [36], [37], [48] in object detection tasks with UAVs. However, despite the immense popularity of fine-tuning techniques, no research focuses to studies the precise fine-tuning effects for object detection tasks with UAVs. Several reasons: i) the current work uses fine-tuning techniques without focus observing the impact of the transferred features or building an efficient fine-tuning strategy. ii) the differences in data sizes along with features are crucial to consider in the fine-tuning process. For example, the COCO [31] dataset and VisDrone [40] dataset have differences, as described in Figure 1(a) VisDrone which dominated by small objects, while in Figure 1(b) COCO dataset is dominated by general objects. The differences in the characteristics of these features can have impacted the performance of the fine-tuning technique, and very important to investigate which is the best fine-tuning strategy in object detection tasks with UAVs.

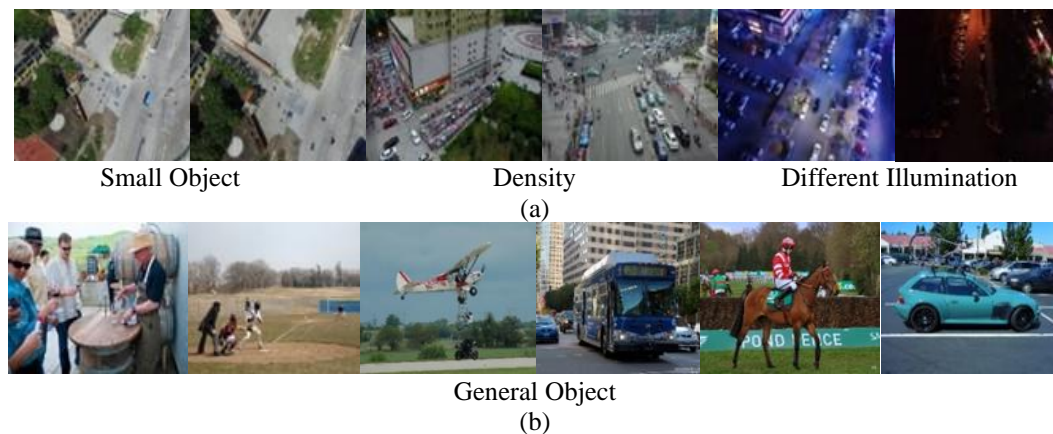


Figure 1. Image from dataset, (a) VisDrone dataset and (b) COCO dataset

In this study, we conduct experimental analysis on every existing fine-tuning approach, such as common fine-tuning (Common F-T) and frozen fine-tuning (Frozen F-T). We also propose a partial half fine-tuning strategy which consists of two techniques: first half fine-tuning (First Half F-T) and final half fine-tuning (Final half F-T). The first half F-T freezes the first half layer in the target network backbone, and the Final half F-T freezes the final half layer. The remaining layers are not frozen or randomly initialized during the training process for the target-task. In the evaluation process, we compared the partial half fine-tuning strategy with Frozen F-T, Common F-T, traditional training (Traditional-T), and one of the state-of-the-art methods of object detection tasks with UAVs. The main contributions of this paper are,

- We propose a partial half fine-tuning strategy: First half F-T and Final half F-T as a fine-tuning strategy for object detection with UAVs. (Section 2)
- We show the experimental results to answer where is the best procedure for transferring features from the base-task to the target-task, whether Common F-T, Frozen F-T, or partial half fine-tuning. (Sections 3.4.1-3.4.2)
- We show that partial half fine-tuning can outperform one of the state-of-the-art methods in object detection tasks with UAVs. (Section 3.4.3).

2. RESEARCH METHOD

In this section, we set up several fine-tuning procedures along with a partial half fine-tuning strategy to transfer a set of learned features in base-task and use them for target-task, as shown in Figure 2. That consists of three concepts: *base-task*, *target-task*, and transfer or *fine-tuning*. In *base-task* is shown in Figure 2(a): given the base-dataset $B_{dataset}$, which is trained with the base-network $B_{network}$ to complete the base-task B_{task} . In *target-task* is shown in Figures 2(b)-2(e): given a target-dataset $T_{dataset}$ to be trained with target-network $T_{network}$ for the target-task T_{task} . Then in the *fine-tuning* process, is do transfer features in the form of weight parameter $B_{network}$ from the pre-trained model to improve $T_{network}$ performance on new task T_{task} . Where $B_{dataset} \neq T_{dataset}$ or $B_{task} \neq T_{task}$

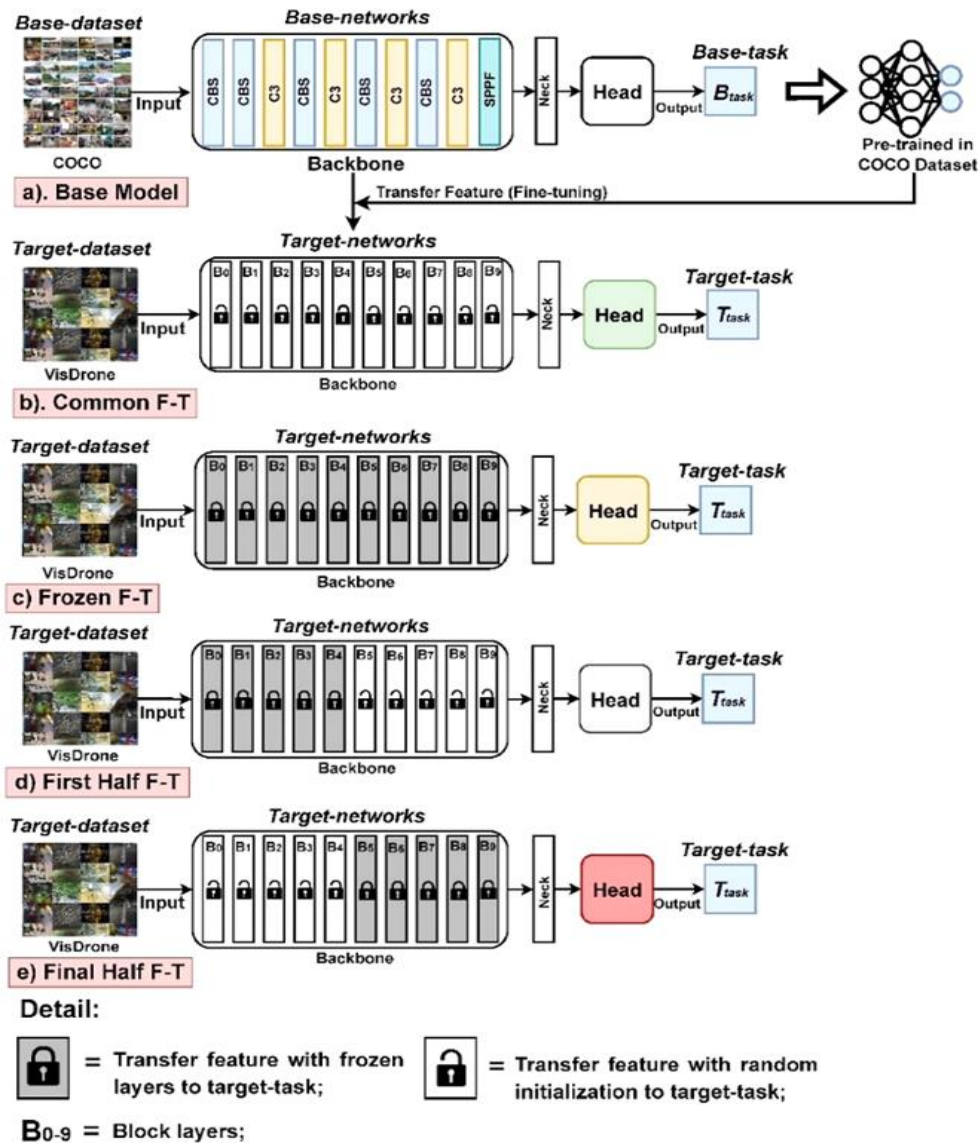


Figure 2. Fine-tuning strategy, (a) The base model in base-task uses a pre-trained model, (b) Common F-T, (c) Frozen F-T, (d) First half F-T, and (e) Final half F-T

In $B_{network}$ we leverage pre-trained YOLOv5 [49] on $B_{dataset}$ COCO [31] in B_{task} . The total layers in the backbone YOLOv5 are 10 blocks which we denote as B_{0-9} . Then a fine-tuning process is carried out to $T_{network}$ by training it to $T_{dataset}$ VisDrone [41] to complete the T_{task} object detection on the UAVs. The details of each fine-tuning strategy, including the spatial half fine-tuning: First half F-T, and Final half F-T that we propose illustrated in Figure 2. With the following details,

- *Common fine-tuning (Common F-T)*: transfer feature $B_{network}$ from pre-trained model to $T_{network}$ with condition parameter feature B_{0-9} on $T_{network}$ initialized randomly when trained with $T_{dataset}$ in specific T_{task} . The details of the illustration are explained in Figure 2(b).
- *Frozen fine-tuning (Frozen F-T)*: transfers feature $B_{network}$ from pre-trained model to $T_{network}$ with condition B_{0-9} on $T_{network}$ frozen, meaning that there is no process of changing parameters in $T_{network}$ during training process with $T_{dataset}$ for T_{task} . The details of the illustration are explained in Figure 2(c).
- *First half fine-tuning (First half F-T)*: transfers feature $B_{network}$ from pre-trained model to $T_{network}$ with condition B_{0-4} on $T_{network}$ frozen. Then B_{5-9} initialized randomly, meaning that during the training process on $T_{dataset}$ on T_{task} there was no process of changing the parameters of $T_{network}$ on B_{0-4} . However, in B_{5-9} there are changes during the training process. The details of the illustration are explained in Figure 2(d).
- *Final half fine-tuning (Final half F-T)*: is the opposite of First half F-T, in Final half F-T transfer feature $B_{network}$ from pre-trained model to $T_{network}$ with condition B_{5-9} on $T_{network}$ in frozen, and B_{0-4} initialized randomly, where during the training process with $T_{dataset}$ on T_{task} parameter changes to $T_{network}$ only occur in B_{0-4} . The details of the illustration are explained in Figure 2(e).

3. EXPERIMENTAL

3.1. Dataset

To validate the effectiveness of the fine-tuning strategy, we used the dataset VisDrone2019-Det [50]. The VisDrone2019-Det dataset is the same as VisDrone2021-Det [40]. The dataset consists of ten object categories: (pedestrian, person, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motorcycle) with a total number of 10,209 images, 6,471 for training, 548 for validation, and 3,190 for testing. For all the fine-tuning techniques in this research, we train that on the VisDrone training set and evaluate them with a validation set.

3.2. Evaluation metrics

For the evaluation process of each fine-tuning strategy, we used several relevant parameters [51]. Including precision (P), recall (R), average precision (AP), and mean average precision (mAP). We measured mAP by setting the threshold value to 0.5 intersections over union (IoU). The details of each parameter are explained in (1)-(4), (1) explains P , (2) explains R , (3) explains AP , and (4) explains mAP .

$$\text{Precision (P)} = \frac{TP}{TP+FP} \quad (1)$$

$$\text{Recall (R)} = \frac{TP}{TP+FN} \quad (2)$$

$$AP = \sum_n (R_{n+1} - R_n) \max_{\tilde{R}: \tilde{R} \geq R_{n+1}} P(\tilde{R}) \quad (3)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (4)$$

Based on (1) and (2), where true positive (TP) is the detection correct from the ground truth bounding box, false positive (FP) is the object that was detected but misplaced, and false negative (FN) is the ground truth of the bounding box not identified. AP is the average value of P and R as shown in (3) $P(\tilde{R})$ is the measured P at R . Then mAP is the average of AP used to measure all class categories in the dataset and is a metric used to measure the accuracy of object detection. As shown in (4) AP_i is the AP in class i , and N is the total number of classes evaluated.

3.3. Experimental details

We use the pre-trained model YOLOv5x [49] as a base-task in the COCO dataset [31] same as that used in [36]. We transfer features from pre-trained models to each fine-tuning technique: Common F-T, Frozen F-T, First half F-T, and Final half F-T. Then we train each that technique to the VisDrone target-dataset [50] as an object detection task in UAVs. In the training phase, we use stochastic gradient descent as optimization

with momentum 0.9, batch size 16, learning rate 0.01, and training iterations of 50 epochs with 640×640 input. The framework for the training and validation process uses PyTorch with a Tesla T4 GPU.

3.4. Results and discussion

To reflect the scalability of each fine-tuning strategy. We analyze each fine-tuning approach with an input scale of 640×640, and we also involve traditional training (Traditional-T), that shown in (section 3.4.1). We analyze the training results with different input scales, shown in (section 3.4.2). In the last, we compare the best results in this study with one of the state-of-the-art methods that used the VisDrone validation set as the evaluation process and with the same input scale, shown in (section 3.4.3).

3.4.1. Analysis effect fine-tuning strategy

Table 1 presents the validation results for each fine-tuning strategy with an input scale of 640×640. Common F-T can achieve detection *mAP* accuracy 8.9% greater than Traditional-T. The deferences result of 8.9% from Traditional-T means Common F-T indicated more increases in generalization of the model than Frozen F-T that only increases by 3.5% *mAP* accuracy compared to Traditional-T. The low improvement from Frozen F-T that because the features transferred from the COCO base-task directly adjusted to the VisDrone target-task without setting random parameters, and no feature update during the training process to the target-task. For partial half fine-tuning: Final half F-T can achieve detection *mAP* accuracy 9.3% greater than Traditional-T, 5.8% from Frozen F-T, and at the same time also outperform Common F-T with a deference 0.4% more height, that results prove Final half F-T is the best strategy for fine-tuning in input scale of 640×640. But, the *mAP* accuracy for the first half F-T result of 5% is slightly lower than the result of Common F-T and 5.4% lower than the Final half F-T. Base on research conducted by Yosinski *et al.* [41] that shown a transition process when the features of the base-task transferred to the target-task. In our study, the low improvement from first half F-T that because the first half layer in the target-network learns general features but is more specific to the COCO dataset than to the VisDrone dataset in the target-task. While the Final half F-T proves that the last half layer is more common or matches the features of the COCO base task and VisDrone target task. The details of the detection from Table 1 described in Table 2. While Figure 3 shows one of the results of the visualization of detection with a validation set.

Table 1. Evaluation results with VisDrone validation set

Training	Input size	Precision	Recall	mAP_0.5
Traditional-T	640×640	43.8	33.8	32.8
Common F-T	640×640	53.3	40.9	41.7
Frozen F-T	640×640	47	36.8	36.3
First half F-T	640×640	48.3	36.6	36.7
Final half F-T	640×640	52.4	41.4	42.1

Table 2. Detection results with VisDrone validation set

Model	Pedestrian	People	Bicycle	Car	Van	Truck	Tricycle	Awn	Bus	Motor
Traditional-T	42.2	33.3	09	73.3	34.9	27.7	17.3	10.1	41	39.4
Common F-T	49.9	39.1	20.2	78.6	43.1	40.2	28.8	15.7	55.6	46.1
Frozen F-T	42.8	35.3	12.3	74.6	38.3	34.1	23.5	12.7	49.4	39.5
First half F-T	42.7	35.5	11.6	75.3	36.8	37.7	23.2	12.7	50.1	41
Final half F-T	49.9	40.1	19.4	78.6	44	40.1	30.2	15.5	54.8	48.3

3.4.2. Analysis on different scale

To obtain a more in-depth analysis, we evaluated each fine-tuning technique with different input scales, namely 416×416, 608×608, 832×832, and 960×960, as described in Table 3. Common F-T outperforms the results of Traditional-T, Frozen F-T, First half F-T, and particularly, Final half F-T by a difference of 0.1% on 416×416 and 608×608 scales. However, on the 832×832 and 960×960 scales, the final half F-T is superior to Traditional-T, Frozen F-T, first half F-T, and especially Common F-T by a difference of 0.5% on both scales. These results indicate that the Common F-T is slightly higher than the Final half F-T with a smaller input scale. However, based on our experiment, the results of Final half F-T are more robust with higher input scales than Common F-T, Traditional-T, Frozen F-T, and First half F-T.

3.4.3. Compare with state-of-the-art

In this section, we compare our proposed partial half fine-tuning: Final half F-T with one of the state-of-the-art methods employed by [48]. We only compared it with one previous work because the authors used the same input scale and VisDrone validation set for the evaluation process. We focus on the 832×832 scale

input because it is the best result from [48]. Such as described in Table 4, the results of Final half F-T are 20.3% greater than SlimYOLOv3-SPP3-50 and 19.7% than YOLOv3-SPP3. That result indicates our proposed Final half F-T is better than one of the previous studies.

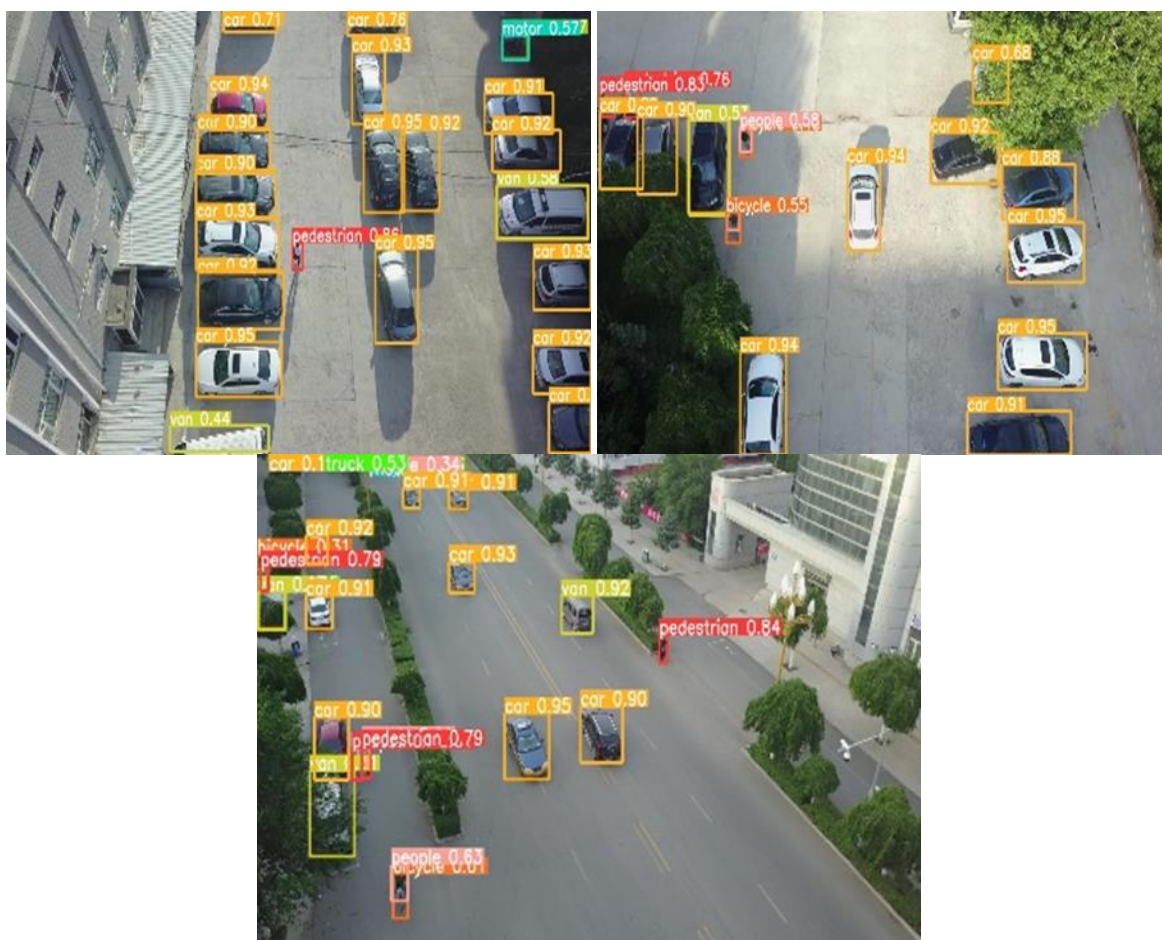


Figure 3. Some visualization results from our research

Table 3. Validation result at different scales using the VisDrone validation set

Model	Input	Precision	Recall	mAP 0.5
Traditional-T	416×416	36.6	25.5	24.2
	608×608	42.5	32.6	31.9
	832×832	44.1	38	36.3
	960×960	45	38.1	37.1
Common F-T	416×416	46.4	31.7	31.9
	608×608	52.7	40.4	41.1
	832×832	56.6	44.3	45.6
	960×960	57.7	45.6	46.8
Frozen F-T	416×416	39.4	26.6	25.9
	608×608	45.6	35.7	35.3
	832×832	49.3	40.9	40.4
	960×960	49.8	43.2	42.5
First half F-T	416×416	40.8	26.9	26.7
	608×608	49.1	35.2	35.9
	832×832	51.9	41.3	41.4
	960×960	53.1	43.5	43.4
Final half F-T	416×416	44.6	31.5	31.8
	608×608	51.6	39.9	41
	832×832	56.4	44.6	46.1
	960×960	55.8	46.3	47.3

Table 4. Results of comparison with one of the state-of-the-art methods

Model	Input	Precision	Recall	mAP 0.5
SlimYOLOv3-SPP3-50	832×832	45.9	36	25.8
YOLOv3-SPP3	832×832	43.5	38	26.4
Final half F-T	832×832	56.4	44.6	46.1

4. CONCLUSION




In this study, we conduct experimental analysis on every existing fine-tuning approach and propose a partial half fine-tuning strategy which consists of two techniques: First half F-T and Final half F-T. In the evaluation process, we used the VisDrone validation set. Here we show that the result of Final half F-T can achieve detection *mAP* accuracy 9.3% greater than Traditional-T, 5.8% from Frozen F-T, and 0.4% from Common F-T in an input scale of 640×640, and its also more accurate at higher scales, such as in scale 832×832 and 960×960. Then we compared the final half F-T with one of the state-of-the-art methods, based on the *mAP IoU* 0.5 and the same 832×832 input scale. Here we show that the results of final half F-T are 20.3% greater than SlimYOLOv3-SPP3-50 and 19.7% than YOLOv3-SPP3. That means our technique is better than other fine-tuning techniques and also better than one of the state-of-the-art methods in object detection with UAVs.

REFERENCES




- [1] P. Gupta, B. Pareek, G. Singal, and D. V. Rao, "Edge device based Military Vehicle Detection and Classification from UAV," *Multimedia Tools and Applications*, vol. 81, no. 14, pp. 19813–19834, 2022, doi: 10.1007/s11042-021-11242-y.
- [2] C. Yuan, Z. Liu, and Y. Zhang, "Aerial Images-Based Forest Fire Detection for Firefighting Using Optical Remote Sensing Techniques and Unmanned Aerial Vehicles," *Journal of Intelligent & Robotic Systems*, vol. 88, no. 2–4, pp. 635–654, Dec. 2017, doi: 10.1007/s10846-016-0464-7.
- [3] U. R. Mogili and B. B. V. L. Deepak, "Review on Application of Drone Systems in Precision Agriculture," *Procedia Computer Science*, vol. 133, pp. 502–509, 2018, doi: 10.1016/j.procs.2018.07.063.
- [4] A. Singh, D. Patil, and S. N. Omkar, "Eye in the Sky: Real-Time Drone Surveillance System (DSS) for Violent Individuals Identification Using ScatterNet Hybrid Deep Learning Network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2018, vol. 2018-June, pp. 1710–17108, doi: 10.1109/CVPRW.2018.00214.
- [5] X. Zhang and K. Kusurini, "Autonomous long-range drone detection system for critical infrastructure safety," *Multimedia Tools and Applications*, vol. 80, no. 15, pp. 23723–23743, Jun. 2021, doi: 10.1007/s11042-020-10231-x.
- [6] W. L. Leong, N. Martinel, S. Huang, C. Micheloni, G. L. Foresti, and R. S. H. Teo, "An Intelligent Auto-Organizing Aerial Robotic Sensor Network System for Urban Surveillance," *Journal of Intelligent & Robotic Systems*, vol. 102, no. 2, p. 33, Jun. 2021, doi: 10.1007/s10846-021-01398-y.
- [7] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
- [8] X. Wu, W. Li, D. Hong, R. Tao, and Q. Du, "Deep Learning for Unmanned Aerial Vehicle-Based Object Detection and Tracking: A survey," *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 1, pp. 91–124, Mar. 2022, doi: 10.1109/MGRS.2021.3115137.
- [9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Neural Information Processing Systems*, 2012, pp. 1106–1114.
- [10] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.
- [11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [12] C. Szegedy et al., "Going deeper with convolutions," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9908 LNCS, pp. 630–645, 2016, doi: 10.1007/978-3-319-46493-0_38.
- [15] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2020, vol. 2020-June, pp. 1571–1580, doi: 10.1109/CVPRW50498.2020.00203.
- [16] M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," May 2019, doi: 10.48550/arXiv.1905.11946.
- [17] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 580–587, 2014, doi: 10.1109/CVPR.2014.81.
- [18] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010, doi: 10.1109/TPAMI.2009.167.
- [19] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun, "Bottom-Up Segmentation for Top-Down Detection," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2013, pp. 3294–3301, doi: 10.1109/CVPR.2013.423.
- [20] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010, doi: 10.1007/s11263-009-0275-4.
- [21] R. Girshick, "Fast R-CNN," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.

- [22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.
- [23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
- [24] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.
- [25] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, [Online]. Available: <http://arxiv.org/abs/1804.02767>.
- [26] W. Pebrianto, P. Mudjirahardjo, and S. H. Pramono, "YOLO Method Analysis and Comparison for Real-Time Human Face Detection," in *2022 11th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, Aug. 2022, pp. 333–338, doi: 10.1109/EECCIS54468.2022.9902919.
- [27] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, [Online]. Available: <http://arxiv.org/abs/2004.10934>.
- [28] C. Y. Wang, A. Bochkovskiy, and H. Y. M. Liao, "Scaled-yolov4: Scaling cross stage partial network," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 13024–13033, 2021, doi: 10.1109/CVPR46437.2021.01283.
- [29] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020, doi: 10.1109/TPAMI.2018.2858826.
- [30] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, pp. 21–37.
- [31] T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8693 LNCS, no. PART 5, pp. 740–755, 2014, doi: 10.1007/978-3-319-10602-1_48.
- [32] X. Zhang, E. Izquierdo, and K. Chandramouli, "Dense and small object detection in UAV vision based on cascade network," *Proceedings - 2019 International Conference on Computer Vision Workshop, ICCVW 2019*, pp. 118–126, 2019, doi: 10.1109/ICCVW.2019.00020.
- [33] J. Dai *et al.*, "Deformable Convolutional Networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, vol. 2017-Octob, pp. 764–773, doi: 10.1109/ICCV.2017.89.
- [34] C. Chen *et al.*, "RRNet: A Hybrid Detector for Object Detection in Drone-Captured Images," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Oct. 2019, pp. 100–108, doi: 10.1109/ICCVW.2019.00018.
- [35] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in Vision: A Survey," *ACM Computing Surveys*, vol. 54, no. 10s, pp. 1–41, Jan. 2022, doi: 10.1145/3505244.
- [36] X. Zhu, S. Lyu, X. Wang, and Q. Zhao, "TPH-YOLOv5: Improved YOLOv5 Based on Transformer Prediction Head for Object Detection on Drone-captured Scenarios," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Oct. 2021, pp. 2778–2788, doi: 10.1109/ICCVW54120.2021.00312.
- [37] Z. Zhang, X. Lu, G. Cao, Y. Yang, L. Jiao, and F. Liu, "ViT-YOLO:Transformer-Based YOLO for Object Detection," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Oct. 2021, vol. 2021-Octob, pp. 2799–2808, doi: 10.1109/ICCVW54120.2021.00314.
- [38] A. Vaswani *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, pp. 5999–6009, 2017.
- [39] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 10778–10787, doi: 10.1109/CVPR42600.2020.01079.
- [40] Y. Cao *et al.*, "VisDrone-DET2021: The Vision Meets Drone Object detection Challenge Results," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Oct. 2021, vol. 2021-Octob, pp. 2847–2854, doi: 10.1109/ICCVW54120.2021.00319.
- [41] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Neural Information Processing Systems*, 2014, vol. 27, pp. 3320–3328.
- [42] W. Ouyang, X. Wang, C. Zhang, and X. Yang, "Factors in Finetuning Deep Model for Object Detection with Long-Tail Distribution," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, vol. 2016-Decem, pp. 864–873, doi: 10.1109/CVPR.2016.100.
- [43] X. Li, Y. Grandvalet, and F. Davoine, "Explicit inductive bias for transfer learning with convolutional networks," in *35th International Conference on Machine Learning, ICML 2018*, Feb. 2018, vol. 6, pp. 4408–4419.
- [44] M. Wortsman *et al.*, "Robust fine-tuning of zero-shot models," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 7949–7961, doi: 10.1109/CVPR52688.2022.00780.
- [45] S. X. Hu, D. Li, J. Stuhmer, M. Kim, and T. M. Hospedales, "Pushing the Limits of Simple Pipelines for Few-Shot Learning: External Data and Fine-Tuning Make a Difference," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 9058–9067, doi: 10.1109/CVPR52688.2022.00886.
- [46] S. Niu, Y. Liu, J. Wang, and H. Song, "A Decade Survey of Transfer Learning (2010–2020)," *IEEE Transactions on Artificial Intelligence*, vol. 1, no. 2, pp. 151–166, Oct. 2020, doi: 10.1109/TAI.2021.3054609.
- [47] G. S. Dhillon, P. Chaudhari, A. Ravichandran, and S. Soatto, "A Baseline for Few-Shot Image Classification," Sep. 2019.
- [48] P. Zhang, Y. Zhong, and X. Li, "SlimYOLOv3: Narrower, Faster and Better for Real-Time UAV Applications," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Oct. 2019, pp. 37–45, doi: 10.1109/ICCVW.2019.00011.
- [49] G. Jocher *et al.*, "ultralytics/yolov5: v6.2 - YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations," Aug. 2022, doi: 10.5281/ZENODO.7002879.
- [50] D. Du *et al.*, "VisDrone-DET2019: The Vision Meets Drone Object Detection in Image Challenge Results," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Oct. 2019, pp. 213–226, doi: 10.1109/ICCVW.2019.00030.
- [51] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Jul. 2020, vol. 2020-July, pp. 237–242, doi: 10.1109/IWSSIP48289.2020.9145130.




BIOGRAPHIES OF AUTHORS

Wahyu Pebrianto    Wahyu Pebrianto received his bachelor degree in Department of Information Technology Politeknik Negeri Jember in 2020. Now he is pursuing Master degree in Universitas Brawijaya. His main research focus is deep learning, computer vision, and pattern recognition. He can be contacted at email: wahyu.pebrianto1@gmail.com.



Panca Mudjirahardjo    Panca Mudjirahardjo received the B.Eng. Degree in Electrical Engineering from Universitas Brawijaya, Indonesia, in 1995. M.Eng. Degree in Electrical Engineering from Universitas Gadjah Mada, Indonesia, in 2001 and Dr. Eng. degree in Control Engineering (Machine Intelligence Lab.) from Kyushu Institute of Technology, Japan, in 2015. He joined as Assistant Engineer of Production Engineering Dept. at PT. Asahi Electronics Indonesia, a Telephone Answering Device manufacturer, in 1995-1998; as Engineering Assistant of Engineering Dept. at PT. Tokyo Pigeon Indonesia, an Audio Mechanism manufacturer, in 1998-1999. Currently he is with Electrical Engineering Dept. at Universitas Brawijaya, since 2002. His current research interests include Digital and Analog Instrumentation System Design, Pattern Recognition, Image Processing and Computer Vision. He can be contacted at email: panca@ub.ac.id.



Sholeh Hadi Pramono    Sholeh Hadi Pramono was born in 1958. He obtained his bachelor degree in 1985 majoring in electrical power system from Universitas Brawijaya. In 1990 and 2009 He obtained his Master and Doctoral degree respectively from Universitas Indonesia majoring in Optoelectrotechniques and laser application. His major research is in optical communication, photovoltaic and artificial intelligence. He has been a lecturer in Universitas Brawijaya since 1986. He can be contacted at email: sholehpramono@ub.ac.id.