Drone Detection†

†Note: This paper is a progress report for Introduction to Computer Vision at Innopolis University for Fall 2021

Ahmad Hamdan*
Robotics Institute
Innopolis University
Russia
a.hamdan@innopolis.university

Hany Hamed*
Robotics Institute
Innopolis University
Russia
h.hamed@innopolis.university

***Abstract*—Object detection refers to the capability of computer and software systems to locate objects in an image/scene and identify each object. Drone detection is identifying whether there is a drone in the image and locating if it exists. Such problem could be challenging due to the wide variety in shapes of drones. This paper presents we are tackling the problem of detecting drones using**

***Index Terms*—component, formatting, style, styling, insert**

I. INTRODUCTION

Understanding the environment through the camera is a vital part to interact with the surroundings. Object detection is a fundamental task to detect and identify the objects in the environment, it includes finding the objects in the image and at the same time classify these objects. Furthermore, performing object detection in real-time is an important aspect for multiple of fields such as autonomous vehicles where the robot must identify the object in the real-time to plan how to interact with that object.

Deep neural networks (DNN) contributed greatly in solving object detection problem, DNN proved robustness and efficiency that make it usable in real-life situations [STE13]. Different architectures were implemented such as YOLO [Red+15] and RCNN [Gir+14], however, in this study we are focusing on YOLOv4 [BWL20], YOLOv5 [ult21], and Faster-RCNN [Ren+15].

In this paper, we are discussing a certain class of object detection, which concerns for the detection and the classification of drones in images. Detecting drones has multiple of applications, for example, in defence systems to detect drones and try to destroy them (e.g. in prey and predator scenario [AAB19] where we have two drones or more that one of them is runner and the other is chaser, thus the chaser must identify and track the other drone which is the runner).

This paper is mainly discussing object detection for drones, thus, this work contributes in 3 main points:

- Collecting a diverse dataset for drones
- Train and benchmark using the collected dataset the different object detection models: YOLOv4-Tiny, YOLOv4-Scaled, YOLOv5, and Faster-RCNN. The benchmark will be based on mAP (Mean Average Precision) mainly mAP@50 (i.e. Mean Average Precision when Intersection over Union (IoU) threshold is 0.5) and inference time.
- Perform domain adaptation (DA) [zha21] for a target domain that is similar to the source domain that the model is trained with. Different domains can be considered such as: training domain is a certain type of drones from a specific company and the target domain is different shapes and types of drones. Another direction to apply DA is using a source domain of light images and target domain with low conditioned lightning and night conditions [SN20] and identifying the drones in that bad conditions.

* Equal contributions

II. RELATED WORK

A. Dataset

Datasets are important for supervised learning, in object detection task, the dataset consists of images and labels such that the labels are the bounding boxes that surrounding the objects (described in top-left and bottom-right coordinates, width and height) and the class label. In our case, the drone is the only class label in our dataset.

We have found two main datasets that dedicated for drone detection.

1) MCL dataset [Che+17], this dataset consists of thermal detection dataset for the drone and the rest of the dataset consists of local footages for a drone inside a campus of the university in addition to snapshots from different videos. Collecting datasets from videos is an approach that multiple datasets has used in order to collect sufficient of data, however, sometimes it lacks the generalization of the scenes, and it suffers from the repetition of the same frame.
2) Amateur dataset [Aks19], this dataset consists of 300 images for a specific type of drones which is DJI-Phantom and 1300 images for snapshots from videos that in some videos the quality of the video and the point of view of the camera is hard for the human to identify the drone. But these images has a diverse set of drones such as tricopter, quadcopter, hexacopter, and octacopter. However, the only downside for this dataset that 200 out of 300 of DJI-phantom images are unlabelled and

approximately 400 out of 1300 for the other images are unlabelled as well.

B. YOLO

YOlO (You Only Looks Once) [Ism+21] is one of the famous architectures due to the real-time performance. For the current moment, there are multiple of versions, however, we are only considering YOLOv4 and YOLOv5.

C. R-CNN

R-CNN (Recursive Convolution Neural Networks)[Gir+14] is as well a known architecture for object detection. A library from Facebook dectron2 [Wu+19] has implemented different models that can be used as pretrained models. Currently, there are multiple of different models, we are only considering Fast-RCNN [Gir15] and Faster-RCNN [Ren+15].

D. Domain Adaptation

Domain adaptation is a branch of machine learning that concerns about training in a source domain and then testing on a different domain but similar to the source [RSF18; SN20]. In this paper, we are applying domain adaptation were our source domain is a dataset for a specific shape of drones (DJI phantom) and the target is a different dataset of different drones from different companies. That will enable us to train on a specific dataset for a drone that is wide-spread globally with less amount of data from other drones, then use the model to detect other different drones. Moreover, this work can be extended in order to perform domain adaptation on drones as well, but when the target domain is in a dark poor light scenarios.

III. Implementation

A. Dataset

TODO: Put roboflow in the citation here We have found multiple data-sets for drones, however, most of them are just screenshots from a video that only contributes for one or two drones with specific backgrounds with low quality and low resolution. There is only one dataset that had good quality and different scenes for the drone, There is a dataset that includes mostly one specific type of drones, Dji phantom. This dataset included more than 200 images that are not labelled and we uploaded the dataset on roboflow and annotated them.

B. Benchmark

We trained YOLO v4 Tiny and dectron2 Faster RCNN on the dataset so far.

1) YOLO: For training YOLO v4 tiny, we followed the exact same structures provided in AlexeyAB and here is our script. For training we used two different batch size just to spot if there is any difference. in the first chart we set the batch size to 128, but for the second time we used 32 batch size, which made the training significantly faster. In both cases, we used the model with the highest MAP. The results are described in IV section

Yolo V4 scaled , Yolo v5 (graphs and comparison)

2) Faster-RCNN: For training Faster-RCNN, we have used https://github.com/facebookresearch/detectron2 and here is our training script. We have used R101-FPN model as it provides less training memory and less inference time, later we will test and benchmark against different models

C. Domain Adaptation

TODO: Explains our hypothesis and the current findings and briefly explain the results in the next section

IV. Experiments And Results

A. Faster-RCNN

In the following figures, we provide the results of testing Faster-RCNN after 20 minutes of training.

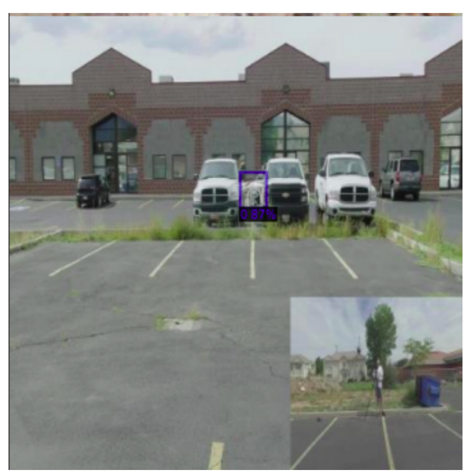
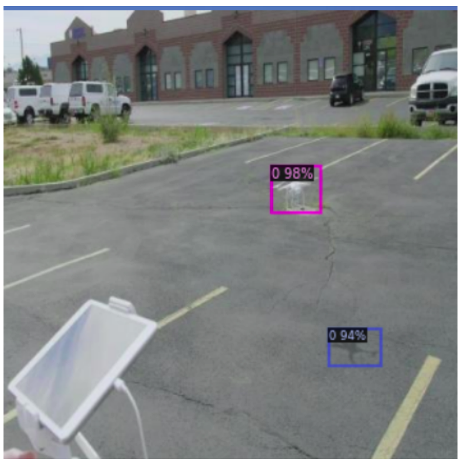


TABLE I
BENCHMARK RESULTS FOR DIFFERENT OBJECT DETECTION MODELS FOR DRONE DETECTION

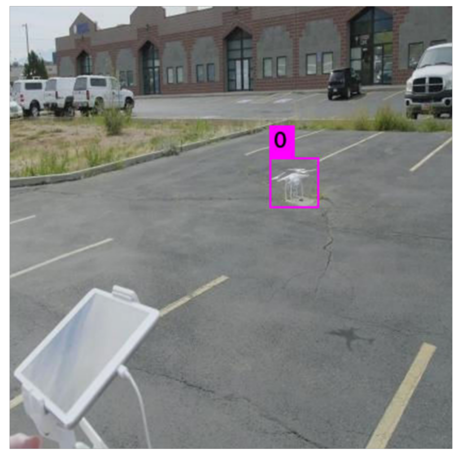
Model	backbone	AP	AP@50	Inference time (sec)
Faster R-CNN [He+16]	ResNet-101-C4	42.711	92.484	0.2546
Faster R-CNN w FPN [Lin+17]	ResNet-101-FPN	45.637	95.128	0.0909
YOLOv4 [Jia+20]	Tiny	**57.47**	91.8	**0.00299**
YOLOv4 [WBL21]	Scaled	36.9	83.6	0.024
YOLOv5 [ult21]	S	52.4	**98.4**	0.006



We can see from the testing output that this method is has a below average accuracy, especially in such cases where there aren't much details in the image or the drone is very far away. For example in one of the tests the model detected the shadow in the image as a drone with 0.94 confidence.

B. YOLO v4 tiny

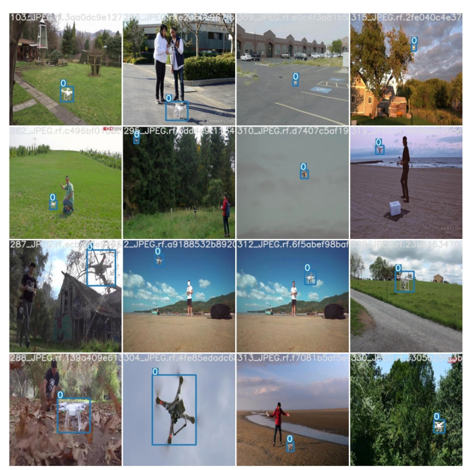
In the following figures, we provide the results of testing Yolo v4 tiny.



We can notice better performance as this model does not detect the shadow or the people as drones in both images.

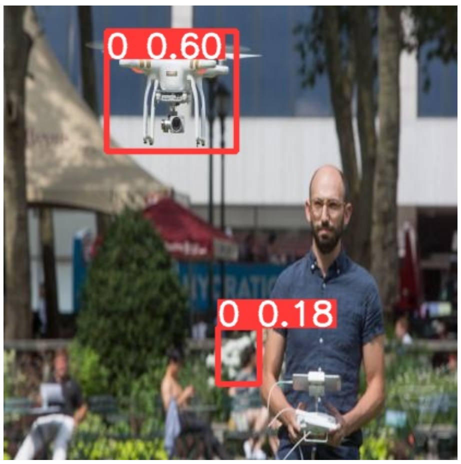
C. YOLO v4 scaled

In the following figures, we provide the results of testing Yolo v4 scaled.



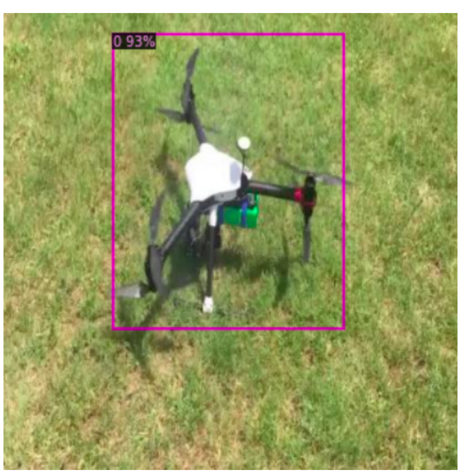
D. YOLO v5

In the following figures, we provide the results of testing Yolo v5.

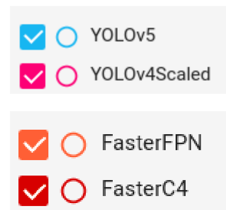


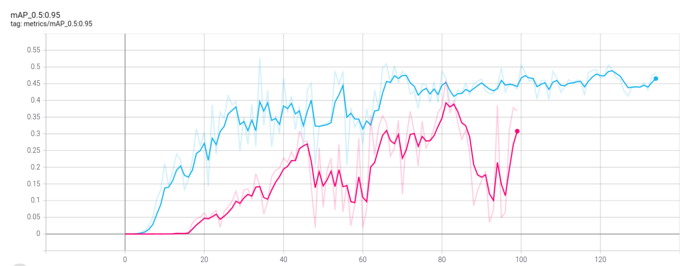
E. Different datasets

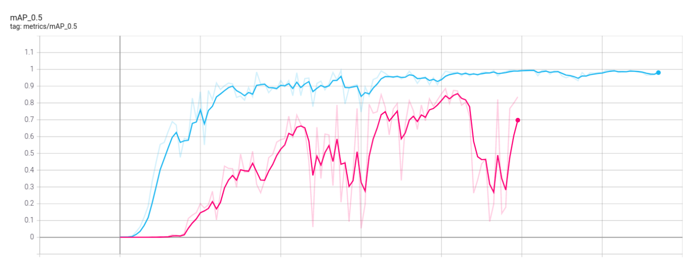
we tried to train all models on DJI phantom dataset only, then we tested it on images that contains different types of drones (different color, shape, number of fans). All models were able to detect the new drone with an acceptable accuracy, therefore, we decided that we will accept this result and continue with domain adaptation as a future work.

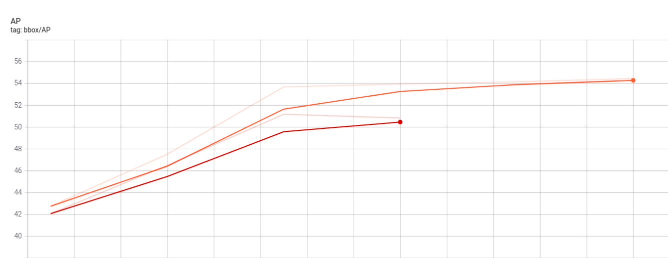


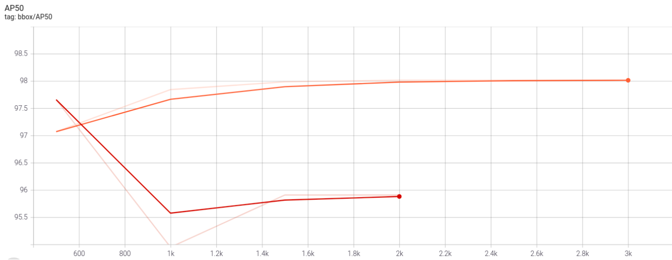
TODO: Modify the plots











V. Future work

VI. Conclusions

This is basically the start of the project, we will continue working and experimenting with the models we have already. On the other hand, we will try YOLO V5 and compare all the results that we will get. More over, we are planning to expand the dataset that we already have as we believe it is not sufficient enough and it will be a good contribution to the open source community anyway. And lastly, we are going to the most interesting part of the project which is domain adaptation and see how it will preform on our problem.

References

[STE13] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. "Deep neural networks for object detection". In: (2013).

[Gir+14] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[Gir15] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[Red+15] J Redmon et al. "You only look once: Unified, real-time object detection. arXiv 2015". In: *arXiv preprint arXiv:1506.02640* (2015).

[Ren+15] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015), pp. 91–99.

[He+16] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[Che+17] Yueru Chen et al. "A deep learning approach to drone monitoring". In: *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE. 2017, pp. 686–691.

[Lin+17] Tsung-Yi Lin et al. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[RSF18] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. "Beyond sharing weights for deep domain adaptation". In: *IEEE transactions on pattern analysis and machine intelligence* 41.4 (2018), pp. 801–814.

[AAB19] Moulay A Akhloufi, Sebastien Arola, and Alexandre Bonnet. "Drones chasing drones: Reinforcement learning and deep search area proposal". In: *Drones* 3.3 (2019), p. 58.

[Aks19] Mehmet Cagri Aksoy. *Amateur Unmanned Air Vehicle Detection*. 2019. URL: https://www.kaggle.com/mcagriaksoy/amateur-unmanned-air-vehicle-detection-dataset.

[Wu+19] Yuxin Wu et al. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019.

[BWL20] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection". In: *arXiv preprint arXiv:2004.10934* (2020).

[Jia+20] Zicong Jiang et al. "Real-time object detection method based on improved YOLOv4-tiny". In: *arXiv preprint arXiv:2011.04244* (2020).

[SN20] Yukihiro Sasagawa and Hajime Nagahara. "Yolo in the dark-domain adaptation method for merging multiple models". In: *European Conference on Computer Vision*. Springer. 2020, pp. 345–359.

[Ism+21] Amir Ismail et al. "Performance Benchmarking of YOLO Architectures for Vehicle License Plate Detection from Real-time Videos Captured by a Mobile Robot." In: *VISIGRAPP (5: VISAPP)*. 2021, pp. 661–668.

[ult21] ultralytics. *ultralytics/yolov5: YOLOv5 in PyTorch ¿ ONNX ¿ CoreML ¿ TFLite*. Oct. 2021. URL: https://github.com/ultralytics/yolov5.

[WBL21] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. "Scaled-yolov4: Scaling cross stage partial network". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 13029–13038.

[zha21] zhaoxin94. *zhaoxin94/awesome-domain-adaptation: A collection of AWESOME things about domian adaptation*. Oct. 2021. URL: https://github.com/zhaoxin94/awesome-domain-adaptation.

Appendix

The tasks that was covered during the project:

- Finished training the models (YOLO v5)
- Finished training the models (Fast-RCNN)
- Prepared the benchmark based on the criteria that was mentioned earlier
- Test the models on the target dataset and check the result and test the hypothesis for the need of domain adaptation