# Drone Detection†

†Note: This paper is a progress report for Introduction to Computer Vision at Innopolis University for Fall 2021 and there is always a place for improvment

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 task is about identifying whether there is a drone in the image and locating if it exists. Such a problem could be challenging due to the wide variety in shapes of drones and the lack of the dataset. This paper presents the findings and the approaches we have used to solve this problem, moreover we provided a benchmark for different object detection architectures.**

***Index Terms*—object detection, YOLO, R-CNN, drones, domain adaptation**

## 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.

The implementation code of this paper alongside the results' logs are available at `https://github.com/hany606/Drone-Detection`

## 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

* Equal contributions

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 that are used for object detection and object recognition. YOLO detects the object by a single forward propagation of the image to the network, it provides the bounding boxes for the objects alongside the classification probability. There are multiple of versions, however, we are only considering YOLOv4-Tiny, YOLOv4-Scaled and YOLOv5.

### C. R-CNN

R-CNN (Recursive Convolution Neural Networks) [Gir+14] is an architecture that is used for object detection task. R-CNN solves the problem of selecting a huge number of regions by proposing a method based on selective search. Furthermore, Fast R-CNN and improves the performance of regular R-CNN by feeding the image to a CNN first then perform the selective search and Faster R-CNN uses a region proposal network to catch the candidate regions. A library from Facebook dectron2 [Wu+19] has implemented different models that can be used as pretrained models. In this paper, we are only considering Faster R-CNN [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]. A previous work [SN20] has discussed the application of domain adaptation with YOLO to detect books with poor lightning conditions. Moreover, another work discussed the usage of Faster R-CNN for domain adaptation [Che+18].

## III. Implementation

### A. Dataset

The mentioned datasets in the previous section have the problem of unlabelled datat and the quatlity and the diversity of the dataset. Thus, we had to label all the unlabelled data in Amateur dataset and used it for our training.

We have used [21] in order to label the data and to export it for different models for object detection. This tool enabled us to annotate our data and use it as fast as possible.

### B. Benchmark

We have used our dataset to train different architectures (Faster R-CNN, YOLOv4-Tiny, YOLOv4-Scaled, and YOLOV5) then evaluate the performance of the training on the testing dataset.

For YOLO, we have prepared training scripts based on darknet library [Ale21] and we have tested YOLOv4-Tiny, YOLOv4-Scaled, and YOLOv5. Moreover, for training Faster-RCNN, we have used dectron2 [Wu+19] and we have tested two models based on ResNet-101-C4 and ResNet-101-FPN.

### C. Domain Adaptation

In this paper, the main intention is to test a hypothesis that answers tha following question: "Do we need to apply domain adaptation to train on a source dataset that consists of a single type of drone and test on a target dataset that includes multiple types of drones?"

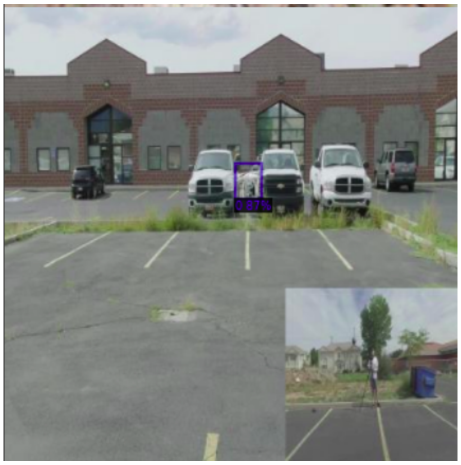
The motivation behind of answering this question was derived from the existing findings of the dataset, as we found that most of the datasets only include quadcopters and especially DJI-Phantom, however, there are multiple of types of drones in the real life.

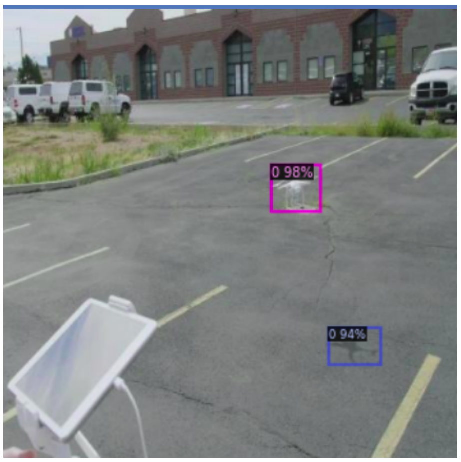
In the following section, we will discuss the findings of this question.

## IV. Experiments And Results

### A. Faster-RCNN

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

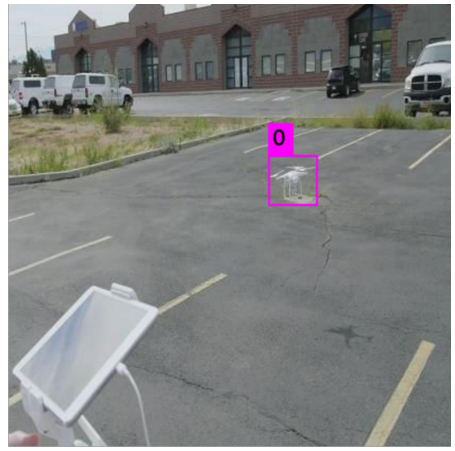





We can see from the testing output that this method is has a below average accuracy, especially in such cases where there are not much details in the image or the drone is very far away. For example in one of the tests the model detected the shadow in the image as a drone with 0.94 confidence.

### B. YOLO v4 tiny

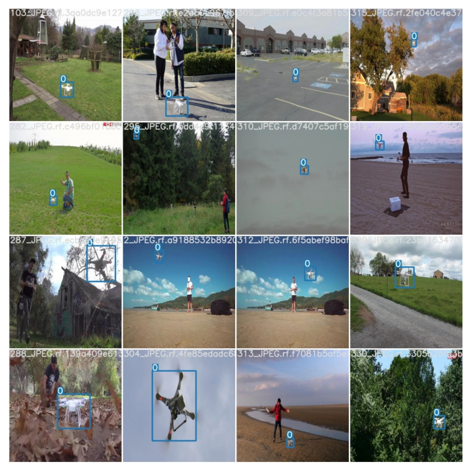
In the following figures, we provide the results of testing Yolo v4 tiny.


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

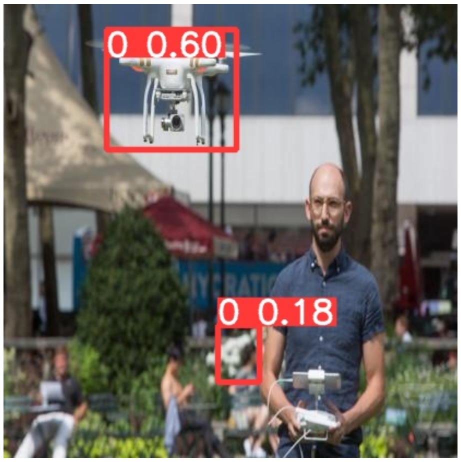
### C. YOLO v4 scaled

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.

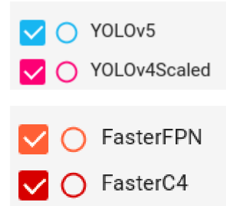



### E. Different datasets

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.



In the following figures, we can see the mean average precision plots during the training for the different models.



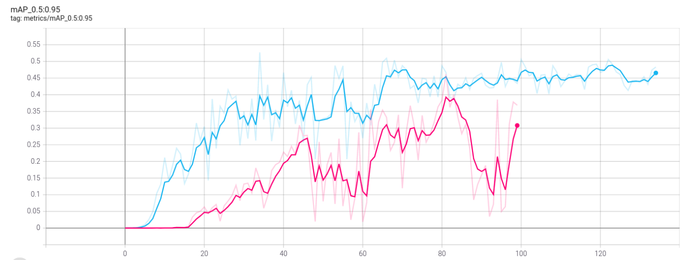




Fig. 1. Mean Average Precision for YOLO

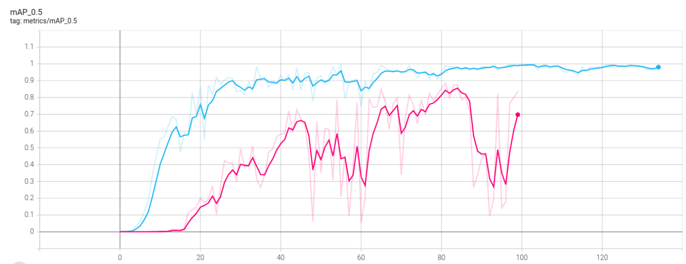



Fig. 2. Mean Average Precision for YOLO at IoU=0.5

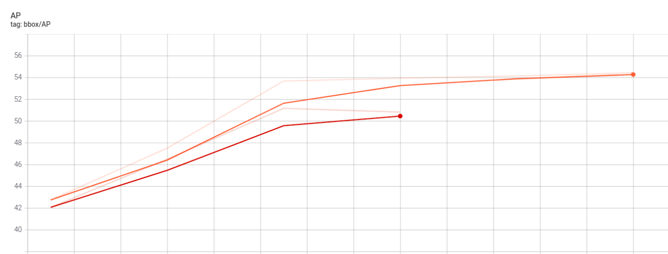



Fig. 3. Mean Average Precision for Faster R-CNN

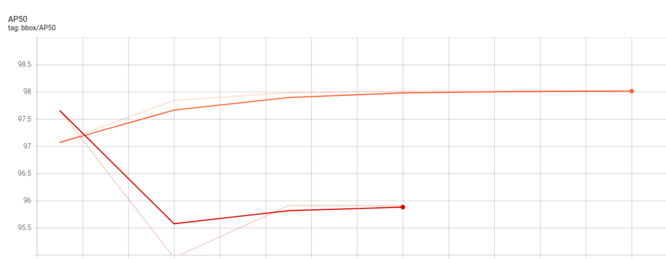



Fig. 4. Mean Average Precision for Faster R-CNN at IoU=0.5

We provide a table I that is evaluated on the source dataset with the following results:

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 |

Furthermore, we have tested the trained model of Faster R-CNN that is trained with the source dataset on the target dataset, and we got the following results in II

As we can see from II, the results have changed in the mean average precision, but still it is a satisfactory result. However, there is a space for improvement and increasing the robustness of the detection using domain adaptation.

## V. FUTURE WORK

The dataset can be extended later with more diverse images for different drones (e.g. VTOL) and different scenes. Moreover, some augmentations can be made to proudce more diverse background, an interesting idea is to extract several drones from scenes and using a background images from a public dataset for indoor navigation [KB18]. Domain adaptation is a interesting domain, thus, we can start to apply DA to this problem as we mentioned earlier.

## 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.

[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.

[Che+18] Yuhua Chen et al. "Domain Adaptive Faster R-CNN for Object Detection in the Wild". In: *Computer Vision and Pattern Recognition (CVPR)*. 2018.

[KB18] Alexandros Kouris and Christos-Savvas Bouganis. "Learning to fly by myself: A self-supervised cnn-based approach for autonomous navigation". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 1–9.

[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.

TABLE II
RESULTS OF FASTER R-CNN ON TARGET DATASET

| Model | backbone | AP | AP@50 | AP@75 | $AP_s$ | $AP_m$ | $AP_l$ |
|---|---|---|---|---|---|---|---|
| Faster R-CNN [He+16] | ResNet-101-C4 | 30.674 | 64.171 | 23.29 | 26.703 | 35.684 | 34.542 |
| Faster R-CNN w FPN [Lin+17] | ResNet-101-FPN | 32.879 | 64.559 | 28.638 | 30.79 | 33.951 | 38.4 |


[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.

[21] 2021. URL: https://roboflow.com/.

[Ale21] AlexeyAB. *AlexeyAB/darknet: YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet )*. Sept. 2021. URL: https://github.com/AlexeyAB/darknet.

[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