Air-to-Air Visual Detection of Micro-UAVs: An Experimental Evaluation of Deep Learning

Ye Zheng, Zhang Chen, Dailin Lv, Zhixing Li, Zhenzhong Lan, Shiyu Zhao

Abstract*—This paper studies the problem of air-to-air visual detection of micro unmanned aerial vehicles (UAVs) by monocular cameras. This problem is important for many applications such as vision-based swarming of UAVs, malicious UAV detection, and see-and-avoid systems for UAVs. Although deep learning methods have exhibited superior performance in many object detection tasks, their potential for UAV detection has not been well explored. As the first main contribution of this paper, we present a new dataset, named Det-Fly, which consists of more than 13,000 images of a flying target UAV acquired by another flying UAV. Compared to the existing datasets, the proposed one is more comprehensive in the sense that it covers a wide range of practical scenarios with different background scenes, viewing angles, relative distance, flying altitude, and lightning conditions. The second main contribution of this paper is to present an experimental evaluation of eight representative deep-learning algorithms based on the proposed dataset. To the best of our knowledge, this is the first comprehensive experimental evaluation of deep learning algorithms for the task of visual UAV detection so far. The evaluation results highlight some key challenges in the problem of air-to-air UAV detection and suggest potential ways to develop new algorithms in the future. The dataset is available at *https://github.com/Jake-WU/Det-Fly*.

***Index Terms*—UAV detection; Visual detection; Deep learning**

I. Introduction

VISUAL detection of micro unmanned aerial vehicles (UAVs) has attracted increasing attention in recent years since it is the core technology for many important applications. For example, visual detection of UAVs is essential to achieve vision-based UAV swarming systems, where each UAV needs to use onboard cameras to measure the relative motion of their neighboring UAVs [1]. In addition, the hostile use of micro UAVs has become a serious threat to public safety and personal privacy nowadays. Visual detection of malicious micro UAVs [2], [3] is a key technology for developing civilian UAV defense systems. Another application is see-and-avoid among UAVs [4]. In particular, as more and more commercial UAVs occupy low-altitude airspace for the purpose of, for example, parcel delivery, how to ensure UAVs to detect other UAVs timely to navigate safely without colliding with each other is an important problem.

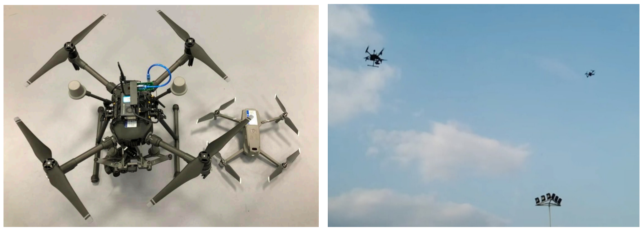


Fig. 1. A DJI M210 platform with XT2 camera was used to acquire images of a flying target UAV (DJI Mavic).

The detection of UAVs could be classified into two application scenarios. The first is ground-to-air, where cameras are placed on the ground to detect flying UAVs. The second scenario is air-to-air, where a flying UAV uses its onboard cameras to detect other flying UAVs (see, for example, Fig. 1). This paper focuses on the air-to-air scenario. In addition, although different types of sensors could be used to detect micro UAVs such as vision, radar [5], and acoustic sensors [6], visual sensors are one of the few suitable options for the air-to-air scenario due to the extremely limited onboard payload of micro UAVs. This paper focuses on the most widely used RGB monocular cameras.

While ground-to-air UAV detection has attracted increasing research attention in recent years (see Section II for a review), the air-to-air case, which is even more challenging, is far from being well solved up to now. In many ground-to-air UAV detection tasks, ground cameras are usually stationary or moving slowly [7], and the background of target UAV images is a clear or cloudy sky. As a comparison, in an air-to-air UAV detection task, a flying UAV may observe the target UAV from top or side view angles. As a result, the background of the target UAV image could be extremely complex scenes such as urban and natural fields (see Fig. 2 for example). Moreover, since the onboard camera is flying dynamically, the appearance of the target UAV such as its shape, scale, and color may vary dramatically. Since micro UAVs are small in size, their images may be extremely small (e.g., less than 10×10 pixels), which thus increases the difficulty of detection.

The existing approaches for UAV detection could be clas-

Manuscript received: September 24, 2020; Revised November 19, 2020; Accepted January 17, 2021.

This paper was recommended for publication by Editor Pauline Pounds upon evaluation of the Associate Editor and Reviewers' comments. This work is supported by the National Natural Science Foundation of China (grant no. 61903308) and the Westlake University and Bright Dream Joint Institute for Intelligent Robotics.*(Corresponding author: Shiyu Zhao.)*

[1]Ye Zheng is with the Department of Computer Science & Technology at Zhejiang University and the School of Engineering at Westlake University, Hangzhou, China. zhengye@westlake.edu.cn

[2]Zhang Chen is with the Department of Automation at Tsinghua University, Beijing, China. cz_da@tsinghua.edu.cn

[3]Zhenzhong Lan and Shiyu Zhao are with the School of Engineering at Westlake University, Hangzhou, China. {lanzhenzhong, zhaoshiyu}@westlake.edu.cn

[4]Dailin Lv and Zhixing Li are with the School of Electronics and Information Engineering at Hangzhou Dianzi University, Hangzhou, China. {hdu17072119, lzx17011319}@hdu.edu.cn

Digital Object Identifier (DOI): see top of this page.

sified into two streams. The first stream is the conventional approaches that are composed of two-step operations. The first step is to extract object features represented by, for example, Histogram of Oriented Gradients (HOG) or Scale Invariant Feature Transform (SIFT). The second step is to classify the features using machine-learning algorithms such as Support Vector Machine (SVM) or Adaboost. The second stream is the deep-learning-based approaches, which directly outputs detection results using end-to-end artificial neural networks. In contrast to the conventional approaches, which use hand-craft features, deep-learning-based approaches rely on deep convolutional neural network (DCNN) features and consequently have a stronger capability to represent complex objects. However, the disadvantage of using DCNN is that it has high computational requirements and it requires large datasets to train. A detailed review of the existing approaches is given in Section II.

Although deep learning methods have exhibited superior performance in many object detection tasks, their potential for UAV detection has not been well explored or evaluated up to now (see Section II-B for a review). As the first step towards establishing a robust approach to air-to-air UAV detection, this paper proposes a new dataset of micro UAV images and presents a comprehensive experimental evaluation of eight representative deep-learning algorithms. It is worth noting that we focus on the case where the target UAVs are known in advance such that a dataset of them could be built up for the purpose of training. This case applies to tasks like vision-based cooperative control multi-UAV systems, which is our main motivation for UAV detection. Although the algorithms exhibit a certain generalization ability to detect unknown UAVs with similar appearances, other measures such as building up datasets of multiple types of UAVs or target motion sensing [2] may be required.

The novelty and contribution of this work are detailed as follows.

First, this paper presents a dataset of 13,271 images of a flying target UAV (DJI Mavic) acquired by another flying UAV (DJI M210). Compared to the existing air-to-air datasets, the proposed one is more systematically designed and comprehensive in the sense that it covers a wide range of practical scenarios with different background scenes, viewing angles, relative distance, flying altitude, and lightning conditions. In particular, the environmental background scenes vary from simple ones such as clear sky to complex ones such as mountain, field, and urban. The relative distance of the target UAV varies from 10 m to 100 m, and the flight altitude from 20 m to 110 m. Since lightning conditions are also important factors in flying UAV detection, the time for data collection varies from morning to evening in different periods of the day. The dataset also covers some challenging scenarios with, for example, strong light, motion blur, and partial target occlusion.

Second, this paper presents an experimental evaluation of eight representative deep-learning algorithms based on our proposed dataset: RetinaNet [8], SSD [9], YOLOv3 [10], FPN [11], Faster R-CNN [12], RefineDet [13], Grid R-CNN [14], and Cascade R-CNN [15]. To the best of our knowledge, this is the first comprehensive evaluation of deep learning algorithms for UAV detection tasks. The evaluation results suggest that the overall performance of Cascade R-CNN and Grid R-CNN is superior compared to the others. We also evaluated the impact of some key factors such as background scene complexity, relative viewing angles, and relative distance on the detection performance.

The proposed dataset could be used as a benchmark to evaluate different UAV detection algorithms (either conventional or deep-learning-based). The evaluation results highlight some key challenges in the problem of air-to-air UAV detection and suggest potential ways to develop new algorithms in the future.

II. Related Work

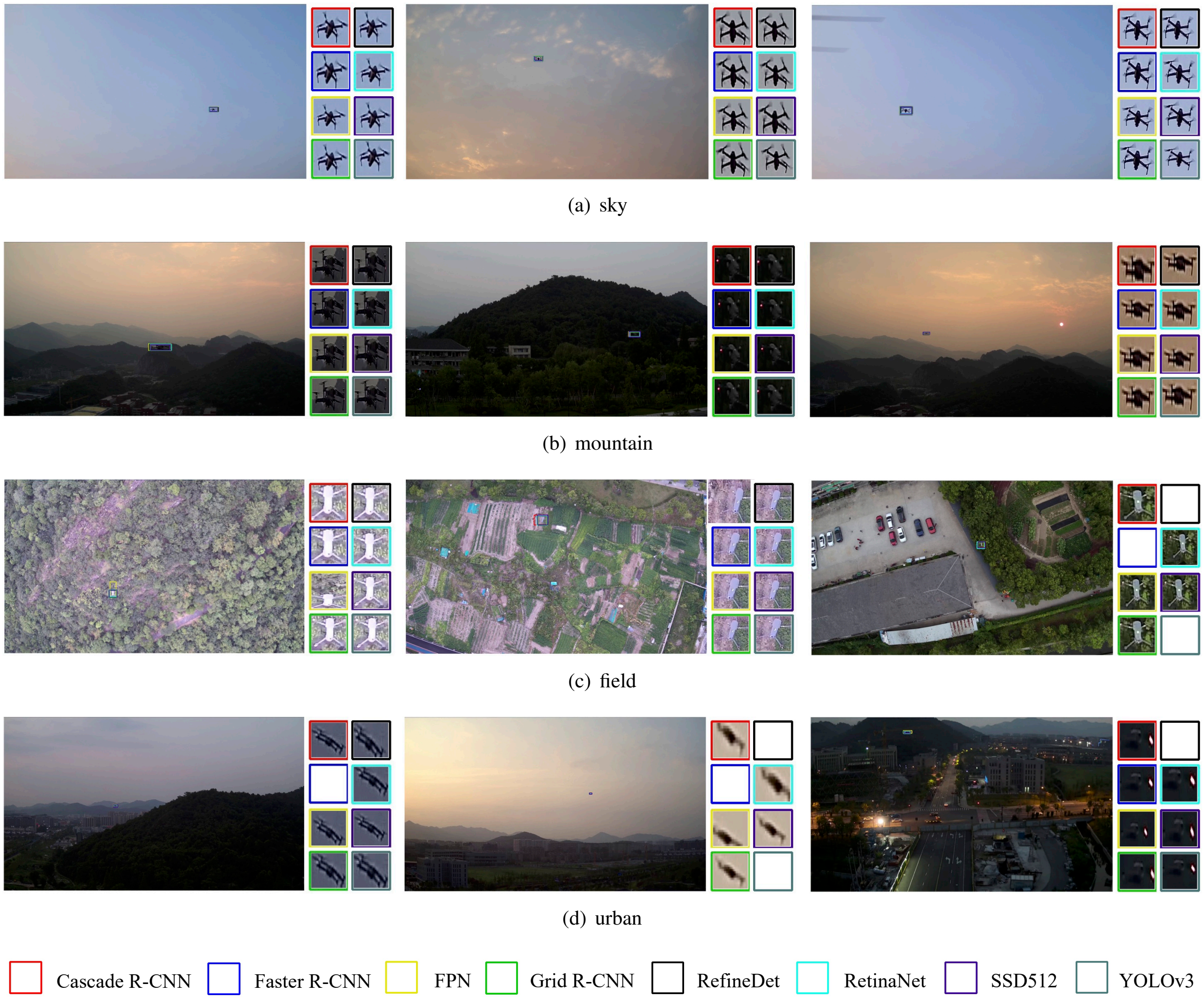
This section gives a review of the existing studies on visual detection of micro UAVs. We only consider the case of using monocular cameras.

A. Conventional approaches

The conventional techniques adopted by existing UAV detection works can be classified into two categories. The first is to use feature extraction methods to obtain target features, and then use a discriminant classifier to determine the target location. The second is to detect moving objects in the image, and then use a generative classifier to determine whether the moving object is the target.

In particular, the work in [16] adopts Haar wavelet based AdaBoost to detect UAVs. The approach is demonstrated by flight experiments to be effective in the simple case of the cloudy sky background. The work in [17] proposes a cascade approach to detect UAVs based on Haar-like features, local binary patterns, and HOG. Since it is a combination of different detection methods, this approach has a low running speed. HOG feature is adopted in [18] for training classical cascade detectors. Although this approach significantly reduces the number of repeated detections by applying non-maximum suppression, the detection accuracy drops rapidly in the case of partial occlusion. Motivated by moving object detection in see-and-avoid tasks, the work in [19] utilizes optical flow matching to integrate spatial and temporal information to track moving targets. This approach requires high-precision motion compensation. The optical flow method is also used to locate moving objects in [20]. The consequent step is to recognize the moving objects by template matching, which is not robust to variance in the target appearance. The work in [21] also adopts template matching as well as morphological filtering for UAV detection. A real-time detection and tracking strategy is proposed in [22] where the object of interest can be automatically detected in a saliency map by computing background connectivity cue at each frame. The work in [23] proposes a pyramidal Lucas-Kanade (PLK) algorithm to detect motion targets in a team of cooperative UAVs. The work in [24] detects moving target by extracting geometry features and dynamic features in the segmentation image, and classifies them by discriminant function derived from the Bayesian theorem.

In summary, although UAV detection has been studied based on many conventional approaches, these approaches are



(a) sky

(b) mountain

(c) field

(d) urban

Fig. 2. Samples of images in the dataset and the corresponding detection results by the eight algorithms. The dataset contains four types of background scenes: sky, mountain, field, and urban. The detected areas by the eight algorithms are given right of each sample image with color-coded boxes. If the corresponding area is blank, it means that the algorithm does not detect any target UAV in this image.

effective only in restricted scenarios where, for example, the background scene is relatively simple or the target appearance does not vary considerably.

B. Deep-learning approaches

Although the methods based on deep learning have made great progress in the field of general object detection, they have not been well explored in the field of UAV detection. Up to now, there are only few studies on visual detecting UAVs by deep learning algorithms. For example, an approach to detect flying objects using motion compensation is proposed in [25], where the features of moving objects are classified by CNNs. This approach leads to high average detection precision, whereas the motion compensation step requires high-precision measurement of the motion of the camera. The work in [26] combines SegNet with bottom-hat morphological processing for detecting large-size aircraft in the air. This approach could detect aircraft within a long-range up to 2800 m, but the accuracy is as low as 13.4%. Although some other studies such as [27], [28] also adopt deep learning algorithms such as YOLOv2 to detect UAVs, the performance of different representative deep learning algorithms for UAV detection have not been evaluated or compared.

C. Existing datasets for UAV detection

Up to now, there are very few comprehensive datasets for the purpose of training deep learning algorithms for UAV detection. The dataset in [29] comprises 20 video sequences and each of them has about 4000 752×480 gray frames. The image of the flying target UAV is captured by a camera mounted on another UAV in indoor and outdoor environments. The dataset proposed in [30] consists of two sub-datasets. The first is a Public-Domain drone dataset that contains 30 video sequences with different drone models captured in indoor and outdoor environments. The other one is the USC drone dataset that contains 30 video clips of the same target UAV. This dataset is acquired on the USC campus and the background of most samples is a clean or cloudy sky, which is relatively simple compared to our proposed dataset. In order to increase the number of images in the dataset, the authors of the USC dataset developed a model-based automatic data augmentation method to paste clipped drone model images into background images. Although the size of data can be expanded in this way, the work in [31] shows that networks trained based on such kind of data may not be significantly improved by the augmentation. Very recently, a new dataset, named MIDGARD, was presented in [32]. This dataset contains different kinds of backgrounds and varying lighting conditions. It also proposed a new method for automatic annotation by using their previous work of UltraViolet Direction And Ranging [33]. A detailed comparison between MIDGARD and our dataset is given in Section V.

III. The Proposed Dataset

The proposed dataset, named *Det-Fly*, consists of 13,271 images of a target micro UAV (DJI Mavic). Each image has 3840 × 2160 pixels. Some images of the dataset are sampled from videos at 5 FPS and the others are captured from desired relative poses. All the images are manually annotated by professionals. Some sample images are given in Fig. 2. The dataset is available at *https://github.com/Jake-WU/Det-Fly*.

Det-Fly covers a wide range of scenarios including different viewing angles, background scenes, relative ranges, and lighting conditions. In particular, Det-Fly involves four types of environmental background: sky, urban, field, and mountain (see Fig. 2). Each type of environmental background occupies nearly the same proportion (about 20%-30%) of the entire dataset. In terms of relative viewing angles, Det-Fly can be split into three categories: front view, top view, bottom view. The data proportion of the three viewing angles are, respectively, 36.4% (front view), 32.5% (top view), and 31.1% (bottom view).

In terms of the image size of the target UAV, the statistics data given in Fig. 3 shows that a large portion of the target UAV images in the dataset are small. In particular, nearly half of them are smaller than 5% of the entire image size. When the height and width of a target UAV image are smaller than 10% of the entire image, it could be regarded as a small object, whose detection is a well-known challenging task. In addition, since the lighting conditions are also important factors in flying UAV detection, the time of image collection varies from morning to evening in different periods of a day. The dataset also covers some challenging scenarios with such as strong/weak lighting (10.8%), motion blur (11.2%), and partial target occlusion (0.8%).

Some remarks about the proposed dataset are given below. First, each image in this dataset only contains one single target UAV. However, the algorithms trained based on the dataset could naturally detect multiple UAVs, which is required by vision-based UAV swarming. Second, although the dataset covers a wide range of environmental scenarios, it is impossible to cover *all* possible scenarios. The primary purpose for establishing the dataset is to evaluate different deep learning algorithms. If one is interested in implementing a deep-learning approach in practice in a specific environmental scenario, the dataset should be adjusted to cover either the specific environment where the UAV detection is performed or more environmental scenarios to enhance the generalization ability. Third, this dataset only covers one single type of UAV (DJI Mavic). If one is interested in detecting more types of UAVs, other measures such as building up datasets of multiple types of UAVs or target motion sensing [2] may be required.

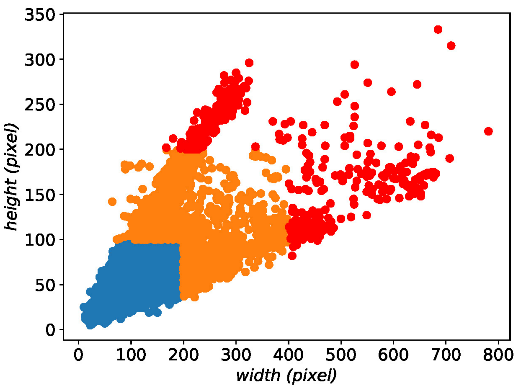


Fig. 3. Statistical data of the target UAV size in our dataset. The blue points correspond to the UAV images whose width and height are less than 5% of the image size. The orange points represent the data that the sizes are less than 10% of the image size. The remaining red points are samples greater than 10%. Since the attitude of the camera may frequently change during flight, there are various height-width ratios of bounding boxes.

IV. Experimental Setup

In this paper, we finely tune and evaluate eight classic deep-learning based object detection methods: SSD [9], RetinaNet [8], YOLOv3 [10], RefineDet [13], Faster R-CNN [12], FPN [11], Cascade R-CNN [15], and Grid R-CNN [14]. These methods generate similar performance in terms of small-scale objects on COCO dataset in which mean Average Precision (mAP) is used as an evaluation metric.

According to the types of detection algorithms, the selected methods can be divided into two categories: one-stage networks and two-stage networks. A one-stage network does classification and regression directly on the feature map to achieve fast object detection. Among the selected methods, SSD, RetinaNet, YOLOv3, and RefineDet are one-stage networks. A two-stage network consists of a region proposal network (RPN) that proposes several candidate boxes and a classification and regression network that achieves recognition and localization for a specified object. Among the selected methods, Faster R-CNN, FPN, Cascade R-CNN, and Grid R-CNN are two-stage networks.

The primary hyper-parameters of the algorithms implemented in our work are given in Table I. Since ResNet achieves state-of-the-art performance on ImageNet, we adopt it as the backbone in most of the algorithms. Generally, ResNet has two versions for common use, named ResNet-50 and ResNet-101. In our work, we choose ResNet-50 over ResNet-101 because it is lite and suitable to be implemented in embedded computers on micro UAVs. Since DarkNet-53 is widely used as the backbone of YOLOv3 and it exhibits similar performance as ResNet-50 [10], [34], we choose DarkNet-53 for YOLOv3 in our experiments. The original optimizers are used. The

TABLE I
THE HYPERPARAMETERS IN OUR IMPLEMENTATION OF THE EIGHT ALGORITHMS.

hyperparameter	backbone	optimizer	input_size	LR	momentum	weight_decay	iteration
Cascade R-CNN	ResNet-50	SGD	[640,640]	1e-2	0.9	1e-4	6,652
FPN	ResNet-50	SGD	[600,600]	1e-3	0.9	1e-5	49,993
Faster R-CNN	ResNet-50	SGD	[1000,600]	1e-2	0.9	1e-4	6,652
Grid R-CNN	ResNet-50	SGD	[600,600]	2e-2	0.9	1e-4	46,564
RefineDet	ResNet-50	SGD	[320,320]	1e-3	0.9	5e-4	30,000
RetinaNet	ResNet-50	SGD	[600,600]	2e-4	0.9	1e-4	13,304
SSD512	ResNet-50	SGD	[512,512]	1e-3	0.9	5e-4	46,564
YOLOv3	DarkNet-53	Adam	[416,416]	1e-3	0.9	5e-4	7,000

learning rate (LR), momentum, weight decay, and iteration are finely tuned based on extensive tests.

Our experiments are implemented on a computer with an Intel i7, 32GB RAM, Nvidia RTX 2080Ti rather than an embedded computer in order to reduce training time. We train the models based on 70% of the images, in which 10% is evaluated for validation, and test them based on the remaining 30% images. In addition, we use non-maximum suppression (NMS) to remove overlapping bounding boxes, so that an object is only contained in one bounding box. As an important parameter in NMS to evaluate the overlapping rate of predicted bounding boxes, IoU is defined as

$$A_o = \frac{area(\mathcal{O}_p \cap \mathcal{O}_{gt})}{area(\mathcal{O}_p \cup \mathcal{O}_{gt})}.$$

In our experiments, the IoU threshold is set to 0.5.

In the training stage, we set the training epoch as eight and save model parameters in each epoch. If the training loss and validation loss remain stable we conclude that the detector is well trained. Otherwise, we modify the setting epoch and resume training until the model is well trained.

Precision is a metric to evaluate missing detection. The calculation of Precision in this paper is the same as the ones in general visual object detection, which traverses all predicted boxes to calculate Precision. If the UAV is successfully detected, then the predicted bounding box will be regarded as true positive (TP). Otherwise, it will be regarded as a false positive (FP). Precision is defined as

$$\text{Precision} = \frac{TP}{TP + FP}.$$

Recall is a metric to measure false detection and defined as

$$\text{Recall} = \frac{TP}{TP + FN}.$$

The performance of an object detector can be evaluated by Precision×Recall (P-R) curve, which considers false detections with respect to missing detections for varying thresholds. However, P-R curves are often zigzag curves going up and down and tend to cross each other frequently, it is usually not easy to compare different curves (different detectors) in the same plot. Instead, numerical metrics called Average Precision (AP) can help us compare different detectors. AP is the area under a curve (AUC) of the Precision×Recall curve. It is easy to make a comparison between areas. Thus, we use AP as the evaluation metrics.

V. EVALUATIONS RESULTS

A. Average Precision

The APs of the eight algorithms are shown in Table II. Grid R-CNN achieves the best performance (82.4%) among all detectors, while RefineDet the worst (69.5%). Among two-stage networks, Cascade R-CNN achieves the best performance (79.4%), whereas Faster R-CNN, which is the main framework of two-stage networks in our experiments, achieves the worst (70.5%). For one-stage networks, SSD512 (78.7%) and RetinaNet (77.9%) both perform well, whereas YOLOv3 achieves only 72.3%. Although one-stage networks sacrifice detection performance to obtain high implementation efficiency, SSD512 achieves the same AP as FPN, which suggests that SSD512 could be a good alternative for tasks requiring high computational efficiency.

To further evaluate the performance of the algorithms, we split the testing set into two sets. One set, named *Det-Fly-Simple*, contains images with a relatively simple background (e.g., clean sky), short sensor-target range, and low flight speed. The other set, named *Det-Fly-Complex*, consists of a more complex background (e.g., complex urban) and small target size. Both datasets contribute about 50% images of the entire dataset. The evaluation results on *Det-Fly-Simple* suggest that the two-stage networks, Cascade R-CNN and Grid R-CNN, achieve the highest AP (more than 82.0%) among all the eight networks. Among one-stage networks, RetinaNet and SSD512 achieve the best performance (nearly 81.0%). Except for RefineDet and YOLOv3, the performance of other algorithms is higher than 80.0%. Compared with *Det-Fly-Simple*, the detection performance of most of the algorithms on *Det-Fly-Complex* drops sharply by nearly an average of 5.0%, due to the high complexity of *Det-Fly-Complex*. The mean Precision of the algorithms could only achieve 74.4%. In particular, Grid R-CNN still achieves the best performance and it is also the only one exceed 80.0%. RetinaNet and SSD512, which have similar performance, still perform best within one-stage networks. In general, two-stage networks perform a little better than one-stage networks in this test.

In summary, Grid R-CNN and Cascade R-CNN show stable and superior performance compared to the others in all evaluation scenarios. One stage networks, SSD512 and RetinaNet, also show stable and good performance. Since they could achieve higher computational speed, SSD512 and RetinaNet may be a good choice for tasks with limited computational resources.

TABLE II
THE AP OF THE EIGHT ALGORITHMS TESTED ON DET-FLY (%).

dataset	Cascade R-CNN	RetinaNet	RefineDet	FPN	Faster R-CNN	Grid R-CNN	SSD512	YOLOv3
Det-Fly	79.4	77.9	69.5	78.7	70.5	**82.4**	78.7	72.3

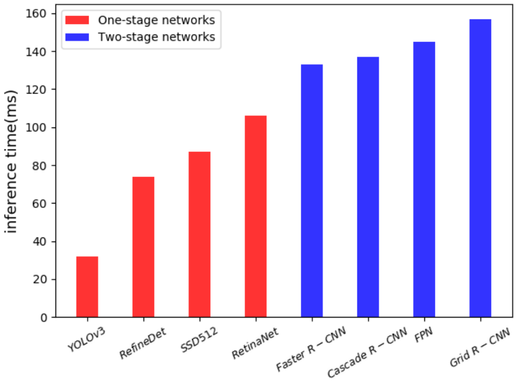


Fig. 4. The inference time of all algorithms in our experiment.

B. Network attributes affecting UAV detection

The inference speed of the algorithms is an important aspect for practical implementation, especially in onboard embedded systems. Figure 4 shows the average inference time of the eight deep learning algorithms in our experiments. As can be seen, one-stage networks have a faster inference speed than two-stage networks. Although Grid R-CNN archives the best AP performance among all algorithms, it is also the most time-consuming one. The inference time of YOLOv3 (32ms) is nearly one-fifth of that of Grid R-CNN (157ms). If computational efficiency is the priority for an application, YOLOv3 is recommended since it is the fastest and its performance is better than the other two algorithms (RefineDet and Faster R-CNN) as shown in Table II.

All the compared models except YOLOv3 were implemented with the ResNet-50 backbone network in our experiments. Although ResNet-50 has already been run in real-time on some embedded devices, one may be interested in the performance with a even lighter backbone network. To this end, we tested SSD512 on our dataset with MobileNetv2 as the backbone. The resulting AP on the dataset is 68.8%, which is nearly 10% less than the result of SSD512 with ResNet-50. However, the inference time of SSD512 with MobileNetv2 (53 ms) is much shorter than SSD512 with ResNet-50 (84 ms). Therefore, lighter backbones such as MobileNet may be considered when the onboard computational resource is extremely limited.

The different performance of FPN and Faster R-CNN suggests that the network structure FPN can improve UAV detection capability significantly. Since Grid R-CNN and Cascade R-CNN have superior and robust performance than Faster R-CNN, it suggests that the grid guided mechanism and multi-stage structures could generate better-regressed bounding boxes. While using multi-stage structures will cost more time, the grid mechanism is highly recommended for future detector design. Furthermore, the performance of RefineDet is weaker than SSD512, which may suggest that high-resolution input also could improve the UAV detection capability. In addition, RetinaNet shows good and stable performance among one-stage networks, which suggests that focal loss may be a recommended method to solve the problem of class imbalance.

TABLE III
THE AP FOR DIFFERENT ENVIRONMENTAL BACKGROUND SCENES (%)

algorithms	F	U	S	M
Cascade R-CNN	67.9	64.7	**93.1**	**84.8**
FPN	71.9	70.4	87.2	71.7
Faster R-CNN	64.1	48.9	87.6	78.9
Grid R-CNN	**78.0**	**76.1**	89.1	83.5
RefineDet	70.4	44.7	84.2	77.2
RetinaNet	73.7	67.5	89.5	77.3
SSD512	73.6	65.6	92.3	78.7
YOLOv3	69.1	58.4	83.5	79.6
mean AP	71.1	62.0	88.3	80.0

* F: field, U: urban, M: mountain, S: sky

C. Image attributes affecting UAV detection

We next evaluate some key aspects of the images such as environmental background, target scales, viewing angles, and other challenging conditions on the detection performance. Since the performance of an algorithm could be affected by many aspects such as insufficient training and different parameters, we take the mean Average Precision (mAP) of these algorithms as the criteria for a fair evaluation.

1) Environmental background: The complexity of the background scene has a great impact on UAV detection performance. Table III shows the APs of the algorithms for different types of environmental background. In particular, the mAP suggests that the sky (88.3%) is the easiest type of background for UAV detection, while urban (62.0%) is the hardest. This is consistent with our intuition that the complex urban background makes visual UAV detection very challenging.

As for the performance of algorithms, Grid R-CNN shows consistent and high Precision across different types of background scenes, whereas the performance of Faster R-CNN and RefineDet drops rapidly when the background complexity increases.

2) Target scales: The size of the target UAV in the image has a great impact on detection performance. Figure 5 shows the APs of all the algorithms with respect to the target size/ratio. As shown in the figure, the APs of all algorithms increase at different rates when the target scale increases. In particular, Grid R-CNN shows the best performance for different target scales, whereas the performance of RefineDet and Faster R-CNN drops rapidly when the target scale becomes small.

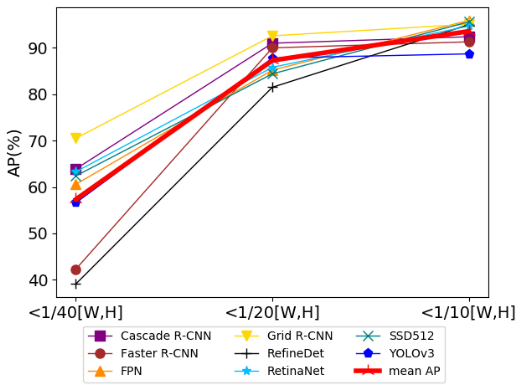


Fig. 5. The AP of the algorithms for different target scales. If both the width and height of the annotated bounding box are, respectively, smaller than x ($x \in \{1/40, 1/20, 1/10\}$) of the width and height of the entire image, then it is classified as $< x$[W,H]. The AP is calculated by the algorithms with data in the internals. The mAP represents the mean AP of the eight algorithms in each scale interval.

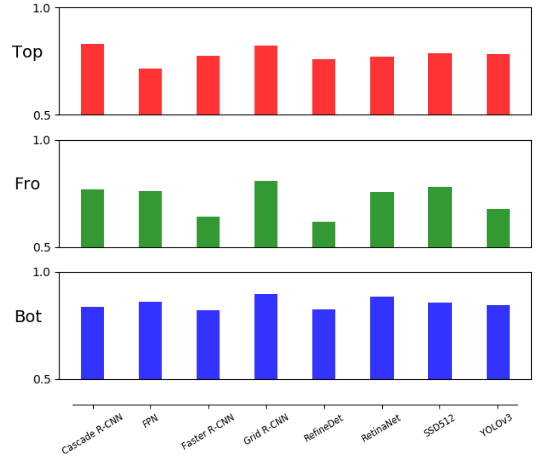


Fig. 6. The AP of different viewing angles. This figure is divided into three parts which are Top (top view), Fro (front view), and Bot (bottom view). The vertical axis of each part, which is the AP of the algorithms, is from 0.5 to 1.0. The mAP of each part is about 0.78 (Top), 0.72 (Fro), and 0.85 (Bot), respectively. The marker in each part represents the performance of the algorithm.

3) Viewing angles: It is noticed from our experiment that the viewing angles of the target UAV also has an impact on the detection performance. Figure 6 shows the AP for different viewing angles. It can be seen that the bottom view leads to the highest Precision, whereas the front view is the lowest. The reason is that, for the bottom-view cases, the target shows rich geometric information, and in the meantime, the background scene is a blue or cloudy sky. However, for the front-view cases, the target is flat and hence shows less geometric information, and in the meantime, the background could be more complex than the bottom view case.

4) Other challenging conditions: The dataset covers some challenging conditions such as strong/weak lighting, motion blur, and partial occlusion. The ratios of the images of the three scenarios in our dataset are 10.8%, 11.2%, and 0.8%, respectively. Here, partial occlusion refers to the case where part of the target UAV is out of the field of view. All the images in these cases can be found online in our dataset.

The testing results of the eight algorithms under the three challenging conditions are reported in Table IV. It is notable that partial occlusion causes much lower AP. Part of the reason is that partially occluded target detection is indeed a challenging task, and in the meantime, the images of this case only occupy a small proportion of the dataset. On the other hand, strong/weak lighting conditions and motion blur do not compromise the performance significantly, which verifies the robustness of the deep learning algorithms.

D. Comparison with the state-of-the-art dataset

To the best of our knowledge, MIDGARD is the latest comprehensive dataset designed for deep-learning-based micro-UAV detection [32]. Compared to MIDGARD, the annotation bounding box of each image in Det-Fly is tighter, because the images in Det-Fly are annotated one by one manually by professionals, whereas the images in MIDGARD are automatically annotated based on UVDAR and relative pose estimation. Moreover, Det-Fly covers a wider range of relative target distances. In particular, the longest relative target distance in Det-Fly reaches more than 100 m, but the longest distance in MIDGARD is less than 20 m. Due to the wide range of relative distances, the scale of the target UAV in Det-Fly is more diverse.

The eight algorithms have been trained and tested on MIDGARD. The testing results are shown in Table V. As can be seen, the results of MIDGARD are 10% better than that of Det-Fly. This might be caused by the complexity and diversity of the samples in Det-Fly.

VI. Conclusion

This paper presented a new dataset, named Det-Fly, for air-to-air UAV detection and evaluated eight representative deep-learning algorithms based on this dataset. Not only the overall performance of the algorithms are carefully evaluated and compared, the impact of environmental background, target scales, viewing angles, and other challenging conditions on the detection performance is also analyzed. According to the experimental results, suggestions on how to design algorithms to achieve better detecting performance in the future are given.

In the future, to detect unknown UAVs in various environments, the dataset should be further enhanced by adding

TABLE IV
THE AP FOR DIFFERENT CHALLENGING CONDITIONS (%)

algorithms	S	M	P
Cascade R-CNN	73.3	81.5	37.3
FPN	67.9	83.1	**43.0**
Faster R-CNN	68.7	76.2	34.6
Grid R-CNN	**83.1**	**84.3**	40.0
RefineDet	68.4	76.0	33.4
RetinaNet	69.1	78.3	37.3
SSD512	68.6	76.0	38.2
YOLOv3	65.3	80.9	33.3
mean AP	70.6	79.5	37.1

* S: Strong/weak light, M: Motion blur, P: Partial occlusion.

TABLE V
THE AP OF THE EIGHT ALGORITHMS TESTED ON MIDGARD (%).

dataset	Cascade R-CNN	RetinaNet	RefineDet	FPN	Faster R-CNN	Grid R-CNN	SSD512	YOLOv3
MIDGARD	89.4	88.8	85.8	85.8	89.1	**90.1**	89.1	87.7

more types of UAVs and background scenarios. Moreover, an ablation study is necessary to design deep-learning algorithms that are specifically suitable for UAV detection tasks and to be implemented onboard. In addition, interpretable technology may be adopted to explain why the recommended network structures or methods could improve detection performance. Algorithms that are able to process high-resolution images also need more attention.

REFERENCES

[1] Y. Tang, Y. Hu, J. Cui, F. Liao, M. Lao, F. Lin, and R. S. H. Teo, "Vision-aided multi-UAV autonomous flocking in GPS-denied environment," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 1, pp. 616–626, 2019.

[2] J. Xie, J. Yu, J. Wu, Z. Shi, and J. Chen, "Adaptive switching spatial-temporal fusion detection for remote flying drones," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 6964–6976, 2020.

[3] R. Mitchell and I. Chen, "Adaptive intrusion detection of malicious unmanned air vehicles using behavior rule specifications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 5, pp. 593–604, 2014.

[4] J. Zhang, C. Hu, R. G. Chadha, and S. Singh, "Maximum likelihood path planning for fast aerial maneuvers and collision avoidance," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2805–2812.

[5] J. Ren and X. Jiang, "Regularized 2-d complex-log spectral analysis and subspace reliability analysis of micro-doppler signature for UAV detection," *Pattern Recognition*, vol. 69, pp. 225–237, 2017.

[6] A. Bernardini, F. Mangiatordi, E. Pallotti, and L. Capodiferro, "Drone detection by acoustic signature identification," *Electronic Imaging*, vol. 2017, no. 10, pp. 60–64, 2017.

[7] R. Yoshihashi, T. T. Trinh, R. Kawakami, S. You, M. Iida, and T. Naemura, "Differentiating objects by motion: Joint detection and tracking of small flying objects," *arXiv: Computer Vision and Pattern Recognition*, 2017.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (CVPR)*, 2017, pp. 2980–2988.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37.

[10] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv*, 2018.

[11] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944.

[12] 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.

[13] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4203–4212.

[14] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, "Grid R-CNN," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7363–7372.

[15] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 6154–6162.

[16] F. Lin, K. Peng, X. Dong, S. Zhao, and B. M. Chen, "Vision-based formation for UAVs," in *Proceedings of the IEEE International Conference on Control & Automation (ICCA)*, 2014, pp. 1375–1380.

[17] F. Gökçe, G. Üçoluk, E. Şahin, and S. Kalkan, "Vision-based detection and distance estimation of micro unmanned aerial vehicles," *Sensors*, vol. 15, no. 9, pp. 23 805–23 846, 2015.

[18] K. R. Sapkota, S. Roelofsen, A. Rozantsev, V. Lepetit, D. Gillet, P. Fua, and A. Martinoli, "Vision-based unmanned aerial vehicle detection and tracking for sense and avoid systems," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 1556–1561.

[19] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman, "Multi-target detection and tracking from a single camera in unmanned aerial vehicles (UAVs)," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4992–4997.

[20] S. Minaeian, J. Liu, and Y.-J. Son, "Effective and efficient detection of moving targets from a UAV's camera," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 497–506, 2018.

[21] R. Opromolla, G. Fasano, and D. Accardo, "A vision-based approach to UAV detection and tracking in cooperative applications," *Sensors*, vol. 18, no. 10, 2018.

[22] Y. Wu, Y. Sui, and G. Wang, "Vision-based real-time aerial object localization and tracking for UAV sensing system," *IEEE Access*, vol. 5, pp. 23 969–23 978, 2017.

[23] S. Minaeian, J. Liu, and Y. Son, "Vision-based target detection and localization via a team of cooperative UAV and UGVs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 1005–1016, 2016.

[24] F. Lin, X. Dong, B. M. Chen, K. Lum, and T. H. Lee, "A robust real-time embedded vision system on an unmanned rotorcraft for ground target following," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 1038–1049, 2012.

[25] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting flying objects using a single moving camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 879–892, 2017.

[26] J. James, J. J. Ford, and T. L. Molloy, "Learning to detect aircraft for long-range vision-based sense-and-avoid systems," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4383–4390, 2018.

[27] C. Aker and S. Kalkan, "Using deep networks for drone detection," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017, pp. 1–6.

[28] A. Schumann, L. Sommer, J. Klatte, T. Schuchert, and J. Beyerer, "Deep cross-domain flying object classification for robust UAV detection," in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017, pp. 1–6.

[29] A. Rozantsev, V. Lepetit, and P. Fua, "Flying objects detection from a single moving camera," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4128–4136.

[30] Y. Chen, P. Aggarwal, J. Choi, and C. C. J. Kuo, "A deep learning approach to drone monitoring," in *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2017, pp. 686–691.

[31] X. Peng, B. Sun, K. Ali, and K. Saenko, "Learning deep object detectors from 3d models," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1278–1286.

[32] V. Walter, M. Vrba, and M. Saska, "On training datasets for machine learning-based visual relative localization of micro-scale UAVs," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 674–10 680.

[33] V. Walter, M. Saska, and A. Franchi, "Fast mutual relative localization of UAVs using ultraviolet led markers," in *Proceedings of the International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, pp. 1217–1226.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.