2017 17th International Conference on Control, Automation and Systems (ICCAS 2017)
Oct. 18–21, 2017, Ramada Plaza, Jeju, Korea

A Comparison of Convolutional Object Detectors for Real-time Drone Tracking Using a PTZ Camera

Jihun Park*, Dae Hoe Kim, Young Sook Shin, Sang-ho Lee

Agency of Defense Development, Daejeon, Korea
{jhpark_a, dhkim17, ysshin, lee.sangho}@add.re.kr *Corresponding author

Abstract: As highly maneuverable drones are available at the low price, the threats that might be caused by the drone attacks has been increased. Recent object detectors have been dramatically improved in accuracy by using convolutional neural networks, and these can be utilized to identify hostile drones. In this paper, we examine state-of-the-arts convolutional object detectors for a real-time drone detection and tracking system using a Pan-Tilt-Zoom (PTZ) camera. In the drone detection and tracking system, an object detector is used to identify whether an image from the PTZ camera contains a drone, and our system generates PTZ actions to track the detected drone. To detect small size drones in real-time, an appropriate object detector should be selected. This paper compares six convolutional object detectors in the accuracy and speed.

Keywords: Object detection, drone defense, object tracking

1. INTRODUCTION

Recently, drone technology has been rapidly developed. Drones can be utilized for delivering products or for taking aerial photographs and videos at a low price, but they might be used in a hostile way. For example, drones can cross over the Military Demarcation Line (MDL) to make an aerial reconnaissance, and they also may carry a small bomb to attack a specific person or a facility. Because aerial surveillance systems, like a radar system, are usually much more expensive than drones and very huge to be installed, the defense system should be at the low cost.

Recent advanced object detection models, which use deep convolutional neural network (CNN) architectures, can be utilized to detect the drone attacks. Accuracy of recent object detection is higher than human level in some case studies, and this technology can automate and improve surveillance systems. Many fast object detection algorithms, such as YOLO [1], have been suggested, and recently, Google released object detection API [2], which includes five deep CNN models and pre-trained weights. According to the previous research[2], there is trade-offs between the speed and accuracy, and we should choose an appropriate model for our usage.

In this paper, we introduce a drone detection and tracking system using a Pan-Tilt-Zoom (PTZ) camera and the object detection algorithm. The system can be installed at core facilities to detect drone attacks by warning users to cope with the drone attack at the early phase. In our system, images are captured through the PTZ camera, and the object detector identifies whether any drone exists in the image. If a drone is identified, PTZ action is generated to track the drone, and approach of the drone is noticed to the user. Our system has two main challenges—1) Drones are small in size, which are usually smaller than 50cm in width, and 2) we should build a real-time detection model to track the drones with pan, tilt, and zoom actions.

We compare six state-of-the-arts convolutional object detectors, including YOLOv2 [1] and five models provided by Google's object detection API—SSD [3] with MobileNet [4], SSD with Inception V2 [5], R-FCN [6] with Resnet 101 [7], Faster R-CNN [8] with Resnet 101, and Faster R-CNN with Inception Resnet [9]. We collect 9,525 labled images of 11 multi-rotor drones for the experiment data set. On the data set, we investigate precision-recall curves as accuracy measurement, and we also investigate the training and testing speed of object detection models.

2. RELATED WORK

Object detection algorithms have been improved by using deep convolutional neural network architectures. As the state-of-the-arts of the *Region-based CNN* (R-CNN) models, Faster R-CNN [8] performs object detection in two stages. In the first stage, a region proposal network (RPN) takes feature maps of a given image at some intermediate layer as input and outputs a set of rectangular object proposals. Then, the object proposals are classified by Fast R-CNN [10] in the second stage.

In Region-based Fully Convolutional Networks (R-FCN) [6], position-sensitive score maps are introduced to consider translation variance. The R-FCN achieves a competitive accuracy with the Faster R-CNN while further improves computational efficiency.

Differing from multi-stage methods (*i*.e., Faster R-CNN, R-FCN) YOLO [1] and SSD [3] propose single detection pipelines that directly predict object locations and corresponding class probabilities. YOLO [1] treats object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. SSD [3] discretizes the output space of bounding boxes

978-89-93215-13-7/17/$31.00 ©ICROS

into a set of default boxes over different aspect ratios and scales per feature map location. Due to the multiscale approach and matching strategy, SSD showed improved accuracies compared to YOLO.

Due to the increased usage of drones, some studies have been conducted for drone detection. In [11], morphological pre-processing and Hidden Markov Model was used to detect and track fixed-wing unmanned aerial vehicle (UAV). In [12], object-centric motion compensation and CNN regressor are utilized for flying object detection with a single moving camera.

3. STUDY APPROACH

This section describes the object detection models, the drone data set, and the experimental setup.

3.1 Object detection models

Six object detection models are used in our experiments. A YOLOv2 [1] detection model is trained using authors' original darknet implementation, and we then test and evaluate the model using the tensorflow version of the darknet (darkflow)[1]. We use five models of Google object detection API, which include SSD [3] with MobileNet [4], SSD with Inception V2 [5], R-FCN [6] with Resnet 101 [7], Faster R-CNN [8] with Resnet 101, and Faster R-CNN with Inception Resnet [9] We use training and testing configurations provided by Google. All models are modified to consider only drone class and not others.

3.2 Data set

The drone data set is made of 11 drone models, which include products of DJI—Phantom 2, Phantom 4, Inspire, and Mavic, and seven newly built models. All drone models are multi-rotor drones, and the sizes vary from 335 mm to 1300mm in the diagonal size. For each drone model, 1080p videos are recorded in different view, different distance, and different background conditions. Images are extracted from the videos, with the frequency of about ten frames per seconds. The images are manually labeled, and we finally collect 9,525 labeled multi-rotor drone images.

The data set is divided into a training set and a testing set. Because the images are captured from videos, the training set and the testing set should be separated in time to assure different background and size conditions. 200 consecutive images (about 20 seconds) are classified to the training set, and 100 following consecutive images (about 10 seconds) are classified to the testing set.

The image size is 1920 × 1080 in width and height, respectively. The minimum size of a drone in an image is 24 × 9 pixels, and the maximum size of a drone is 799 × 491 pixels in width and height, respectively. Most of images contain one drone in an image, and 21 images contain two drones. In our data set, most of drones are small in size. Figure 1 shows the number of drones varying width in pixels. 75% of drones have widths smaller than 100 pixels.

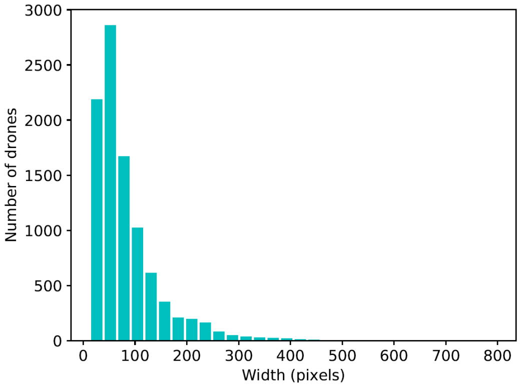

Fig. 1 The number of drones varying the width of drone boxes.

3.3 Experimental setup

We use a workstation to train and test object detection models. The machine contains an Intel Xeon E5-2640 v4 CPU, and two Quadro M6000 24GB GPUs. All models are trained for about 100,000 iterations. Similar with many previous work, if a predicted box has the intersection over union (IoU) greater than 0.5 with any ground truth box, the prediction is regarded as a true positive.

4. DRONE TRACKING SYSTEM

Threats of drone attacks have been increasing as the drone technology has been rapidly developed. Drone invasion should be identified at the early phase of the attack, and the defense system should be implemented at the low cost because drone is very cheap compared to usual aerial surveillance systems. In this section, we introduce a drone tracking system using a PTZ camera to detect and track drones.

Figure 2 shows overview of the drone tracking system. The drone tracking system uses a PTZ camera, and the object detection algorithm is used to identify whether an image from the PTZ camera contains a drone or not. If a drone is identified, our system generates pan-tilt-zoom action and sends it to the PTZ camera to track the drone.

If a drone is identified, our system obtains current horizontal and vertical Field of View (FoV) from the camera, and then calculates the angle difference between the drone and the center point of the camera. The following is formulas of relative degrees of pan and tilt status.

$$\Delta pan = (0.5 - \frac{drone\ x\ center}{image\ width}) \times H\_Fov$$

$$\Delta tilt = (0.5 - \frac{drone\ y\ center}{image\ height}) \times V\_Fov$$

Zoom actions are calculated to keep the size of a drone to the predefined desired portion in an image. Portion of a specific box is the area divided by the image size

[1] https://github.com/thtriew/darkflow

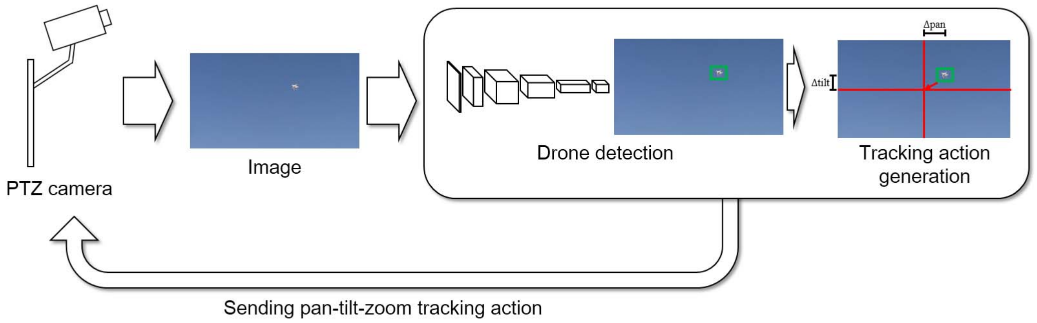
PTZ camera
Image
Drone detection
Δtilt
Δpan
Tracking action generation
Sending pan-tilt-zoom tracking action

Fig. 2 Overview of drone tracking system

(1920 × 1080). We calculate new zoom setting of the PTZ camera as the current zoom multiplied by the ratio between the desired portion and the current portion. For example, if the current portion of a drone is 2% of the image, and our desired portion of a drone is 4%, then we zoom in 2X. The following formula describes how our system calculates the new zoom setting.

$$zoom' = \frac{desired\_portion}{current\_portion} \times current\_zoom$$

5. STUDY RESULTS

This section describes study results in accuracy and speed of the convolutional object detection models.

5.1 Accuracy analysis

We investigate precisions and recalls varying confidence score thresholds at the granurality of 5%. Figure 3 shows the results.

First of all, SSD models show poor results compared to other models. This results might be caused that the majority of drones in our data set is small, which are less than 100 pixels in width. Previous work of Huang et al. [2] mentioned that SSD models showed poor performance on small objects.

R-FCN with Resnet 101 shows higher accuracy than Faster R-CNN with Resnet 101, eventhough R-FCN is faster than the Faster R-CNN. Results of YOLOv2 spreads a wider range than others—from 42% precision with 73% recall (5% confidence) to 100% precision with 3% recall (90% confidence).

In terms of the F-measure, which considers both the precision and the recall, faster R-CNN with Inception Resnet of 85% confidence threshold is the highest (74.3%), followed by R-FCN with Resnet 101 of 90% confidence threshold (73.2%) and YOLOv2 of 50% confidence threshold (72.8%).

5.2 Speed analysis

The drone tracking system requires real-time recognition and tracking of a hostile drone, thus the speed of object detector is an important factor in the system. Table 1 summarize the results. SSD MobileNet is the fastest model (20.8 fps), followed by YOLOv2 (13.0 fps) and SSD Inception V2 (12.0 fps). Considering real-time constraints, these top three fastest models should be used in our system.

Table 1 Speed of detection models

Model	Time taken per image	FPS
SSD MobileNet	0.048	20.8
SSD Inception V2	0.084	12.0
RFCN Resnet 101	0.320	3.1
FRCNN Resnet 101	0.423	2.4
FRCNN Inception Resnet	1.455	0.7
YOLOv2	0.077	13.0

To consider updated data set of new drone models and new background conditions, training time should not be too long to update regularly the object detection model.

We investigate the time taken to train a model for 100,000 iterations. Table 2 shows the results. The R-FCN with Resnet 101 model takes the shortest time for training. Faster R-CNN with Resnet 101 and SSD MobileNet model also take less than 20 hours for training 100,000 iterations. Faster R-CNN with Inception Resnet and YOLOv2 take longer than 30 hours.

Table 2 Time taken for training 100,000 iteration

Model	Time (hours)
SSD MobileNet	19
SSD Inception V2	22
RFCN Resnet 101	12
FRCNN Resnet 101	14
FRCNN Inception Resnet	50
YOLOv2	36

5.3 Discussion

In the experiment, Faster R-CNN with Inception Resnet model shows the highest accuracy, but it is the slowest in detection and training. Considering speed-accuracy trade-offs, YOLOv2 might be the most appropriate model in our usage—YOLOv2 shows comparable accuracy with Faster R-CNN and R-FCN models, and it is much faster than those models.

As the future work, we will investigate how different configurations, such as the number of object proposals in R-FCN models and Faster R-CNN models, affects the speed and accuracy. When operating the system, the

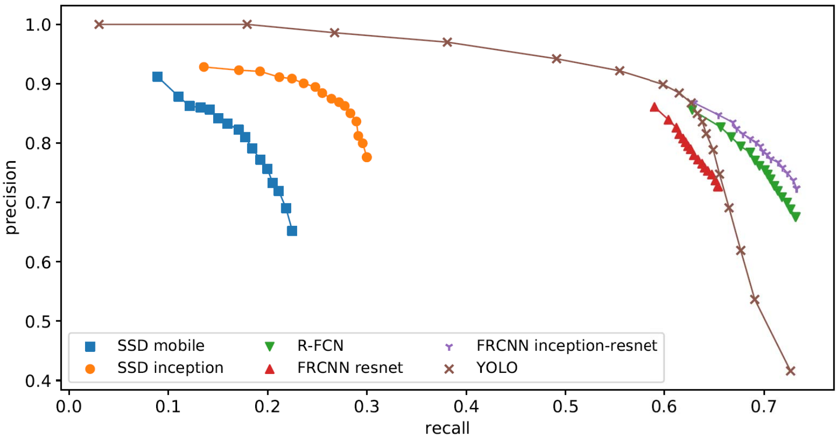

Fig. 3 Precision and recall of studied models.

training time should be considered for re-training false negative examples. The current training time might be adequate for weekly or monthly re-training.

6. CONCLUSION

Drones can be used in a hostile way, which might threat the security of a person or a facility. To detect drones that are flying near core facilities like a nuclear power plant, we introduce the drone tracking system. Our system uses a PTZ camera and the object detection algorithm. Object detection algorithm identifies a drone in an image that is captured from the PTZ camera, and PTZ actions are generated to track the drone. Challenges of our system are the small size of drones and real-time constraints. On the labeled multi-rotor drone data set, we train six convolutional object detection models and evaluate the accuracy and the speed. Our study shows that Faster R-CNN Inception Resnet scores the highest accuracy in terms of F-measure, followed by R-FCN and YOLOv2. In testing speed, SSD with MobileNet model is the fastest, and SSD inception and YOLOv2 models also process more than 10 frames per second. Considering speed and accuracy trade-offs, YOLOv2 might be the most appropriate detection model in our system. As the future work, we will further investigate different configurations of object detection models, such as the number of proposals or the input image resolution.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[2] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," *arXiv preprint arXiv:1611.10012*, 2016.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *arXiv preprint arXiv:1512.02325*, 2015.

[4] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[5] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[6] Y. Li, K. He, J. Sun *et al.*, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 379–387.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[9] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, 2017, pp. 4278–4284.

[10] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[11] L. Mejias, S. McNamara, J. Lai, and J. Ford, "Vision-based detection and tracking of aerial targets for uav collision avoidance," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 87–92.

[12] 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, pp. 879 – 892, 2017.