

DEEP LEARNING TECHNIQUE - BASED DRONE DETECTION AND TRACKING

Nguyen Minh Quang¹, Nguyen Tran Hiep¹, Nguyen Son Hai²,
Do Nam Thang³, Truong Xuan Tung^{1*}

Abstract: The usage of small drones/UAVs is becoming increasingly important in recent years. Consequently, there is a rising potential of small drones being misused for illegal activities such as terrorism, smuggling of drugs, etc. posing high-security risks. Hence, tracking and surveillance of drones are essential to prevent security breaches. This paper resolves the problem of detecting small drones in surveillance videos using deep learning algorithms. Single Shot Detector (SSD) object detection algorithm and MobileNet-v2 architecture as the backbone were used for our experiments. The pre-trained model was re-trained on custom drone synthetic dataset by using transfer learning's fine-tune technique. The results of detecting drone in our experiments were around 90.8%. The combination of drone detection, Dlib correlation tracking algorithm and centroid tracking algorithm effectively detects and tracks the small drone in various complex environments as well as is able to handle multiple target appearances.

Keywords: UAV; Drone detection and tracking; SSD-MobileNet-v2; Correlation tracker; Centroid tracker.

1. INTRODUCTION

In recent years, the usage of unmanned aerial vehicles (UAVs), which are publicly known as drones, has significantly increased. Because of their accessibility and ease of use, UAVs are widely used for many purposes, such as the delivery of goods and medicines, surveying, the monitoring of public places, agriculture, etc. However, the wide and rapid spread of UAVs causes danger when the illegal flight of drones is used for crimes such as smuggling (the illegal transportation of goods at borders, in restricted areas, prisons, etc.), illegal video surveillance, and interference with aircraft flying [1]. The development of drone surveillance systems is necessary and one of the most important request for drone detection and tracking system is real-time performance.

To create a robust, efficient drone detection and tracking system, many researchs were presented. Michael Jian *et al.*, 2018 [2] described a system based on phase-interferometric Doppler Radar. Dongkyu 'Roy' Lee *et al.*, 2018 [3] introduced a system based on machine learning and OpenCV library. Janousek *et al.*, 2019 [4] created an autonomous system that detects and recognizes the moving UAV using YOLO method and Mean square error (MSE) image comparison method. Ulzhalgas Seidalieva *et al.*, 2020 [1] addressed the problem of real-time drone detection based background subtraction and convolutional neural network (CNN).

In this paper, we introduce an efficient drone detection and tracking algorithm by combining the SSD-MobileNet-v2 [5] object detection and Dlib correlation tracker [6] as well as centroid tracking algorithm [7]. We first convert the video into a sequence of frames and set it as input of the trained model which was fine-tuned on the our synthetic drone dataset to recognize the drone in frames and achieve the bounding box around targets. The information of the targets bounding box is transmitted to Dlib correlation tracker and centroid tracker to track targets on sequence frames along with their ID.

The remainder of this paper is organized as follows. First, we introduce the object detection algorithm, the process from dataset prepare along with setting parameters for training to get the trained model as well as the Dlib correlation tracker and centroid tracker algorithms in Section II. We present our experiment results and evaluate the performance of our drone detection and tracking system in Section III. Finally, we end our paper with a conclusion in Section IV.

2. METHODS

2.1. SSD-MobileNet-v2

Object detection is an important task in computer vision applications. The SSD MobileNet-v2 model detector, and its drone detection capabilities, are analyzed and discussed in this paper.

The SSD-MobileNet-v2 is divided into two parts, of which MobileNet is for object prediction, and Single Shot MultiBox Detector (SSD) is to determine the classification results [8]. MobileNet-v2 works as a features extractor. Features are fed into the SSD network to determine the class and location of the detected objects on the captured images. The advantage of the SSD-MobileNet-v2 is that it provides a more balanced relationship of speed and accuracy when compared to other state-of-the-art models with similar network architecture such as YOLO and Faster-RCNN [9]. The SSD-MobileNet-v2 model is a part of the Tensor-Flow Object Detection API and is modelled on the MS-COCO Dataset that consists of more than 300,000 images and 80 object classes but this dataset does not include object drone.

2.2. Drone detection using SSD-MobileNet-v2

2.2.1. Data preparation

Dataset preparation is one of the most important step of deep learning model training process. It is crucial and can significantly affect the overall performance and usability of trained model. Drone Dataset is not popular or does not be provided freely, so we have to create a custom Synthetic Drone Dataset to train the model using transfer learning. Transfer Learning's fine-tuning technique is used to re-train the model with the assistance of a custom dataset which includes 25,000 synthetic drone images that the original model was not trained on. The drone images were captured on video which presented the drone with various perspectives and angles along with pre-processed to get drone images consisting of multiple sizes of the drone on white backgrounds. The background images were collected from many sources with the aim of increasing the complexity of the environment in which the drone was operating. The dataset generation program [10] are originally used for create synthetic fruits dataset.

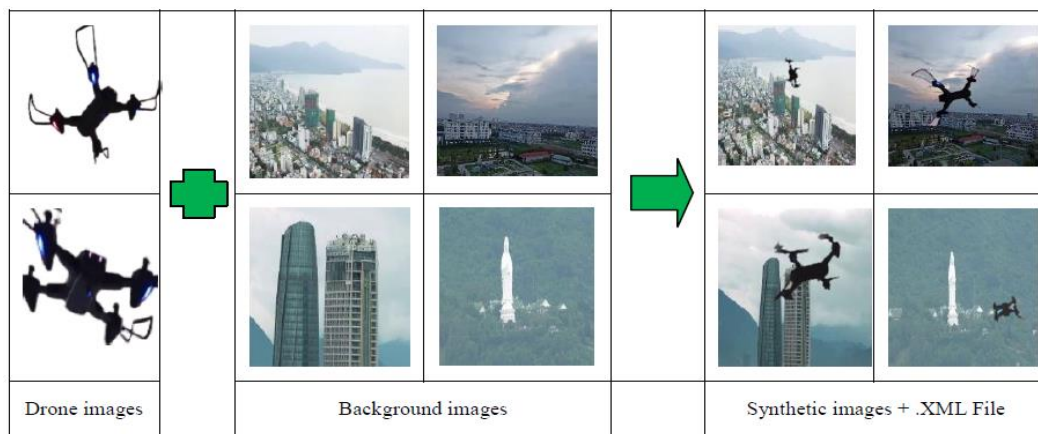


Fig.1. Create synthetic dataset process.

The XML files stored our annotations. We have chosen to do this in VOC XML format which means we will create one XML file for each generated image. These tell our model where we placed the drone.

2.2.2. Training process

The training process took place on Google's Colab application. The training process with more detail steps is shown as fig.2.

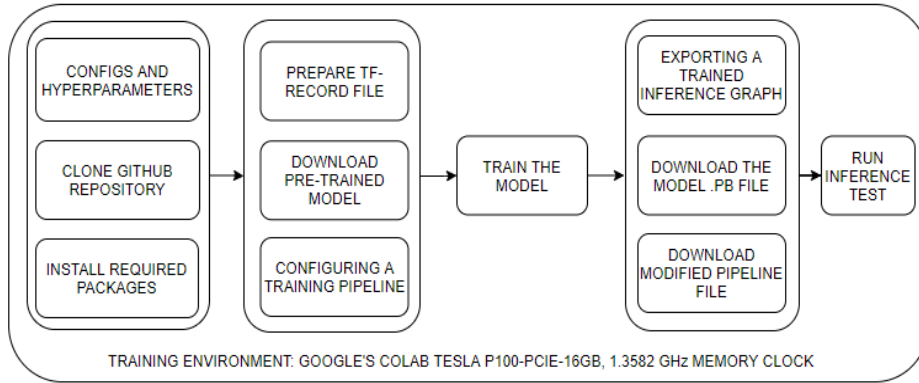


Fig.2. Google Colab training process.

The dataset is splitted into two sets included 80% for training and 20% for testing. It is extremely important that the training set and testing set are independent of each other and do not overlap. TF-Records file was generated for the custom dataset which was needed for training process.

The batch_size and epochs number were set in different values for training process.

The pre-trained model was downloaded from TensorFlow Detection [11] Model Zoo which we used as initial checkpoint for transfer learning. In this paper the model SSD_MobileNet_v2_coco was used.

The training process run automatically and finished when the pre-setting epochs number is reached. The value of batch_size and epoch is got by the experiment. First, we set the bath_size value then training with different value of epoch. In the training process, the training error is decreased gradually. The optimal value of epoch that make the training process stop before the training error increase again is optimal value. The Exporting step gave us an inference graph that we used for testing the trained model. The .pb file and .config file are used for running the model in OpenCV.

2.3. Drone tracking using Dlib correlation tracker and object centroid tracking algorithm

When a target is located in one frame of a video, it is often useful to track that object in subsequent frames. Every frame in which the target is successfully tracked provides more information about the identity and the activity of the target [12]. This paper used Dlib's correlation tracker combined with an object centroid tracking algorithm to implement drone tracking and counting in the video.

Dlib correlation tracker is widely used in image processing techniques for object tracking. Separate filters for translation and scale estimation are learnt by the tracker, which gives a performance advantage over the other existing tracking by detection approaches [13]. Correlation tracking method attempts to find the position and scale of an object in the current frame by using a known object bounding box in the previous frame.

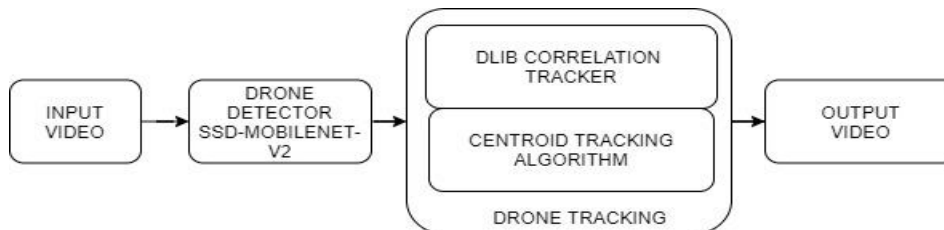


Fig.3. Diagram of drone detection and tracking system.

On the other hand, centroid tracking algorithm is used to track the centroid of the detected object for each subsequent video frame. The Euclidean distances between each pair of centroids are used to associate the new object's centroid with the previous object's centroid. This approach for object detection and tracking in a video is shown in fig.3.

The detected objects from the SSD-MobileNet-v2 Drone Detector are treated initially as targets in the first frame. The target is tracked by correlating the filter in next frame. The objects are recognized by a deep learning neural network in subsequent frames and are then used for tracking. The maximum correlation output value indicates the target and its new position. The coordinates of the object's location are then updated based on the new location. The output of SSD-MobileNet-v2 based Drone Detector is a class of object and bounding box coordinates. The bounding box, and its centroid coordinates, are used to initiate the object tracker. The output of the object tracker is the aforementioned bounding box and tracked centroids as well as an object ID (multiple object detection case).

If the system that the object detector combines with is successfully coupled with an object tracking system [14] (object detection is not run on each individual frame) it can achieve a quicker overall process and therefore provides a more viable option for real time requests. The value of skip_frame is the period that drone detector is ran one time.

2.4. Evaluation Dataset

To evaluate the performance of drone detector and drone detection and tracking system, the custom evaluation dataset is used. This dataset includes videos and images that were captured by smart phone camera.

Tab.1. Videos for algorithm evaluation.

No.	Number of frame	Resolution	Usage
Video1.mp4	320	1080*1920	Single object detection and tracking (SOT)
Video2.mp4	300		
Video3.mp4	320		
Video4.mp4	300		Multiple object detection and tracking (MOT)
Video5.mp4	330		
Video6.mp4	300		Test for different skip frame value
Images	500		Test Drone detector

Intersection Over Union (IOU) [15] value are used to evaluate the accuracy of the drone detector and combine algorithm in the case of single object tracking (SOT). We compute the Intersection of the area of the predicted bounding box, and the area of the ground-truth bounding box, and divide by the Union of the two areas. The accuracy is then the average of IOU for all the frames. For multiple object tracking performance evaluate, the Multiple Object Tracking Accuracy (MOTA) [15] is used.

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDS_t)}{\sum_t GT_t}$$

Where, FN (False negative) is the number of time that target is missed. FP (False positive) is the number of time that the tracking results are wrong. IDS are the number of time that target's IDs are switched. GT are the number of ground-truth box in all frames.

3. EXPERIMENTAL RESULTS

The small drone that we used for creating the training and testing data sets for the drone detector and drone detection and tracking system is a mini quadcopter drone. This type of drone has a popular design and is widely used in amateur photography.

The testing process was run on Dell Inspiron with Intel(R) Core (TM) i5-3210 CPU; 8.00 GB RAM and Geforce GT 640M NVIDIA Graphic Card. Ubuntu 18.04 operation system is installed. The programming process used is Python 3.6 and OpenCV 4.0 version. We also test the algorithm on Intel(R) CEON E3-1231 v3 CPU; 8.00 GB RAM and ZOTAC-1060, 6GB Graphic Card which was installed using Ubuntu 18.04 operation with CUDA 10.1 to compare the processing speed and accuracy of algorithm in different hardware configuration.

A video with our experimental results can be found at the link: <http://youtu.be/KKfrk30EY4M>.

3.1. Drone Detection

The training results with different set of parameters are shown in tab.2. The fine-tuned model was test on 500 images which the model had not been trained before. The results showed that with the specific dataset, the value of batch_size and epoch number which controls the accuracy of the estimate of the error gradient when training neural networks are the important hyperparameters that influence the dynamics of the learning algorithm.

Tab.2. Drone detection training results with diffrent setting parameters.

Dataset	Epochs	Batch_size	Test images	Confidence	Correct detection	False detection	Percent
25,000 images 80% for train 20% for test	100,000	4	500	≥ 0.5	335	165	67%
	60,000	4			390	110	78%
	150,000	8			454	46	90.8%
	200,000	8			370	130	74%

The best result achieved was 90.8% of accurate detection when the hyperparameters were set as 8 for batch_size and 150,000 for training epochs.



Fig. 4. Drone detector evaluation.

We use the IOU (Intersection Over Union) value to evaluate the Drone detector with above setting confidence value. The fig.4 shows the Drone detection evaluation results, the aqua bounding box is ground truth box that is achieved by handcraft; the red bounding box is prediction box that is generated by drone detector with confidence value is set as 0.5.

Tab.3. *The average of IOU.*

Number of Image	Confidence	Average of IOU
454	0.5	0.67

Tab.3 shows the average of IOU value. Normally, if the IOU value is higher than 0.5 then the Detector is considered as a “good” Detector.

On the other hand, from fig.4, we can see that the confidence value and IOU value depended on the size of target when compare with the size of frames. We use the size_compare value to measure the relation between the size of drone and the size of frame in percentages. When the target is small compared with the frame size (about 1/16 the size of frame), the IOU and confidence value are lower. When the target size is larger enough compared with the frame size, those values are higher.



a) Strong light condition



b) complex background



c) Drone flying closed trees



d) Drone flying closed building

Fig. 5. *Drone detector test result.*

The Drone Detector was tested on images that were captured from previously unseen video footage. The detected result is shown in fig.5. It can be seen that the Drone Detector effectively recognized small drones in strong light fig.5.(a), complex background fig.5.(b), drone fly close the trees fig.5.(c) and drone fly close the buildings fig.5.(d).

3.2. The combination of drone detection and tracking algorithm

3.2.1. Algorithm testing with different value of skip frame

Tab.3. shows the object detection and tracking with different Skip_frame value, the results are

achieved when algorithms are run on both CPU and GPU configuration. We can see that, with the CPU configuration, different values of skip frame directly affect the accuracy and running speed of the system. For the GPU configuration, we can see that, different value of Skip_frame affect the running speed of the system and provide the same multi object tracking accuracy. The selection pair of Skip_frame value and the confidence threshold are important for improving the running speed while maintaining the detection and tracking accuracy.

Tab.4. Drone detection and tracking with different Skip_frame value.

Input video	Confidence	Skip Frames	CPU		GPU	
			FPS	MOTA	FPS	MOTA
Video6. mp4	0.5	1	9.22	0.808	17.62	0.878
		3	16.07	0.908	25.52	0.945
		5	18.21	0.793	29.39	0.923
		9	19.81	0.868	34.37	0.966
		13	21.22	0.963	39.51	0.987

For the GPU configuration, we can see that, different values of Skip_frame affect the running speed of the system and provide the same multi object tracking accuracy. The selection pair of Skip_frame value and the confidence threshold are important for improving the running speed while maintaining the detection and tracking accuracy.

The drone detection and tracking system was also tested on video and managed to detect a small drone with the use of a smart phone camera. The results are shown in fig.7.



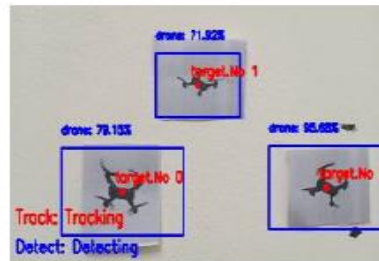
a) Drone tracking



b) Drone detecting and tracking



c) Drone detecting and tracking in strong light and closed trees condition.



d) Multi-Drone detection and tracking

Fig. 6. Drone detection and tracking system test result.

The number of targets and red dots detected indicate a tracking result while the blue bounding boxes indicate the class probability. Fig.6(a) shows the tracking result without object detection, fig.6(b, c) show both the object detection and tracking results in different background condition, fig.6(d) shows the multiple object detection and tracking result.

From these results we can see that when combining detection and tracking algorithms in a system, we can achieve the system with the improvement of system perform in both running speed and tracking accuracy.

3.2.2. Single object tracking and multiple object tracking

Tab.5. Single object tracking.

	Input videos	Confidence	Skip Frames	FPS	Accuracy (IOU)
Run on CPU	Video1.mp4	0.5	13	21.78	0.55
	Video2.mp4			22.44	0.58
Run on GPU	Video1.mp4	0.5	13	41.61	0.74
	Video2.mp4			41.39	0.76

Tab.5 shows the single object tracking result when algorithms are run on both CPU and GPU configuration. We can see that, with the same input videos and setting parameters, the achieved of FPS value and tracking accuracy when the algorithm is ran on GPU configuration are higher than CPU configuration's.

Tab.6. Multiple object tracking.

	Input videos	SkipFrame	FPS	MOTA
Run on CPU	Video3.mp4	13	22.53	0.660
	Video4.mp4		21.57	0.773
	Video5.mp4		19.38	0.796
Run on GPU	Video3.mp4	13	40.50	0.940
	Video4.mp4		40.95	0.933
	Video5.mp4		39.39	0.954

Tab.6 shows the multi object tracking result when algorithms are run on both CPU and GPU configuration. We can see that, with the same input videos and setting parameters, the achieved of FPS value and tracking accuracy when the algorithm is ran on GPU configuration are higher than CPU configuration's. The result shows that the combination of object detection and object tracking algorithms provides an effective solution for real-time small drone detection and tracking. The system also performed good characteristic for handle multiple object tracking.

4. CONCLUSIONS

In this paper, we present a drone detection and tracking system based on deep learning algorithms. We can see that, by leveraging existed convolution neural network model and transfer learning technique as well as Google's Colab application, we can develop the robust system for recognizing moving objects in input videos using a small custom dataset. The combination of object detection model and object tracking algorithm provides an effective solution for real-time small drone detection and tracking as well as handles multi-target

tracking problem.

However, there are some problems in the proposed system that will take more research to improve. The first problem is rate of false detection (FP, FN) in some cases. This problem can lead to the bad effects for the performance of whole system. Secondly, the number of video in dataset for evaluate is small, it causes the evaluation results which just reflect a local meaning.

As further research, dealing with following problems and extension task will be focused on. Firstly, the quality of the synthetic dataset that directly affect the performance of the whole system is needed to improve. Base on this, the dataset for create a multi-type of drone detection and tracking system will be expanded. Secondly, the research should involve the problem of data fusion where the information of camera-based drone detection will be associated to the information from other detection method such as radar-based method, acoustic-based method or RF-based method. Additionally, the development of deployable application that applies to realizable Anti-drone system will be done complete.

REFERENCES

- [1]. Ulzhalgas Seidaliyeva, Daryn Akhmetov, Lyazzat Ilipbayeva, Eric T. Matson “*Real-Time and Accurate Drone Detection in a Video with a Static Background*”, Sensors 2020, 20, 3856; doi:10.3390/s20143856.
- [2]. Michael Jian, Zhenzhong Lu and Victor C. Chen, “*Drone Detection and Tracking Based on Phase-Interferometric Doppler Radar*”, 2018 IEEE Radar Conference.
- [3]. Dongkyu 'Roy' Lee, Woong Gyu La, and Hwangnam Kim, “*Drone Detection and Identification System using Artificial Intelligence*”, 2018 International Conference on Information and Communication Technology Convergence (ICTC).
- [4]. J. Janousek, P. Marcon, J. Pokorny, and J. Mikulka, “*Detection and Tracking of Moving UAVs*”, 2019 Photonics Electromagnetics Research Symposium.
- [5]. A. Howard et al., “*MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*”, Computing Research Repositor, arXiv:1704.04861, 2017.
- [6]. Dlib C++ Library, (2018) “*Correlation Tracker*,” [Online]. Available: <http://dlib.net/imaging.html#correlationtracker>.
- [7]. Adrian Rosebrock, *Simple object tracking with OpenCV*, Available at: <https://www.pyimagesearch.com/2018/07/23/simple-object-tracking-with-opencv>.
- [8]. Yujie Du, Mingyu Gao, Yuxiang Yang, Jing Zhang², Zhongfei Yu, “*A Target Detection System for Mobile Robot Based On Single Shot Multibox Detector Neural Network*”, 2018 IEEE 4th International Conference on Control Science and Systems Engineering.
- [9]. Hashir Ali, Mahrukh Khursheed, Syeda Kulsoom Fatima, “*Object Recognition for Dental Instruments Using SSD-MobileNet*”, 2019 International Conference on Information Science and Communication Technology (ICISCT).
- [10]. Brad Dwyer, “*How to Create a Synthetic Dataset for Computer Vision*”, <https://blog.roboflow.com>.
- [11]. Priya Dwivedi (2017). “*Is Google Tensorflow Object Detection API the easiest way to implement image recognition?*”. Available at: <https://towardsdatascience.com/is-google-tensorflow-object-detection-api-the-easiest-way-to-implement-image-recognition-a8bd1f500ea0>.
- [12]. G. Gamage, I. Sudasingha, I. Perera, D. Meedeniya, “*Reinstating Dlib Correlation Human Trackers Under Occlusions in Human Detection based Tracking*”, 2018 International Conference on Advances in ICT for Emerging Regions (ICTer) : 092 – 098.
- [13]. Lasitha Mekkayil, Hariharan Ramasangu, “*Object Tracking with Correlation Filters using Selective Single Background*”, arXiv:1805.03453v1 [cs.CV] 9 May 2018.
- [14]. Adrian Rosebrock, *OpenCV People Counter* Available at : <https://www.pyimagesearch.com/https://www.pyimagesearch.com/2018/08/13/opencv-people-counter>.
- [15]. B. Keni and S. Rainer, “*Evaluating multiple object tracking performance: the clear mot metrics*”, EURASIP J. Image Video Process, Dec. 2008.

TÓM TẮT

HỆ THỐNG TỰ ĐỘNG PHÁT HIỆN VÀ THEO DÕI DRONE SỬ DỤNG KỸ THUẬT HỌC SÂU TIỀN TIẾN

Cùng với sự phát triển của công nghiệp sản xuất, các loại thiết bị bay không người lái kích thước nhỏ (còn được gọi là drone) ngày càng được sử dụng rộng rãi trong nhiều lĩnh vực. Tuy nhiên, việc sử dụng drone một cách thiếu kiểm soát có thể mang đến những nguy cơ tiềm ẩn như: sử dụng drone cho mục đích khủng bố, vận chuyển chất cấm, các hoạt động trinh thám, xâm nhập khu vực cấm bay,... Xây dựng hệ thống tự động phát hiện và theo dõi các thiết bị bay không người lái là một nhiệm vụ quan trọng trong bài toán giám sát, bảo vệ an ninh trên không. Bài báo sử dụng kỹ thuật học chuyển tiếp (transfer learning) để huấn luyện lại mạng nơ-ron học sâu SSD-MobileNet-v2 trên tập dữ liệu nhân tạo, kết quả nhận dạng chính xác mục tiêu đạt được là 90.8%. Kết hợp thuật toán nhận dạng drone với thuật toán bám đối tượng theo thuật toán bám tương quan và thuật toán bám tâm đối tượng có thể nhận dạng và theo dõi hiệu quả đối tượng drone với kích thước nhỏ trong các điều kiện khác nhau cũng như có khả năng phát hiện và theo dõi nhiều mục tiêu cùng lúc.

Từ khóa: UAV; Phát hiện và theo dõi drone; SSD-MobileNet-v2; Thuật toán bám tương quan; Bám tâm đối tượng.

Received April 07th 2021

Revised June 04th 2021

Published June 10th 2021

Author affiliations:

¹Faculty of Control Engineering, Le Quy Don Technical University;

²East Asia University of Technology;

³Academy of Military Science and Technology.

*Corresponding author: xuantung.truong@gmail.com.