Multi-model deep learning drone detection and tracking in complex background conditions

Kim-Phuong Phung, Thai-Hoc Lu, Trung-Thanh Nguyen, Ngoc-Long Le, Van-Phuc Hoang, Huu-Hung Nguyen
Le Quy Don Technical University, no. 236 Hoang Quoc Viet Str., Hanoi, Vietnam

***Abstract*—The recent popularity of drones with quadcopter layouts is threatening public safety and personal privacy. With the ability to hover and perform complex maneuvers even in indoor conditions, equipped with video cameras as well as capable of carrying hazardous materials, drones can truly become a security threat, especially to vulnerable organizations. Therefore, detecting and tracking drones in secured areas poses an urgent task for the surveillance system. In this paper, we design a real-time drone detection and tracking system with the combination of multiple deep learning and computer vision techniques: 1) Yolo-v4 model for detecting drones and 2) visual models for tracking drones. Besides, we have collected and labeled a larger drone dataset by mixing the existing datasets with our collected images. We evaluated three deep learning models for drone detection on this dataset and acquired the Yolo-V4 model to be the highest detection performance with AP = 34.63%. Combining this detection model and the existing visual tracking modules can boost the drone tracking up to more than 20fps for different backgrounds at around 700m by using an usual PC without GPU.**

***Index Terms*—Drone Detection, Drone Tracking, Convolutional Neural Network, Yolo-V4.**

I. Introduction

Drones are vehicles which can fly remotely or autonomously without a human operator. Thanks to recent technological advances, drones have been developed and used for a wide range of applications from government authorities to commercial related tasks used by civilians such as border security, agriculture, construction, law enforcement,wildfire surveillance, and general cinematography [1]. However, their characteristics such as versatility, ease of use, cheap price as well as wide availability also bring serious security threats by malicious use for criminal activities. A recent report shows that drones have been used for evil purposes, such as collision hazards, deployment of explosive weapons, smuggling of illegal substances and privacy violations. To deal with these existing as well as future threats, the governments have to develop the right equipment against illegal drones. Therefore, the development of a system that regulates drone usages is extremely urgent.

Usually, drones have been possible to be detected, tracked and located by analyzing the signature of appearance captured by the individual or integrated equipments, such as radar [2], radio frequency (RF) sensor [3], acoustic sensor [4] and cameras [5]. With the development of deep learning, recently, researchers gradually improve the performance of drone surveillance systems from the detection stage to tracking, jamming and countermeasures. In the field of visual detection and tracking using surveillance cameras, there have been many improvements by applying deep learning models in the detection stage, which makes visual detection become an essential part of anti-drone surveillance systems.

In this paper, we propose the multimodel of deep learning and computer vision techniques. Several kinds of one-stage deep-learning-based drone detection methods were evaluated to choose the best performance (Yolo-v4). Then, combining the best deep-learning-based drone detection method with the visual tracking modules increases both tracking accuracy and frame rate. The proposed approach is analyzed by our collection drone dataset to seek a trade-off point between computational complexity and accuracy. The system can detect drones with a distance of at least 700m and provide continuous tracking at more than 20fps using an usual PC without GPU.

The rest of this paper is organized as follows. In Section II, we summarize the state-of-the-art drone detection and drone tracking approaches with computer vision and deep learning. Our design of multiple models for drone detection and tracking is introduced in Section III. The experimental results are given in Section IV with different scenarios and comparison to the state-of-the-art methods. Finally, Section V concludes this paper.

II. Related work

A. Drone detection methods using image processing and computer vision

Video-based technique determines the location and moving direction of a drone via visual motion. Imaging systems and cameras can be used both in the visual and infra-red spectrum to detect and classify drones. Not typically a primary detection source, electro-optical sensors use a visual signature to detect drones, while infrared sensors use a heat signature. High-performance camera systems provide images as forensic evidence. They are often equipped with a high zoom capability to show small objects at a distance; however, they have range limitations. Several researchers have suggested methods to detect a drone and its trajectory by using motion cues [6], visual marks [7], and shape descriptors [8]. Rozantsev [9] recovered drone trajectories by using multiple fixed ground cameras in a dynamic environment. Opromolla [10] exploited template matching and normalized cross-correlation metrics for drone detection. Gokcce [11] employed drone detection and distance estimation using conventional visual features such as histogram of gradients (HOG). The above mentioned methods can accurately locate and identify drones. However, since many similarities between the movements of drones and

other small flying object as birds exist; there are high false positives on the one hand combined with high false negative rates on the other due to the increasing number of drone types and atmospheric opacity

Higher detection and recognition accuracy of these computer vision approaches can be improved by innovative deep learning techniques. In [12], Recurrent Correlational Networks (RCN) including four networks with specific tasks was proposed by Yoshihashi to together detect and track small drones. The representation of non-target and target appearances from individual frames was determined by a convolutional layer. A ConvLSTM was used to learn the representations of motion from multiple frames. After that, correlation maps between the template and each subsequent frame were generated by cross-correlation layers to localize the target in the frame. Finally, fully connected layers were used to generate the confidence scores. A serial of YOLO network based on deep CNN has been used for detecting drone [13] [14] [15] [16]. Aker proposed an extension of YOLO, that is a single shot object detector. Saqib took advantage of fine tuning technique in the new version, YOLOv2 [14], to train a regressor for drone detection . Peng investigated different pre-trained CNN models including Zeiler and Fergus (ZF) and VGG16 coupled with the Faster R-CNN model for the detection of drones from video data [17]. The VGG16 and the ZF model were used as a transfer learning to compensate for the lack of sufficient dataset and to ensure the convergence during training for the model. However, multiple small objects in large space are challenging the approaches to distinctly detect each of them.

B. Drone tracking methods

Drone video object tracking is an important application of visual tracking technology. In recent years, there have been two mainstream methods for the development of visual tracking, tracking with correlation filters, and tracking with attention mechanisms.

Tracking with correlation filters

Exploiting CF for object tracking started with the method called the minimum output sum of squared error, MOSSE tracker [18]. The tracker is constructed and trained using gray-scale samples in the frequency domain for efficiency. Kernel cross-correlator (KCC) [19] provides a novel solution for the CF-based framework with high expandability and brief formulation. For the task of visual object tracking in drone videos, several algorithms have been proposed based on correlation filtering. In [20], a fast-tracking stability measurement metric was designed, based on the peak-to-sidelobe ratio values, which made the DCF algorithms more robust to complicated appearance variations. In [21], a novel approach to repress the aberrances happening during the detection process was proposed, i.e., aberrance repressed correlation filter (ARCF). By enforcing the restriction on the rate of alteration in response maps generated in the detection phase, the ARCF tracker suppresses aberrances, and thus is more robust and accurate for tracking objects. By integrating three kinds of attention, namely contextual attention, dimensional attention, and spatiotemporal attention, into the correlation filter tracking framework, a drone tracker TACF [22] with multilevel visual attention was proposed, improving the robustness to challenging visual factors such as partial occlusion and clutter background.

Tracking with attention mechanism

In visual tracking, CNN-based methods with attention mechanisms can integrate different visual information to improve tracking accuracy. J. Choi et al. [23] proposed an attention network to switch among different features to select the suitable tracking mode. Z. Zhu et al. [24] incorporated optical flow based on deep learning into the tracking pipeline. Other tracking methods [25] used feature maps extracted from deep neural networks to select the appropriate tracking mode for better performance.

III. Proposed Method

A. Dataset Description

Data collection and preprocessing is a very important step when implementing a problem applying artificial intelligence. Deep learning models cannot work without data. When the dataset is too small, the common phenomenon easily leads to overfitting when the model cannot fully learn the attributes for the generalization. In this study, we built ourselves a dataset collected from the camera, internet images of drones, then manually labeled them using the LabelImg labeling tool and standardized label data file as YOLO annotation format. Figure 1 shows several images in the dataset.

Fig. 1: Sample images in the dataset

From public datasets, we collect labeled quadcopter drone images with different sizes and different background conditions. The collected images was mixed with our images obtained using a long range surveillance camera. As result, we have obtained a dataset of about 4,500 color images of drones that is used to train and evaluate the proposed algorithm later. This amount of data is sufficient for ensuring comprehensive and diversity for training and testing detection models.

B. The object detection models

In the literature, there are mainly two types of state-of-the-art object detectors. On one hand, we have two-stage detectors, such as Faster R-CNN (Region-based Convolutional Neural Networks) [26] or Mask R-CNN [27], that (i) use a Region Proposal Network to generate regions of interests in the first stage and (ii) send the region proposals down the pipeline for object classification and bounding-box regression. Such models achieve better accuracy performance, but are

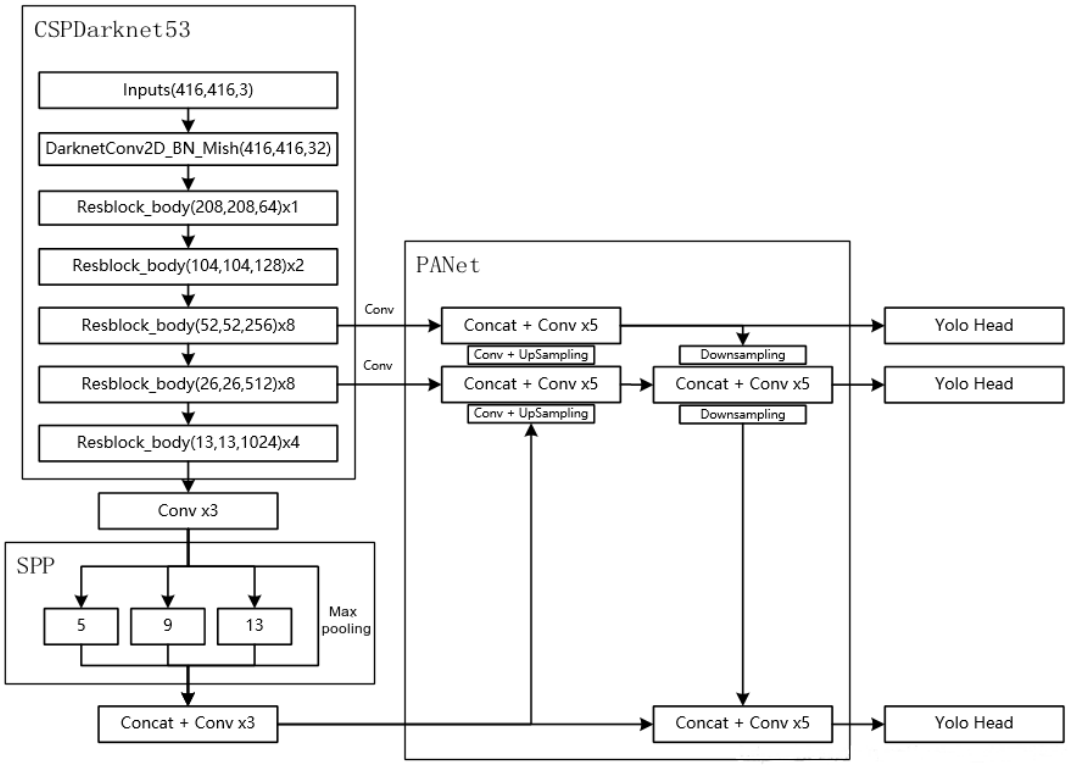

Fig. 2: YoloV4 architecture consists of three parts: CSPDarknet53 as the backbone, SPP is used as an additional module of Neck and PANet is used as a feature fusion module of Neck, YOLOv3 serves as the Head.

typically slower. On the other hand, we have single-stage detectors, such as YOLO (You Only Look Once) [13] and SSD (Single Shot MultiBox Detector) [28], that treat object detection as a simple regression problem by taking an input image and learning the class probabilities and bounding box coordinates. Such models reach lower accuracy rates, but they are much faster than two-stage object detectors.In this study, we apply models of single-stage detectors to perform object detection to ensure the execution speed that meets the real-time requirement. Specifically, we perform training and compare drone detection results of three models: SSD MobileNetV2 [29], EfficientDet-D3 [30] and YoloV4 [16]. Finally, we have YoloV4 as the most efficient drone detection model for usages with real world complex background conditions and chose this model to evaluate in combination with tracking models.

YoloV4 Details

Yolo architecture includes: base networks are convolution networks that perform feature extraction. The next part is the extra layers applied to detect objects on the base network feature map. Yolo's base network uses mainly convolutional layers and fully connected layers. Before YoloV4 [16], there were 3 versions of Yolo: YoloV1 [13], YoloV2 (Yolo9000) [14], YoloV3 [15] which made strong strides in object detection. The appearance of YoloV4 has made great strides compared to the previous 3 versions. YoloV4's architecture has made object detection more accessible to people without powerful computing resources. The architecture of YoloV4 consists of 3 main parts: Backbone, Neck and Head. YoloV4 uses CSPDarknet53 as backbone with the task of extracting features from input images. The next step is to mix and combine the features formed in the ConvNet backbone to prepare for the detection step, YoloV4 considers a few options for the neck and finally chose to combine SPP (Spatial Pyramid Pooling) [31] and PANet [32]. YoloV4 deploys the same Yolo head as YoloV3 for detection with the anchor-based detection steps, and three levels of detection granularity. Some new features in YoloV4: Weighted residual connection (WRC), Cross Stage Partial (CSP), Cross minibatch Batch Norm (CmBN), Mish activation, Mosaic data augmentation, DropBlock regularization, and CIoU loss. Figure 2 shows the general architecture of YoloV4.

Besides, bag of freebies and bag of specials are methods applied to backbone and detector to increase the accuracy of YoloV4. Bag of freebies are methods that only change the training strategy or only increase the training cost (nothing to do with inference). Bag of specials are plugin modules and post-processing methods that only increase the inference cost by a small amount but can significantly improve the accuracy. In addition to the modified to the state-of-the-art methods including CBN (Cross-iteration Batch Normalization), PAN (Path aggregation network),etc., are now more efficient and suitable for single GPU training.

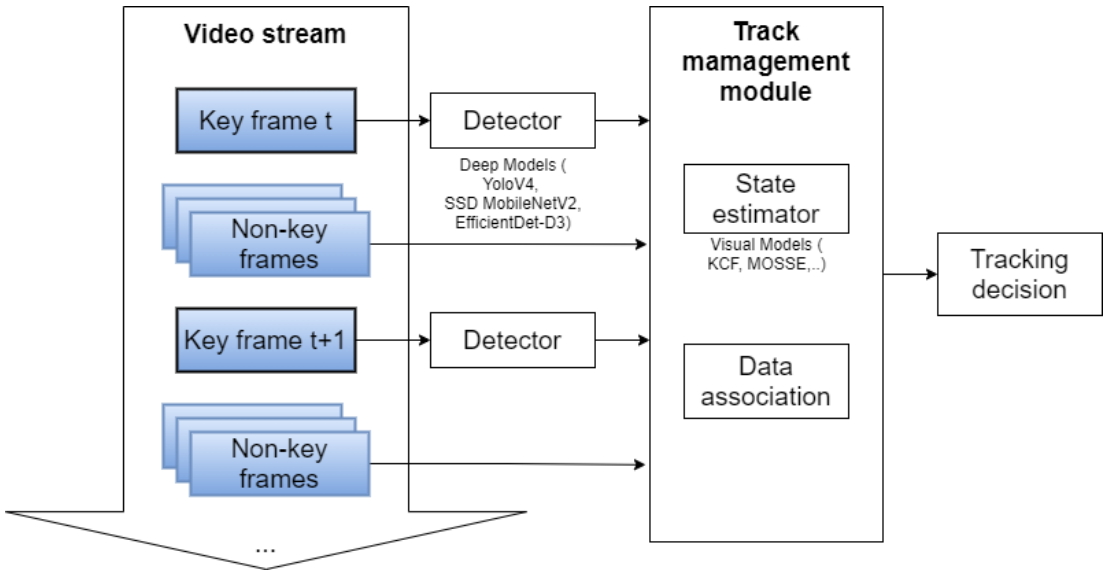

Fig. 3: Combination of detection and tracking with key frames

C. Object tracking algorithms

Object tracking is one of the most trendy and under research topics of Computer Vision that implies several issues that should be considered while creating tracking systems, such as visual appearance, occlusions, camera motion, and so on. In several tracking algorithms Convolutional Neural Network (CNN) has been applied to utilize its effectiveness in feature extraction that convolutional layers can characterize the object from different perspectives and prevent tracking process from misclassification.

The goal of object tracking is to estimate the state of the selected object in the subsequent frames. The object being tracked is usually marked using a rectangle to indicate its location in the initial frame. When there are no changes in the environment, object tracking is not overly complex, but this is rarely the case. Various disturbances are presented in real world scenarios. These disturbances might include occlusion, variations in illumination, changes of viewpoint, rotation, blurring due to motion, etc. The task of designing a robust and efficient tracker is known to be a very challenging one.

We use object tracking methods available in the open-source Computer Vision (OpenCV) library to track drones. OpenCV basic tracking algorithms are chosen for its versatility and simplicity of use. The OpenCV library includes eight algorithms for object tracking, which are available through OpenCV tracking API. In the table I we provide general information about the current-existing algorithms in the OpenCV library with their publication years and references to research papers detailing their implementation. In general, tracking an object in a video stream involves several steps: a) choosing the tracker, b) selecting the object (target) from the initial frame with the bounding box, c) initializing the tracker with information about the frame and bounding box, and d) processing the remaining frames and find the new bounding box of the object. The last step usually implements those above steps in a loop.

In our system, we combined the drone detection module with the tracking module to continuously monitor the detected drones in real time using the existing visual tracking module. It helps us to greatly reduce the processing time for object tracking algorithms that usually require much less computation power than detection models. Combined with the tracking algorithm, the detection model doesn't have to be deployed on the whole video, instead, we just need to detect objects on every n-frame and track objects on the remaining frames of the video. In the case that the detection model may not detect the object at some frames or cannot detect the object when it is obscured; using the tracking algorithms can help to predict the position of the target on those frames, thus ensuring that the object is monitored continuously.

IV. Experimental Results

In this section, we evaluate the proposal method in two parts. First, we evaluate the object detection models on a test set of 1350 drone images built by the research team, which is independent of 3150 training images. Then, based on the selection of the detection model, the object tracking algorithms will be evaluated on each model.

A. The object detection models

Training details: We trained 3 models including SSD MobileNetV2, EfficientDet-D3 and YoloV4 on the training dataset of 3150 drone images. The training dataset is divided into 80/20 scale for training/validation. Table II shows the training details of three models.

Evaluation metrics: In this study, we used AP (Average Precision) as a metric to evaluate the detection models. The AP is an arithmetic that computes the inclusion of both precision and recall. AP is computed by calculating the

TABLE I: OpenCV single object trackers sorted by the year of their publication. Google Scholar Citations are accessed on April 21th 2020 .

No	Tracker Full Name (Abbreviation)	Publication Title and Reference	Publication Year (Google Scholar Citations)
1.	**Boosting**	Real-time tracking via on-line boosting [33]	2006 (1432)
2.	Multiple Instance Learning (**MIL**)	Visual tracking with online multiple instance learning [34]	2009 (2095)
3.	**MedianFlow**	Forward-backward error: Automatic detection of tracking failures [35]	2010 (802)
4.	Minimum Output Sum of Squared Error (**MOSSE**)	Visual object tracking using adaptive correlation filters [18]	2010 (1839)
5.	Tracking Learning Detection (**TLD**)	Tracking-learning-detection [36]	2011 (3275)
6.	Kernelized Correlation Filter (**KCF**)	High-speed tracking with kernelized correlation filters [37]	2014 (3131)
7.	**GOTURN** (Generic Object Tracking Using Regression Networks)	Learning to track at 100 fps with deep regression networks [38]	2016 (648)
8.	**CSRT** (Channel and Spatial Reliability Tracker)	Discriminative Correlation Filter with Channel and Spatial Reliability [39]	2017 (444)

TABLE II: Training details of three models

Model	input_image_size	initial_learning_rate	base_learning_rate	batch_size	framework	check_point
SSD MobileNetV2	320 x 320	0.027	0.079	32	Tensorflow Object Detection API	SSD MobileNet V2 FPNLite 320x320
EfficientDet-D3	512 x 512	0.027	0.0001	8	Monk Object Detection	EfficientDet-D3
YoloV4	416 x 416	none	0.001	64	Darknet of author AlexeyAB	yolov4.conv.137

average interpolation of the precision values over the recall values in $[0, 1]$. We use the interpolation method at 101 recall points which is the method that the authors of the dataset for object detection of COCO dataset proposed. We also use the AP metric proposed by these authors to evaluate the detection models on the COCO dataset, which includes: $AP^{\text{IoU=.50}}$ is the AP value at threshold of IoU is 0.5, $AP^{\text{IoU=.75}}$ is the AP value at threshold of IoU is 0.75 and AP is the average AP value for the threshold of IoU from 0.5 to 0.95 with each increment of 0.05. AP^{small}, AP^{medium}, AP^{large} are AP values calculated for objects less than 32^2, larger than 32^2 and smaller than 96^2 , larger than 96^2, respectively.

AP is based on the precision-recall curve. To reduce the impact of the wiggles in the curve, we first interpolate the precision at multiple recall levels before actually calculating AP. The interpolated precision p_{interp} at a certain recall level r is defined as the highest precision found for any recall level $r^{'} \geqslant r$:

$$p_{interp}(r) = \max_{r' \geq r} p(r') \tag{1}$$

Then, we divide the recall value from 0 to 1.0 with each increment of 0.01 into 101 points. Next, we compute the average of maximum precision value for these 101 recall values.

$$AP = \frac{1}{101} \sum_{r \in \{0.0, \ldots, 1.0\}} p_{interp}(r) \tag{2}$$

Compare results: In this section, the research team evaluates the test results on the models presented in subsection III-B on the test data set of 1350 images containing the drone that have been processed, labeled and separated from the training data set. From there, compare the performance of the best object detection models today. Table III shows the results with the measures presented above.

TABLE III: Compare detection result of three models.

Model	$AP(\%)$	$AP_{50}(\%)$	$AP_{75}(\%)$	$AP_S(\%)$	$AP_M(\%)$	AP_L
SSD MobileNetV2	20.78	45.78	15.09	3.16	15.66	27.91
EfficientDet-D3	24.03	52.24	16.19	0	19.72	30.70
YoloV4	**34.63**	**78.03**	**21.26**	**10.56**	**36.26**	**36.95**

As the results can be seen above, YoLoV4 gives the best results of the 3 models on all metrics with AP=34.63% approximately 10% more than the 2nd best model, EfficientDet-D3. Especially, the YoLoV4 model is able to detect small sized drones with AP_S = 10.56%, while the SSD MobileNetV2 is only 3% and even EfficientDet-D3 cannot detect. Figure 4 show some predicted images of the three models.

B. Combine object detection model and tracking algorithms for drone detection

It's very difficult to even for human eyes to detect drones in complex background of urban objects. For this reason, a single drone cannot be detected continuously in each image frame. We propose combining detection and visual tracking for better monitoring information. In this experiment, we evaluate the drone monitoring results in 2 ways: 1) using only the object detection model method and 2) using the detection model combined with different tracking algorithms. We used the method as shown in the Fig.3.

Evaluation dataset: We performed this test on two self-recorded videos visualized in Fig.5 : 1) One with a simple background of the sky, 2) another with a complex background (tree, road). Each video describes a moving drone in the distance from 500m to 700m. The system's task is to track the drone in the video. We have labeled the coordinates of the drone in the video and used them to compare it with predicted coordinates.

Evaluation methodology: Evaluation metric ensures valuable feedback about the algorithm being evaluated, hence choosing the right metric is essential in the evaluation process.

(a) SSD MobileNetV2 (b) EfficientDet-D3 (c) YoloV4

Fig. 4: Predicted results of the three models. Green boxes are the ground-truth and red boxes are predicted results.

Fig. 5: Testing condition. Simple background (left) with 2042 frames and complex background (right) with 2000 frames

We use the **Precision Plot** as our evaluation metrics. The precision plot evaluation metric is based on an average Euclidean distance between center locations of the tracked object and the manually labeled ground truth of all the frames in the test sequence. It represents how far the tracker drifts away from the actual target. It should be noted that when the tracker fails, if the Euclidean distance between center locations of the tracked object and the manually labeled ground-truth is greater than a threshold, we say the tracker fails. We set a thresholding in our measurements that are 10px. Along with that, we also evaluate the FPS, number of true prediction locations out of the total number of the manual labeled ground truth of all the frames in the test sequence.

Evaluation result: For each video, we make a series of test runs on a Linux based machine with i7-7500U @ 2.7GHz. For the object detection model, we use YOLOv4 because it is the best model selected in previous experience. The demonstration video of the experiment can be found by this link: https://www.youtube.com/watch?v=wOEogESMG80. We do the test with two methods: using only the detection model and using the combination of the object detection model and the tracking algorithm. The tracking algorithms used for valuation include KCF, CSRT, BOOSTING, MIL, TLD, MEDIANFLOW, MOSSE.

TABLE IV: Result on the first video with simple background

	True predict	Scale (%)	Precision plot	FPS
YOLOV4 only	1784	87.37	4.0568	6.5249
YOLOV4 + KCF	1415	69.29	4.8872	13.1470
YOLOV4 + CSRT	**1654**	**81.00**	**3.9768**	**10.5775**
YOLOV4 + BOOSTING	1623	79.48	4.3145	9.2950
YOLOV4 + MIL	1319	64.59	5.1119	9.0195
YOLOV4 + TLD	1210	59.26	5.3636	8.1447
YOLOV4 + MEDIANFLOW	1527	74.78	4.3786	13.7125
YOLOV4 + MOSSE	**1676**	**82.08**	**4.6404**	**14.4114**

In the first video, with a simple background of the sky, the YOLOv4 model can easily detect drones on video with the rate of true predictions being 1784/2042 equal 87.37% and precision plot equal 4.0568, but the resulting FPS is quite low (6.52 FPS). With the combination of the object detection model and the tracking algorithm, the system achieved 14.41 FPS with YOLOv4+MOSSE, the true predictions rate being 1467/2042 equal 82.08% and precision plot equal 4.640. With YOLOv4+CSRT the true predictions rate is 1654/2042 equal 81.00%, 8.3330 FPS and precision on plot is 3.9768. Using the combination of the object detection model and the tracking algorithm, we observed that the resulting differences of true prediction rate and precision plot between the two methods are minor, but the system can obtain much higher FPS.

TABLE V: Result on the second video with complex background.

	True predict	Scale (%)	Precision plot	FPS
YOLOV4 only	411	20.13	6.0670	5.6134
YOLOV4 + KCF	587	28.75	5.5274	11.2871
YOLOV4 + CSRT	**1268**	**62.10**	**5.4798**	**8.3330**
YOLOV4 + BOOSTING	947	46.38	6.2284	9.1414
YOLOV4 + MIL	857	41.39	6.5114	5.8678
YOLOV4 + TLD	681	33.35	6.1973	4.5715
YOLOV4 + MEDIANFLOW	782	38.30	6.2111	10.6527
YOLOV4 + MOSSE	**603**	**29.53**	**6.6368**	**12.9229**

In the second video with a complex background and the drone in the video is quite small in size, the YOLOv4 model can only detect with true prediction rate being 411/2000 (equal 20.13%) and FPS is 5.61, precision plot equal 6.0607. But when we use the combination of YOLOv4+CSRT, the true prediction rate is increased up to 1268/2000 (equal 62.10%), 8.3330 FPS and the precision plot is 5.4798. As we can see, the combination of the object detection model and tracking algorithm can help both drastically increase the accuracy in predicting the object's position and also increase the FPS. In this video, the model lost track of the drone in various frames. However, by using the tracking algorithm, we can track and predict the position of the drone on these frames, so the system's ability to continuously monitor the drone is greatly increased.

V. Conclusion

In this study, we have proposed drone detection and tracking using several recent neural network detection models. We have trained and evaluated these models to estimate the efficiency of applying deep learning in drone surveillance. We also prepare several real-world videos with simple and complex background conditions. As detector performance was significantly reduced in complex background conditions, we propose a combination of detection and tracking algorithms. The proposed method demonstrates stable detection and tracking with a long range pan tilt surveillance camera via experiment on the real-world videos. In comparison, despite using a modest requirement of computing power, the method was practically useful for automatic processing surveillance video from a pan tilt camera with minimum of 5 degrees in field of view and 720p resolution and detection range up to 800m. For future works, we intend to implement the methods on specially designed cameras to reach better range and speed performance. Further works will consider extending the model for multimodal (video, radar, audio, and RF data) networks, which can enhance the drone detection and classification performance of surveillance systems.

References

[1] UAV Commercial. Commercial drone market analysis by product (fixed wing, rotary blade, nano, hybrid), by application (agriculture, energy, government, media & entertainment) and segment forecasts to 2022. *Grand View Research: San Francisco, CA, USA*, 2016.

[2] Byung Kwan Kim, Hyun-Seong Kang, and Seong-Ook Park. Drone classification using convolutional neural networks with merged doppler images. *IEEE Geoscience and Remote Sensing Letters*, 14(1):38–42, 2016.

[3] Martins Ezuma, Fatih Erden, Chethan Kumar Anjinappa, Ozgur Ozdemir, and Ismail Guvenc. Micro-uav detection and classification from rf fingerprints using machine learning techniques. In *2019 IEEE Aerospace Conference*, pages 1–13. IEEE, 2019.

[4] Xianyu Chang, Chaoqun Yang, Junfeng Wu, Xiufang Shi, and Zhiguo Shi. A surveillance system for drone localization and tracking using acoustic arrays. In *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 573–577. IEEE, 2018.

[5] Shuowen Hu, Geoffrey H Goldman, and Christoph C Borel-Donohue. Detection of unmanned aerial vehicles using a visible camera system. *Applied optics*, 56(3):B214–B221, 2017.

[6] Shuowen Hu, Geoffrey H. Goldman, and Christoph C. Borel-Donohue. Detection of unmanned aerial vehicles using a visible camera system. *Appl. Opt.*, 56(3):B214–B221, Jan 2017.

[7] Lucas Vago Santana, Alexandre Santos Brandão, Mário Sarcinelli-Filho, and Ricardo Carelli. A trajectory tracking and 3d positioning controller for the ar.drone quadrotor. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 756–767, 2014.

[8] Eren Unlu, Emmanuel Zenou, and Nicolas Rivière. Using Shape Descriptors for UAV Detection. In *Electronic Imaging 2017*, pages pp. 1–5, Burlingam, United States, January 2018.

[9] Artem Rozantsev, Sudipta Sinha, Debadeepta Dey, and Pascal Fua. Flight dynamics-based recovery of a uav trajectory using ground cameras. *30Th Ieee Conference On Computer Vision And Pattern Recognition (Cvpr 2017)*, pages 10. 2482–2491, 2017.

[10] Roberto Opromolla, Giancarmine Fasano, and Domenico Accardo. A vision-based approach to uav detection and tracking in cooperative applications. *Sensors*, 18(10):3391, 2018.

[11] Fatih Gökçe, Göktürk Üçoluk, Erol Şahin, and Sinan Kalkan. Vision-based detection and distance estimation of micro unmanned aerial vehicles. *Sensors*, 15(9):23805–23846, 2015.

[12] Ryota Yoshihashi, Tu Tuan Trinh, Rei Kawakami, Shaodi You, Makoto Iida, and Takeshi Naemura. Differentiating objects by motion: Joint detection and tracking of small flying objects. *arXiv preprint arXiv:1709.04666*, 2017.

[13] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[14] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016.

[15] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

[16] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934, 2020.

[17] Muhammad Saqib, Nabin Sharma, Sultan Khan, and Michael Blumenstein. A study on detecting drones using deep convolutional neural networks. 08 2017.

[18] David Bolme, J. Beveridge, Bruce Draper, and Yui Lui. Visual object tracking using adaptive correlation filters. pages 2544–2550, 06 2010.

[19] Chen Wang, Le Zhang, Lihua Xie, and Junsong Yuan. Kernel cross-correlator. *CoRR*, abs/1709.05936, 2017.

[20] Yong Wang, Lu Ding, and Robert Laganiere. Real-time uav tracking based on psr stability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[21] Ziyuan Huang, Changhong Fu, Yiming Li, Fuling Lin, and Peng Lu. Learning aberrance repressed correlation filters for real-time uav tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[22] Yujie He, Changhong Fu, Fuling Lin, Yiming Li, and Peng Lu. Towards robust visual tracking for unmanned aerial vehicle with tri-attentional correlation filters, 08 2020.

[23] Jongwon Choi, Hyung Jin Chang, Sangdoo Yun, Tobias Fischer, Yiannis Demiris, and Jin Young Choi. Attentional correlation filter network for adaptive visual tracking. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4828–4837, 2017.

[24] Zheng Zhu, Wei Wu, Wei Zou, and Junjie Yan. End-to-end flow correlation tracking with spatial-temporal attention. *CoRR*, abs/1711.01124, 2017.

[25] Ning Wang, Wengang Zhou, Qi Tian, Richang Hong, Meng Wang, and Houqiang Li. Multi-cue correlation filters for robust visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[26] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[27] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[28] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[29] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[30] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. *CoRR*, abs/1911.09070, 2019.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *CoRR*, abs/1406.4729, 2014.

[32] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. *CoRR*, abs/1803.01534, 2018.

[33] Helmut Grabner, Michael Grabner, and Horst Bischof. Real-time tracking via on-line boosting. volume 1, pages 47–56, 01 2006.

[34] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 983–990, 2009.

[35] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Forward-backward error: Automatic detection of tracking failures. In *2010 20th International Conference on Pattern Recognition*, pages 2756–2759, 2010.

[36] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1409–1422, 2012.

[37] João F. Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(3):583–596, 2015.

[38] David Held and Silvio Savarese. Learning to track at 100 fps with deep regression networks. volume 9905, pages 749–765, 10 2016.

[39] Alan Lukežic, Tomáš Vojír, Luka Cehovin Zajc, Jirí Matas, and Matej Kristan. Discriminative correlation filter with channel and spatial reliability. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4847–4856, 2017.