SIP (2019), vol. 8, e5, page 1 of 13 © The Authors, 2019.

doi:10.1017/ATSIP.2018.30

ORIGINAL PAPER

Towards Visible and Thermal Drone Monitoring with Convolutional Neural Networks

YE WANG, YUERU CHEN, JONGMOO CHOI AND C.-C. JAY KUO

This paper reports a visible and thermal drone monitoring system that integrates deep-learning-based detection and tracking modules. The biggest challenge in adopting deep learning methods for drone detection is the paucity of training drone images especially thermal drone images. To address this issue, we develop two data augmentation techniques. One is a model-based drone augmentation technique that automatically generates visible drone images with a bounding box label on the drone's location. The other is exploiting an adversarial data augmentation methodology to create thermal drone images. To track a small flying drone, we utilize the residual information between consecutive image frames. Finally, we present an integrated detection and tracking system that outperforms the performance of each individual module containing detection or tracking only. The experiments show that, even being trained on synthetic data, the proposed system performs well on real-world drone images with complex background. The USC drone detection and tracking dataset with user labeled bounding boxes is available to the public.

Keywords: Deep learning, Detection, Tracking, Drone, Integrated system

Received 5 October 2018; Revised 5 December 2018

I. INTRODUCTION

There is a growing interest in the commercial and recreational use of drones. This in turn imposes a threat to public safety. The Federal Aviation Administration (FAA) and NASA have reported numerous cases of drones disturbing the airline flight operations, leading to near collisions. It is therefore important to develop a robust drone monitoring system that can identify and track illegal drones. Drone monitoring is however a difficult task because of diversified and complex background in the real-world environment and numerous drone types in the market.

Generally speaking, techniques for localizing drones can be categorized into two types: acoustic and optical sensing techniques. The acoustic sensing approach achieves target localization and recognition by using a miniature acoustic array system. The optical sensing approach processes images or videos to estimate the position and identity of a target object. In this work, we employ the optical sensing approach by leveraging the recent breakthrough in the computer vision field.

The objective of video-based object detection and tracking is to detect and track instances of a target object from image sequences. In earlier days, this task was accomplished by extracting discriminant features such as the scale-invariant feature transform (SIFT) [1] and the histograms of oriented gradients (HOG) [2]. The SIFT feature vector is attractive since it is invariant to object's translation, orientation, and uniform scaling. Besides, it is not too sensitive to projective distortions and illumination changes since one can transform an image into a large collection of local feature vectors. The HOG feature vector is obtained by computing normalized local histograms of image gradient directions or edge orientations in a dense grid. It provides another powerful feature set for object recognition.

In 2012, Krizhevsky *et al.* [3] demonstrated the power of the convolutional neural network (CNN) in the ImageNet grand challenge, which is a large-scale object classification task, successfully. This work has inspired a lot of follow-up work on the developments and applications of deep learning methods. A CNN consists of multiple convolutional and fully connected layers, where each layer is followed by a non-linear activation function. These networks can be trained end-to-end by back-propagation. There are several variants in CNNs such as the R-CNN [4], SPPNet [5], and Faster-RCNN [6]. Since these networks can generate highly discriminant features, they outperform traditional object detection techniques by a large margin. The Faster-RCNN includes a Region Proposal Network (RPN) to find object proposals, and it can reach nearly real-time object detection.

University of Southern California, Los Angeles, CA 90089, USA

Corresponding author:
C.-C. Jay Kuo,
Email: cckuo@sipi.usc.edu

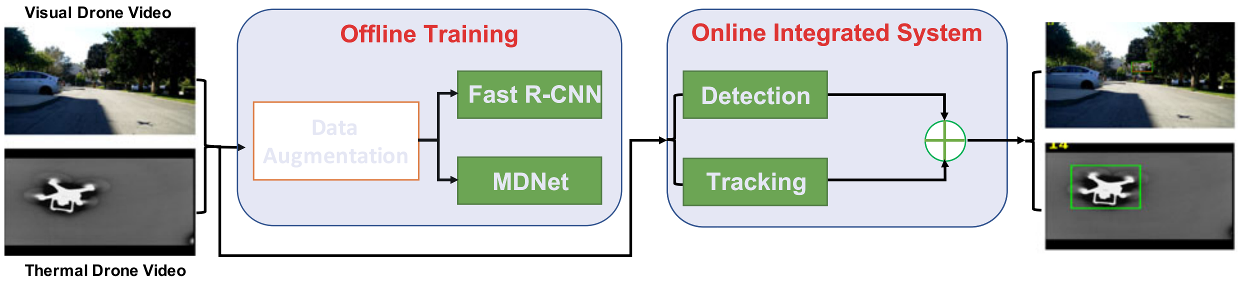


Fig. 1. Overview of proposed approach. We integrate the tracking module and detector module to set up an integrated system. The integrated system can monitor the drone during day and night with exploiting our proposed data augmentation techniques.

In particular, our proposed model integrates the detector module and tracker module to set up a drone monitoring system as illustrated in Fig. 1. The proposed system can monitor drones during both day and night. Due to the lack of the drone data and paucity of thermal drone diversities, we propose model-based augmentation for visible drone data augmentation and design a modified Cycle-GAN-based generation approach for thermal drone data augmentation. Furthermore, a residual tracker module is presented to deal with fast motion and occlusions. Finally, we demonstrate the effectiveness of the proposed integrated model on USC drone dataset and attain an AUC score of 43.8% on the test set.

The contributions of our work are summarized below.

- To the best of our knowledge, this is the first one to use the deep learning technology to solve the challenging drone detection and tracking problem.
- We propose to exploit a large number of synthetic drone images, which are generated by conventional image processing and 3D rendering algorithms, along with a few real 2D and 3D data to train the CNN.
- We develop an adversarial data augmentation technique, a modified Cycle-GAN-based generation approach, to create more thermal drone images to train the thermal drone detector.
- We propose to utilize the residue information from an image sequence to train and test an CNN-based object tracker. It allows us to track a small flying object in the cluttered environment.
- We present an integrated drone monitoring system that consists of a drone detector and a generic object tracker. The integrated system outperforms the detection-only and the tracking-only sub-systems.
- We have validated the proposed system on USC drone dataset.

The rest of this paper is organized as follows. Related work is reviewed in Section II. The collected drone datasets are introduced in Section III. The proposed drone detection and tracking system is described in Section IV. Experimental results are presented in Section V. Concluding remarks are given in Section VI.

II. RELATED WORK

A) Object detection

Current state-of-the-art CNN object detection approaches include two main streams: two-step and one-step object detection.

Two-step object detection approaches are based on R-CNN [4] framework, the first step generates candidate object bounding box and the second step classifies each candidate bounding box as foreground or background using a CNN. The R-CNN method [4] trains CNNs end-to-end to classify the proposal regions into object categories or background. SPPnet [5] develops spatial pyramid pooling on shared convolutional feature maps for efficient object detection and semantic segmentation. Inspired by SPPnet, Fast R-CNN [7] enables shared computation on the entire image and then the detector network evaluates the individual regions which dramatically improves the speed. Faster R-CNN [6] proposes a Region Proposal Network (RPN) to generate candidate bounding boxes followed by a second-step classifier which is the same as that of Fast R-CNN. The two-step framework consistently achieves top accuracy on the challenging COCO benchmark [8].

One-step object detection approaches predict bounding boxes and confidence scores for multiple categories directly without the proposal generation step in order to reduce the training and testing time. OverFeat [9], a deep multiscale and sliding window method, presents an integrated framework to implement classification, localization, and detection simultaneously by utilizing a single shared network. YOLO [10] exploits the whole topmost feature map to predict bounding boxes and class probabilities directly from full images in one evaluation. SSD [11] utilizes default boxes of different aspect ratios and scales on each feature map location. At prediction time, the classification scores are determined in each default box and the bounding box coordinates are adjusted to match the object shape. To handle objects of various sizes, multiple feature maps with different resolutions are combined to perform better predictions.

B) Object tracking

Object tracking is one of the fundamental problems in computer vision and CNN-based tracking algorithms have

https://doi.org/10.1017/ATSIP.2018.30 Published online by Cambridge University Press

developed very fast due to the development of deep learning. The trackers can be divided to three main streams: correlation filter-based trackers [12, 13], Siamese network-based trackers [14, 15], and detection-based trackers [16, 17].

Correlation filter-based tracker can detect objects very fast in the frequency domain. Recent techniques incorporate representations from CNN with discriminative correlation filters to improve the performance. BACF [12] designs a new correlation filter by learning from negative examples densely extracted from background region. ECO [13] proposes an efficient discriminative correlation filter for visual tracking by reducing the number of parameters and designing a compact generative model.

The Siamese network is composed of two-branch CNNs with tied parameters, and takes the image pairs as input and predict their similarities. Siamese network-based trackers learn a matching function offline on image pairs. In the online tracking step, the matching function is exploited to find the most similar object region compared with the object in the first frame. SiamFC [15] trains a fully convolutional Siamese network directly without online adaptation, and it achieves 86 fps with GPU but its tracking accuracy is not state-of-the-art. SINT [14] utilizes optical flow to deal with candidate sampling and it achieves higher tracking accuracy but lower speed. CFNet [18] interprets the correlation filter as a differentiable layer in a deep neural network to compute the similarity between the two input patches. The experimental results show comparable tracking accuracy at high frame rates.

Tracking-by-detection approaches train a classifier to distinguish positive image patches with negative image patches. MDNet [16] finetunes a classification network to learn class-agnostic representations appropriate for visual tracking task. It proposes a multi-domain learning framework to separate the domain-independent information from the domain-specific one. Although MDNet demonstrates state-of-the-art tracking accuracies on two benchmark datasets, the tracking speed is about 1 fps. RT-MDNet [17] utilizes improved RoIAlign [19] technique to improve the tracking speed by extracting representations from the feature map instead of the image. This approach achieves similar accuracy with MDNet with real-time tracking speed.

C) Generative Adversarial Network

Paucity of thermal drone training data forms a major bottleneck in training deep neural networks for drone monitoring. To address the problem, we propose an adversarial data generation approach to augment the existing thermal drone data.

Generative Adversarial Networks (GANs) simultaneously train two models: a generator and a discriminator. The generator tries to generate data from some distributions to maximize the probability of the discriminator making a mistake, while the discriminator distinguishes the sample comes from the training data rather than the generator. GANs have showed impressive results in a wide range of tasks, such as generating high-quality images [20], semi-supervised learning [21], image inpainting [22], video prediction and generation [23], and image translation [24]. Current image-to-image translation approaches have drawn more and more attentions due to the development of GANs. Pix2Pix [24] exploits a regression loss to guide the GAN to learn pairwise image-to-image translation. Due to the lack of the paired data, Cycle-GAN [25] utilizes a combination of adversarial and cycle-consistent losses to deal with unpaired data image-to-image translation. Taigman *et al.* [26] exploits cycle consistency in the feature map with the adversarial loss to transfer a sample in one domain to an analog sample in another domain. In our paper, a new unpaired image-to-image translation algorithm is proposed to augment the thermal drone images.

III. DATA COLLECTION AND AUGMENTATION

A) Data collection

The first step in developing the drone monitoring system is to collect drone flying images and videos for the purpose of training and testing. We collect two drone datasets as shown in Fig. 2. They are explained below.

- Public-domain drone dataset.
  It consists of 30 YouTube video sequences captured in an indoor or outdoor environment with different drone models. Some samples in this dataset are shown in Fig. 2(a). These video clips have a frame resolution of 1280 × 720 and their duration is about 1 minute. Some video clips contain more than one drone. Furthermore, some shoots are not continuous.
- USC drone dataset.
  It contains 30 visible video clips shot at the USC campus. All of them were shot with a single drone model. Several examples of the same drone in different appearance are shown in Fig. 2(b). To shoot these video clips, we consider a wide range of background scenes, shooting camera angles, different drone shapes, and weather conditions. They are designed to capture drone's attributes in the real world such as fast motion, extreme illumination, occlusion, etc. The duration of each video is approximately 1 minute and the frame resolution is 1920 × 1080. The frame rate is 30 frames per second.
- USC thermal drone dataset.
  It contains 10 thermal video clips shot at the USC campus and sample images are demonstrated in Fig. 7. All of them were shot with the same drone model as that for USC drone dataset. Each video clip is approximately 1 minute and the frame resolution is 1920 × 1080. The frame rate is 30 frames per second.

We annotate each drone sequence with a tight bounding box around the drone. The ground truth can be used in CNN training. It can also be used to check the CNN performance when we apply it to the testing data.

https://doi.org/10.1017/ATSIP.2018.30 Published online by Cambridge University Press

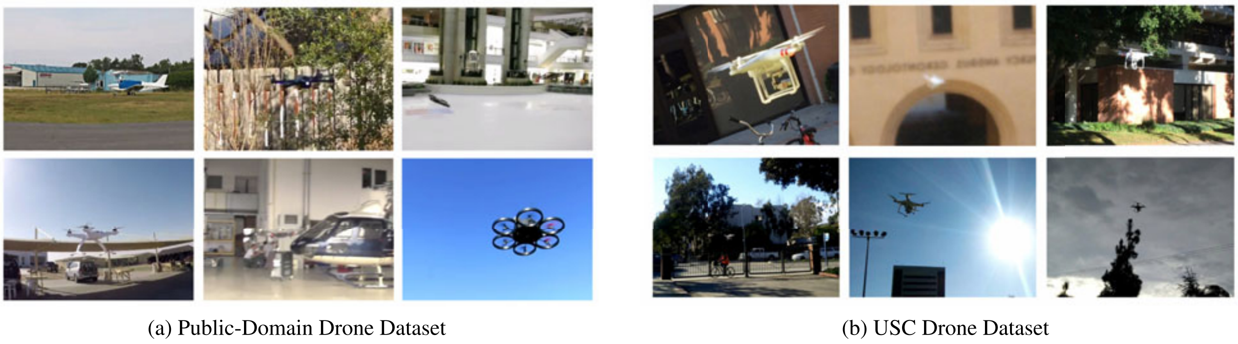


Fig. 2. Sampled frames from two collected drone datasets. (a) Public-Domain Drone Dataset, (b) USC Drone Dataset.

B) Data augmentation

The preparation of a wide variety of training data is one of the main challenges in the CNN-based solution. For the drone monitoring task, the number of static drone images is very limited and the labeling of drone locations is a labor-intensive job. The latter also suffers from human errors. All of these factors impose an additional barrier in developing a robust CNN-based drone monitoring system. To address this difficulty, we develop a model-based data augmentation technique that generates training images and annotates the drone location at each frame automatically.

The basic idea is to cut foreground drone images and paste them on top of background images as shown in Fig. 3. To accommodate the background complexity, we select related classes such as aircrafts and cars in the PASCAL VOC 2012 [27]. As to the diversity of drone models, we collect 2D drone images and 3D drone meshes of many drone models. For the 3D drone meshes, we can render their corresponding images by changing the view-distance, viewing-angle, and lighting conditions of the camera. As a result, we can generate many different drone images flexibly. Our goal is to generate a large number of augmented images to simulate the complexity of background images and foreground drone models in a real-world environment. Some examples of the augmented drone images of various appearances are shown in Fig. 5.

Specific drone augmentation techniques are described below.

- Geometric transformations
  We apply geometric transformations such as image translation, rotation, and scaling. We randomly set the width of the foreground drone in the range (0.1, 0.5) of the background image width, and keep the height–width ratio unaltered. We randomly select the angle of rotation from the range (−30°, 30°). Furthermore, we conduct uniform scaling on the original foreground drone images along the horizontal and the vertical directions. Finally, we randomly select the drone location in the background image.
- Illumination variation
  To simulate drones in the shadows, we generate regular shadow maps by using random lines and irregular shadow maps via Perlin noise [28]. In the extreme lighting environments, we observe that drones tend to be in monochrome (i.e. the gray-scale) so that we change drone images to gray level ones.
- Image quality
  This augmentation technique is used to simulate blurred drones caused by camera's motion and out-of-focus. We use some blur filters (e.g. the Gaussian filter, the motion Blur filter) to create the blur effects on foreground drone images.

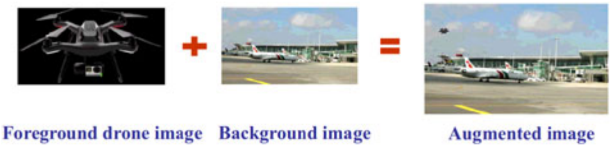


Fig. 3. Illustration of the data augmentation idea, where augmented training images can be generated by merging foreground drone images and background images.

We use the model-based augmentation technique to acquire more training images with the ground-truth labels and show several exemplary synthesized drone images in Fig. 5, where augmented drone models are shown in Fig. 4.

C) Thermal data augmentation

In real-life applications, our systems are required to work in both daytime and nighttime. To monitor the drones efficiently during the nighttime, we train our CNN-based thermal drone detector using infrared thermal images. It is more difficult to acquire enough training data for training the thermal drone detector. We can therefore apply data augmentation methods as mentioned in the previous section to generate thermal drone images with drone bounding box annotations. As illustrated in Fig. 3, we collect thermal images as the background from public thermal datasets and the Internet.

However, it is difficult to directly apply the visible data augmentation techniques since the thermal drone models are very limited and we cannot collect enough foreground thermal drone models with large diversity. This problem can be solved if we can successfully translate a visible drone image to a corresponding thermal drone image, where we face an unsupervised image-to-image translation problem. To address this issue, we provide two approaches for generating thermal foreground drone images. One is specifically

https://doi.org/10.1017/ATSIP.2018.30 Published online by Cambridge University Press

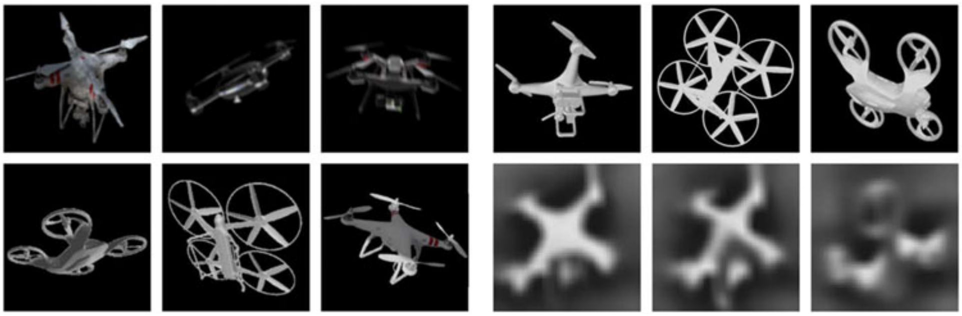


Fig. 4. Illustration of augmented visible and thermal drone models. The left three columns show the augmented visible drone models using different augmentation techniques. The right three columns show the augmented thermal drone models with the first row exploiting 3D rendering technique and the second row utilizing Generative Adversarial Networks.

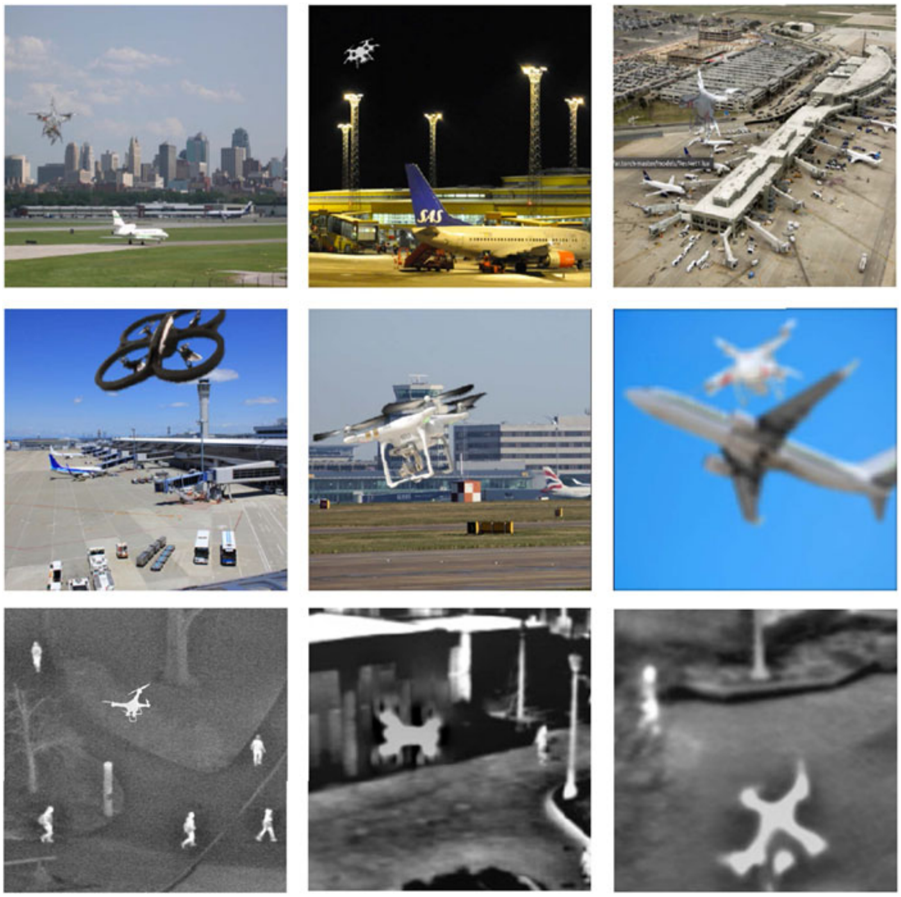


Fig. 5. Synthesized visible and thermal images by incorporating various illumination conditions, image qualities, and complex backgrounds.

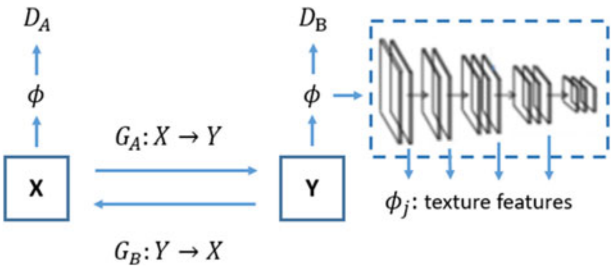


Fig. 6. The architecture of the proposed thermal drone generator.

targeting at translating thermal drone images using traditional image processing techniques, and the other is the proposed image translation approach using GANs.

- Traditional image processing techniques

  From observation of USC thermal drone dataset in Fig. 7, thermal drones have nearly uniform gray color in most cases. Therefore, a post-processing section is added to convert visible drones to monochrome drones.

https://doi.org/10.1017/ATSIP.2018.30 Published online by Cambridge University Press

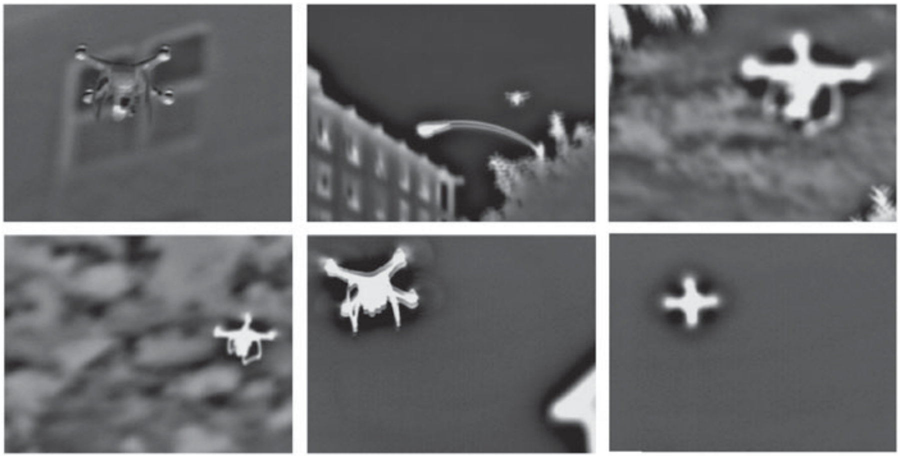


Fig. 7. Sampled frames from collected USC thermal drone dataset.

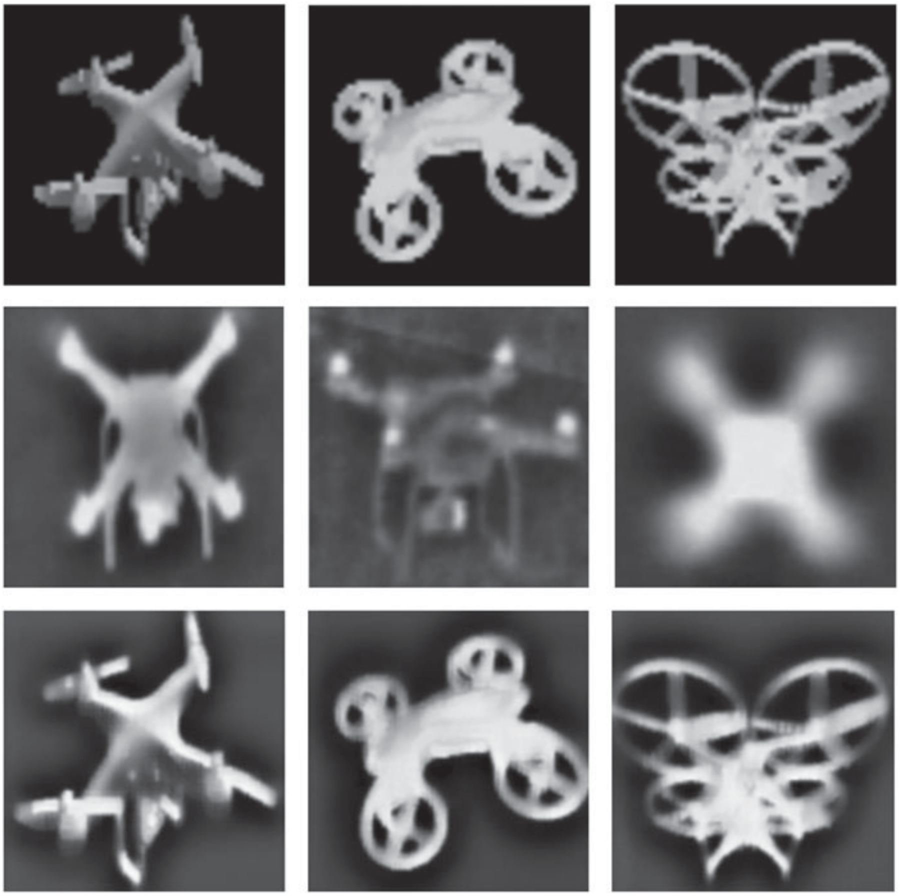


Fig. 8. Comparison of generated thermal drone images of different methods: 3D rendering (first row), Cycle-GAN (second row), proposed method (third row).

- Modified Cycle-GAN

  Our goal is to learn mapping functions between two domains X and Y given the unbalanced training samples. In our case, there are enough samples in visible domain X but very few samples in thermal domain Y, and applying the learned mapping function helps to generate a large diversity of samples in domain Y.

  Cycle-GAN provides a good baseline for unpaired image-to-image translation problem, however the training images in two domains are heavily imbalanced in our case. As demonstrated in the second row in Fig. 8, Cycle-GAN cannot increase the diversity of drone foreground images. Their proposed cycle consistency loss is, however, necessary to solve our problem. We utilize this cycle consistency loss to constrain the object shape consistency in two domains. The objective also contains the perceptual texture loss for only learning the texture translation between two domains, which helps to address the failures of Cycle-GAN.

As illustrated in Fig. 6, our methods learn two generators $G_A : X \to Y$ and $G_B : Y \to X$ with corresponding discriminator D_A and D_B. Image $x \in X$ is translated to domain Y as $G_A(x)$ and translated back as $G_B(G_A(x))$ which should be the reconstruction of x. Similarly, image $y \in Y$ is translated to domain X as $G_B(y)$ and translated back as $G_A(G_B(y))$. The cycle consistency loss is defined as the sum of the two reconstruction loss:

$$L_{cycle}(G_A, G_B, X, Y) = E_{x\sim P_x}[\|G_B(G_A(x)) - x\|] + E_{y\sim P_y}[\|G_A(G_B(y)) - y\|]. \quad (1)$$

We extract texture features of images as inputs of discriminators which aims to distinguish the texture styles between images x and translated images $G_B(y)$, images y and translated images $G_A(x)$. We exploit the texture features proposed by Gatys *et al.* [29], which is the gram matrix G of network feature map of layer j. The perceptual texture GAN loss is defined as:

$$L_{tex}(G_A, D_B, X, Y) = E_{y\sim P_y}[\log D_B(G(\phi_j(y)))] + E_{x\sim P_x}[\log(1 - D_B(G_A(G(\phi_j(x)))))]. \quad (2)$$

The full loss function is:

$$L_{loss}(G_A, D_A, G_B, D_B) = \lambda L_{cycle}(G_A, G_B, X, Y) + L_{tex}(G_A, D_B, X, Y) + L_{tex}(G_B, D_A, Y, X), \quad (3)$$

where λ controls the relative importance of the cycle consistency loss and perceptual texture GAN loss.

IV. DRONE MONITORING SYSTEM

To achieve the high performance, the system consists of two modules; namely, the drone detection module and the drone tracking module. Both of them are built with the deep learning technology. These two modules complement each other, and they are used jointly to provide the accurate drone locations for a given video input.

A) Drone detection

The goal of drone detection is to detect and localize the drone in static images. Our approach is built on the Faster-RCNN [6], which is one of the state-of-the-art object detection methods for real-time applications. The Faster-RCNN utilizes the deep convolutional networks to efficiently classify object proposals. To achieve real-time detection, the Faster-RCNN replaces the usage of external object proposals with the Region Proposal Networks (RPNs) that share convolutional feature maps with the detection network. The RPN is constructed on the top of convolutional layers. It consists of two convolutional layers, one encodes conv feature maps for each proposal to a lower dimensional vector and the other provides the classification scores and regressed bounds. The Faster-RCNN achieves nearly cost-free region proposals and it can be trained end-to-end by back-propagation. We use the Faster-RCNN to build the drone detector by training it with synthetic drone images generated by the proposed data augmentation technique for the daytime case and by the proposed thermal data augmentation method for the nighttime case as described in Section III.

B) Drone tracking

The drone tracker attempts to locate the drone in the next frame based on its location at the current frame. It searches around the neighborhood of the current drone's position. This helps track a drone in a certain region instead of the entire frame. To achieve this objective, we use the state-of-the-art object tracker called the Multi-Domain Network (MDNet) [16] as the backbone. Due to the fact that learning a unified representation across different video sequences is challenging and the same object class can be considered not only as a foreground but also background object. The MDNet is able to separate the domain-specific information from the domain-independent information in network training.

The network architecture includes three convolution layers, two general fully connected layers and a N-branch fully connected layer, where N is the number of training sequences. To distinguish the foreground and background object in the tracking procedure, each of the last branches includes a binary softmax classifier with cross-entropy loss. As compared with other CNN-based trackers, the MDNet has fewer layers, which lowers the complexity of an online testing procedure and has a more precise localization prediction. During online tracking, the N-branch fully connected layer is replaced by a single-branch layer. Besides, the weights in the first five layers are pretrained during the multi-domain learning, and random initialization is exploited to the weights in the new single-branch layer. The weights in the fully connected layer are updated during online tracking whereas the weights in the convolutional layers are frozen. Both the general and domain-specific features are preserved in this strategy and the tracking speed is improved as well.

To control the tracking procedure and weights update, long-term and short-term updates are conducted, respectively, based on length of the consistent positive examples intervals. Besides, hard negative example mining [30] is performed to reduce the positive/negative example ratio and improve the binary classification difficulty to make the network more discriminative. Finally, bounding box regression is exploited to adjust the accurate target location.

To improve the tracking performance furthermore, we propose a video pre-processing step. That is, we subtract the current frame from the previous frame and take the absolute values pixelwise to obtain the residual image of the current frame. Note that we do the same for the R,G,B three channels of a color image frame to get a color residual image. Three color image frames and their corresponding color residual images are shown in Fig. 9 for comparison. If there

https://doi.org/10.1017/ATSIP.2018.30 Published online by Cambridge University Press

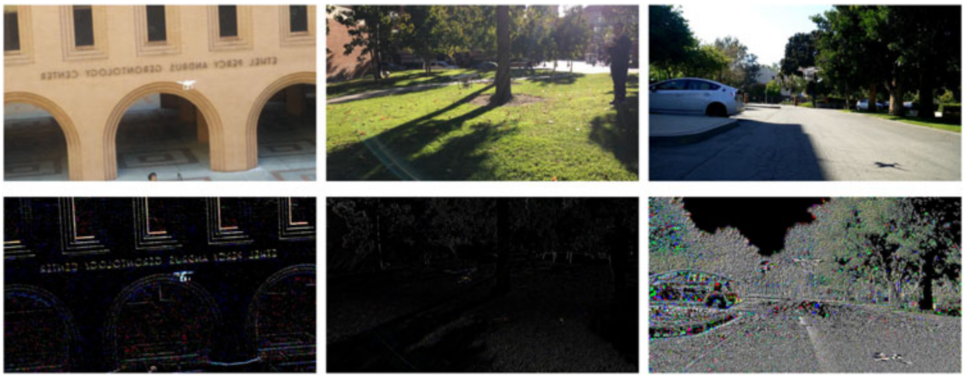


Fig. 9. Comparison of three raw input images (first row) and their corresponding residual images (second row).

is a panning movement of the camera, we need to compensate the global motion of the whole frame before the frame subtraction operation.

Since there exists strong correlation between two consecutive images, most background of raw images will cancel out and only the fast moving object will remain in residual images. This is especially true when the drone is at a distance from the camera and its size is relatively small. The observed movement can be well approximated by a rigid body motion. We feed the residual sequences to the MDNet for drone tracking after the above pre-processing step. It does help the MDNet to track the drone more accurately. Furthermore, if the tracker loses the drone for a short while, there is still a good probability for the tracker to pick up the drone in a faster rate. This is because the tracker does not get distracted by other static objects that may have their shape and color similar to a drone in residual images. Those objects do not appear in residual images.

C) Integrated detection and tracking system

There are limitations in detection-only or tracking-only modules. The detection-only module does not exploit the temporal information, leading to huge computational waste. The tracking-only module attempts to track the drone by leveraging the temporal relationships between video frames without knowing the object information, but it cannot initialize the drone tracker when failed to track for a certain time interval. To build a complete system, we need to integrate these two modules into one. The flow chart of the proposed drone monitoring system is shown in Fig. 10.

Generally speaking, the drone detector has two tasks – finding the drone and initializing the tracker. Typically, the drone tracker is used to track the detected drone after the initialization. However, the drone tracker can also play the role of a detector when an object is too far away to be robustly detected as a drone due to its small size. Then, we can use the tracker to track the object before detection based on the residual images as the input. Once the object is near, we can use the drone detector to confirm whether it is a drone or not.

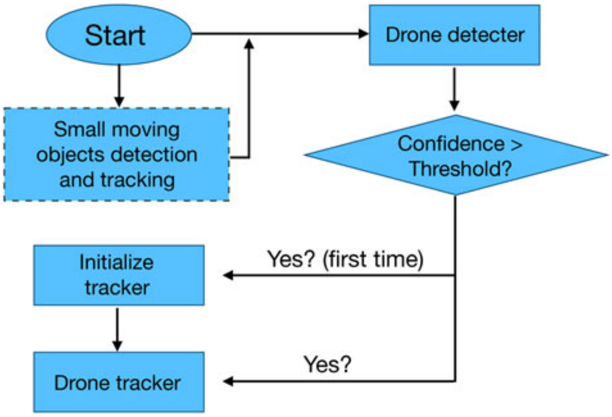


Fig. 10. A flow chart of the drone monitoring system.

An illegal drone can be detected once it is within the field of view and of a reasonable size. The detector will report the drone location to the tracker as the start position. Then, the tracker starts to work. During the tracking process, the detector keeps providing the confidence score of a drone at the tracked location as a reference to the tracker. The final updated location can be acquired by fusing the confidence scores of the tracking and the detection modules as follows.

For a candidate bounding box, we can compute the confidence scores of this location via

$$S'_d = 1/(1 + e^{-\beta_1(S_d-\alpha_1)}), \tag{4}$$

$$S'_t = 1/(1 + e^{-\beta_2(S_t-\alpha_2)}), \tag{5}$$

$$S' = max(S'_d, S'_t), \tag{6}$$

where S_d and S_t denote the confidence scores obtained by the detector and the tracker, respectively, S' is the confidence score of this candidate location and parameters $\beta_1, \beta_2, \alpha_1, \alpha_2$ are used to control the acceptance threshold.

We compute the confidence score of a couple of bounding box candidates, where S'_i denotes the confidence score and BB_i denotes the bounding box position, $i \in C$, where C denotes the set of candidate indices. Then, we select the one

https://doi.org/10.1017/ATSIP.2018.30 Published online by Cambridge University Press

with the highest score:

$$i^* = \underset{i\in C}{argmax}\, S'_i, \tag{7}$$

$$S_f = \underset{i\in C}{max}\, S'_i, \tag{8}$$

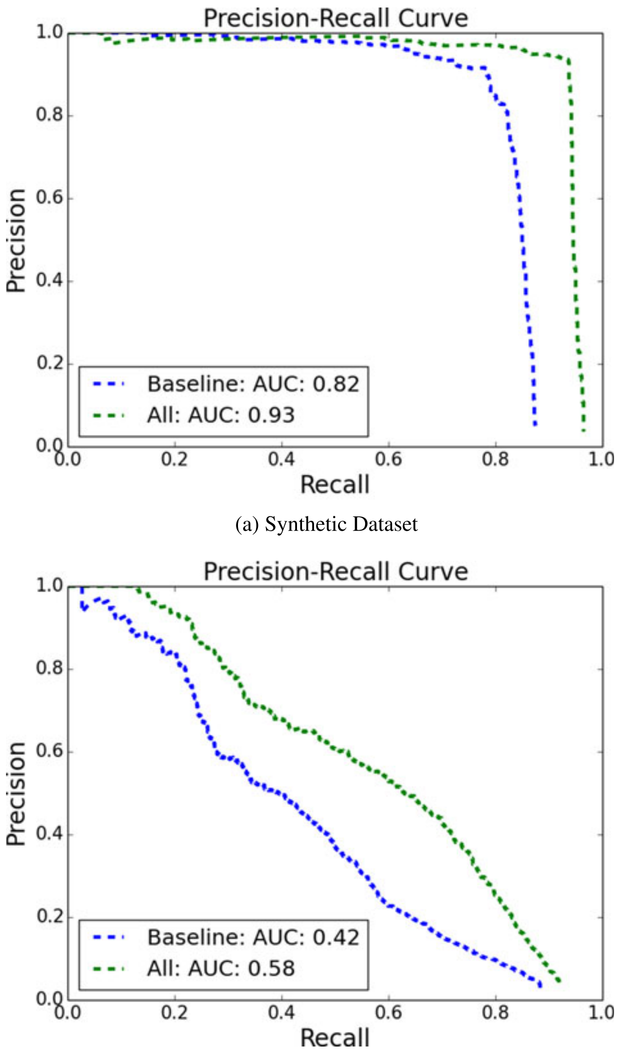
$$BB_{i^*} = BB_{\underset{i\in C}{argmax}\, S'_i} \tag{9}$$

where BB_{i^*} is the finally selected bounding box and S_f is its confidence score. If $S_f = 0$, the system will report a message of rejection.

V. EXPERIMENTAL RESULTS

A) Drone detection

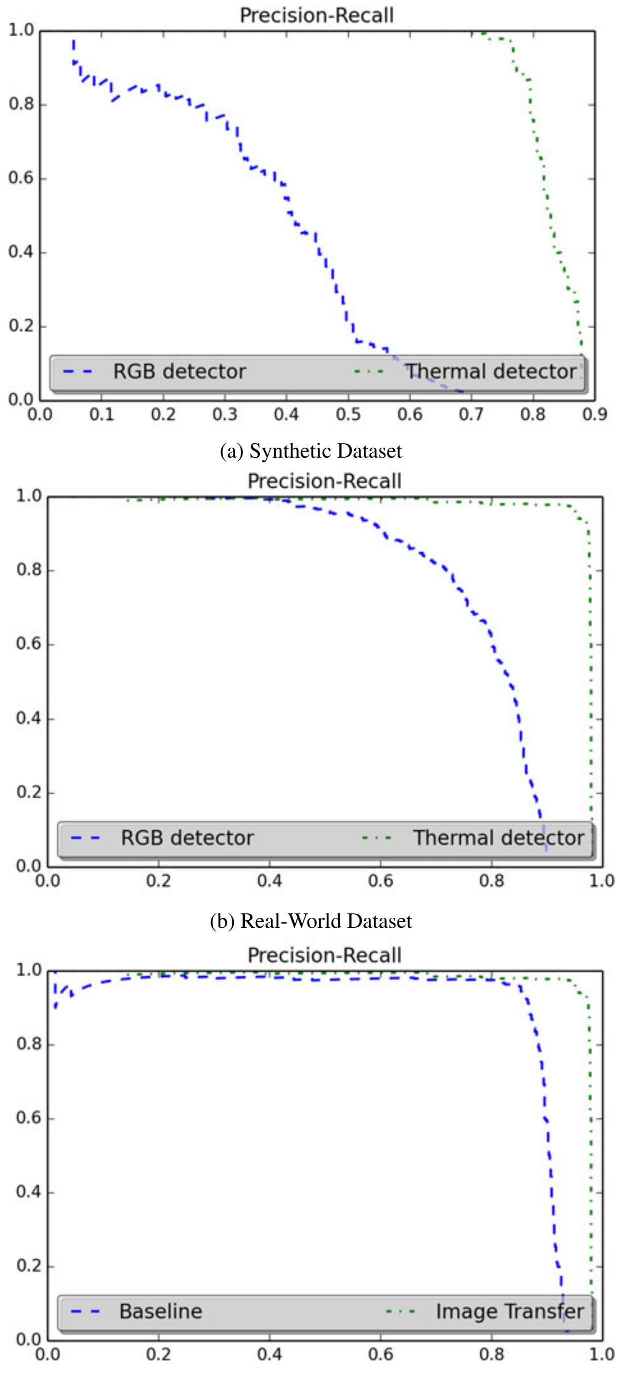
We test the visible and thermal drone detector on both the real-world and the synthetic visible or thermal datasets. Each of them contains 1000 images. The images in the real-world dataset are sampled from videos in the USC Drone dataset and the USC Thermal Drone dataset. The images in the synthetic dataset are generated using different foreground and background images with those in the training dataset. The detector can take any size of images



(a) Synthetic Dataset

(b) Real-World Dataset

Fig. 11. Comparison of the visible drone detection performance on (a) the synthetic and (b) the real-world datasets, where the baseline method refers to that using geometric transformations to generate training data only while the All method indicates that exploiting geometric transformations, illumination conditions and image quality simulation for data augmentation. (a) Synthetic dataset. (b) Real-world dataset.



(a) Synthetic Dataset

(b) Real-World Dataset

(c) Real-World Dataset

Fig. 12. Comparison of the visible and thermal drone detection performance on (a) the synthetic and (b) the real-world datasets; (c) shows the comparison of different thermal data augmentation techniques, where the baseline method refers to that using geometric transformations, illumination conditions, and image quality simulation for data augmentation, and image transfer refers to that utilizing the proposed modified Cycle-GAN data augmentation technique. (a) Synthetic dataset. (b) Real-world dataset. (c) Real-world dataset.

https://doi.org/10.1017/ATSIP.2018.30 Published online by Cambridge University Press

as the input. These images are then re-scaled such that their shorter side has 600 pixels [6]. To evaluate the drone detector, we compute the precision-recall curve. Precision is the fraction of the total number of detections that are true positive. Recall is the fraction of the total number of labeled samples in positive class that are true positive. The area under the precision-recall curve (AUC) [31] is also reported.

As for the visible detector, the effectiveness of the proposed data augmentation technique is illustrated in Fig. 11. In this figure, we compare the performance of the baseline method that uses simple geometric transformations only and that of the method that uses all mentioned data augmented techniques, including geometric transformations, illumination conditions, and image quality simulation. Clearly, better detection performance can be achieved by more augmented data. We see around 11 and 16% improvements in the AUC measure on the real-world and the synthetic datasets, respectively.

The experiment results of thermal drone detector are presented in Fig. 12. In both real-world and synthetic thermal datasets, the thermal detector achieves better performance than the visible detector by a large margin. We further compare the proposed modified Cycle-GAN data augmentation approach with the traditional data augmentation techniques to present the necessity of the proposed approach. The comparison between baseline methods which uses the 3D rendering techniques and image transfer methods which exploits the modified Cycle-GAN model for generating foreground images are demonstrated in Fig. 12. We observe there is 8% performance gain in the AUC measure on the real-world datasets.

B) Drone tracking

The MDNet is adopted as the object tracker. We take three video sequences from the USC drone dataset as testing ones. They cover several challenges, including scale variation, out-of-view, similar objects in background, and fast motion. Each video sequence has a duration of 30–40 seconds with 30 frames per second. Thus, each sequence contains 900–1200 frames. Since all video sequences in the USC drone dataset have relatively slow camera motion, we can also evaluate the advantages of feeding residual frames (instead of raw images) to the MDNet.

The performance of the tracker is measured with the area-under-the-curve (AUC) measure. We first measure the intersection over union (IoU) for all frames in all video sequences as

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}, \tag{10}$$

where the “Area of Overlap" is the common area covered by the predicted and the ground truth bounding boxes and the “Area of Union" is the union of the predicted and the ground truth bounding boxes. The IoU value is computed at each frame. If it is higher than a threshold, the success rate is set to 1; otherwise, 0. Thus, the success rate value is either 1 or 0 for a given frame. Once we have the success rate values for all frames in all video sequences for a

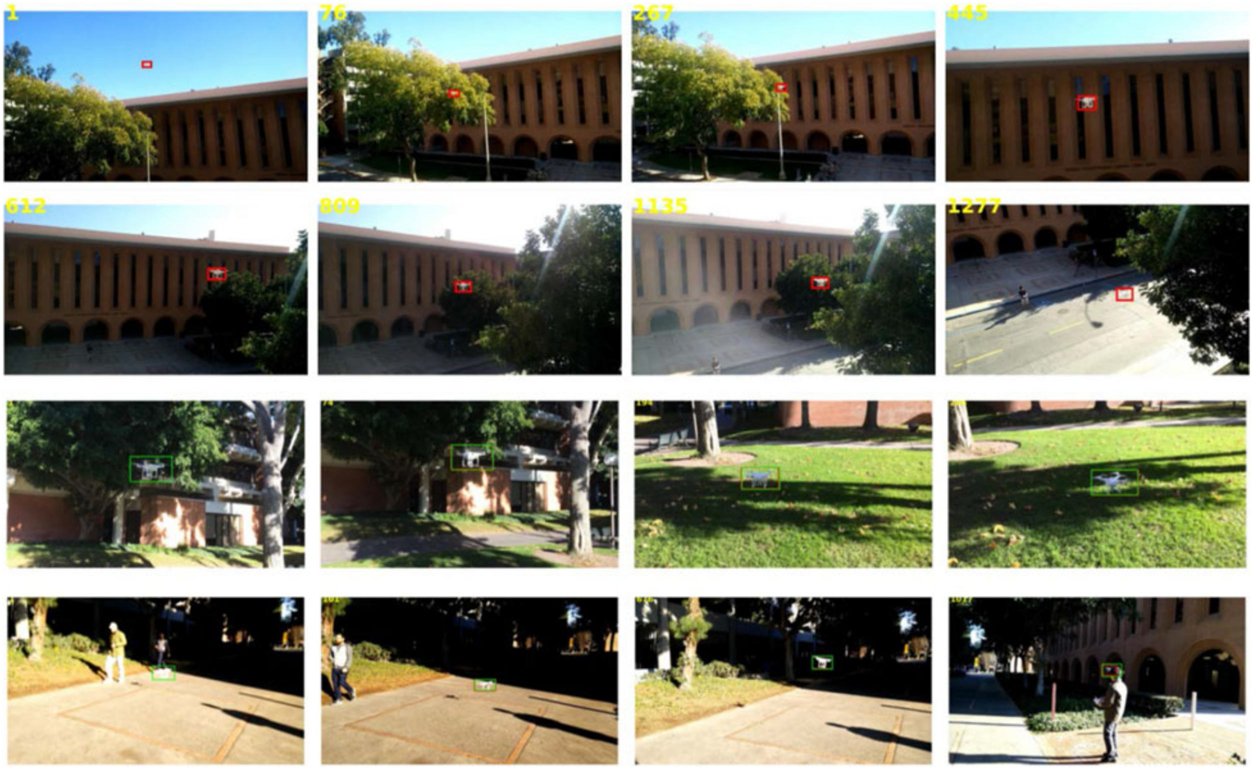


Fig. 13. Qualitative results on USC drone datasets. Our algorithm performs well on small object tracking and long sequence (first and second row), complex background (third row), and occlusion (forth row). The bounding boxes in red are integrated system results and the bounding boxes in green are tracking-only results.

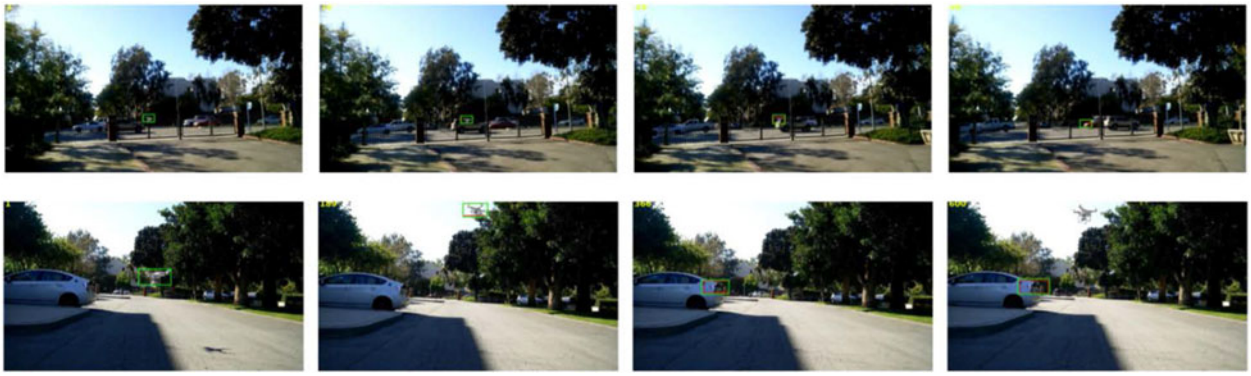


Fig. 14. Failure cases. Our algorithm fails to track the drone with strong motion blur and complex background (top row), and fails to re-identify the drone when it goes out-of-view and back (bottom row). The bounding boxes in red are integrated system results and the bounding boxes in green are tracking-only results.

particular threshold, we can divide the total success rate by the total frame number. Then, we can obtain a success rate curve as a function of the threshold. Finally, we measure the AUC which gives the desired performance measure.

We compare the success rate curves of the MDNet using the original images and the residual images in Fig. 15. As compared to the raw frames, the AUC value increases by around 10% using the residual frames as the input. It collaborates the intuition that removing background from frames helps the tracker identify the drones more accurately. Although residual frames help improve the performance of the tracker for certain conditions, it still fails to give good results in two scenarios: (1) movement with fast changing directions and (2) co-existence of many moving objects near the target drone. To overcome these challenges, we have the drone detector operating in parallel with the drone tracker to get more robust results.

Our algorithm performs well for most of the sequences on USC drone datasets as shown in Fig. 13. The qualitative results show that our algorithm performs well for tracking drones in distances and a long video (first and second rows). The third row shows that both tracking algorithm (in green) and integrated system (in red) works well when meeting with complex background. The forth row shows our approach performs well with motion blur and occlusions. Especially the integrated system predicts more accurate and tight bounding boxes in red compared with those of the tracking-only algorithm in green in the fourth column.

We present failure cases on USC drone datasets in Fig. 14. The first row shows both tracking algorithm and integrated system fail when dealing with strong motion blur, complex background in the fourth column. The second row shows the failure case when the drone is out-of-view for a long time where the drone is going out of view in the frame 189 and fully comes back in the frame 600. Both the two approaches cannot track the drone when the drone reappears in the video. Long-term memory should be applied in this case and optical flow could be exploited to re-identify the drone to initialize the tracker.

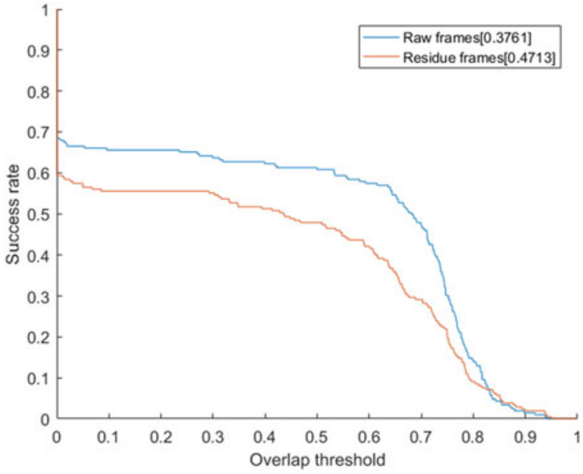


Fig. 15. Comparison of the MDNet tracking performance using the raw and the residual frames as the input.

C) Fully integrated system

The fully integrated system contains both the detection and the tracking modules. We use the USC drone dataset to evaluate the performance of the fully integrated system. The performance comparison (in terms of the AUC measure) of the fully integrated system, the conventional MDNet (the tracker-only module), and the Faster-RCNN (the detector-only module) is shown in Fig. 18. Note that the tracker-only module needs human annotation for the first frame to perform the drone tracking, while the integrated system is an autonomous system without relying on the first frame annotation. The fully integrated system outperforms the other benchmarking methods by substantial margins. This is because the fully integrated system can use detection as the means to re-initialize its tracking bounding box when it loses the object. It is worth pointing out that the overall detection performance is not good, moreover the ground truth label is more accurate than the detection results which leads to that tracking has better success rate when the overlap threshold is larger than 0.8.

https://doi.org/10.1017/ATSIP.2018.30 Published online by Cambridge University Press

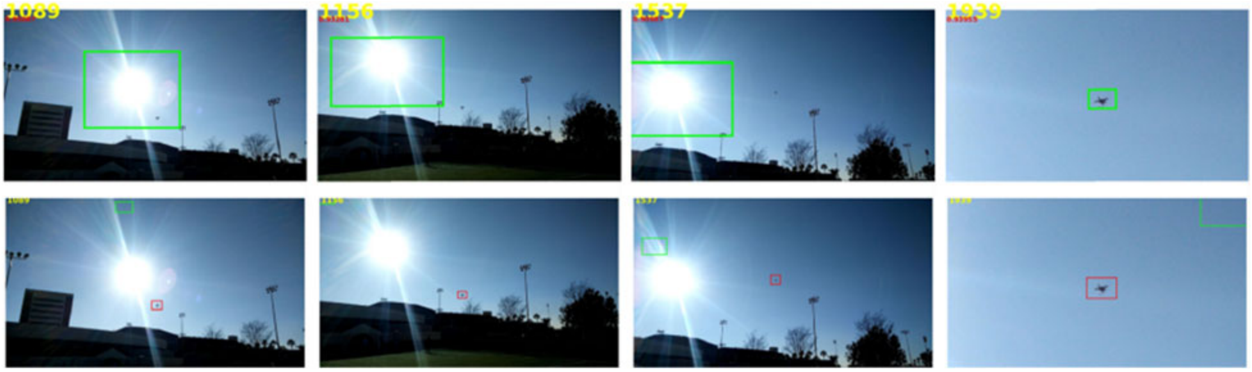


Fig. 16. Comparison of qualitative results of detection, tracking, and integrated system on the *drone_garden* sequence. The detection results are shown in the first row. The corresponding tracking and integrated system results are shown in the second row with tracking bounding boxes in green and integrated system bounding boxes in red, respectively.

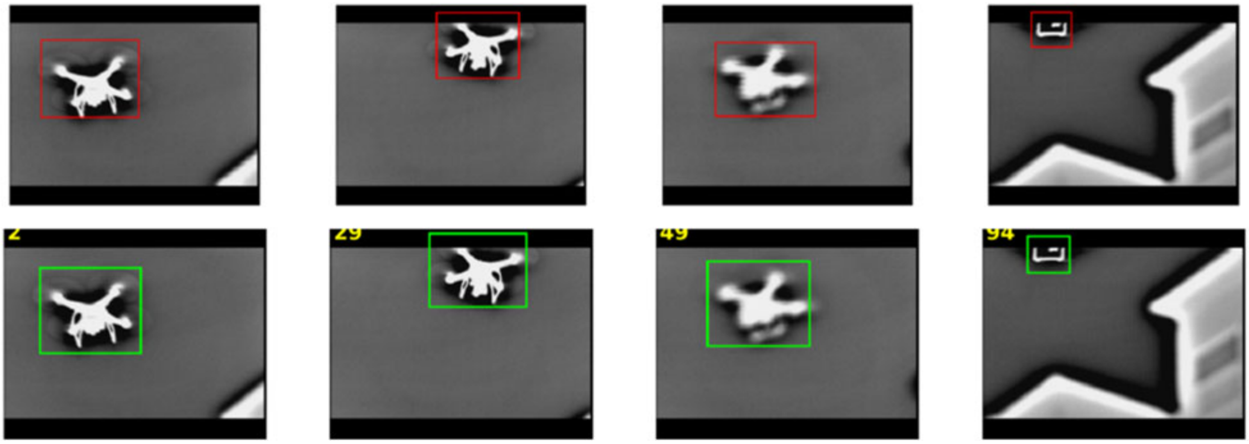


Fig. 17. Comparison of qualitative results of detection (first row), integrated system (second row) on the thermal drone sequence.

We show the comparison of qualitative results of detection, tracking, and integrated system in Fig. 16. The drone detection-only results in the first row demonstrate that detection-only performs bad when there are strong illumination changes. The drone tracker in green in the second row fails to track the drones back when it loses tracking at the very beginning. The proposed integrated system has higher tolerance against the illuminance and scale changes which are in the red bounding boxes in the second row. It outperforms the detection-only results and tracking-only results since it learns from both of them. Furthermore, we present the comparison of qualitative results of the thermal detection and integrated system in Fig. 17. Both the two approaches perform well to monitor the drone at night. Since the thermal drone detector has over 8% performance gain in the AUC measure over the visible drone detector and it almost achieves "perfect" precision-recall curve. We apply the object detection algorithm every other n frames, and re-initialize the tracker if the detection confidence score is higher than the threshold. Due to the high detection performance, the tracklet is updated nearly every n frames by the detection. Therefore, the integrated system has similar performance with detection results in this case.

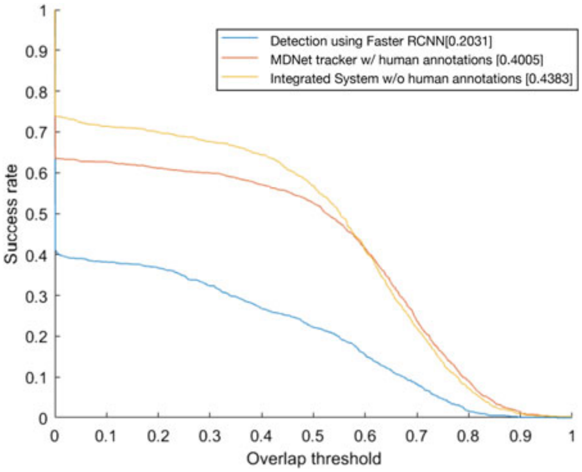


Fig. 18. Detection only (Faster RCNN) vs. tracking only (MDNet tracker) vs. our integrated system: The performance increases when we fuse the detection and tracking results.

VI. CONCLUSION

A video-based drone monitoring system was proposed in this work to detect and track drones during day and night. The system consists of the drone detection module and the drone tracking module. Both of them were designed based on deep learning networks. We developed a model-based data augmentation technique for visible drone monitoring to enrich the training data. Besides, we presented an adversarial data augmentation methodology to create more thermal drone images due to the lackage of the thermal drone data. We also exploited residue images as the input to the drone tracking module. The fully integrated monitoring system takes advantage of both modules to achieve high performance monitoring. Extensive experiments were conducted to demonstrate the superior performance of the proposed drone monitoring system.

ACKNOWLEDGMENT

This research is supported by a grant from the Pratt & Whitney Institute of Collaborative Engineering (PWICE).

REFERENCES

[1] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, **60** (2004), 91–110.

[2] Dalal, N.; Triggs, B.: Histograms of oriented gradients for human detection, in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conf. on* 2005, 886–893.

[3] Krizhevsky, A.; Sutskever, I.; Hinton, G.E.: Imagenet classification with deep convolutional neural networks, In Advances in neural information processing systems, 2012, 1097–1105.

[4] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2014, 580–587.

[5] He, K.; Zhang, X.; Ren, S.; Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition, in *European Conf. on Computer Vision*, 2014.

[6] Ren, S.; He, K.; Girshick, R.; Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural. Inf. Process. Syst.*, (2015), 91–99.

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

[8] Lin, T.-Y. *et al.*: Microsoft coco: common objects in context, in *European Conf. on Computer Vision*, 2014, 740–755.

[9] Sermanet, P.; Eigen, D.; Zhang, X.; Mathieu, M.; Fergus, R.; LeCun, Y.: Overfeat: integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[10] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A.: You only look once: unified, real-time object detection, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, 779–788.

[11] Liu, W. *et al.*: Ssd: single shot multibox detector, in *European Conf. on Computer Vision*, 2016, 21–37.

[12] Galoogahi, H.K.; Fagg, A.; Lucey, S.: Learning Background-Aware Correlation Filters for Visual Tracking, in *ICCV*, 2017, 1144–1152.

[13] Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M.: ECO: Efficient Convolution Operators for Tracking, in CVPR, 2017, 3.

[14] Tao, R.; Gavves, E.; Smeulders, A.W.: Siamese instance search for tracking, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, 1420–1429.

[15] Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.: Fully-convolutional siamese networks for object tracking, in *European Conf. on Computer Vision*, 2016, 850–865.

[16] Nam, H.; Han, B.: Learning multi-domain convolutional neural networks for visual tracking, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, 4293–4302.

[17] Jung, I.; Son, J.; Baek, M.; Han, B.: Real-Time MDNet. *arXiv preprint arXiv:1808.08834*, 2018.

[18] Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H.: End-to-end representation learning for correlation filter based tracking, in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conf. on*, 2017, 5000–5008.

[19] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.: Mask r-cnn, in *Computer Vision (ICCV), 2017 IEEE Int. Conf. on*, 2017, 2980–2988.

[20] Radford, A.; Metz, L.; Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[21] Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X.: Improved techniques for training gans, In Advances in Neural Information Processing Systems, 2016, 2234–2242.

[22] Yeh, R.; Chen, C.; Lim, T.Y.; Hasegawa-Johnson, M.; Do, M.N.: Semantic image inpainting with perceptual and contextual losses. arXiv preprint. *arXiv preprint arXiv:1607.07539*, 2016, 2.

[23] Mathieu, M.; Couprie, C.; LeCun, Y.: Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[24] Isola, P.; Zhu, J.-Y.; Zhou, T.; Efros, A.A.: Image-to-image translation with conditional adversarial networks. *arXiv preprint*, 2017.

[25] Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint*, 2017.

[26] Taigman, Y.; Polyak, A.; Wolf, L.: Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016.

[27] Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A.: The pascal visual object classes challenge: a retrospective. *Int. J. Comput. Vis.*, **111** (2015), 98–136.

[28] Perlin, K.: An image synthesizer. *ACM Siggraph Computer Graphics*, **19** (1985), 287–296.

[29] Gatys, L.A.; Ecker, A.S.; Bethge, M.: Image style transfer using convolutional neural networks, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, 2414–2423.

[30] Shrivastava, A.; Gupta, A.; Girshick, R.: Training region-based object detectors with online hard example mining, in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2016, 761–769.

[31] Huang, J.; Ling, C.X.: Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data. Eng.*, **17** (2005), 299–310.