A Novel Spatial and Temporal Context-Aware Approach for Drone-Based Video Object Detection

Zhaoliang Pi, Yanchao Lian, Xier Chen, Yinan Wu, Yingping Li, Licheng Jiao
School of Artificial Intelligence, Xidian University
Xi'an, Shaanxi Province, 710071, China
zlpi@stu.xidian.edu.cn lchjiao@mail.xidian.edu.cn

Abstract

Nowadays, with the advent of Unmanned Aerial Vehicles (UAV), drones equipped with cameras have been fast deployed to a wide range of applications. Consequently, automatic and effective object detection plays an important role in understanding and analysis of visual data collected from the drones, which could be further applied to civilian and military fields. However, various challenges still exist in object detection of drone-based videos, such as defocus, motion blur, occlusion and various variations (e.g., illumination, view and size), leaving too weak visual clues for successful detections. In this paper, we propose a novel approach for object detection in drone-based videos, which includes the multi-model fusion detection, an efficient tracker and a new evaluation method for confidence of the track, and the false positive analysis with scene-level context information and inferences. The experimental results on VisDrone2018-VID [44] dataset demonstrate the effectiveness of the proposed approach.

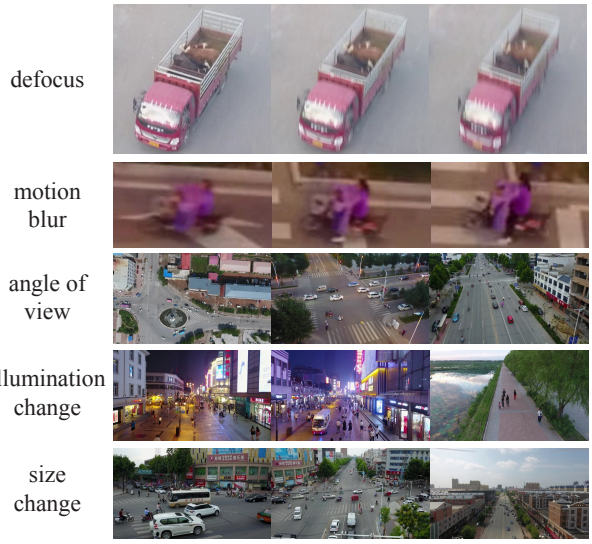

Figure 1. Examples of the challenging frames in videos of VisDrone2018-VID dataset.

1. Introduction

Drones equipped with cameras have been fast deployed to a wide range of applications, which includes agriculture, aerial photography, fast delivery, surveillance, etc. Consequently, automatic understanding of visual data collected from these platforms becomes highly demanding, involving recognizing the categories of objects in the scene, locating the objects and determining exact boundaries of each object, which brings computer vision to drones more and more closely [44]. The three corresponding research tasks in computer vision are image classification [35], object detection [34], and semantic segmentation [28]. Object detection is the most common task and has been attracting increasing number of attention.

Over the past few years, with the rapid development of deep learning, the convolutional neural network (CNN) has proven to be successful in detecting objects. Instead of designing handcrafted features, CNN architecture has a powerful ability of learning feature representations. Many CNN-based detection frameworks are proposed and achieve state-of-the-art results on PASCAL VOC [13] and COCO [26]. Generally, they can be divided into two classical technology solutions, which are region-based methods [15, 34, 16, 6] and single shot methods [30, 27, 12, 37]. However, directly utilizing these still image detectors on video objects remains a great challenge. The common frame degeneration problem usually appears in videos, which is more frequently in aerial videos taken by moving cameras (e.g., cameras equipped on drones). As shown in Figure 1, the challenging frames in drone-based videos (VisDrone2018-VID [44]) may suffer from defocus, motion blur, occlusion and various variations (e.g., illumination, view and size), leaving too weak visual clues to successful detections.

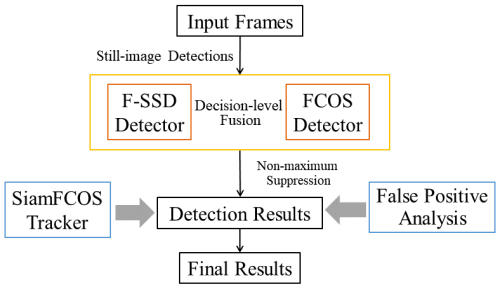

Figure 2. Architecture of our detection system. It is a multi-stage framework for video object detection task.

To tackle the challenges of object detection in drone-based videos, one of the straightforward solutions is to consider the spatial and temporal coherence in videos and utilize the information from adjacent frames. Consecutive video frames are highly similar, as well as their high-level convolutional features [47, 23, 42]. Deep Feature Flow [46] suggests to reuse the features of nearby frames to avoid redundant feature computation, which can be exploited to reduce time cost. This method involves a motion estimation to propagate feature, which needs to predict per-pixel motion by optical flow [18, 11, 5, 40, 19]. However, such pixel-level feature propagation approach would be inaccurate and sacrifice detection accuracy when the appearance of object dramatically changes, especially when the object is occluded, which occurs quite frequently in drone-based videos. On the other hand, several existing methods exploit temporal coherence on feature level by aggregating features of adjacent frames [45, 39], which could enhance the features of the low-quality frames in video but also need to predict motion paths by flow estimation.

Our philosophy is that better and more efficient using of temporal information is of great importance in drone-based video object detection. Instead of exploring better flow estimation methods, we introduce the strategy of tracking to assist detection. Current tracking-by-detection methods [1, 3, 29] for multi-pedestrian tracking indicate that temporal information could be utilized to regularize the detection results [20]. Therefore Kai [20] incorporates object tracking into detection framework for ImageNet [10] object detection in video (VID) task dataset. They track high-confidence detection proposals bidirectionally across the whole video clip, randomly perturb the boundaries of tubelet boxes and utilize CNN-based detectors to rescore all the candidate boxes. However, due to the large number of challenging frames in drone-based videos, the tracker may be easier to drift to background or other objects in a long tracking interval, and it is not efficient to rescore all the boxes by CNN-based detectors.

In this paper, we propose a novel approach for object detection in drone-based videos, which includes a deep CNN detection method, an efficient tracking process and the false positive analysis. The framework can be divided into three stages: 1) CNN-based detectors are trained and utilized to get the detection result of each frame with multi-model decision fusion; 2) We exploit the strategy of short-term tracking and a new evaluation method for the confidence of track to recall false negative objects; 3) False positive analysis is conducted to remove wrong alarms with scene-level context information and inferences.

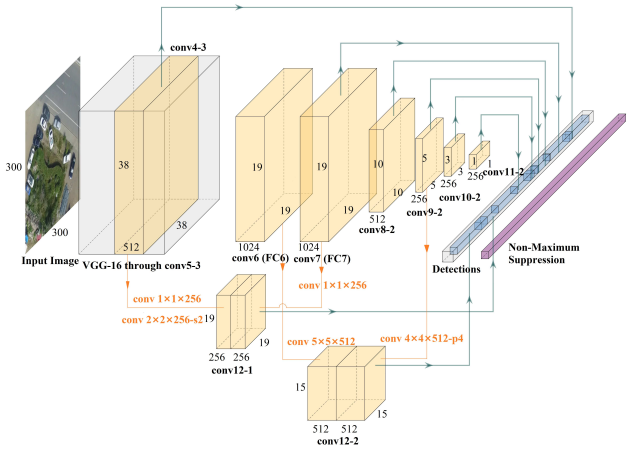

Figure 3. Architecture of F-SSD. The basic feature extractor of F-SSD is VGG-16 and it is constructed with two multi-scale feature fusion modules added to the original SSD. The two fusion layers are conv12_1 and conv12_2. F-SSD generates locations of bounding boxes and classifies objects from multiple feature maps in different layers densely and respectively. conv12_1 and conv12_2 are added to predict bounding boxes and object categories. Conv $1\times1\times256$ denotes the convolutional operation with the size of kernel 1×1, and the number of output channels is 256. p1 denotes $pad = 1$, s2 denotes $stride = 1$.

2. Related Works

2.1. Object Detection in Still Images

Existing state-of-the-art methods for still image object detection are mainly based on deep CNNs. They can be simply divided into two categories based on whether extra region proposal modules are required, i.e., two-stage and single-stage detectors. Two-stage object detectors have been the leading paradigm of object detection in recent years. The final detection result is generated by two stage: First, generate a large number of region proposals that likely contain objects of interest, and then classify the region proposals as well as refine the coordinates [15]. Because the region proposal generation with selective search [38] is time-consuming, Faster R-CNN [34] utilizes Region Proposal Network (RPN) instead and merges the proposal generation, classification and bounding box regression into an end-

to-end architecture by sharing convolutional features. However, the computation and run-time memory cost is relatively large for two-stage object detectors to generate region proposals. On the other hand, one-stage detectors regard object detection as a regression problem that directly predicts the locations and scores of bounding boxes in one evaluation, such as YOLO series models [31, 32, 33], Single Shot MultiBox Detector(SSD) [27] and RetinaNet [25]. YOLO [31] can easily make use of the spatial context information from the full image to reduce the false positives, but may not get an effective detection of small objects. YOLOv2 [32] involves the pre-defined anchors and achieves a higher recall than its precedent. SSD generates anchors densely from several different feature maps and thus has much better performance on object detection with multi scales. Most of the above-mentioned detectors are anchor-based object detectors, but the pre-defined sizes and aspect ratios of anchors may reduce the generalization ability of the model. FCOS [37], CenterNet [12] and CornerNet [21] are anchor-free detectors, of which FCOS makes full use of all points in the ground truth bounding box and suppresses the low-quality boxes by the proposed "center-ness" branch, which brings comparable recall with anchor-based detectors.

2.2. Object Detection in Videos

There have been video object detection methods that consider the spatial and temporal coherence in videos, and employ information from adjacent frames. Deep Feature Flow [46] considers to reuse the features from nearby frames to avoid redundant feature computation, which should involve motion estimation to propagate features and predict per-pixel motion by optical flow [18, 11, 5, 40, 19]. However, such pixel-level feature propagation approach would be inaccurate and time-consuming, especially when the appearance of object dramatically changes. This phenomenon is quite frequently in drone-based videos. On the other hand, several existing methods exploit temporal coherence on feature level by aggregating features from adjacent frames [45, 39], which could enhance the features of the low-quality frames in videos but also need to predict motion paths by flow estimation.

2.3. Object Tracking

As for visual object tracking, the mainstream models contain two types: 1) correlation filter based trackers [17, 8, 41, 9, 7], and 2) Siamese network based trackers [4, 2, 22]. For correlation filter based trackers, correlation operation is conducted to calculate the maximum response in the sub-region of current frame around object's location in the previous frame and get the updated location of the object. They are extremely fast, and always do well in natural videos. However, for drone-based videos, the task is still challenging because of the complexity and diversity of scenes. Due to the powerful representation of feature, the Siamese network based trackers have received increasing attentions for their well-balanced tracking accuracy and efficiency, which could deal better with rotations, occlusions, deformations and other appearance changes to avoid the drift.

3. Method

Our detection system for drone-based videos employs CNN-based detectors, tracking process and false positive analysis. As shown in Figure 2, the entire system is a multi-stage framework for object detection task. A detailed description is given in the following sections.

3.1. Still Image Object Detection

The CNN-based detectors of our proposed framework are derived from SSD [27] and FCOS [37] that predict bounding boxes and corresponding object categories of each frame with the multi-model decision fusion strategy, which is more robust compared to the single model that may generate much more false negative objects. As shown in Figure 2, we utilize FCOS model for its great performance in detecting small objects(e.g. pedestrian, person and motor), and a multi-scale feature fusion technique is applied to original SSD (just called F-SSD). F-SSD generates locations of bounding boxes and identifies the category of objects from multiple feature maps of different layers densely and respectively. In order to aggregate low-level features with more accurate details and high-level features with semantic information, we implement a feature fusion module that concatenates multi-scale feature maps, of which the specific details are described in Figure 3.

As shown in Figure 3, we add two multi-scale feature fusion modules to the original SSD and the basic feature extractor is VGG-16 [36]. In feature fusion module 1, we add convolution layer conv12_1_1 after layer conv4_3 and deconvolution layer conv12_1_2 after layer fc7, then the first fusion layer conv12_1 is generated by the concatenation of conv12_1_1 and conv12_1_2. In feature fusion module 2, convolution layers conv12_2_1 and deconvolution layers conv12_2_2 are added after layer fc6 and layer conv8_2 respectively, and the same concatenation method is utilized to construct the second fusion layer conv12_2 from conv12_2_1 and conv12_2_2. Subsequently, the two fusion layers are added to predict locations and categories, which improves the feature representation capacity of model to cover kinds of objects with different scales and shapes.

At the multi-model decision fusion stage, we conduct a decision fusion on the detection results obtained from F-SSD and FCOS, which could improve the accuracy compared with the single model in our experiments. The decision fusion ratio of the two models is 1:1. Afterwards, we

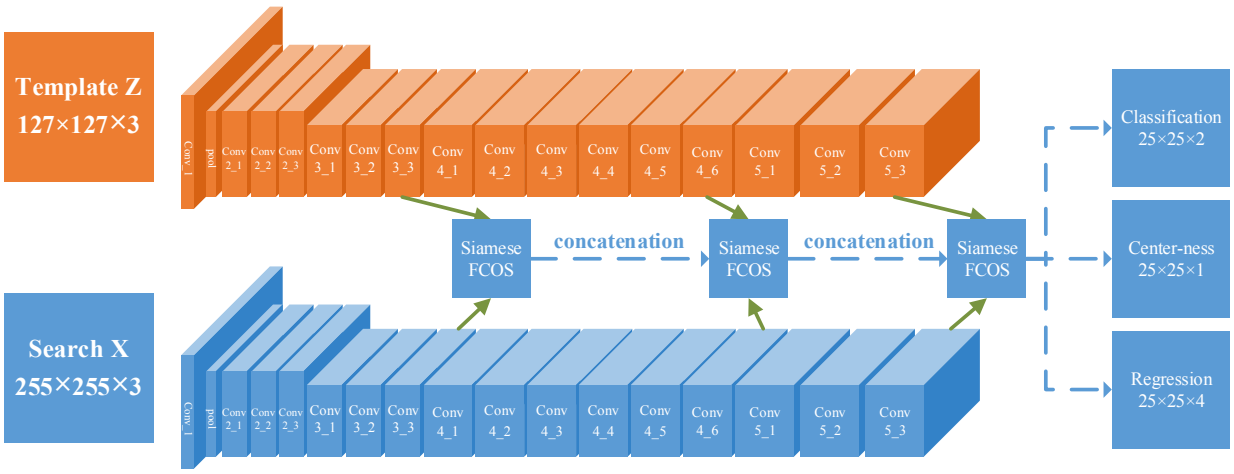

Figure 4. The architecture of SiamFCOS. It is an end-to-end fully convolutional network based on the structure of SiameseFC [4] and three Siamese FCOS modules. The input feature maps of the three modules are from layers conv3_3, conv4_6 and conv5_3 of the backbone network ResNet-53 respectively. Each Siamese FCOS module has three branches for different prediction tasks of regression, center-ness and classification. We concatenate the last layer's feature maps of all the regression branches from the three modules and involve in one 1×1 convolutional operation to generate an new branch for regression, and we also get an new branch for center-ness and an new branch for classification in the same manner.

exploit non maximum suppression(NMS) to reduce the redundancy of predicted boxes, and the thresholds of NMS for interclass and intraclass objects are different.

3.2. Object Tracking

We propose a one-stage fully convolutional network for tracking based on the structure of SiamFC [4], called SiamFCOS. Figure 4 shows its overall structure and the backbone network is ResNet-53, which is the same as SiamRPN++ [22]. Both shallow and deep features of the network are considered of equal importance, so the backbone network has three multi-level feature outputs in the layers conv3_3, conv4_6 and conv5_3 respectively. Each of the three layers is utilized subsequently as the input for the Siamese FCOS module. As shown in Figure 4, we replace the original anchor-based regression branch in SiamRPN++ with an anchor-free regression branch, and regress the distances from each location to the four sides of bounding box. Moreover, a "Center-ness" sub-branch [37] is added to infer the center of object. In [37], the Center-ness branch is parallel to the classification branch and used to suppress low-quality boundary boxes. We hope that the predicted value corresponding to the pixels near the center point of object will approach to 1 and the pixels far from the center point of object will approach to 0 in the Center-ness branch. Therefore, the predicted value of this branch will be (1).

$$centerness = \sqrt[2]{\frac{min(L,R)}{max(L,R)} * \frac{min(T,B)}{max(T,B)}} \qquad (1)$$

where L, T, R, and B represent the distances of the corresponding pixel in the input image from the left, upper, right, and lower boundary of the ground truth bounding box respectively.

In Figure 4, the object template Z is a small rectangular image block that contains the object and the search area X is an almost two times larger image block in current frame to find the designated object in. After the object template Z and the search area X get three levels of feature maps through the backbone network ResNet-53, the feature maps are separately utilized as the input of three Siamese FCOS modules. In one of the three modules, *F(z)* and *F(s)* denote the corresponding level of feature maps from Z and X. The structure of Siamese FCOS module is shown in Figure 5, of which the left describes all the components of the module and the connection relationship between them, such as Adj_1, Corr_1, Box head and so on. These components are described in detail in the right of Figure 5 with the corresponding color. The correlation operation is the same as that in SiamRPN++ [22]. (*L, T, R, B*), S and C denote the output feature maps of the regression, classification and center branch of one Siamese FCOS module respectively. Then, we concatenate (*L, T, R, B*), S and C from the three modules and employ one 1×1 convolutional operation to generate

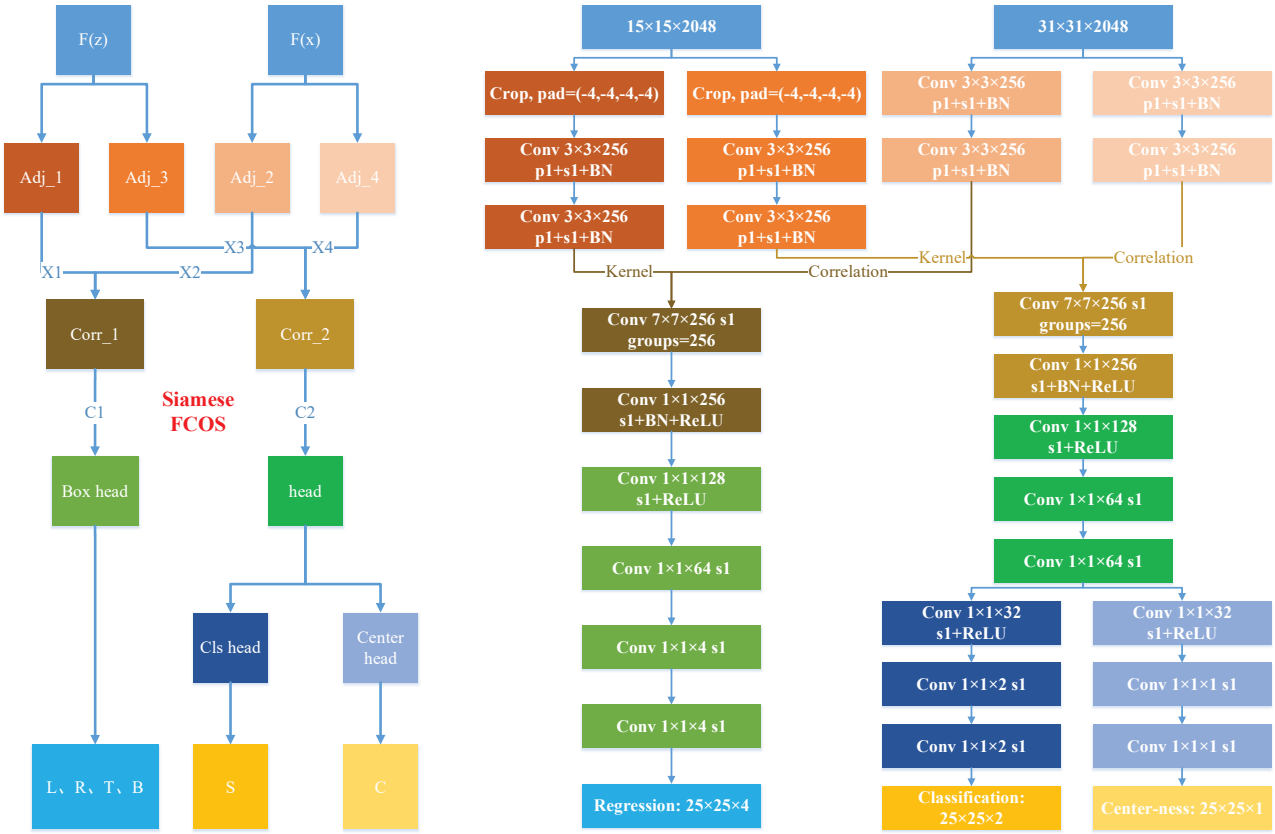

Figure 5. The structure of Siamese FCOS module. The left describes all the components of the module and the connection relationship between them. The right shows the specific details of the components with the corresponding color. Conv 3×3×256 p1 + s1 + BN denotes the convolutional operation with the size of kernel 3×3, and the number of output channels is 256. $pad = 1$, $stride = 1$. BN and ReLU denote the operation of Batch Normalization and ReLU motivation.

new output feature maps to predict the locations and categories of objects. The process of prediction is the same as FCOS [37].

In this paper, we utilize the strategy of short-term tracking, which is different from T-CNN [20]. The objects with detection confidence larger than a pre-defined threshold are chosen as the starting point for tracking, and the tracking process continues for K frames($K \leq 25$). We calculate the score S_t of trajectory to determine whether the tracking is valid by:

$$S_t = \frac{1}{2}(\lambda_1 I_{\mathrm{K}} + \lambda_2 I_{\mathrm{K\text{-}m}} + \lambda_3 I_{\mathrm{K\text{-}2m}}) \quad s.t. \lambda_1 + \lambda_2 + \lambda_3 = 1 \tag{2}$$

where I_{K}, $I_{\mathrm{K\text{-}m}}$ and $I_{\mathrm{K\text{-}2m}}$ denote the maximum value of intersection over union (IoU) between the Kth, *(K-m)*th, *(K-2m)*th bounding box of the tracking object along the trajectory and the detections of same class in the corresponding frame of video respectively. λ_1, λ_2, λ_3 are the weights. The equation (2) has a premise that the confidence of the Kth, (*K-m*)th, or (*K-2m*)th bounding box of the tracking object should be larger than 0.7. In this paper, if $S_t \geq 0.6$, the tracking trajectory is valid and the added bounding boxes will be fused with the detection result by NMS.

3.3. False Positive Analysis

Removing outliers by inferring video shooting direction. The camera on a drone generally takes images or videos of the objects of interest at a relatively long distance. Therefore, we can infer the shooting direction of video through observing the changes in size of the objects belonging to the same class in different regions of the frames. Based on the distribution of size along the shooting direction, we can eliminate the false detections with abnormal size. By comparing the sizes of boxes of each category in different areas of the same frame, it can be determined whether the camera's shooting direction is overlooked or not. In the former situation, there is no distinct difference

Figure 6. A part of object detection results with the proposed method. In one given frame, bounding boxes in different colors are used to mark different objects detected by the proposed method.

in the sizes of boxes of same category. However, in the latter situation, the sizes of boxes are smaller if they are far away from the camera and the objects with abnormal size in specific regions can be found.

Removing false detections by considering context information of video. Directly applying still-image object detectors to video will waste the context information. The statistics of detection result is useful for us to suppress false positive detections. Given a drone-based video, we count the number of objects of a certain category from the result of detection by still image object detectors . If the number of objects of the specified category is samller than 10 percent of the video frames, it indicates a very high probability of the objects of this category not existing in the scene of video (e.g. the truck or bus doesn't appear in pedestrian streets).

The objects in a video should be strongly correlated, which can help us to find out the false detections in the background. For each object in a frame, we calculate the number of its neighbors within a rectangular region of size 225×225 centered on it. If more than 80 percent of the objects have at least 3 neighbors, it means the objects in the frame are concentrated. Under this premise, the isolated object whose number of neighbors is less than 3 and detection score is lower than 0.5 can be regarded as a false detection.

4. Experiments

We propose an efficient approach for object detection in drone-based videos, which includes still image object detection, object tracking and false positive analysis. To demonstrate the performance of the proposed method, we empirically evaluate it on the publicly available dataset: VisDrone2018-VID [43]. The dataset is challenging since the objects are multi-category and multi-scale with complex backgrounds.

4.1. Dataset

VisDrone2018-VID [43] consists of 96 challenging video clips for the detection task, including 56 clips for training (24201 frames in total), 7 for validation (2819 frames in total) and 33 for testing (12968 frames in total). The videos in the three subsets are captured by various drone platforms at different cities in China, and share similar environments and attributes. The maximal resolution of video clips is 3840×2160 and ten object categories of interest are mainly defined, including pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle. The models in this paper are evaluated on the test set.

(a) (b)

Figure 7. The influence of tracking process. (a) shows examples of detection results before using the proposed tracking strategy. It is found that some people on the motor or tricycle are not detected due to their small size, as well as the buses in the distance. (b) shows examples of detection results after using the proposed tracking strategy and we can find that the missing objects are recalled through the tracking process.

4.2. Parameter Settings or Implementation Details

Data augmentation is performed to increase the number of annotated images with corresponding objects, which plays an important role in reducing over-fitting in the training process and improving the generalization ability of model. In our experiments, random flip horizontally or vertically and random rotation are adopted. For VisDrone2018-VID, we crop in the provided images of train set with the object as the center and balance positive samples among each class with data augmentation. The models were trained on Linux workstation with Intel Xeon E5-2630 v3 2.4 GHz CPU and two NVIDIA GeForce GTX 1080 GPUs.

F-SSD predicts locations and categories of objects directly using an end-to-end neural network. We choose the VGG-16 as a feature extractor, which was pretrained on ImageNet [10] classification task. The image input size is 300×300 and multiple feature maps from different layers are used to make predictions respectively. We utilize the stochastic gradient descent (SGD) with 0.9 momentum and weight decay of 0.0005. The learning rate starts from 10^{-3} and then decays by a factor of 5 at the iteration step of 50k, 80k and 110k. We set the maximum training iteration step as 120k and use mini-batch size of 25. Besides, we train the anchor-free model of FCOS with the input size of 1000×1000 and the batchsize of 28. The base model is ResNet-50 pretrained also on ImageNet classification task. The learning rate is initialized to 0.01 and decays by a factor of 10 at the iteration step of 60k and 80k. The maximum training iteration step for FCOS is 90k. When testing, at the multi-model decision fusion stage, we employ non maximum suppression (NMS) process to fuse the detections from F-SSD and FCOS. The NMS threshold is set as 0.6 for objects belonging to different categories and 0.4 for objects of same categories.

SiamFCOS is trained based on Resnet-53, which was pretrained on ImageNet datasets. The objects from detection with the confidence higher than 0.85 are chosen separately as the start tracking objects. The sizes of object template Z and search X are 127×127 and 255×255 respectively. Z is a small rectangular image block that contains the object and the search X is an almost two times larger image block in current frame to find the designated object in. You can find more detailed information in SiamFC[4] about how to select them as the input of a siamese network for tracking. SiamFCOS is finetuned on the augmented train dataset from VisDrone2018-VID. The initial learning rate is 0.01 and decreases exponentially with the step. The batchsize is 32 and the maximum training iteration step is 150k. When testing, the trajectory's length K is set as 20, and m is 5. λ_1, λ_2 and λ_3 are 0.5, 0.3 and 0.2 respectively.

5. Results

In this section, the proposed approach is compared with CenterNet [12], CornerNet [21], FPN [24], D&T [14], FGFA [45] and Faster-RCNN [34]. The mean Average Precision (mAP) is used to quantitatively evaluate the performance of the proposed method and comparison algorithms. Following the evaluation protocol in MS COCO [26], we use $AP^{IoU=0.50:0.05:0.95}$, $AP^{IoU=0.50}$, and $AP^{IoU=0.75}$ metrics to evaluate the results of detection algorithms [43].

Method	mAP(%)	AP(0.5)(%)	AP(0.75)(%)
CenterNet[12]	15.75	34.53	12.1
CornerNet[21]	16.49	35.79	12.89
FPN[24]	16.72	39.12	11.8
D&T[14]	17.04	35.37	14.11
FGFA[45]	18.33	39.71	**14.39**
Faster-RCNN[34]	14.46	31.8	11.2
ours	**18.73**	**44.38**	12.68

Table 1. The mean Average Precision of different methods on the VisDrone2018-VID test dataset. AP(0.5) denotes the mean Average Precision computed at the IoU threshold of 0.5.

Specifically, $AP^{IoU=0.50:0.05:0.95}$ is computed by averaging among all 10 intersection over union (IoU) thresholds (i.e., in the range of [0.50 : 0.95] with the uniform step size of 0.05) of all categories, which is used as the primary metric for ranking [43]. $AP^{IoU=0.50}$ or $AP^{IoU=0.75}$ is computed at the single IoU threshold of 0.50 or 0.75 over all categories respectively.

Our work is based on a multi-stage structure to detect objects with different sizes in drone-based videos. TABLE 1 shows the mAP of our method and comparison algorithms on the test set of VisDrone2018-VID and Figure 6 shows the object detection results in examples of frames using the proposed approach.

As observed in Table 1, in terms of mAP over all ten object categories, the proposed approach outperforms all the comparison algorithms. Figure 7 shows the influence of tracking process through the comparison of example frames, of which the left displays the results before using the proposed tracking strategy. It can be found that the people riding the motor or tricycle is recalled through the tracking process, as well as the bus in Figure 7(b), which indicates that the tracker is able to alleviate the problem of missed detection caused by motion blur, illumination change, and dimensional change. Figure 8 and 9 show the process of false positive analysis to eliminate the false detections by considering context information of video. As shown in Figure 8, the view of frame is overlooking, which can be inferred through comparing the size of cars. Therefore, the object in the red circle is a false positive detection of bus since its size is an outlier in the scene. In addition, the shooting direction of the frame in Figure 9 can be inferred through observing the changes in size of cars in different regions. The objects in red circle are far away from the concentrated region and have abnormal sizes. Subsequently, these false positive objects will be filtered out if their confidence are lower than the threshold.

6. Conclusion and Future Work

In this paper, we propose a novel approach for object detection in drone-based videos, which includes deep CNN detection, efficient tracking process and false positive analysis. The deep CNN detection exploits multi-model decision fusion strategy from F-SSD and FCOS. The efficient tracking process involves the tracker named SiamFCOS and an evaluation method for confidence of the track to recall false negative objects. At last, we utilize false positive analysis with scene-level context information and inferences to remove wrong alarms. The proposed framework presents a remarkable performance on the publicly available VisDrone2018-VID dataset. In future work, we will continue to improve the proposed approach for better detection performance. A better detector is needed in the first stage, based on which the performance of detection can be improved greatly through the tracking process and false positive analysis. Some small objects are lost due to the implementation of pooling. Therefore, we will consider designing a module to obtain richer information for accurate localization and classification of small objects.

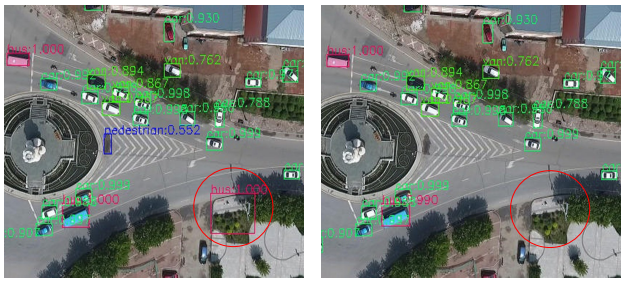

Figure 8. The influence of false positive analysis. The view of frame is overlooking and the false positve detection of bus in the red circle can be found based on whether its size is in a specific range.

Figure 9. The influence of false positive analysis. The shooting direction of frame can be inferred through observing the changes in size of cars in different regions. The false positive object in the red circle can be found and filtered out if the size of it is abnormal and the confidence is lower than the threshold.

References

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2008.

[2] W. W. Z. Z. B. Li, J. Yan and X. Hu. High performance visual tracking with siamese region proposal network. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2018.

[3] S. H. Bae and K. J. Yoon. Robust online multi-object tracking based on tracklet confidence and online discriminative appearance learning. In *Computer Vision & Pattern Recognition*, 2014.

[4] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr. *Fully-Convolutional Siamese Networks for Object Tracking*. 2016.

[5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *Eccv*, 3024(:10):25–36, 2004.

[6] Z. Cai and N. Vasconcelos. Cascade r-cnn: Delving into high quality object detection. 2017.

[7] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Eco: Efficient convolution operators for tracking. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2017.

[8] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg. Discriminative scale space tracking. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 39(8):1561–1575, 2017.

[9] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg. Beyond correlation filters: Learning continuous convolution operators for visual tracking. 2016.

[10] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li. Imagenet: A large-scale hierarchical image database. *Proc of IEEE Computer Vision & Pattern Recognition*, pages 248–255, 2009.

[11] A. Dosovitskiy, P. Fischery, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision*, 2015.

[12] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian. Centernet: Keypoint triplets for object detection. 2019.

[13] M. Everingham and J. Winn. The pascal visual object classes challenge 2007 (voc2007) development kit. *International Journal of Computer Vision*, 111(1):98–136, 2006.

[14] C. Feichtenhofer, A. Pinz, and A. Zisserman. Detect to track and track to detect. 2017.

[15] R. Girshick. Fast r-cnn. *Computer Science*, 2015.

[16] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, PP(99):1–1, 2017.

[17] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista. High-speed tracking with kernelized correlation filters. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 37(3):583–596, 2015.

[18] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1980.

[19] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2017.

[20] K. Kai, W. Ouyang, H. Li, and X. Wang. Object detection from video tubelets with convolutional neural networks. *IEEE Transactions on Circuits & Systems for Video Technology*, PP(99):1–1, 2016.

[21] H. Law and D. Jia. Cornernet: Detecting objects as paired keypoints. 2018.

[22] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. 2018.

[23] S. Lin, K. Jia, T. H. Chan, Y. Fang, W. Gang, and S. Yan. Dlsfa: Deeply-learned slow feature analysis for action recognition. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2014.

[24] T. Y. Lin, P. Dollár, R. Girshick, K. He, and S. Belongie. Feature pyramid networks for object detection. 2016.

[25] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, PP(99):2999–3007, 2017.

[26] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. 2014.

[27] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. 2015.

[28] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 39(4):640–651, 2014.

[29] H. Possegger, T. Mauthner, P. M. Roth, and H. Bischof. Occlusion geodesics for online multi-object tracking. In *Computer Vision & Pattern Recognition*, 2014.

[30] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. 2015.

[31] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Computer Vision & Pattern Recognition*, 2016.

[32] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *IEEE Conference on Computer Vision & Pattern Recognition*, 2017.

[33] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. 2018.

[34] S. Ren, R. Girshick, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 39(6):1137–1149, 2017.

[35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014.

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014.

[37] Z. Tian, C. Shen, H. Chen, and T. He. Fcos: Fully convolutional one-stage object detection. 2019.

[38] Uijlings, R. R. J., V. D. Sande, E. A. K., Gevers, Smeulders, and W. M. A. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, 2013.

[39] S. Wang, Y. Zhou, J. Yan, and Z. Deng. *Fully Motion-Aware Network for Video Object Detection*. 2018.

[40] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. Deepflow: Large displacement optical flow with deep matching. In *IEEE International Conference on Computer Vision*, 2014.

[41] L. Yang and J. Zhu. A scale adaptive kernel correlation filter tracker with feature integration. 2014.

[42] Z. Zhang and D. Tao. Slow feature analysis for human action recognition. *IEEE Trans Pattern Anal Mach Intell*, 34(3):436–450, 2012.

[43] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu. Vision meets drones: A challenge. *CoRR*, abs/1804.07437, 2018.

[44] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, and Q. H. et al. Visdrone-vdt2018: The vision meets drone video detection and tracking challenge results. In *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part V*, pages 496–518, 2018.

[45] X. Zhu, Y. Wang, J. Dai, Y. Lu, and Y. Wei. Flow-guided feature aggregation for video object detection. 2017.

[46] X. Zhu, Y. Xiong, J. Dai, Y. Lu, and Y. Wei. Deep feature flow for video recognition. In *Computer Vision & Pattern Recognition*, 2017.

[47] W. Y. Zou, S. Zhu, A. Y. Ng, and Y. Kai. Deep learning of invariant features via simulated fixations in video. In *International Conference on Neural Information Processing Systems*, 2012.