remote sensing

MDPI

Article

Object Tracking in Unmanned Aerial Vehicle Videos via Multifeature Discrimination and Instance-Aware Attention Network

Shiyu Zhang [1], Li Zhuo [1,2,*], Hui Zhang [1,2] and Jiafeng Li [1,2]

[1] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China; S201739017@emails.bjut.edu.cn (S.Z.); huizhang@bjut.edu.cn (H.Z.); lijiafeng@bjut.edu.cn (J.L.)

[2] Beijing Key Laboratory of Computational Intelligence and Intelligent System, Beijing University of Technology, Beijing 100124, China

* Correspondence: zhuoli@bjut.edu.cn

Received: 7 July 2020; Accepted: 13 August 2020; Published: 17 August 2020

Abstract: Visual object tracking in unmanned aerial vehicle (UAV) videos plays an important role in a variety of fields, such as traffic data collection, traffic monitoring, as well as film and television shooting. However, it is still challenging to track the target robustly in UAV vision task due to several factors such as appearance variation, background clutter, and severe occlusion. In this paper, we propose a novel two-stage UAV tracking framework, which includes a target detection stage based on multifeature discrimination and a bounding-box estimation stage based on the instance-aware attention network. In the target detection stage, we explore a feature representation scheme for a small target that integrates handcrafted features, low-level deep features, and high-level deep features. Then, the correlation filter is used to roughly predict target location. In the bounding-box estimation stage, an instance-aware intersection over union (IoU)-Net is integrated together with an instance-aware attention network to estimate the target size based on the bounding-box proposals generated in the target detection stage. Extensive experimental results on the UAV123 and UAVDT datasets show that our tracker, running at over 25 frames per second (FPS), has superior performance as compared with state-of-the-art UAV visual tracking approaches.

Keywords: visual tracking; two-stage framework; instance-aware network; unmanned aerial vehicle (UAV) videos

1. Introduction

In recent years, low-altitude aerial photography technology based on unmanned aerial vehicle (UAV) has been widely used as an effective supplement to aviation remote sensing and satellite remote sensing for traffic data collection. This technology can clearly capture ground targets through airborne high-resolution cameras, and the imaging resolution can reach the centimeter level. Correspondingly, the amount of traffic image data generated by low-altitude aerial photography of unmanned aerial vehicles has also increased dramatically. It is not only laborious but also inefficient to process a large amount of data manually. Therefore, the intelligent processing of UAV imagery has gradually become a research hotspot. Single object tracking of UAV images, as one of the technologies, provides the basis for subsequent tasks such as vehicle traffic assessment and road conflict prediction. Single object tracking involves manually setting the initial position of the target in the first frame, and continuously predicting the target's bounding box in the subsequent frame. Although great progress has been made in recent years, robust tracking is still a very challenging problem for scenes in UAV scenarios such as appearance variation, background clutter, and severe occlusion.

The existing UAV visual trackers are mostly migrated from the general framework of visual tracking, and the representative approaches present the following two main streams: tracks based on discriminative correlation filter (DCF) and tracking based on deep learning. In general, the former method models the tracking task as the ridge regression and utilizes the property of circulant matrices to obtain the closed-form solution [1]. In order to improve the computing speed and tracking performance of the algorithm, fast Fourier transform (FFT) is applied to DCF-based trackers to carry out complicated calculations in the frequency domain rather than in the spatial domain. In a study by [2], kernel functions were used to further optimize the algorithm. Due to the simplicity of the model and the fast-tracking speed of hundreds of frames per second (FPS), DCF-based trackers are widely used to perform object tracking in UAV videos. Unfortunately, the most obvious disadvantage of these algorithms is that they only use coarse multiscale inputs to simulate possible target size changes, therefore, it is difficult to obtain more robust performance in drone scenes where the imaging viewpoint changes dramatically and frequently.

With the rapid development of deep learning, visual tracking based on deep neural network (DNN) has proliferated and the performance has also been significantly improved as compared with the traditional approaches, showing its impressive development prospects. On the one hand, Siamese network-based algorithms [3–7] address visual object tracking as universal similarity learning between candidates and templates, and achieve state-of-the-art performance without model updating. On the other hand, along with the rapid development of object detection algorithms, in the last two years, visual trackers based on DNNs designed for object detection task have demonstrated significant performance improvement in terms of accuracy and robustness on several benchmarks [8–12]. At the same time, inspired by multitask learning, researchers have separated the tracking task into target classification and localization estimation; this innovative idea [13] has improved tracker performance to a new level. However, in UAV scenarios which face many huge challenges, such as relatively small object sizes, severe viewpoint changes, and so on, the above algorithms show varying degrees of performance degradation, and can even cause tracking drift. A well-designed and robust algorithm is still needed to handle object tracking tasks in UAV videos.

Inspired by intersection over union (IoU), IoU-Net [14], we propose a novel two-stage scheme for object tracking in UAV videos, including a DCF-based target detection stage and an IoU-Net-based bounding-box estimation stage. Specifically, in the target detection stage, considering that the size of the target in the drone image is relatively small, it is even as small as only one ten-thousandth of the entire field of view area in some scenarios, we propose a feature representation scheme for small targets that integrates low-level deep features, high-level deep features, and handcrafted features. The low-level deep features capture fine-grained information such as edge, shape, and color which are useful for target localization, while the high-level deep features contain more semantic abstraction and are robust to changes in target appearance.The handcrafted features complement the deep features by tackling the difficult challenge of extracting (tiny) small target features from a deep neural network. Then, DCF is employed to roughly predict the target location. In the following bounding-box estimation stage, first, we construct an instance-aware attention network for embedding the target template prior, it can provide more discriminative representation for the instance-aware IoU-Net network. Finally, we evaluate the target bounding-box proposals generated in the target detection stage through the instance-aware IoU-Net, and therefore accurately estimate the size of the target. We perform comprehensive experiments on two benchmarks, UAV123 [15] and UAVDT [16]. The experimental results demonstrate that our tracker achieves high area under curve (AUC) scores and runs at over 25 FPS on a general-purpose single-board GPU(graphics processing unit) computing platform.

The contributions of our work are three-fold as follows: (I) For UAV tracking tasks, a novel two-stage visual tracking framework is proposed. First, the target detection stage (TDS) is used to roughly locate the target, and then bounding-box estimation stage (BES) completes the accurate estimation of the target position. (II) An instance-aware deep network with a channel attention module is constructed for embedding the prior of target template, and it achieves a more discriminative target

representation in the clutter scenarios. (III) A feature representation scheme for small target is proposed which integrates handcrafted features, low-level deep features, and high-level deep features.

2. Related Work

UAV 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, one based on correlation filtering, and the other based on deep learning. Therefore, we only introduce the works most related to ours. For a more comprehensive and detailed review, please refer to [15–17].

2.1. Discriminative Correlation Filter (DCF)-Based Tracker

The pioneering work of applying correlation filtering in visual object tracking is the minimum output sum of squared error (MOSSE) [1] filter method. Until 2013, the circulant structure of tracking-by-detection with kernels (CSK) [2] and the kernelized correlation filters (KCF) [18] based on MOSSE [1] used the property of circulant matrices to deduce closed-form solution and utilized fast Fourier transform (FFT) to carry out complicated calculations in the frequency domain rather than in the spatial domain, which minimized computational cost and achieved high tracking speed, usually hundreds of frames per second. At the same time, the kernel function was applied to map the samples to a high-dimensional space, which improved the tracking performance. Discriminative scale space tracking (DSST) [19] is a novel scale adaptive tracker which is implemented by learning separate discriminative correlation filters for translation and scale estimation. The explicit scale filter is learned online using target appearance sampled at a set of different scales. Bertinetto et al. [20] designed a tracking method for the deformation and illumination challenges that often occur in the tracking process. By using color features to characterize the targets, their experimental results showed that better performance was obtained for dealing with the deformation and motion blur problems in the tracking process. However, color feature is very sensitive to illumination changes. At this time, the histogram of gradient (HOG) feature is preferable which can obtain better tracking results in illumination variation cases. Spatially regularized discriminative correlation filters (SRDCF) [21] can effectively suppress the boundary effects by adding spatial regularization but destroy the closed-form solution. Therefore, the iterative Gauss–Seidel method is used to obtain the optimal correlation filter parameters, which makes the speed very slow but significantly improves the robustness.

For the task of visual object tracking in UAV videos, several algorithms have been proposed based on correlation filtering. In [22], 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 [23], 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 UAV tracker TACF [24] with multilevel visual attention was proposed, improving the robustness to challenging visual factors such as partial occlusion and clutter background.

2.2. Deep Learning (DL)-Based Tracker

In recent years, Siamese networks have drawn great attention in the community of visual object tracking with balanced accuracy and speed. A Siamese network consists of two branches, which implicitly encodes the original patches into another space, and then operates them with a specific tensor to produce a response map. SiameseFC [6] uses AlexNet [25] as the backbone network for feature extraction and, first, introduces the correlation layer as fusion tensor to obtain good performance. CFNet [3] adds a correlation filter to the template branch based on SiameseFC [6], which makes the Siamese network shallower but more effective. However, both SiameseFC [6] and CFNet [3] need

multiscale input for the multiscale test, which makes it unable to better cope with the constant change of target scale in UAV videos. Currently, Siamese network-based trackers still have an accuracy gap compared with state-of-the-art algorithms, and they cannot benefit from deep networks, such as ResNet-50 [26] or deeper.

Recently, a new residual module, called a cropping-inside residual (CIR) unit [27], has been proposed to eliminate the negative effect of padding, and to further improve the SiamFC [6] using the modules with controlled receptive field size and network stride. The limitation of a shallow network can be relieved by using the CIR unit [27], and the performance of the trackers based on the Siamese network is greatly improved. In the last two years, because of the rapid development of the target detection algorithms, researchers in the field of visual object tracking have tried to combine a target detection algorithm with a target tracking algorithm. SiamRPN [11] formulates the tracking problem as a local one-shot detection, through combining a Siamese network and a region proposal network (RPN), which obtains the target position by classifying the foreground and background of each region proposals, as well as predicting bounding-box regression. SiamRPN++ [12] proposes a simple yet effective sampling strategy to break the spatial invariance restriction of Siamese tracker and successfully train a ResNet-driven Siamese tracker with significant performance gain. Accurate tracking by overlap maximization (ATOM) [8] decomposes the tracking task into two tasks of classification and estimation. For the estimation task, it modifies IoU-Net [14] to perform target-specific prediction to estimate the target bounding-box, which is suitable for tracking tasks. For the classification task, it consists of a two-layer fully convolutional network head, which is trained online using a conjugate gradient-based strategy to guarantee processing speed.

2.3. Combination of DL- and DCF-Based Tracker

In recent years, we have witnessed continuous development of deep learning in the field of visual object tracking. Several works have attempted to combine deep features with traditional features to obtain a superior tracking performance. The authors of [28] used the semantic information of high-level features and the texture information of low-level features to obtain a tracker with better performance, where the feature maps from conv5-4, conv4-4, and conv3-4 of VGG-19 [29] were exploited. These features from three layers were resized to a uniform resolution, and then used as samples for training three correlation filters, respectively. Finally, three response graphs were merged by the linear weighting with fixed weights of 1, 0.5, and 0.02 as the final response graph, and the maximum response point was the position of the target. For the DeepSRDCF tracker, [30] carried out experiments on the features of different layers of VGG-19 and proved that the features of low-level convolutional neural networks (CNN) had a better effect on solving the tracking problem, and excessive semantic information was not necessary for tracking. It was concluded that the convolutional feature of the first layer of VGG-19 was the best. Therefore, the DeepSRDCF tracker [30] replaced the traditional features used in SRDCF [21] with the deep features extracted from the ImageNet-VGG-2048 network [31] which used principal component analysis (PCA) to reduce the dimension of the features. C-COT [32] employs an implicit interpolation model to solve the problem of multiresolution features fusion and its formulation enables efficient integration of multiresolution deep feature maps in the continuous spatial domain. ECO [33] is further optimized based on C-COT [32]. The author considered that the number of deep feature channels was quite large as well as massively redundant. Therefore, a factorized convolution operator was proposed to greatly reduce the useless feature channels for object tracking. Meanwhile, ECO [33] also proposed a conservative model update strategy and a compact generation model to improve the robustness of the algorithm in complex scenes. In [34], an attentional correlation filter network (ACFN) was proposed, which adopted an attentional mechanism to adaptively select the best fitting subset from all available correlation filters.

3. Proposed Method

3.1. Two-Stage Tracking Structure

We decomposed the visual object tracking into two dedicated subtasks, i.e., target detection and bounding-box estimation, which was inspired by the framework adopted in seminal works [8,11,12]. The structure of the proposed method is shown in Figure 1, which contains two components, i.e., the target detection stage (TDS) and the bounding-box estimation stage (BES). TDS is used to locate the object based on discriminative correlation filter (DCF), and BES is used to estimate the accurate region of the target with an instance-aware IoU-Net. These two components comprise a two-stage object tracking framework for the typical challenges of background clutter, scale variation, and viewpoint changes in UAV videos. In the TDS, the fundamental function of the DCF tracker is to roughly locate the target position. In order to obtain a better performance and a more robust DCF tracker in the UAV scenarios, we used a variety of handcrafted features and multilevel deep features. Furthermore, considering that the size of the target in UAV videos was relatively small, we used the shallow network ResNet-18 [26] trained on ImageNet as the backbone for feature extraction, in order to avoid the loss of informative feature caused by pooling and large convolution steps in the deeper network. This stage is detailed in Section 3.2.

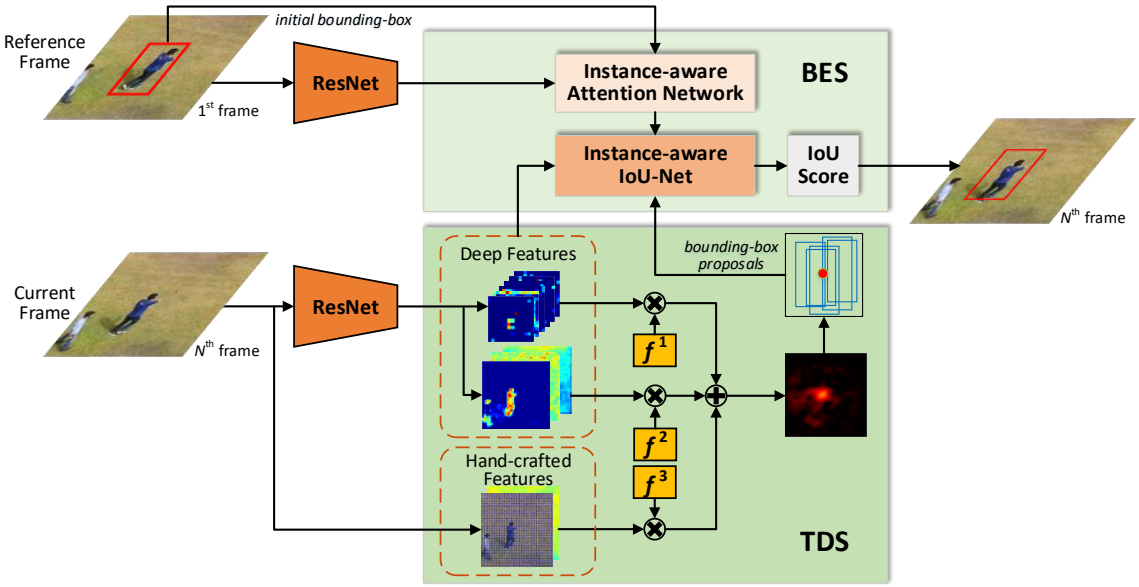


Figure 1. Overview of our proposed tracking architecture. The target detection stage (TDS) locates the target position in each frame, and the bounding-box estimation state (BES) takes the position, candidate regions, as well as the first frame with ground truth as input, and then outputs the final result.

The BES accurately estimates the target bounding-box based on the proposals given by the TDS. Visual object tracking in a scene usually focuses on specific instances of a given object rather than a class of objects. Therefore, we designed two sub-networks, one for modeling the instance-based prior knowledge of the given object, and the other for estimating the IoU scores against different proposals in the current frame, namely instance-aware attention network and instance-aware IoU-Net, respectively. The BES accepts the following four inputs: (1) deep features of the current frame, (2) the region proposals generated from the current frame, (3) deep features of the reference frame, and (4) the target bounding box in the reference frame. It outputs the predicted IoU scores of the region proposals in each frame. In the tracking process, the final bounding box was obtained by maximizing the score value with gradient ascent. We discuss this stage in detail, in Section 3.3.

3.2. Target Detection Stage

Considering that the correlation filter-based trackers reached a better balance between computational efficiency and robustness of target discrimination, we explored a target localization model for UAV video tracking scenarios. The purpose of the discriminative correlation filter inference process was to learn multiple filters from a set of labeled samples $(x, y) = \{(x_k, y_k)\}_{k=1}^{T}$. Each training sample x_k contained a c-dimensional feature map extracted from the region of interest (ROI) of the image. Assume that all samples have the same spatial size $W \times H$, then, at each spatial position $(w, h) \in \Omega := \{0, \ldots, W-1\} \times \{0, \ldots, H-1\}$, thus we have a c-dimensional feature vector $x_k(w, h) \in \mathbb{R}^C \cdot x_k^c$ is used to represent the c^{th} feature layer of x_k. The desired filter F contained a set of $W \times H$ convolution filters f^c learned by each feature layer. The response map $R(x_k)$ obtained by convolving the filter F with the sample x_k of size $W \times H$, which is given by:

$$R(x_k) = \sum_{c=1}^{C} x_k^c \otimes f^c \tag{1}$$

where $\otimes$ represents the circular convolution operation, C denotes the total number of features extracted from the sample x_k, and f^c represents the filter parameter of size $W \times H$ corresponding to the c^{th} feature layer. The filter parameters are obtained by minimizing the L2 error between the response map $R(x_k)$ on the training sample x_k and the label y_k, which can be formalized as:

$$\varepsilon_T(f) = \sum_{k=1}^{T} \alpha_k \|R(x_k) - y_k\|^2 + \lambda \sum_{c=1}^{C} \|f^c\|^2 \tag{2}$$

where the coefficient $a_k \geq 0$ determines the influence of each training sample, and $\lambda \geq 0$ is the weight of the regular term. The desired output y_k is a two-dimensional Gaussian distribution in the frequency domain, which includes the labels for each position in the sample x_k. Obviously, Equation (2) is the linear least square error (LSE) problem, therefore, we can use Parseval's theorem to convert it to the frequency domain solution to improve the speed of the algorithm.

Considering the complexity of the UAV scene and the challenge incurred by the small target, we used a variety of feature fusion methods to construct rich target representation. Specifically, in the implementation process, on the one hand, we adopted two handcrafted features, a color histogram score [20] that was insensitive to spatial structure and a HOG descriptor that was robust to the illumination variation. On the other hand, we also used the features of the first and third convolution layers in ResNet-18 [26] to make full use of representation capabilities of the deep features. The handcrafted features complemented deep features to tackle the difficult problem of extracting tiny small target features from a deep neural network. It is worth noting that when the target and the scene changed only slightly between adjacent frames, the abovementioned DCF-based tracker could accurately estimate the new position of the target. However, in real UAV videos, there are also huge changes in viewpoint, illumination, background, and target scale. Therefore, we only used the target localization results at this stage as guidance and further combined the bounding-box prediction scheme in the next stage to accurately estimate the position of the target.

3.3. Bounding-Box Estimation Stage

IoU-Net [14] is used to predict intersection over union (IoU) values between the region proposals and the target bounding box, which shows superiority in object detection tasks. We migrated it to perform visual object tracking in UAV videos by pruning category prediction branches and using a Siamese network architecture. Considering that the low-level features of CNN contain more texture information, and the high-level features contain more semantic information, we combined the features of first convolutional layer and high-level features of the ResNet-18 [26] network to avoid the loss of target information. As shown in Figure 2, in order to make the network focus on the target rather than

the background, we used the reference frame to generate an instance-aware attention value, which gave different weights to different feature channels. During the tracking process, instance-aware IoU-Net was guided by these weights to predict the IoU scores of the bounding-box proposals given by TDS. Then, the bounding-box proposals were refined by using gradient ascent maximizing IoU and the final bounding box was obtained by fusing the proposals whose IoU scores were the top three.

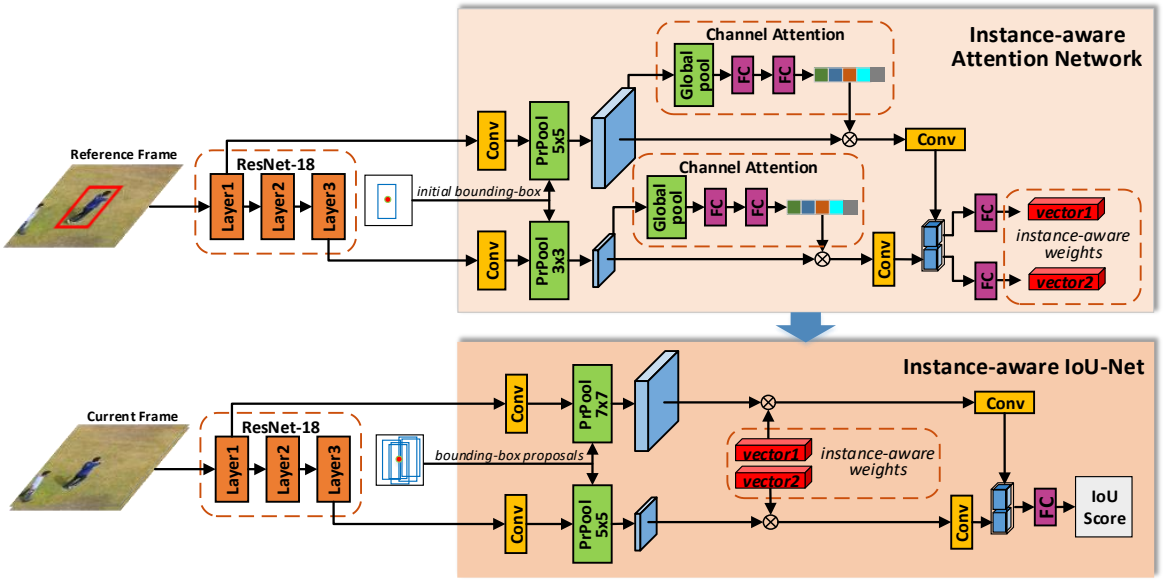


Figure 2. Schematic diagram of bounding-box estimation stage. The instance-aware attention network (top row) takes the features of first frame and initial bounding box as input, then, generates instance-aware vector. The instance-aware IoU-Net (bottom row) takes the features of subsequent frames and bounding-box proposals from the TDS as input, then, predicts the IoU score. ⊗ means channel-wise multiplication operation. All convolutional and fully connected (FC) layers are followed by BatchNorm and ReLU.

3.3.1. Instance-Aware Attention Network

As shown in Figure 2, the generation process of a instance-aware vector representing the prior information of an instance can be formalized as:

$$\omega(z, B_0) = \nu(\text{PrPool}(\beta(z), B_0) \cdot W_{chn}) \tag{3}$$

where $\beta(z)$ represents the convolutional features of a reference frame z; B_0 is the normalized initial bounding box in the reference frame; W_{chn} is the weights obtained by channel attention; $\text{PrPool}(\cdot)$ is precise ROI pooling operation [14]; and $\nu(\cdot)$ represents the convolution, concatenate, and fully-connected layer operations after feature channel attention is applied. Specifically, $\text{PrPool}(\cdot)$ can pool the features based on ROIs in each channel, it can be calculated by the average pooling of features β in any given region B, i.e.,

$$\text{PrPool}(\beta, B) = \frac{\int_{y_1}^{y_2}\int_{x_1}^{x_2} g(x, y)dxdy}{(x_2 - x_1) \times (y_2 - y_1)} \tag{4}$$

where (x_1, y_1) and (x_2, y_2) are the continuous coordinates of the top-left and bottom-right of the bounding box B, and $g(\cdot)$ is a continuous feature value after performing a bilinear interpolation operation on β.

The channel attention module assigns weights to the feature channels by calculating W_{chn} to suppress feature channels that are not related to the target. This channel attention module contains three operations. Firstly, each channel performs a global pooling (implemented by max pooling, in this paper) operation which makes the results contain a global receptive field [35] to characterize the global response

distribution characteristics of the feature channels. Then, considering the flexibility and simplicity of feature modeling, we used two cascaded fully connected layers, the first compressed the number of channels and the second restored the number of channels to the original value, to further learn a set of parameters to model the correlation between feature channels. Next, a convolution operation was performed on the weighted features, and the size of convolution kernel was consistent with the feature dimensions of each channel to obtain a feature quantity with a dimension of $1 \times 1 \times C_z$. We performed the above operations on the low-level and high-level convolutional features, respectively, and used the concatenate operation, as well as the fully connected layer to obtain an instance-aware vector.

3.3.2. Instance-Aware IoU-Net

As shown in Figure 2, this sub-network is used to evaluate the IoU score of i^{th} bounding-box proposal B_t^i in the current frame x_t, which can be formulated as follows:

$$s_{IoU}\left(x_t, B_t^i\right) = p\left(\omega \cdot \mathrm{PrPool}\left(\beta(x_t), B_t^i\right)\right) \tag{5}$$

where $\beta(\cdot)$ represents the convolutional features of sample x_t; ω contains the prior information of the instance, i.e., the instance-aware vector obtained by Equation (3) in the reference image; $p(\cdot)$ represents the operations of a subsequent convolutional layer and a fully connected layer. In particular, this sub-network is similar to the instance-aware attention network. The convolutional features of the current frame are first subjected to a convolution operation to adapt to the specific tracking task. Then, features are input to the PrPool layer along with the bounding-box proposals, so the PrPool layer operating in continuous space can subsequently maximize the IoU score to obtain more accurate tracking results. We used the concatenate operation on different convolutional features of the sample, and applied a fully connected layer to fuse the features and predict the IoU score. It is worth noting that, since the proposals given by the TDS may not completely include the target during the tracking process, in order to retain as much target information as possible, the hyperparameters of the PrPool layer in the instance-aware IoU-Net adopted larger values as compared with those of the instance-aware attention network.

3.4. Implementation Details

3.4.1. Offline Training

In order to train the network, we minimized the error between the prediction value S_{IoU} in Equation (5) and the ground-truth by given the image patch pairs. For the training dataset, we used the recently released large tracking datasets, namely TrackingNet [36] and VisDrone2018 [37]. In our proposed two-stage architecture, the backbone network used for deep feature extraction was the pretrained ResNet-18 [26] network on ImagNet. Because the TDS is updated online, only the parameters of the instance-aware attention network and instance-aware IoU-Net are optimized from offline training. For clarity and completeness of the BES descriptor, as shown in Tables 1 and 2, we briefly introduce the configuration of the deep neural network below. Our instance-aware attention network consists of two sets of symmetrical structures. The configuration of the instance-aware attention network and parameters are shown in Table 1. The convolution kernel size of all convolutional layers is set as 3×3, the two hyperparameters of PrPool layer are 5×5 and 3×3, respectively, and the rest of the fully connected layers are the parameters, as shown in Table 1.

Table 1. The configuration of instance-aware attention network and parameters.

Layers	Kernel Size	Stride	Output Size
Conv-1_1	3×3	1	$72 \times 72 \times 256$
PrPool-1	5×5	-	$5 \times 5 \times 256$
Conv-1_2	5×5	1	$1 \times 1 \times 256$
MaxPool-1	5×5	1	$1 \times 1 \times 16$
FC-1_1	256×16	-	1×16
FC-1_2	16×256	-	1×256
FC-1_3	512×256	-	1×256
Conv-2_1	3×3	1	$18 \times 18 \times 256$
PrPool-2	3×3	-	$3 \times 3 \times 256$
Conv-2_2	3×3	1	$1 \times 1 \times 256$
MaxPool-2	3×3	1	$1 \times 1 \times 16$
FC-2_1	256×16	-	1×16
FC-2_2	16×256	-	1×256
FC-2_3	512×256	-	1×256

Table 2. The configuration of instance-aware IoU-Net and parameters.

Layers	Kernel Size	Stride	Output Size
Conv-3_1	3×3	1	$72 \times 72 \times 256$
PrPool-3	7×7	-	$7 \times 7 \times 256$
Conv-3_2	7×7	1	$1 \times 1 \times 256$
Conv-4_1	3×3	1	$18 \times 18 \times 256$
PrPool-4	5×5	-	$5 \times 5 \times 256$
Conv-4_2	5×5	1	$1 \times 1 \times 256$
FC	512×1	-	1×1

For our instance-aware IoU-Net as shown in Table 2, it is mainly composed of four convolutional layers with a convolution kernel size of3×3, two PrPool layer with hyperparameters of 7×7 and 5×5, respectively, and a fully connected layer with parameter of 521×1. All convolutional layers and fully connected layer are followed by BatchNorm and ReLU. During the training process, a pair of image patches are sampled from different frames of a video, and the interval between the two frames is set below 40. Specifically, for obtaining instance-aware vector branch, we take the target as the center and sample a square image patch on the original image with 5^2 times the size of the target. Then, we adjusted the crop regions to a fixed size of 288×288. For the IoU score prediction branch, we used the same strategy to sample a patch, but added some perturbations to the position and scale to simulate the movement of the target. For each pair of image patches, Gaussian noise was added to the ground truth to generate 16 candidate images, and the minimum IoU between them and the ground truth is 0.1. Throughout the training process, we used image flipping and color dithering for data augmentation.

3.4.2. Online Tracking

The TDS first crops an 150×150 image patch as the target search area according to the target position at the previous frame, and then extracts the HOG features with 4 cells in this area, and the feature map size is $37 \times 37 \times 28$. Then, the TDS extracts the color histogram with 32 bins, and the size of the feature map is $32 \times 32 \times 32$. Secondly, the TDS samples a square patch which is based on the target position at the previous frame, and its area is about 5 times that of the target area, then, input to ResNet-18 [26] to obtain the deep features of the target. Next, the correlation filtering operation is performed on each feature map to obtain the response maps. Then, all response maps are linearly interpolated to the same resolution of 75×75 and fused to obtain the final position of the target. Here, the fusion method has not been studied more specifically, only linear weighting is used as the fusion manner. For the BES, the instance-aware vector is calculated by using Equation (3) in the first frame, whose dimension is $1 \times 1 \times 256$. For subsequent frames x_t, the target bounding-box proposals are

evaluated by the instance-aware IoU-Net. Note that all the network weights of the BES are fixed while performing online tracking. We employ five gradient ascent processes to maximize S_{IoU}, and finally obtain the tracking results by fusing the results of the top three proposals.

4. Experiments

4.1. Experimental Details and Setup

4.1.1. Datasets

We provide a comprehensive evaluation of our proposed tracker, which are performed on extensive challenging tracking datasets, including UAV123 [15] and UAVDT [16]. UAV123 [15] is an extremely large dataset which contains 123 aerial video sequences comprising more than 110K frames. These sequences contain 12 common attributes for visual tracking. UAVDT [16] is composed of about 80 K representative frames which are selected from 10 hours of raw videos, as well as fully annotated with bounding boxes for the following three fundamental computer vision tasks: object detection, single object tracking, and multiple object tracking. For single object tracking task, there are 50 video sequences and 8 attributes are annotated for each sequence.

4.1.2. Metrics

In this paper, we use a one-pass evaluation (OPE) process mentioned in [17] as the metric, which means to run trackers throughout a test sequence with initialization from the ground-truth position in the first frame and report the success rate plot or the precision plot. The success rate plot shows the ratios of number of successful frames to the total number of frames in the sequence when the thresholds varied from 0 to 1. The precision plot shows the percentage of frames whose center location error is within the given threshold. In the success rate plot, we use the area under curve (AUC) to rank all the tracking algorithms. In the precision plot, we use the precision value when the threshold of location error is set as 20 pixels to sort all the trackers.

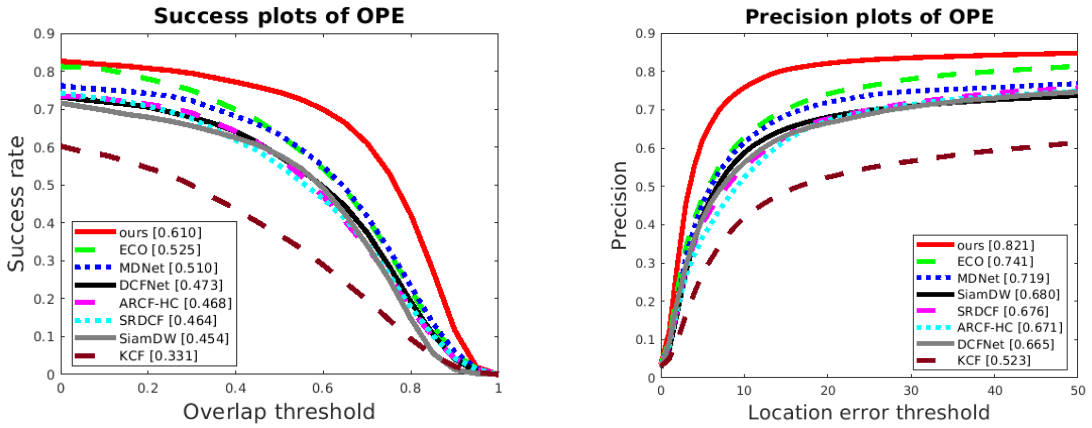
4.1.3. Experimental Details

Our proposed tracker uses the ResNet-18 [26] pretrained on ImageNet as the backbone network with the parameters all fixed while training. The parameters in our instance-aware IoU network architecture, are initialized using [14], which are obtained by minimizing the error of the prediction value in Equaiton (5) with ADAM [38]. We train for 40 epochs with 64 image pairs per mini-batch. While training the hyperparameters of ADAM optimizer, the initial learning rate is set as 0.02 and a factor 0.2 decay every 10 epochs. Our proposed method is implemented on PyTorch 0.4.1 platform, using Ubuntu 16.04 with an Intel Xeon(R) E5-2602 v4, 16G RAM, and Nvidia RTX 2080Ti.

4.2. Quantitive Evaluation

4.2.1. Evaluation on UAV123 Benchmark

Overall Evaluation: Figure 3 demonstrates the overall performance of our proposed method compared with several state-of-the-art trackers on UAV123 [15], including three DCF-based trackers with handcrafted features, i.e., KCF [18], SRDCF [21], and ARCF-HC [23]; three trackers based on deep learning, i.e., SiamDW [27], DCFNet [4], MDNet [39]; and ECO [33] which jointly uses DCF and deep learning. More specifically, on UAV123 [15], the performance of the proposed method is 8% and 10.2% higher than that of the second and third best tracker, i.e., ECO (74.1%) and MDNet (71.9%), respectively, in precision when threshold of center location error equals 20 pixels, as well as 8.5% and 10% higher than that of the second and third best tracker, i.e., ECO (52.5%) and MDNet (51.0%) respectively, in AUC.



(**a**) (**b**)

Figure 3. Comparison results of several state-of-the-art trackers on the UAV123 dataset [15]. The values close to the name of the methods represent the area under curve (AUC) or the precision when the threshold of location error is set as 20 pixels. (**a**) The success rate plot; (**b**) The precision plot.

Attribute-Based Evaluation: We also perform an attribute-based analysis for the proposed method on UAV123 [15]. For 123 videos in this dataset, each video sequence has 12 attributes including aspect ratio change (ARC), background clutter (BC), camera motion (CM), fast motion (FM), full occlusion (FOC), illumination variation (IV), low resolution (LR), out-of-view (OV), partial occlusion (POC), similar object (SOB), scale variation (SV), and viewpoint change (VC). As shown in Table 3, in scale variation cases, DCF-based trackers use multiscale inputs to solve the scale change problem, but these methods are not optimal because the scale change of objects is continuous. Therefore, we benefited from our proposed instance-aware IoU-Net, and obtained the AUC score, at least 15.4% higher than that of the DCF-based trackers, as well as 10.4% higher than that of deep learning based trackers. Meanwhile, in the video sequences with other attributes, the proposed tracker has also achieved a higher performance. It is worth noting that the performance of the proposed method for similar object (SOB) attribute is not significantly improved as compared with other attributes, which shows that our model is not more effective in distinguishing objects with similar appearance than targets and backgrounds.

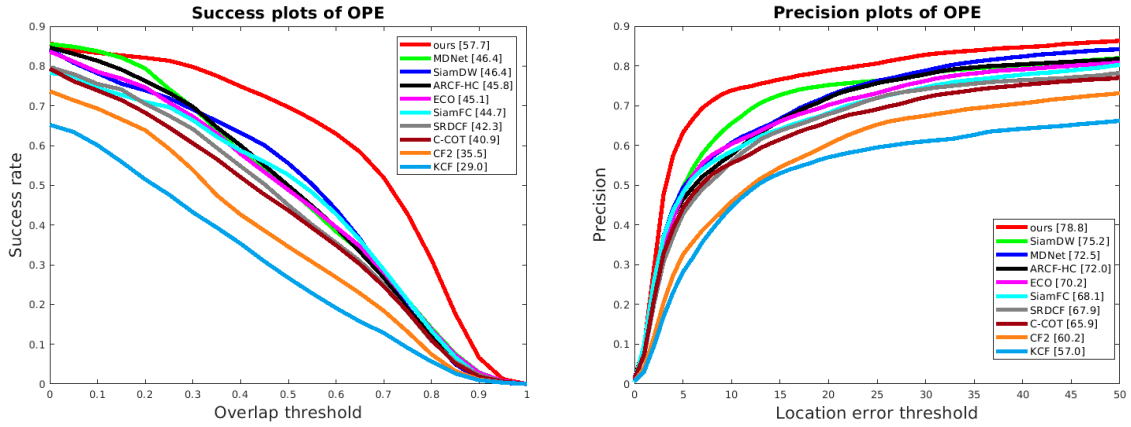
Table 3. Tracker performance analysis based on challenge attributes for the proposed method and the state-of-art methods on the UAV123 dataset [15]. Red, green, and blue fonts indicate the first, second, and third place.

Tracker	SV	VC	SOB	POC	OV	LR	IV	FOC	FM	CM	BC	ARC
ours	58.9	60.6	55.9	53.4	53.2	46.4	59.7	37.6	56.5	62.6	46.9	57.5
ECO [33]	49.6	47.3	51.8	45.6	42.5	39.6	45.9	30.8	41.5	50.6	38.8	44.5
MDNet [39]	48.5	47.0	48.5	43.9	43.5	38.5	40.1	33.3	33.8	50.9	42.9	45.1
DCFNet [4]	44.8	42.5	47.8	40.9	38.3	35.4	39.0	27.1	30.5	45.3	37.4	41.6
SiamDW [27]	42.9	42.1	44.0	37.3	42.0	30.4	34.7	27.3	34.6	45.4	29.3	39.7
ARCF-HC [23]	43.5	41.0	45.8	39.4	39.4	33.5	39.2	23.5	34.2	45.5	33.8	40.8
SRDCF [21]	43.5	41.0	45.2	39.6	38.3	31.0	39.5	26.2	34.0	43.9	32.1	39.4
KCF [18]	29.1	27.4	34.2	28.2	25.7	18.0	27.0	18.5	18.4	31.1	27.2	26.7

4.2.2. Evaluation on UAVDT Benchmark

Overall Evaluation: Figure 4 demonstrates the overall performance of our proposed tracker compared with several state-of-the-art trackers on UAVDT [16]. In addition to the seven trackers mentioned in Section 4.2.1), another two trackers which combine DCF and deep learning together, i.e., C-COT [32] and CF2 [28], are also involved in the comparison experiments. Specifically, on the

UAVDT [16] dataset, the precisions of our method achieve 78.0% when the location error threshold is equal to 20 pixels, 2.8% and 5.5% higher than those of the second and third best tracker, i.e., SiamDW (75.2%) and MDNet (72.5%), respectively. The AUC scores achieve 56.6%, and 10.2% higher than those of the second (SiamDW, 46.4%) and third best trackers (MDNet, 46.4%), respectively.



(**a**) (**b**)

Figure 4. Comparison results of several state-of-the-art trackers on the UAVDT dataset [16]. The values close to the name of the methods represent the AUC or the precision when the threshold of location error is set as 20 pixels. (**a**) The success rate plot; (**b**) The precision plot.

Attribute-Based Evaluation: Again, we perform an attribute-based analysis for the proposed method on UAVDT [16]. There are 8 attributes for each video sequence, including background clutter (BC), camera rotation (CR), object rotation (OR), small object (SO), illumination variation (IV), object blur (OB), scale variation (SV), and large occlusion (LO). As shown in Table 4, in the case of small objects, the AUC score of the proposed method achieves 51.8%, 5% higher than that of ARCF-HC which is designed for UAV target tracking, benefit from the fusion of shallow and deep features. In the case of scale variation, we improve the AUC score by at least by 16.7% as compared with the DCF-based trackers, 13.5% higher than deep learning based trackers, and 17.4% higher than the trackers which combine DCF and deep learning. The proposed tracker also achieves higher ranking performance in the video sequences with other attributes.

Table 4. Tracker performance analysis based on challenge attributes for the proposed method and the state-of-art methods on the UAVDT dataset [16]. Red, green, and blue fonts indicate the first, second, and third place.

Table	BC	CR	OR	SO	IV	OB	SV	LO
ours	50.0	56.9	54.9	51.8	56.2	52.3	60.5	55.3
ECO [33]	38.9	42.2	39.5	46.1	47.3	43.7	43.1	36.0
C-COT [32]	34.0	39.0	34.1	44.2	41.6	37.2	37.9	33.5
CF2 [28]	29.2	34.1	29.7	35.6	38.7	35.8	29.0	28.3
MDNet [39]	39.7	43.0	42.7	44.4	48.5	47.0	46.2	38.1
SiamFC [6]	38.6	40.9	38.4	43.9	47.4	45.3	42.4	35.9
SiamDW [27]	44.2	44.6	44.8	42.1	45.6	41.2	47.7	43.2
ARCF-HC [23]	39.4	43.0	39.9	46.8	45.7	44.4	43.8	38.7
SRDCF [21]	35.3	39.0	36.5	42.1	45.1	41.7	40.2	32.7
KCF [18]	23.5	26.7	24.4	25.1	31.1	29.7	25.4	22.8

4.3. Efficiency Analysis

To validate the effectiveness of the proposed method, we evaluate the performance of the UAV videos in the presence of challenges such as small objects, viewpoint changes, scale variation, and camera

motion. As shown in Figures 5 and 6, our method achieved better robustness and practicability in handling those cases. For example, in the *S010* sequence, KCF [18] loses tracking in the early stage and SiamDW [27] fails to cover the body of the object. CF2 [28] gives a tracking result covering the target, but its size is much larger than the target size, while our tracker performs well in the entire sequence. Because the target size of this sequence is very large at the beginning and it is reduced to one seventh of the original size in only 100 frames, the methods that only use multiscale input to solve the target scale change easily lose the target. The same situation happens in the *Person21* sequence. In the *S0103*, *S0307* and *S1301* sequences, there are different degrees of viewpoint change and object rotation. Although most of the trackers do not lose the target, the prediction accuracy of target size is uneven, while our method can achieve more robust results. In the sequence *Car15* where two trucks interlaced, the trackers are likely to be distracted by the similar object. SiamDW [27] loses tracking in the early stage because it is based on an offline end-to-end architecture, whereas our method has both online positioning module and offline training bounding-box prediction module, therefore, it can always cling to the target.

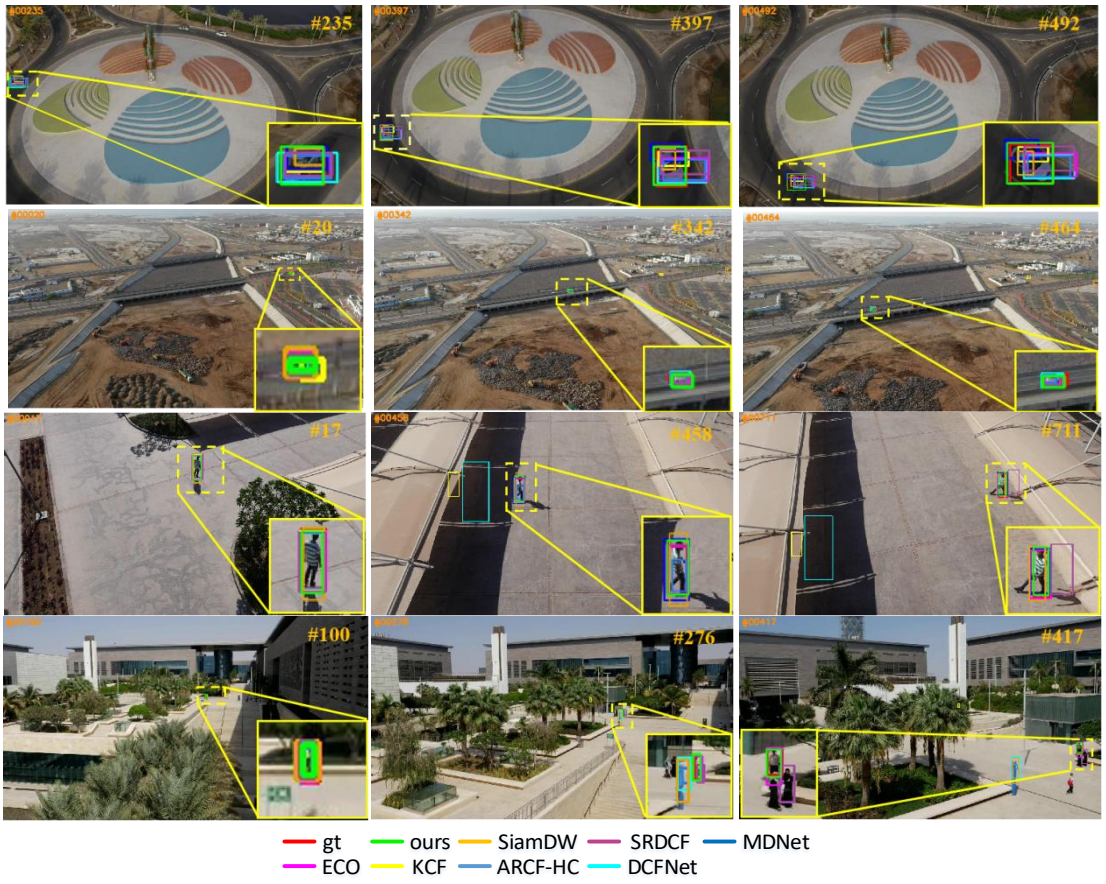


Figure 5. Qualitative evaluation of the proposed tracker and other state-of-art trackers on the UAV123 [15] dataset. From left to right and top to down are the screenshots of the tracking results on the videos of *Car1_3*, *Car15*, *Person12_2*, and *Person21*.

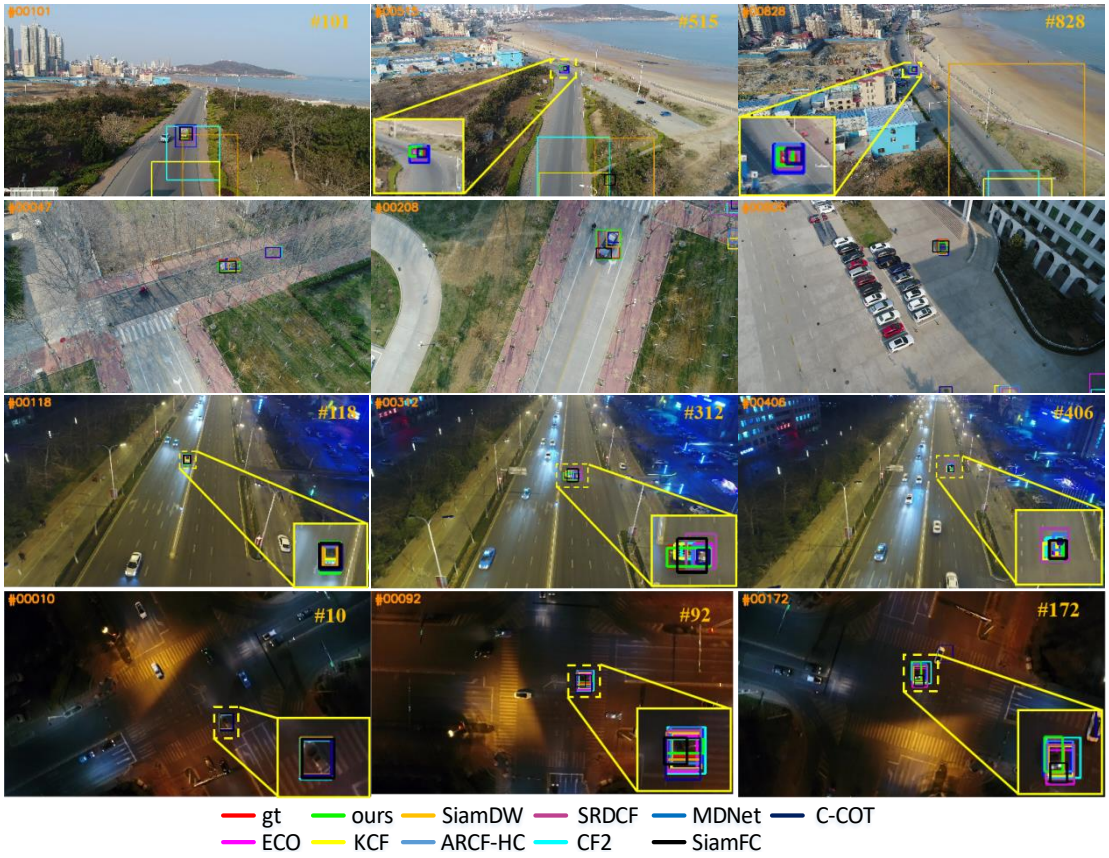


Figure 6. Qualitative evaluation of the proposed tracker and other state-of-art trackers on the UAVDT dataset [16]. From left to right and top to down are the screenshots of the tracking results on the videos of *S0101*, *S0103*, *S0307*, and *S1301*.

In addition to the satisfactory accuracy of tracking results, our method can carry out real-time processing on a general-purpose single-board GPU computing platform. The comparison results in terms of average frames per second (FPS) and milliseconds per frame (MSPF) are shown in Table 5. It can be seen that the proposed method achieve 25.6 FPS and 39.1 MSPF, respectively, which meet the requirements of real-time processing. This is because our proposed method adopts a correlation filtering operation, which achieves high tracking speed. Notably, the real-time performance, here, refers to the UAV tracker that relies on a general GPU parallel computing platform, which is compatible with real-time tracking of offline video and wireless backhaul video. The existing airborne computing platforms of miniaturized low-altitude drones have difficulty meeting the board-level real-time processing requirement.

Table 5. Average frames per second (FPS) and milliseconds per frame (MSPF)23 of the trackers on two benchmarks. Red, green, and blue fonts indicate the first, second, and third place, respectively. All results are generated solely under the same test environments.

	Ours	**ARCF-HC**	**SiamDW**	**MDNet**	**ECO**	**C-COT**	**SRDCF**	**DCFNet**	**SiamFC**	**KCF**
FPS	25.6	14.2	143.1	2.5	8.3	1.2	5.7	125.8	58.7	169.6
MSPF	39.1	70.4	6.9	40.0	12.0	83.3	17.5	7.9	17.0	5.9

4.4. Ablation Studies

To validate the effectiveness of our method, we compare two variants of our method with the original method that integrates all modules on the UAV123 [15] dataset, as shown in Table 6. Specifically, the TDS locates the target by only using the correlation filtering algorithm without multiscale input.

TDS + BES - NoIA uses the correlation filtering algorithm to locate the target and generates 16 candidate regions, and then uses IoU-Net to predict the IoU score for each candidate region. In this process, the coordinates of the candidate regions are modified continuously through five gradient optimizations, and finally the bounding box is obtained through fusing the results of the top three proposals. It should be noted that the IoU-Net in TDS + BES - NoIA does not add instance-aware vector at this time. TDS + BES is our proposed two-stage object tracking method, which uses correlation filter to locate the target position, and then use instance-aware IoU-Net to predict the scale of the target to obtain the final bounding box.

Table 6. Inner module ablation study by comparing the proposed method and two variants of our method on the UAVDT dataset [15]. The AUC is the overall area under curve score. The precision is the value when the threshold of location error is set as 20 pixels. Others are attribute-based AUC scores.

	AUC	Precision	BC	CR	OR	SO	IV	OB	SV	LO
TDS	45.3	71.3	36.3	44.9	38.8	46.1	46.1	43.3	42.7	37.0
TDS + BES - NoIA	50.3	73.8	41.4	49.8	43.4	48.8	50.3	44.9	48.2	44.0
TDS + BES	57.7	78.8	50.0	56.9	54.9	51.8	56.2	52.3	60.5	55.3

As shown in Table 6, TDS + BES obtains 7.4% and 12.4% improvement over TDS + BES - NoIA and TDS, according to the success plot AUC on the UAVDT dataset. Moreover, TDS + BES - NoIA has an5.0% improvement over the method only using TDS. Hence, our two-stage object tracking method is more effective, and the instance-aware network can further improve the accuracy as compared with the method only using IoU-Net.

More specifically, in the case of scale variation (SV), TDS + BES - NoIA gets about 5.5% higher performance gain than only using TDS. This proves that using IoU-Net can better track the targets with scale variation than using multiscale input, and can further improve the performance by introducing instance-aware IoU-Net. The final version of our method achieves 7.8% improvement over only using TDS. For the sequence with other attributes, the AUC score obtained by using the final version of our method is higher than by the other two variants of our method, which proves the validity that instance-aware IoU-Net does help TDS + BES - NoIA with lower sensitivity to various interferences by introducing the weights on the feature maps.

At the same time, the handcraft features and deep features are tested separately in the case of using instance-aware IoU-Net. The results are shown in Table 7, where DCF_DH_IOU stands for the original method, DCF_D_IOU represents a method using only deep features, and DCF_ H_ IOU represents a method that uses only handcraft features. It can be seen that the method using only handcraft features has the lowest AUC score in all-attribute video sequences except the case of small target. On the six-attribute video sequences, the AUC scores of the original methods are all higher than the methods only using depth features

Table 7. Feature ablation study by comparing the proposed method and two variants of our method on the UAVDT dataset [15]. The AUC is the overall area under curve score. The precision is the value when the threshold of location error is set as 20 pixels. Others are attribute-based AUC scores.

	AUC	Precision	BC	CR	OR	SO	IV	OB	SV	LO
DCF_DH_IoU	57.7	78.8	50.0	56.9	54.9	51.8	56.2	52.3	60.5	55.3
DCF_D_IoU	56.6	78.0	48.2	55.7	52.8	52.8	55.4	52.3	58.8	52.7
DCF_H_IoU	49.7	70.1	40.6	45.7	43.3	52.2	50.9	49.3	46.5	39.6

5. Conclusions

In this paper, we exploit a two-stage scheme that combines a detection-based network (IoU-Net) with a DCF-based tracker for object tracking in UAV videos. We integrate multilayer deep features

and handcrafted features to build a discriminative target representation, and use the instance-aware attention network to further embed the prior knowledge of the target. In the tracking process, the DCF-based tracking stage is responsible for locating the target, and the instance-aware IoU-Net is responsible for further predicting the refined target bounding box. We extensively evaluate the proposed tracker on two UAV tracking datasets (i.e., UAV123 [15] and UAVDT [16]). The experimental results show that our approach can improve the overall performance with a large margin and run at over 25 FPS on a general-purpose single-board GPU computing platform.

We summarize the potential directions for improving the proposed method and clarified our future work. First, in the bounding-box estimation stage, the instance-aware IoU-Net only gives the IoU score between the ground-truth bounding box and the candidate objects, and the spatial position relationship between the two is not accurately described. This problem eventually causes the online update of the correlation filtering to be disturbed with the background area around the target. In the future, we should improve the loss function to suppress the expansion of bounding-box anomalies in the optimization process. Second, our method has not fully utilized the inter-frame timing information in the video, which leads to the lack of robust tracking when interfering targets with similar appearances exist. In the future, we plan to explore a priori information embedding method combining temporal context in the instance-aware attention network.

Author Contributions: All the authors made significant contributions to the work. S.Z.'s contribution is conceptualization, investigation, methodology and writing-original draft. L.Z.'s contribution is conceptualization, data curation, funding acquisition, methodology and writing-review & editing. H.Z. is working for formal analysis, validation and writing-review & editing. J.L. is working for funding acquisition, methodology and resources. All authors have read and agreed to the published version of the manuscript.

Funding: This work research was funded by the Beijing Municipal Natural Science Foundation Cooperation Beijing Education Committee (no. KZ 201810005002) and the National Natural Science Foundation of China (no. 61602018).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
2. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In Proceedings of the European Conference on Computer Vision (ECCV), Florence, Italy, 7–13 October 2012; pp. 702–715.
3. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H.S. End-to-End Representation Learning for Correlation Filter Based Tracking. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5000–5008.
4. Wang, Q.; Gao, J.; Xing, J.; Zhang, M.; Hu, W. Dcfnet: Discriminant correlation filters network for visual tracking. *arXiv* **2017**, arXiv:1704.04057.
5. Held, D.; Thrun, S.; Savarese, S. Learning to Track at 100 FPS with Deep Regression Networks. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 749–765.
6. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-Convolutional Siamese Networks for Object Tracking. In Proceedings of the European Conference on Computer Vision Workshops (ECCVW), Amsterdam, The Netherlands, 8–16 October 2016; pp. 850–865.
7. Tao, R.; Gavves, E.; Smeulders, A.W.M. Siamese Instance Search for Tracking. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1420–1429.
8. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ATOM: Accurate Tracking by Overlap Maximization. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4655–4664.

9. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-Aware Siamese Networks for Visual Object Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 103–119.
10. Huang, L.; Zhao, X.; Huang, K. Bridging the gap between detection and tracking: A unified approach. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 29 October–1 November 2019; pp. 3999–4009.
11. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High Performance Visual Tracking with Siamese Region Proposal Network. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 8971–8980.
12. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4277–4286.
13. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H.S. Fast Online Object Tracking and Segmentation: A Unifying Approach. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1328–1338.
14. Jiang, B.; Luo, R.; Mao, J.; Xiao, T.; Jiang, Y. Acquisition of Localization Confidence for Accurate Object Detection. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 816–832.
15. Mueller, M.; Smith, N.; Ghanem, B. A Benchmark and Simulator for UAV Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 445–461.
16. Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; Tian, Q. The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 375–391.
17. Wu, Y.; Lim, J.; Yang, M. Online Object Tracking: A Benchmark. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
18. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596. [CrossRef] [PubMed]
19. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Discriminative Scale Space Tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1561–1575. [CrossRef] [PubMed]
20. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H.S. Staple: Complementary Learners for Real-Time Tracking. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.
21. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Learning Spatially Regularized Correlation Filters for Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
22. Wang, Y.; Ding, L.; Laganiere, R. Real-Time UAV Tracking Based on PSR Stability. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), Seoul, Korea, 8 October–2 November 2019.
23. Huang, Z.; Fu, C.; Li, Y.; Lin, F.; Lu, P. Learning aberrance repressed correlation filters for real-time uav tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Seoul, Korea, 29 October–1 November 2019; pp. 2891–2900.
24. He, Y.; Fu, C.; Lin, F.; Li, Y.; Lu, P. Towards Robust Visual Tracking for Unmanned Aerial Vehicle with Tri-Attentional Correlation Filters. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24–30 October 2020.
25. Deng, J.; Dong, W.; Socher, R.; Li, L.; Kai, L.; Li, F.-F. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

27. Zhang, Z.; Peng, H. Deeper and Wider Siamese Networks for Real-Time Visual Tracking. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4586–4595.
28. Ma, C.; Huang, J.; Yang, X.; Yang, M. Hierarchical Convolutional Features for Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 3074–3082.
29. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
30. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Convolutional Features for Correlation Filter Based Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), Santiago, Chile, 7–13 December 2015; pp. 621–629.
31. Chatfield, K.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Return of the Devil in the Details: Delving Deep into Convolutional Nets. *arXiv* **2014**, arXiv:1405.3531v4.
32. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond Correlation Filters: Learning Continuous Convolution Operators for Visual Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 472–488.
33. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. ECO: Efficient Convolution Operators for Tracking. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6931–6939.
34. Choi, J.; Chang, H.J.; Yun, S.; Fischer, T.; Demiris, Y.; Choi, J.Y. Attentional Correlation Filter Network for Adaptive Visual Tracking. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4828–4837.
35. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
36. Müller, M.; Bibi, A.; Giancola, S.; Alsubaihi, S.; Ghanem, B. TrackingNet: A Large-Scale Dataset and Benchmark for Object Tracking in the Wild. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 310–327.
37. Zhu, P.; Wen, L.; Bian, X.; Ling, H.; Hu, Q. Vision meets drones: A challenge. *arXiv* **2018**, arXiv:1804.07437.
38. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
39. Nam, H.; Han, B. Learning Multi-domain Convolutional Neural Networks for Visual Tracking. In Proceedings of then IEEE Conference Computer on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302.