Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2022.020471

ARTICLE

Visual Object Tracking via Cascaded RPN Fusion and Coordinate Attention

Jianming Zhang[1,2,*], Kai Wang[1,2], Yaoqi He[1,2] and Lidan Kuang[1,2]

[1]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, 410114, China

[2]Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Changsha University of Science and Technology, Changsha, 410114, China

*Corresponding Author: Jianming Zhang. Email: jmzhang@csust.edu.cn

Received: 26 November 2021 Accepted: 20 January 2022

ABSTRACT

Recently, Siamese-based trackers have achieved excellent performance in object tracking. However, the high speed and deformation of objects in the movement process make tracking difficult. Therefore, we have incorporated cascaded region-proposal-network (RPN) fusion and coordinate attention into Siamese trackers. The proposed network framework consists of three parts: a feature-extraction sub-network, coordinate attention block, and cascaded RPN block. We exploit the coordinate attention block, which can embed location information into channel attention, to establish long-term spatial location dependence while maintaining channel associations. Thus, the features of different layers are enhanced by the coordinate attention block. We then send these features separately into the cascaded RPN for classification and regression. According to the two classification and regression results, the final position of the target is obtained. To verify the effectiveness of the proposed method, we conducted comprehensive experiments on the OTB100, VOT2016, UAV123, and GOT-10k datasets. Compared with other state-of-the-art trackers, the proposed tracker achieved good performance and can run at real-time speed.

KEYWORDS

Object tracking; deep learning; coordinate attention; cascaded RPN

1 Introduction

Visual object tracking is a basic but challenging task in computer vision. In recent years, object tracking has received widespread attention due to its extensive applications, e.g., in intelligent surveillance [1], human-computer interaction, autonomous driving [2], and other vision fields. In reality, object tracking is the process of finding the region of interest in a current frame based on subsequent video frames through feature matching. However, because occlusion, deformation, background blur, and scale-change problems during movement inevitably occur [3], it is challenging to accurately predict the scale size and position of a target.

Recently, target trackers based on a Siamese network have received significant attention because of their excellent tracking performance and robustness. The Siamese network usually transforms the tracking task into a similarity-matching problem. The tracking network consists of template and search branches. We can obtain the similarity by extracting features from these two branches and performing cross-correlation operations of these features. The above process is actually an end-to-end method that directly uses extracted features for calculations, but not for modeling. This idea was introduced for the first time in 2016 in fully convolutional Siamese networks (SiamFC) [4], which treated target tracking as a similarity problem between the template and search area. Although this network laid the foundation of the overall framework for object tracking, it requires intensive window sliding and has difficulty coping with changes in target scale. With the development of the Siamese Region Proposal Network (SiamRPN) [5], a region-proposal-network (RPN) [6] structure of detection was subsequently introduced. The SiamRPN converted the similarity calculation into a cross-correlation operation and obtained two branches, i.e., classification and regression. Classification is used for distinguishing foreground from background, and regression is used for determining the offset of the bounding box. However, up to this point, the backbone network of the tracker generally utilized a shallow network, namely AlexNet [7]. The feature-extraction capability of AlexNet is not as good as that of the deep residual network (ResNet), which greatly affects the performance of the tracker. However, if ResNet [8] is directly used as the backbone network of the tracker, the performance of the tracker could decrease. After conducting numerous experiments, with the emergence of the Deeper and Wider Siamese Networks (SiamDW) [9] and SiamRPN++ [10], the influence of padding was finally removed, and the deeper backbone network was successfully applied to Siamese tracking. This method greatly improves the performance of the tracker.

An attention mechanism is a commonly used method in deep learning. It has powerful functions and plug-and-play features, especially in detection segmentation and natural language processing. Generally speaking, the attention mechanism will tell the model what information is involved in the calculation and where it is. Among others, the squeeze-and-excitation network (SENet) [11] and dual attention networks (DANets) [12] inspired our work. SENet uses the information between channels to generate mappings and assigns higher weight to important information. A DANet can adaptively combine local features and global dependencies. A location attention block weights the features at all locations to correlate similar features with each other, while a channel attention block can integrate the mapping of all channel relationships. This helps obtain more useful feature information for classification and regression. Based on this, in our approach, we encode the location and channel information together on the basis of SENet. While capturing the channel-information association, the long-term dependence of spatial location is established.

In SiamRPN tracking, only the deep-level information of the backbone network is used, which is the fifth-level feature. The deep-level feature contains a large amount of semantic information but lacks appearance information. The original tracking network simply extracts features through the backbone network for classification and regression. These features are not strong enough to deal with complex environmental disturbances or to perform identification well enough to locate fast-moving objects in, for example, a drone scene. Our aim is to enhance the feature representation through an attention mechanism and then fuse the tracking results of the different layers of features to be able to accurately locate the target location. Thus, on the basis of upgrading the backbone network to ResNet, we have added an RPN block to the third layer to handle the appearance information of the object. Then, the extracted features are fused to obtain robust features that contain both rich appearance information and sufficient semantic information. The fused features are used for improving the performance of succeeding classification and regression. In addition, SENet only considers the information between channels but ignores the location information that plays an important role in generating the spatial selective response map. We further introduced a coordinated attention (CA) [13] method that embeds position information into channel attention. After this, part of the coded features is sent to the RPN block, where the feature information can be better utilized.

To summarize, the overall contributions of this paper are as follows:

(1) While maintaining channel associations, we introduce a CA mechanism that embeds location information into channel information to establish long-term spatial location dependence.

(2) We propose a cascaded RPN fusion structure that contains two RPN blocks to process the features of different layers. Weighted fusion is then performed based on the results of the two classifications and regressions to obtain the accurate position of the target.

(3) Without adjusting the parameters too much, the proposed method exhibits good performance on the OTB100, VOT2016, UAV123 and GOT-10k datasets, and achieve the best performance on UAV123. This dataset comprises objects that were all photographed by drones and that exhibit fast-moving speed and large-scale changes. This result shows that the proposed tracker can deal with this kind of problem well.

The structure of our paper can be expressed as follows. First, we give a general description of related work in Section 2. We give the detailed description of our proposed method in Section 3. We give the experimental environment and detailed experimental settings in Section 4. We conduct comprehensive experiments of our proposed method and state-of-art methods on datasets in Section 5. Section 6 gives the summary and prospect of this work.

2 Related Work

In this section, we mainly review the relevant literature, the tracking frameworks commonly used in recent years, and content related to our work.

2.1 Siamese Tracking

In recent years, tracking algorithms based on Siamese networks [4–5,9,10,14] have attracted considerable attention because of their excellent accuracy and efficiency. It is undisputed that SiamFC [4] is considered the beginning of end-to-end tracking. The entire tracking network is composed of template and search branches. The tracking problem is often transformed into a similarity-matching problem. By performing cross-correlation operations on the features extracted by the two branches, the approximate location of the target is obtained. Because this method requires intensive window sliding, it has difficulty coping with the size change of the target. The RPN block was introduced into the SiamRPN [5] by RCNN, who discussed the use in detection. The RPN [6] block sets the anchor frame ratio of five scales in advance, generally 1/3, 1/2, 1, 2, and 3. Next, k anchor boxes are generated at each position to predict where the object may exist. To better train the model, a distractor-aware Siamese network (DaSiamRPN) [15] classified the samples roughly into positive and negative samples. When the degree of overlap between the ground truth and bounding box of the sample is greater than a certain threshold, we define it as a positive sample; otherwise, it is defined as a negative sample. Moreover, another situation arises in which the degree of overlap is high, but the object and target are not in the same category. Such samples are defined as hard negative samples. A hard negative sample is difficult to remove, but is particularly important for the robustness of the training model.

Although the SiamRPN method greatly improves tracking performance, the backbone of the network still uses the shallow AlexNet, and the feature-extraction ability is not strong. SiamDW, SiamRPN++, and SiamMask [16] remove the effect of padding in different ways and introduce several important deep networks, such as MobileNet [17] and ResNet [8], to Siamese tracking. In particular, SiamDW spends a significant amount of space to determine the influence of field, output size, stride length, and the presence or absence of padding and provides the optimal values of these influencing factors. This is especially important for understanding and applying deep backbone networks as feature extractors. In addition, the output features of the convolutional layer are related to the perceptual field. The perceptual field of a neuron is the size of the pixel point of the feature map output from each layer of the convolutional neural network corresponding to the region on the original image [10]. As the number of convolutional layers increases, the receptive field of neurons also gradually increases, and the output layer at this time contains deeper semantic features [15,18,19]. This is good for challenging tracking. The shallow features, although they experience less convolution, contain more localized and detailed information, e.g., appearance color as well as location information.

Since 2020, anchor-free frameworks have become popular in object tracking. Anchor-free, as the name implies, does not use an anchor box to obtain the possible location of the target. The previous method is to generate k anchor boxes at each pixel to capture the possible position and size of the target. The current idea is to generate a possible position of the target with each pixel as the center. Because these anchor boxes are not applied in the entire tracking process, the number of parameters and the amount of calculation in the calculation process are greatly reduced. SiamCAR [18], SiamBAN [19], and SiamFC++ [20] are typical representatives of this idea. Anchor-free is a new trend in object tracking.

Most of the mainstream Siamese tracking algorithms mentioned above are not updated online. Compared with some methods that perform online update correlation filtering, or combine Siamese tracking with correlation filtering, Siamese tracking has difficulty dealing with the problem of background updates. Some studies report algorithms that have done a good job in this regard, and not only consider the correlation between the previous and next frames, but also dynamically update the template [21–24]. The CFNet [25] adopts the SiamFC framework and introduces related filtering into Siamese tracking, which improves accuracy. These are all effective methods of improving tracking accuracy.

2.2 Convolutional Features Fusing

In convolutional neural networks, the convolutional features contained in different levels of networks are different. Generally speaking, a shallow network contains more appearance information of the target, and the deep network contains more semantic information after layered feature extraction [10,26–28]. Semantic information is robust to significant appearance changes but cannot accurately locate the target due to the coarse spatial resolution of semantic information. However, shallow features can be used for more precise positioning.

Therefore, a copious amount of work has been directed to fusing features of different layers. For example, in C-RPN [27,29], the extracted features of different layers are first input to the feature-fusion module, and then, the fused features are processed in subsequent steps. Several methods [30,31] exist that combine manual and deep features, and they have achieved good performance. We call the above approaches "early fusion," in which different characteristics are fused.

Another method is to fuse the results of each processing, which we call "late fusion." Some two-stage networks, such as SPM [29], first perform coarse matching and then fine matching and obtain the final target position based on the results of the two matches. Some similar methods, such as FPN [30,32,33], first use their respective features to make predictions and then provide the final result based on the results of multiple predictions.

2.3 Attention Mechanism

Attention mechanisms have been widely studied and applied in visual tasks in the past few years [11–12,34] because they are simple, powerful, and plug-and-play. An attention block can be simply regarded as a computing unit to enhance feature-representation ability. A typical attention mechanism is SENet [11].

SENet converts the feature tensor into a single feature vector by simply squeezing each two-dimensional feature map and then effectively builds the dependency between channels. However, SENet only measures the importance of each channel by modeling the channel relationship, ignoring the location information used to generate spatial selectivity.

Another commonly used attention mechanism is a non-local information statistical attention mechanism called a simplified non-local (SNL) block [34]. Because the convolutional network must capture long-term dependence by gathering global statistics, the network will be deep and the learning efficiency very low. Non-locality is used to obtain the autocorrelation of features through mapping and dot-multiplication operations on features. However, the non-local method involves a very large amount of calculation and may not run normally when memory is limited.

3 Methods

We now introduce the proposed network model. First, we use the CA block to embed the position information into the channel attention to enhance the feature representation. The CA block can use location information to establish long-term spatial location dependence while maintaining channel associations (see Section 3.3). Subsequently, the features of different layers are enhanced by the CA block and input to the cascaded RPN block. By fusing the results of the two classifications and regressions, the accurate position of the target in the current frame is obtained.

3.1 Overview

We roughly divide the entire tracking process into three parts: the feature extraction sub-network that extracts features of different layers, the coordinate attention block that enhances the feature representation, and the cascaded RPN multiple classifications and regressions used to accurately target the location.

Fig. 1 shows the entire tracking process. We used ResNet-50 [8] as our backbone network. ResNet-50 has stronger feature-extraction capabilities than AlexNet [7], and can better mine the hidden information in an image. The feature-extraction part can extract the features of template Z and search area X. Most trackers simply feed deeper features with more semantic information into the RPN block. We believe this is insufficient to make full use of the extracted feature information. Therefore, we first merge the feature of the third and fourth layers through the add operation to fully obtain the appearance information of the target. Second, we put the fused information into the coordinate attention blocks. A CA block can enhance the feature representation, which is more conducive to tracking. Third, we send the fused feature to the first RPN block to initially obtain the approximate position of the target. In addition, we also carry out the above-mentioned process for the features of the fifth layer. Finally, the results of the two classifications and regressions are weighted and fused to obtain a more accurate position of the target in the current frame.

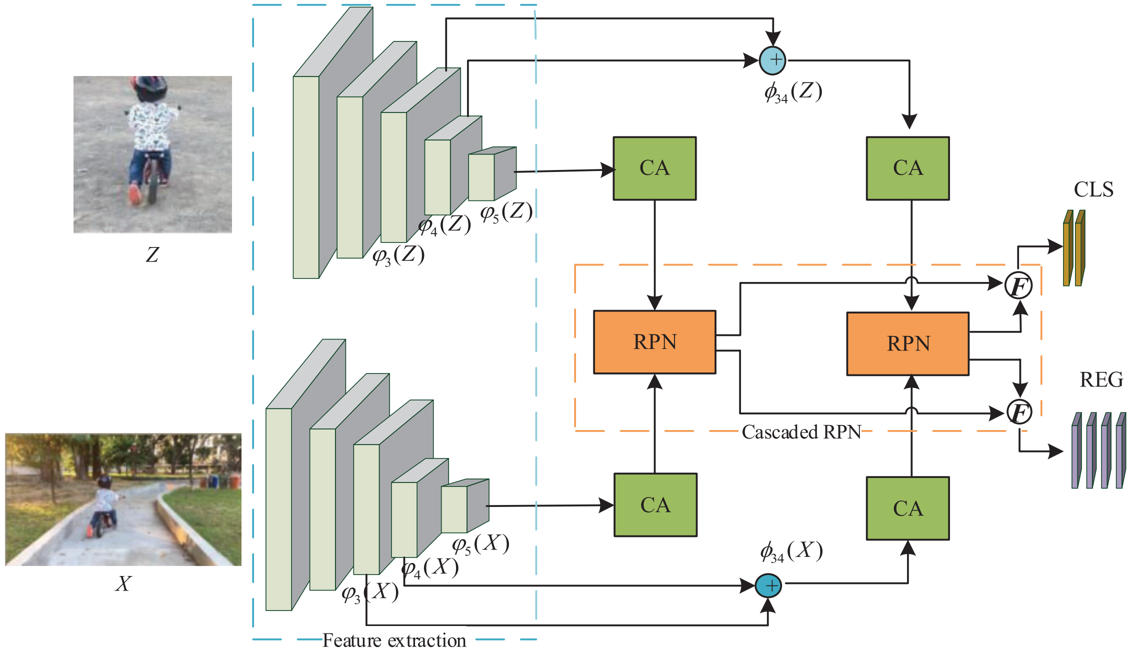


Figure 1: Illustration of our network.$\varphi_i(Z)$ and $\varphi_i(X)$ represent the features of the template and search area extracted from each layer. The feature extraction network uses ResNet-50. CA represents the coordinate attention block and RPN the region proposal network block. + is an add operation. F is expressed as a weighted fusion of the two results. CLS and Reg represent the results of classification and regression, respectively

3.2 Cascaded RPN Fusion

In SiamRPN, only the features of the last layer extracted are used. High-level features have stronger semantic information, which has great advantages in challenging scenes such as motion blur and huge deformation. However, the low resolution of high-level features leads to poor perception of details, while the shallow features, mainly containing some low-level information, e.g., appearance and position information, are all necessary for positioning but contain less semantic information and a large amount of noise.

Therefore, we designed a cascaded RPN structure to use the features of different layers and then performed weighted fusion on the classification and regression results of each RPN. Combining the classification and regression results of these two processes, a more accurate position of the target in the current frame is obtained. The process of weighted fusion for each result is represented by F in Fig. 1. The weight of the classification branch is 1.0 and that of the regression branch is 1.2. To better use the appearance features before inputting to the RPN block, we use the add operation to simply merge the features of the third and fourth layers.

The RPN structure used is shown in Fig. 2. For each template Z, the extracted features are denoted by $\varphi(Z)$ and are used for classification and regression. The classification branch is denoted by $\varphi(Z)_{cls}$ and the regression branch by $\varphi(Z)_{reg}$ for subsequent calculations. The corresponding search area X also extracts the feature $\varphi(X)$, and the classification and regression branches are denoted by $\varphi(X)_{cls}$ and $\varphi(X)_{reg}$, respectively. Conv is a 1 × 1 transformation used to change the number of channels of the feature. We performed a depthwise correlation (DW_corr) operation on the features extracted from these two branches to obtain the classification score and regression offset of each anchor box.

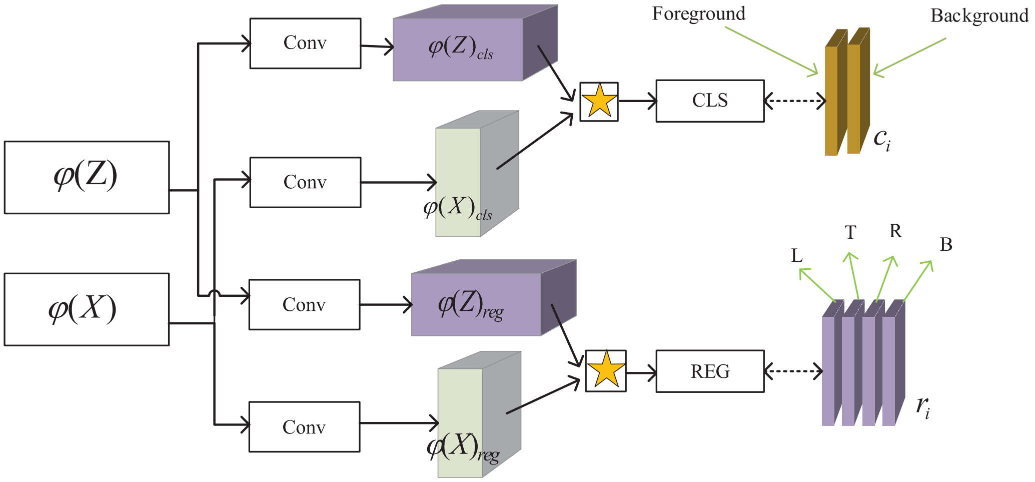


Figure 2: Schematic of RPN structure. The block consists of template and classification branches. ★ denotes DW_corr operation. Finally, the foreground and background classification and regression offsets L, T, R, and B of the object are obtained

The add operation is a parallel strategy that combines the two aforementioned feature vectors into a complex vector. This operation does not change the dimension of the features but increases the information of each dimension. This is beneficial to final image classification. The process of feature fusion is expressed as follows:

$$\phi_{34}(Z) = add(\varphi_3(Z), \varphi_4(Z)), \tag{1}$$

$$\phi_{34}(X) = add(\varphi_3(X), \varphi_4(X)), \tag{2}$$

where Z is the template, X is the search area,$\varphi_i(Z)$ is the extracted feature of the ith ($i = 3$, 4) layer template, and $\phi_{34}(Z)$ is the feature after the third and fourth layers are fused. The fused features are sent to the first RPN block for preliminary classification and regression. The first RPN block mainly deals with the appearance features of the target, and the second RPN block mainly deals with the semantic features of the target. We can obtain the target's position through these two regression steps, even of objects with extreme shapes. We have

$$\{c_i\} = DW\_corr([\phi_l(Z)]_{cls}, [\phi_l(X)]_{cls}), \tag{3}$$

$$\{r_i\} = DW\_corr([\phi_l(Z)]_{reg}, [\phi_l(X)]_{reg}), \tag{4}$$

where c_i is a two-dimensional vector representing the classification result of the ith anchor box, that is, positive and negative samples. r_i is a four-dimensional vector that represents the offset of the anchor box compared to the ground-truth bounding box. DW_corr is different from ordinary convolution, as each convolution kernel is only responsible for the calculation of one channel. Through channel-by-channel calculation, more abundant features are obtained for classification and regression.

Correspondingly, we can obtain the loss of the entire cascaded block as follows:

$$L_{RPN}(\{c_i\}, \{r_i\}) = \sum_i L_{cls}(c_i, c_i^*) + \lambda \sum_i c_i^* L_{loc}(r_i, r_i^*), \tag{5}$$

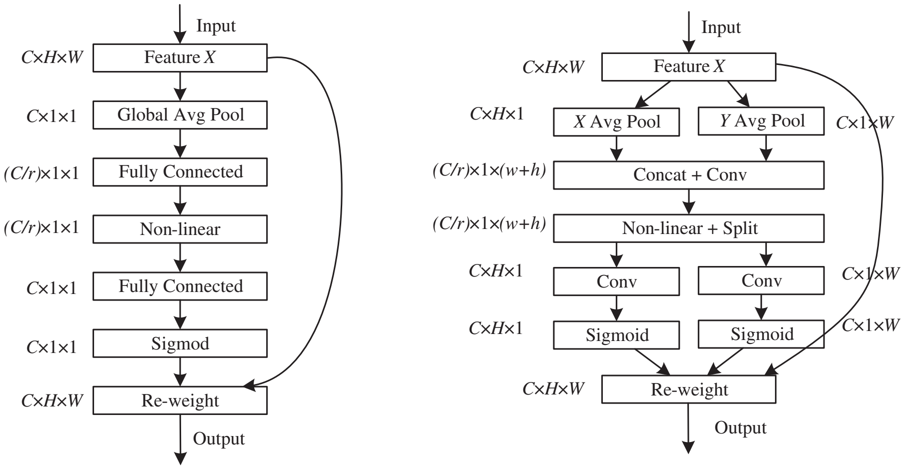
where i refers to the anchor box with index i. c_i is the predicted value of the tracker, c_i^* is the classification label, and λ is the weight of the regression part. In this formula, we use Softmax loss for classification and smooth L_1 loss for regression.

3.3 *Coordinate Attention*

The attention mechanism is well-known because it is helpful for most visual tasks. The most classic one is SE attention [11]. SE attention uses a simple squeeze for each two-dimensional feature map to effectively construct the interdependence between channels. Generally speaking, the SE block contains two parts, i.e., squeeze and excitation. The squeeze operation is used to obtain global information, and excitation is used to obtain the correlation between channels. Given the input feature X, the cth channel can be expressed as

$$Z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} X_c(i,j). \tag{6}$$

The C channel can be expressed as Z_c through a squeeze, which is the output related to the C channel. In Eq. (6), the input feature X comes from different layer features extracted from the backbone network. W and H are the width and height, respectively, of the input feature. This process corresponds to the "Global Avg Pool" operation in Fig. 3a.



(a)SE block (b)CA block

Figure 3: (a) SE block and (b) CA block. "X Avg Pool" and "Y Avg Pool" represent pooling operations in horizontal and vertical directions. "Concat" represents the connection of features and is sent to 1 × 1 convolution. "Split" will decompose the features obtained in the previous step. Then, "Split" is sent into the convolution to change the number of channels, which is activated as the weight of the CA block and added to the input feature

To obtain the correlation between channels, the information after the squeeze must be activated. The excitation process can be expressed as

$$\widehat{Z} = T_2(RELU(T_1(Z)), \tag{7}$$

where two linear transformations are performed on the obtained squeeze results through the full connections T_1 and T_2 to capture the importance of each channel. $\widehat{Z}$ is the result after excitation.

The channel dependence is obtained after processing the transformation as follows:

$$\widehat{X} = X \cdot Sigmod(\widehat{Z}), \tag{8}$$

where X refers to the input feature and the $\cdot$ operation to the channel-wise multiplication. Eq. (8) is reflected in the last step of the SE block. Therefore, the enhanced feature X can be obtained in SENet.

However, the SE block only considers the importance of the channel and ignores the position information. The position information is very important for generating spatial selective attention response maps. Therefore, we introduce coordinate attention [13], which obtains channel and location information. While maintaining channel associations, the CA block embeds location information into CA and uses location information to establish long-term spatial location dependence. This process is mainly divided into two steps: coordinate information embedding and coordinate generation.

Because global pooling compresses global information into channel descriptions, it is difficult to save location information. To obtain new remote spatial interactions with precise locations, we process both horizontal and vertical global pooling through two one-dimensional global pooling operations. Then, two independent directional perception feature maps are generated. Next, we encode these two direction-information-containing feature maps and generate two attention maps to capture the dependence of each spatial direction. In this way, the position information will be retained in the attention map.

Given the input X, we performed two spatial decompositions, encoding each channel along the horizontal and vertical coordinates. The output of the cth channel with height h is written as

$$Z_c^h(h) = \frac{1}{W} \sum_{0 \le i \le W} X_c(h, j). \tag{9}$$

The similar output of channel c with width w is written as

$$Z_c^w(w) = \frac{1}{H} \sum_{0 \le i \le H} X_c(j, w). \tag{10}$$

Eqs. (9) and (10) represent the X Avg pool and Y Avg pool in Fig. 3b, respectively. The above two transformations decompose two spatial directions separately to obtain a pair of directional perception feature maps. This conversion enables attention to capture long-term dependence along a certain spatial direction and precise information along another spatial direction, which is helpful for target positioning. Through the above formula decomposition, the global receptive field can be obtained and accurate position information can be encoded.

To make full use of the expression information, CA generation transformation is performed. This process is similar to the activation part of the SE block. When changing, the correlation of the channel information should be preserved as much as possible. In addition, the captured position information should be fully utilized to facilitate determine the location of the region of interest. We call this process CA generation.

First, we connect the horizontal and vertical decomposition features and then send them to a 1×1 convolution, which is expressed as

$$f = RELU(conv([z^h, z^w])), \tag{11}$$

with the square brackets [,] representing the Concat operation along the spatial dimension; conv is the 1×1 convolution mentioned above. Non-linear operations generally use the rectified linear unit (RELU) function, $f \in \mathbb{R}^{(C/r)\times(H+W)}$. It is the intermediate feature information that is encoded along the horizontal and vertical directions. r is a reduction factor that controls the size of the SE block. Then, the intermediate feature f is decomposed along the spatial dimension directions, $f^h \in \mathbb{R}^{(C/r)\times(H+W)}$ and $f^w \in \mathbb{R}^{(C/r)\times(H+W)}$. After performing a 1 × 1 transformation on f to obtain the same dimension as the input X, the attention weight in two directions is then obtained by the activation function:

$$g_c^h = Sigmod(Conv(f^h)), \tag{12}$$

$$g_c^w = Sigmod(Conv(f^w)), \tag{13}$$

Conv represents a 1 × 1 transformation used to change the number of channels. The obtained result is applied to the input as the weight of the attention block, and the output is obtained. Eq. (14) is reflected in the last step of the CA block,

$$y_c(i,j) = x_c(i,j) \times g_c^h(i) \times g_c^w(j). \tag{14}$$

Finally, the difference between the SE and CA blocks is compared, as shown in Fig. 3.

4 Experimental Setup

The method method was implemented in Python with Pytorch on three RTX2080Ti processors; the operating system was Ubuntu 16 with 32 Gb of memory. The backbone network was the modified ResNet-50. Instead of starting training from scratch, the backbone network was initialized with the ImageNet [35] pre-trained weights, and the parameters of the first two layers were frozen. During training and testing, we used an image with a size of 127 pixels as a template and an image with a size of 255 pixels as the search area. The training set included ImageNet VID, ImageNet DET, YOUTUBEBE [36], and COCO [37]. In the training process, the batch size was set to 28, the optimizer used the stochastic gradient descent (SGD) method, and the initial learning rate was 0.005. In the last 10 batches, we unfroze the last three layers of the network and loaded the backbone network for training. Including the loading of the above four datasets for training, the entire training process lasted approximately 48 h. After training, comprehensive experiments were carried out on the VOT2016 [38], OTB100 [39], UAV123 [40], and GOT-10k [41] datasets. The process of testing the training model on each dataset and then comparing the results obtained with the existing algorithm took approximately 10 h. In addition, for the sake of fairness, the experiment on GOT-10k was tested at the address provided on its official website. We compared roughly 10 trackers on each dataset, but the available trackers were not tested on every dataset, and not all of them provided tracking results. Therefore, we chose some of the better typical tracker results for comparison as much as possible, which contain both correlation-filter-based and deep-learning-based trackers. The results of the comparison between OTB100 and UAV123 were also plotted.

5 Results and Discussion

5.1 Evaluation Results on OTB100

OTB100 is a widely used public dataset containing 100 challenging videos with significant changes. These challenges come from the following aspects, mainly including background clutter, lighting changes, scale changes, motion blur, occlusion, and rotation and deformation. Among them, OTB100 contains the OTB50 dataset, which is more comprehensive. During the test, we used success and precision plots to evaluate different trackers in a one-time evaluation of OPE. The centroid distance between the prediction and real boxes was calculated in OTB100, and the percentage of the total sequences with a distance less than a certain threshold was counted. The threshold value was then plotted as the horizontal coordinate and the corresponding precision as the vertical coordinate for the precision plot. The same success plot calculates the intersection ratio of the predicted plot to the real frame area and counts the percentage of video sequences with an intersection ratio area greater than a certain threshold to the total sequences. However, the precision plot cannot reflect the change of target scale and size.

In this test set, we compared DaSiamRPN [15], Eco_HC [42], SiamRPN [5], BACF [43], Staple [44], and SiamFC [4]. It can be seen in Fig. 4 that the proposed tracker achieves satisfactory performance on OTB100, which is close to the effect of DaSiamRPN. The proposed tracker attains a score of 0.654 in the success plot and 0.875 in the precision plot. It is 1.1% higher than ECO-HC in the success plot and 1.9% higher than ECO-HC in the precision plot. Compared with SiamRPN, the proposed tracker's score increases by approximately 1.7% in the success plot and by 2.4% in the precision plot. Because we retain the position information and spatial dependence in the coordinate attention, the proposed tracker achieves a favorable effect under the challenge of fast motion and scale transformation. It can be seen in Fig. 5 that, in the fast-motion challenge, the proposed tracker achieves a good score of 0.856, which is 3.6% higher than that of DaSiamRPN.

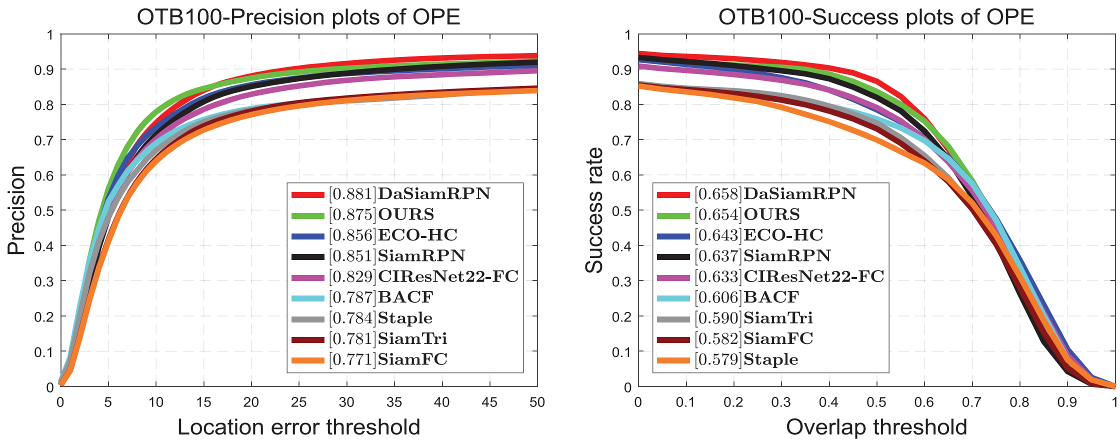


Figure 4: Success and precision plots on OTB100

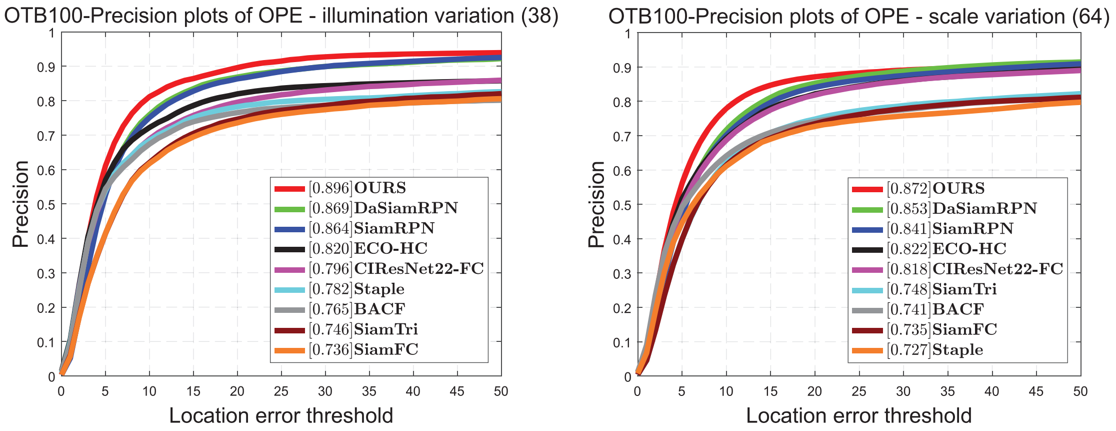


Figure 5: (Continued)

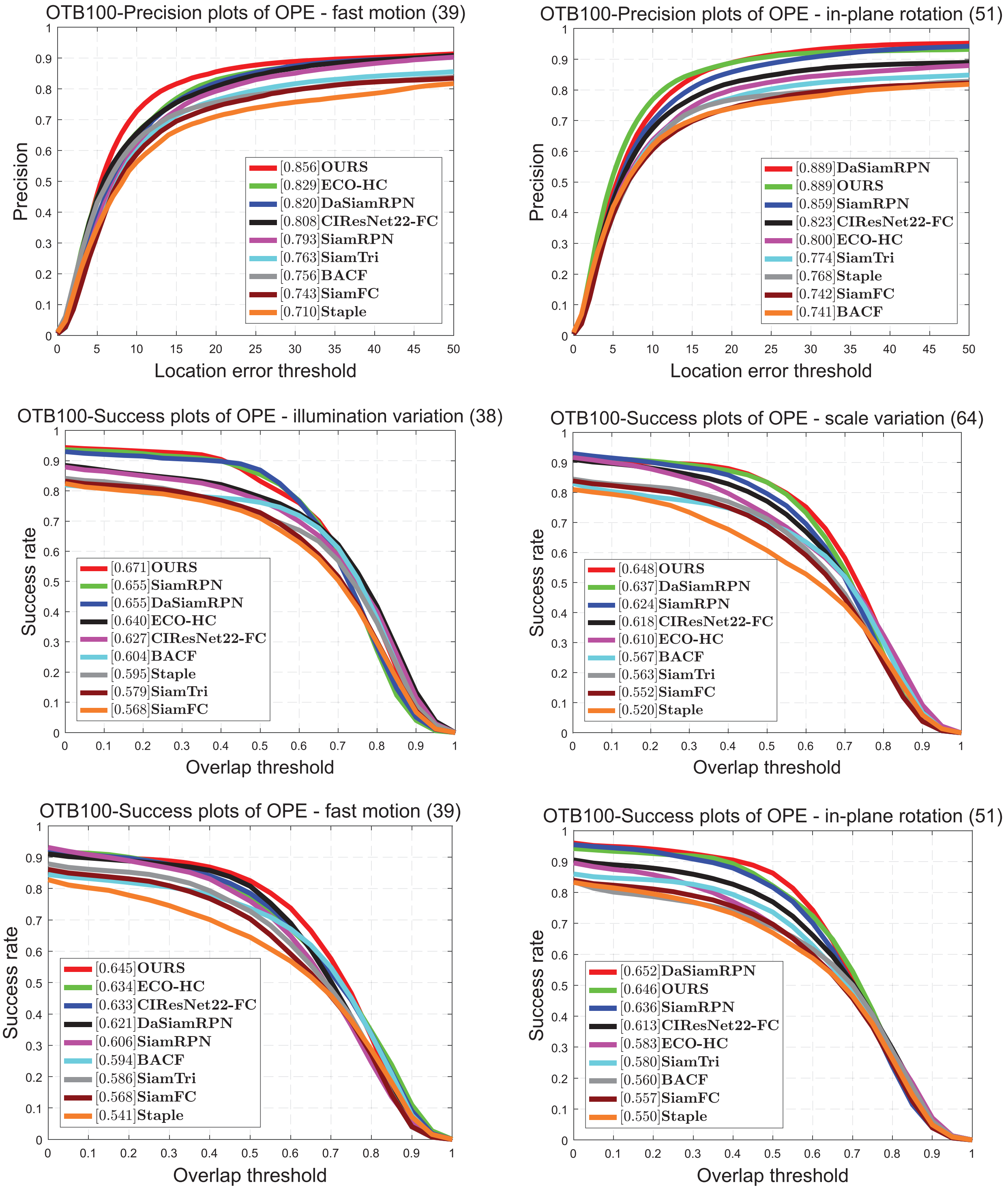


Figure 5: Comparisons on OTB100 with challenging attributes: illumination variation, scale variation, fast motion, and in-plane rotation. Proposed tracker achieves the best performance for these aspects

5.2 *Evaluation Results on VOT2016*

VOT2016 is a dataset that has been used more and more popular in recent years. It contains 60 public sequences with different challenge factors. The sequence of the VOT2016 dataset is the same as the sequence of the VOT2015 dataset. However, the ground-truth boxes of VOT2016 are more accurate than the ground-truth boxes of the VOT2015 dataset. In VOT2016, we used the officially given evaluation indicators Expected Average Overlap (EAO), accuracy, and robustness to compare different trackers. When the overlap degree between the prediction box and the ground truth box is zero, we think that the tracker fails. In OTB100, only the initial frame of the video is used to initialize the tracker, and whenever the target is lost, subsequent frames will not be tracked. Moreover, whenever the target is followed and lost in the VOT2016 dataset, five frames are spaced, and the model is re-initialized. Robustness is used to measure the failure rate of the tracker and accuracy to calculate the overlap rate between the predicted and real boxes. EAO is a composite metric that takes into account the accuracy and robustness and is commonly used to rank trackers.

Listed in Table 1 are common trackers used in recent years, such as ROAM and SPM. Despite the great difficulties, the proposed tracker still achieves good scores on VOT2016, i.e., an accuracy of 0.638, robustness of 0.238, and an EAO of 0.414. Among all trackers, the proposed tracker achieved the best accuracy among those listed, and the third-highest EAO. We believe this result is closely related to our fusion of appearance and semantic features. Compared with SiamRPN, which achieves an accuracy of 0.56, robustness of 0.26, and an EAO of 0.344, the performance in these measures is improved by 7.8%, 2.2%, and 7%, respectively, which is a great improvement. Compared with C-RPN's EAO, that of the proposed tracker has increased by approximately 5%. We then compared the proposed tracker with DaSiamRPN and found that the proposed tracker increased by 2.8% in accuracy and by 0.3% in EAO, but that the failure rate increased by 1.8%.

Table 1: Comparisons on VOT-2016. The three best results are highlighted in red, green, and blue

Tracker	ROAM [45]	SPM [28]	DaSiam RPN [15]	ASRCF [46]	C-RPN [27]	Siam RPN [5]	UDT [47]	TADT [48]	SiamFC [4]	OURS
Accuracy	0.599	0.620	0.610	0.563	0.594	0.56	0.530	0.550	0.53	0.638
Robustness	0.174	0.210	0.220	0.187	-	-	-	-	0.46	0.238
EAO	0.441	0.434	0.411	0.391	0.363	0.344	0.301	0.299	0.235	0.414

5.3 *Evaluation Results on UAV123*

UAV123 is an aerial test dataset released in 2016 containing 123 video sequences obtained from low-altitude aerial photography, with an average sequence length of 915 frames. All sequences in the video are fully annotated with upright bounding boxes. The targets in this dataset mainly have challenges such as fast motion, large scale changes, long videos, and out-of-view targets. These all bring great challenges to tracking; so, the difficulty in testing this dataset is higher than that in testing the other datasets. We compare SiamCAR [18], SiamRPN [5], BACF [43], Staple [44], and SRDCF [49] all mainstream trackers, with the proposed tracker. As can be seen in Fig. 6, the proposed tracker achieved a success-plot score of 0.615 and a precision-plot score of 0.815. Compared with SiamCAR, the proposed tracker obtains a similar result in success plot and a decrease of 0.8% in precision plot. Compared with SiamRPN, it exhibits increases of 5.8% in success plot and of 4.7% in precision plot. The objects in the UAV123 dataset tend to exhibit fast motion and are small targets, and the proposed tracker is very good at dealing with such challenges. Because it adds coordinate attention, which retains the position and spatial information of the object, the proposed tracker has a better effect in dealing with this problem. The effect of the OTB100 dataset in single-challenge fast motion also demonstrates this point.

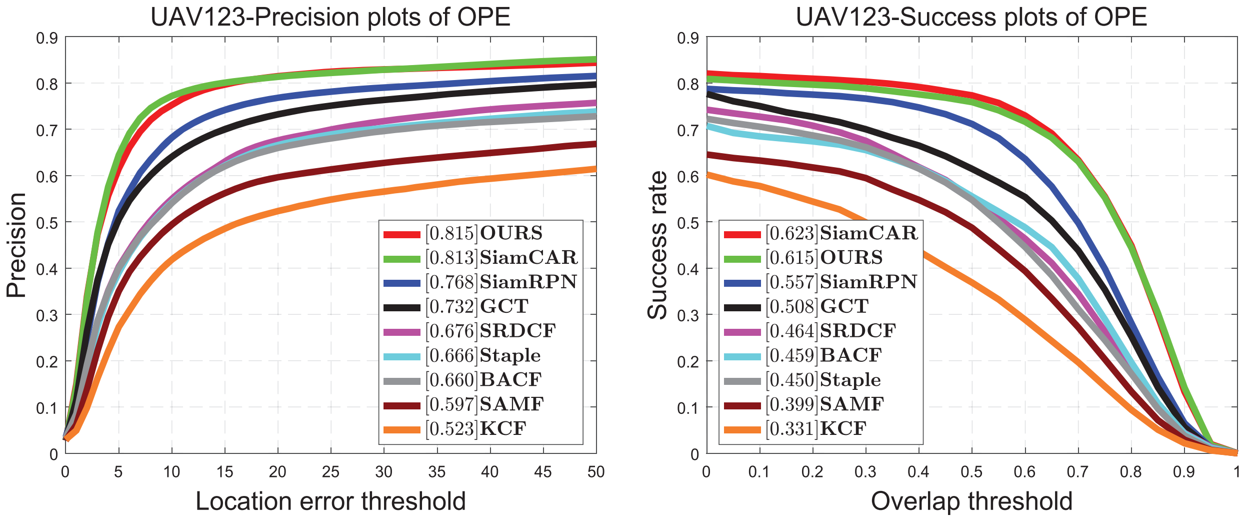


Figure 6: Comparison of the proposed tracker with state-of-the-art trackers on the UAV123 dataset on success and precision plots

5.4 Evaluation Results on GOT-10k

GOT-10k is a general dataset issued by the Chinese Academy of Sciences for object tracking in the field. It contains more than 10,000 videos and 560 categories, which are divided into training and testing datasets. The objects are real objects moving in the wild. To compare these deep trackers more fairly, GOT-10k stipulates that all trackers are trained using the same dataset. One must train the model on a given training dataset and then submit the model for testing on the official website. After uploading the model, the website automatically provides the tracking result. The evaluation indicator of this dataset is the average overlap (AO) and the success rate (SR). AO represents the average degree of overlap between the predicted and ground-truth bounding boxes. SR represents the proportion of frames that are successfully tracked, and $SR_{0.5}$ and $SR_{0.7}$ represent the proportions of frames with overlap ratios of successfully tracked frames exceeding 0.5 and 0.7, respectively. Frames per second (FPS) is used to measure the speed of the tracker.

Of the trackers listed in Table 2, the $SR_{0.5}$ and $SR_{0.75}$ of the proposed tracker both achieved the highest values. Its AO improved by nearly 2% compared to that of SiamRPN_R18, and it achieved a result approximately 5% higher than that of THOR. The proposed tracker's increases in $SR_{0.75}$ were the most, 3.8% higher than the second-place tracker, SPM. Because the proposed tracker joins the CA block and requires additional calculations, the speed will decrease. However, the FPS of the proposed tracker still reached 39.27, surpassing most Siamese trackers, and it can run in real time.

Table 2: Comparisons on GOT-10k. The best three results are highlighted in red, green, and blue

Tracker	AO	$SR_{0.5}$	$SR_{0.75}$	FPS	Hardware	Language
BACF [43]	0.260	0.262	0.101	14.44	CPU	Matlab
CFNet [25]	0.293	0.265	0.087	35.62	Titan X	Matlab
MDnet [50]	0.299	0.303	0.099	1.52	Titan X	Python
ECO [42]	0.316	0.309	0.111	2.62	CPU	Matlab
CCOT [51]	0.325	0.328	0.107	0.68	CPU	Matlab
SiamFC [4]	0.374	0.404	0.144	25.81	Titan X	Matlab
THOR [52]	0.447	0.538	0.204	1.00	RTX 2070	Python
SiamRPN_R18	0.483	0.581	0.270	97.55	Titan X	Python
SPM [28]	0.513	0.593	0.359	72.30	Titan Xp	Python
OURS	0.502	0.597	0.394	39.27	RTX 2080Ti	Python

5.5 *Ablation Studies*

To illustrate the effectiveness of the proposed innovations, we conducted a series of experiments on UAV123, and the results are shown in Table 3.

Table 3: Ablation studies of the proposed tracker on UAV123

Method	Success	Δs	Precision	Δp
Baseline	0.581	-	0.767	-
Baseline + RPN	0.597	+1.6%	0.783	+1.6%
Baseline + CA	0.605	+2.4%	0.804	+3.7%
Baseline + CA + RPN	0.615	+3.4%	0.815	+4.8%

Note: RPN and CA represent the RPN block and the CA block, respectively. Δs and Δp represent increases in success and precision, respectively.

Our baseline used the SiamRPN tracker upgraded from the backbone network to ResNet. This shows the performance of adding an RPN block and CA and adding both. RPN represents the RPN block added in the shallow position layer and CA the CA block. The structure of cascade RPN is formed by adding the RPN, and the results of the two regressions are fused. Compared with the baseline, it can be seen that adding an RPN leads to a 1.6% improvement in success plot and a 1.6% improvement in precision plot. The CA block not only pays attention to location information but also preserves channel information, which enhances feature representation. It can be seen that, compared to the baseline, adding CA achieved a 2.4% improvement in success plot and a 3.7% improvement in precision plot. Finally, we added both of these methods to obtain our tracker. It can be seen that the performance of the proposed tracker has been greatly improved as it exhibits increases of 3.4% in success plot and of 4.8% in precision plot.

6 Conclusion

In this paper, we propose an object-tracking algorithm combining cascaded RPN fusion and coordinate attention and use large-scale image pairs for end-to-end training. The coordinate attention embeds location information into channel attention while maintaining channel associations and uses location information to establish long-term spatial location dependence. A CA block is used to enhance the representation of features. We also added the RPN block in the shallow layers to form a cascaded RPN structure for fusing the processing results of different layers, to make the tracker more robust. In addition, we conducted a series of experiments on four datasets and achieved good performance. In particular, the proposed tracker exhibits good efficacy in dealing with fast motion and large-scale changes. In the future, we plan to improve the tracking network based on anchor freeness, because anchor-free tracking has fewer parameters, and the entire tracking process is concise. We believe that the cross-entropy loss function used for classification in the tracking network requires improvement, since it only classifies the foreground and background of the samples and does not take into account the positive and negative sample imbalance and the hard and easy sample imbalance that exist during the training process. Meanwhile, the intersection-over-union loss function used for regression only considers the area of the prediction and real frames, without paying attention to the distance between the two, along with the similarity of the shape, which is not comprehensive enough and must be improved.

Acknowledgement: We are grateful to our colleagues for their suggestions. We thank LetPub (www.letpub.com) for its linguistic assistance during the preparation of this manuscript.

Availability of Data and Materials: We used publicly available dataset in order to illustrate and test our methods. The OTB100 dataset can be found in http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html and the VOT2016 dataset can be found in https://www.votchallenge.net/vot2016/. The UAV123 dataset can be found in https://cemse.kaust.edu.sa/ivul/uav123 and the GOT-10k dataset can be found in http://got-10k.aitestunion.com/.

Funding Statement: This work was supported in part by the National Natural Science Foundation of China under Grants 61972056 and 61901061, the Science Fund for Creative Research Groups of Hunan Province under Grant 2020JJ1006, the Natural Science Foundation of Hunan Province under Grant 2020JJ5603, the Postgraduate Training Innovation Base Construction Project of Hunan Province under Grant 2019-248-51, the Basic Research Fund of Zhongye Changtian International Engineering Co., Ltd. under Grant 2020JCYJ07, and the Scientific Research Fund of Hunan Provincial Education Department under Grant 19C0031.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Frueh, C. (2016). Modeling impacts on space situational awareness PHD filter tracking. *Computer Modeling in Engineering & Sciences, 111(2),* 171–201. DOI 10.3970/cmes.2016.111.171.
2. Lee, K. H., Hwang, J. N. (2015). On-road pedestrian tracking across multiple driving recorders. *IEEE Transactions on Multimedia, 17,* 1429–1438. DOI 10.1109/TMM.2015.2455418.
3. Wu, Y., Lim, J., Yang, M. H. (2013). Online object tracking: A benchmark. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2411–2418. Portland, USA, DOI 10.1109/CVPR.2013.312.
4. Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., Torr, P. H. (2016). Fully-convolutional siamese networks for object tracking. *European Conference on Computer Vision*, pp. 850–865. Springer, Amsterdam, Netherlands, Cham. DOI 10.1007/978-3-319-48881-3_56.

5. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X. (2018). High performance visual tracking with siamese region proposal network. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8971–8980. Salt Lake City, USA. DOI 10.1109/CVPR.2018.00935.
6. Ren, S., He, K., Girshick, R., Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Neural Information Processing Systems, 28,* 91–99. DOI 10.1109/TPAMI.2016.2577031.
7. Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems, 25,* 1097–1105. DOI 10.1145/3065386.
8. He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778. Las Vegas, USA. DOI 10.1109/CVPR.2016.90.
9. Zhang, Z., Peng, H. (2019). Deeper and wider siamese networks for real-time visual tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4591–4600. Long Beach, USA. DOI 10.1109/CVPR.2019.00472.
10. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J. et al. (2019). SiamRPN++: Evolution of siamese visual tracking with very deep networks. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4282–4291. Long Beach, USA. DOI 10.1109/CVPR.2019.00441.
11. Hu, J., Shen, L., Sun, G. (2018). Squeeze-and-excitation networks. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141. Salt Lake City, USA. DOI 10.1109/TPAMI.2019.2913372.
12. Fu, J., Liu, J., Tian, H. (2019). Dual attention network for scene segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3146–3154. Long Beach, USA. DOI 10.1109/CVPR.2019.00326.
13. Hou, Q., Zhou, D., Feng, J. (2021). Coordinate attention for efficient mobile network design. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13713–13722. DOI 10.1109/CVPR46437.2021.01350.
14. Zhang, J., Liu, Y., Liu, H., Wang, J. (2021). Learning local–global multiple correlation filters for robust visual tracking with Kalman filter redetection. *Sensors, 21(4),* 1129. DOI 10.3390/s21041129.
15. Zheng, Z., Wang, Q., Li, B., Wu, W., Yan, J. et al. (2018). Distractor-aware siamese networks for visual object tracking. *European Conference on Computer Vision*, pp. 101–117. Munich, Germany. DOI 10.1007/978-3-030-01240-3_7.
16. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P. H. (2019). Fast online object tracking and segmentation: A unifying approach. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1328–1338. Long Beach, USA. DOI 10.1109/CVPR.2019.00142.
17. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W. et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. https://arxiv.org/abs/1704.04861.
18. Guo, D., Wang, J., Cui, Y., Wang, Z., Chen, S. (2020). SiamCAR: Siamese fully convolutional classification and regression for visual tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6269–6277. Seattle, USA. DOI 10.1109/CVPR42600.2020.00630.
19. Chen, Z., Zhong, B., Li, G., Zhang, S., Ji, R. (2020). Siamese box adaptive network for visual tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6668–6677. Seattle, USA. DOI 10.1109/CVPR42600.2020.00670.
20. Xu, Y., Wang, Z., Li, Z., Yuan, Y., Yu, G. (2020). SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines. *AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12549–12556. New York, USA. DOI 10.1609/aaai.v34i07.6944.
21. Zhang, J., Sun, J., Wang, J., Yue, X. G. (2021). Visual object tracking based on residual network and cascaded correlation filters. *Journal of Ambient Intelligence and Humanized Computing, 12(8),* 8427–8440. DOI 10.1007/s12652-020-02572-0.
22. Zhang, J., Feng, W., Yuan, T., Wang, J., Sangaiah, A. K. (2022). SCSTCF: Spatial-channel selection and temporal regularized correlation filters for visual tracking. *Applied Soft Computing*, *118,* 108485. DOI 10.1016/j.asoc.2022.108485.

23. Zhang, J., Jin, X., Sun, J., Wang, J., Li, K. (2019). Dual model learning combined with multiple feature selection for accurate visual tracking. *IEEE Access, 7,* 43956–43969. DOI 10.1109/ACCESS.2019.2908668.
24. Zhang, J., Jin, X., Sun, J., Wang, J., Sangaiah, A. K. (2020). Spatial and semantic convolutional features for robust visual object tracking. *Multimedia Tools and Applications, 79(21–22),* 15095–15115. DOI 10.1007/s11042-018-6562-8.
25. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P. H. (2017). End-to-end representation learning for correlation filter based tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2805–2813. Honolulu, USA. DOI 10.1109/CVPR.2017.531.
26. Long, J., Shelhamer, E., Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440. Boston, USA. DOI 10.1109/CVPR.2015.7298965.
27. Fan, H., Ling, H. (2019). Siamese cascaded region proposal networks for real-time visual tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7952–7961. Long Beach, USA. DOI 10.1109/CVPR.2019.00814.
28. Wang, G., Luo, C., Xiong, Z., Zeng, W. (2019). Spm-tracker: Series-parallel matching for real-time visual object tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3643–3652. Long Beach, USA. DOI 10.1109/CVPR.2019.00376.
29. Fan, H., Ling, H. (2020). CRACT: Cascaded regression-align-classification for robust visual tracking. https://arxiv.org/abs/2011.12483.
30. Zhang, J., Lu, C., Li, X., Kim, H., Wang, J. (2019). A full convolutional network based on densenet for remote sensing scene classification. *Mathematical Biosciences and Engineering, 16(5),* 3345–3367. DOI 10.3934/mbe.2019167.
31. He, S., Li, Z., Tang, Y., Liao, Z., Li, F. et al. (2020). Parameters compressing in deep learning. *Computers, Materials & Continua, 62(1),* 321–336. DOI 10.32604/cmc.2020.06130.
32. Guo, D., Yang, Q., Zhang, Y., Zhang, G., Zhu, M. et al. (2021). Adaptive object tracking discriminate model for multi-camera panorama surveillance in airport apron. *Computer Modeling in Engineering & Sciences, 129(1),* 191–205. DOI 10.32604/cmes.2021.016347.
33. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B. et al. (2017). Feature pyramid networks for object detection. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125. Honolulu, USA. DOI 10.1109/CVPR.2017.106.
34. Cao, Y., Xu, J., Lin, S., Wei, F., Hu, H. (2019). Gcnet: Non-local networks meet squeeze-excitation networks and beyond. *IEEE International Conference on Computer Vision Workshops*, Seoul, South Korea. DOI 10.1109/ICCVW.2019.00246.
35. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S. et al. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision, 115,* 211–252. DOI 10.1007/s11263-015-0816-y.
36. Real, E., Shlens, J., Mazzocchi, S., Pan, X., Vanhoucke, V. et al. (2015). Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5296–5305. Honolulu, USA. DOI 10.1109/CVPR.2017.789.
37. Lin, T. Y., Michael, M., Belongie, S. (2014). Microsoft coco: Common objects in context. *European Conference on Computer Vision*, pp. 740–755. Cham, Springer. DOI 10.1007/978-3-319-10602-1_48.
38. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R. et al. (2017). The visual object tracking vot2017 challenge results. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 1949–1972. Venice, Italy. DOI 10.1109/ICCVW.2017.230.
39. Wu, Y., Lim, J., Yang, M. H. (2013). Object tracking benchmark. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2411–2418. Portland, USA. DOI 10.1109/TPAMI.2014.2388226.
40. Mueller, M., Smith, N., Ghanem, B. (2016). A benchmark and simulator for uav tracking. *European Conference on Computer Vision*, pp. 445–461. Cham, Springer. DOI 10.1007/978-3-319-46448-0_27.

41. Huang, L., Zhao, X., Huang, K. (2019). GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43(5),* 1562–1577. DOI 10.1109/TPAMI.2019.2957464.

42. Danelljan M., Bhat, G., Shahbaz Khan, F., Felsberg, M. (2017). Eco: Efficient convolution operators for tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6638–6646. Honolulu, USA. DOI 10.1109/CVPR.2017.733.

43. Hamed, K. G., Fagg, A., Lucey, S. (2017). Learning background-aware correlation filters for visual tracking. *IEEE International Conference on Computer Vision*, pp. 1135–1143. Venice, Italy. DOI 10.1109/ICCV.2017.129.

44. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P. H. (2016). Staple: Complementary learners for real-time tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1401–1409. Las Vegas, USA. DOI 10.1109/CVPR.2016.156.

45. Yang, T., Xu, P., Hu, R., Chai, H., Chan, A. B. (2020). ROAM: Recurrently optimizing tracking model. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6718–6727. Seattle, USA. DOI 10.1109/CVPR42600.2020.00675.

46. Dai, K., Wang, D., Lu, H., Sun, C., Li, J. (2019). Visual tracking via adaptive spatially-regularized correlation filters. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4670–4679. Long Beach, USA. DOI 10.1109/CVPR.2019.00480.

47. Wang, N., Song, Y., Ma, C., Zhou, W., Liu, W. et al. (2019). Unsupervised deep tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1308–1317. Long Beach, USA. DOI 10.1109/CVPR.2019.00140.

48. Li, X., Ma, C., Wu, B., He, Z., Yang, M. H. (2019). Target-aware deep tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1369–1378. Long Beach, USA. DOI 10.1109/CVPR.2019.00146.

49. Danelljan, M., Hager, G., Shahbaz Khan, F., Felsberg, M. (2015). Learning spatially regularized correlation filters for visual tracking. *IEEE International Conference on Computer Vision*, pp. 4310–4318. Santiago, Chile. DOI 10.1109/ICCV.2015.490.

50. Nam, H., Han, B. (2016). Learning multi-domain convolutional neural networks for visual tracking. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4293–4302. Las Vegas, USA. DOI 10.1109/CVPR.2016.465.

51. Danelljan, M., Robinson, A., Khan, F. S. (2016). Beyond correlation filters: Learning continuous convolution operators for visual tracking. *European Conference on Computer Vision*, pp. 472–488. Cham, Springer. DOI 10.1007/978-3-319-46454-1_29.

52. Sauer, A., Aljalbout, E., Haddadin, S. (2019). Tracking holistic object representations. https://arxiv.org/abs/1907.12920.