Hindawi
Mobile Information Systems
Volume 2021, Article ID 6645629, 11 pages
https://doi.org/10.1155/2021/6645629

Research Article

Research on Target Tracking Algorithm Based on Siamese Neural Network

Haibo Pang,[1] Qi Xuan ,[1] Meiqin Xie,[1] Chengming Liu ,[1] and Zhanbo Li[2]

[1]*School of Software, Zhengzhou University, Zhengzhou 450002, China*
[2]*Network Management Center, Zhengzhou University, Zhengzhou 450001, China*

Correspondence should be addressed to Chengming Liu; cmliu@zzu.edu.cn

Received 18 December 2020; Revised 7 February 2021; Accepted 11 March 2021; Published 24 March 2021

Academic Editor: Xiaoxian Yang

Target tracking is a significant topic in the field of computer vision. In this paper, the target tracking algorithm based on deep Siamese network is studied. Aiming at the situation that the tracking process is not robust, such as drift or miss the target, the tracking accuracy and robustness of the algorithm are improved by improving the feature extraction part and online update part. This paper adds SE-block and temporal attention mechanism (TAM) to the framework of Siamese neural network. SE-block can refine and extract features; different channels are given different weights according to their importance which can improve the discrimination of the network and the recognition ability of the tracker. Temporal attention mechanism can update the target state by adjusting the weights of samples at current frame and historical frame to solve the model drift caused by the existence of similar background. We use cross-entropy loss to distinguish the targets in different sequences so that their distance in the feature domains is longer and the features are easier to identify. We train and test the network on three benchmarks and compare with several state-of-the-art tracking methods. The experimental results demonstrate that the algorithm proposed is superior to other methods in tracking effect diagram and evaluation criteria. The proposed algorithm can solve the occlusion problem effectively while ensuring the real-time performance in the process of tracking.

1. Introduction

Target tracking is a research hotspot and basic topic in digital image processing. It has important applications in many fields, such as military field, traffic monitoring, human-computer interaction, video monitoring, precision guidance, and so on [1, 2]. The task of target tracking is to predict the position and motion state of the target in the subsequent frames of video according to the motion trajectory and posture changes of the target given the target size and position in the initial frame of a video sequence [3]. Due to the change of target and environment information in the process of target tracking, the characteristics of target are changing constantly, and the problem that speed and accuracy requirements of target tracking is also discussed. There are several difficulties in target tracking, such as background clutter, deformation, scale variation, and occlusion. In addition to the above common challenges, there are other challenging factors such as illumination change, motion blur, rotation, out of view, and fast motion. All these challenges together determine that target tracking is a very complex task in computer vision [4]. In order to solve these practical problems, researchers have proposed many tracking methods in recent years.

Most of the methods are to solve the tracking problem by establishing the model, which can distinguish the target from the background. Because the specific information of target is available for tracking, it is difficult to learn the target model in the process of offline training, such as in target detection. On the contrary, the target model must be constructed by using the target information given during the test. The unconventional nature of the target tracking problem brings significant challenges when pursuing an end-to-end learning solution [5].

These problems have been solved by Siamese neural network successfully [6–8]. These methods learn a feature

embedding to calculate the similarity between two image regions through simple cross-correlation. Then, choose the image region that is the most similar to the template to be tracked. Because the model only corresponds to the template features extracted from the target area, the tracker can make use of the annotated images for end-to-end training easily. Although Siamese neural network has been successful in target tracking in recent years, there are still limitations seriously. Firstly, lacking of the offline training datasets can lead the measurement standard of similarity to have errors sometimes, resulting in the poor generalization. Secondly, Siamese neural network only uses the appearance of the target when inferring the target model but ignores the information of background appearance that is necessary to distinguish the target from similar objects. Thirdly, Siamese neural network lacks of a powerful model updating strategy. All these limitations make the robustness of Siamese neural network need to be improved [9].

The contributions in this paper are as follows:

First, we add the SE-block substructure [10] to the Siamese neural network, which can enhance feature representation of effective channels and improve feature discrimination by modeling the correlation between each channel of the image. Thereby, we can reduce the computational cost of extracting features.

Second, in order to solve the problem that the target is easy to be occluded in the tracking process, we add the temporal attention mechanism in Siamese neural network framework. Temporal attention mechanism can help the parameter to update of loss function by adjusting the weights of samples at current frame and historical frames.

Furthermore, we use the cross-entropy loss to distinguish the targets in different sequences of video, which makes the distance in the feature domain keep longer and the features have more resolution to classify the target and background.

For testing the effectiveness of the proposed algorithm, we perform comprehensive experiment on three benchmarks, respectively. The results demonstrate that the proposed approach can have a wonderful effect on three benchmarks which is superior to other contrast methods through the qualitative and quantitative evaluation. This paper can verify the feasibility of the network that we proposed and alleviate the problem of target occlusion effectively while ensuring real-time performance.

2. Related Work

2.1. Depth Models. The target tracking methods based on deep learning can track the target through the powerful representation ability of deep learning models. In 2012, the convolutional network AlexNet [11, 12] was first proposed and many networks based on convolutional were generated for target tracking subsequently, such as VGGNet [13], Google Inception Net [14], ResNet [15], and DenseNet [16]. The development of convolutional networks has solved a series of problems about gradient diffusion in the back-propagation process, and the extracted semantic information is more robust to larger changes. These models can have significant effects on target detection and recognition [17, 18] and image classification [19]. However, the effect of tracking is subtle due to the factors such as limited datasets and real-time performance.

According to the way of deep learning model feature extraction, target tracking can be divided into tracking based on pretrained deep features and tracking based on offline training deep features.

In target tracking based on pretrained depth models, the ImageNet [17] was the earliest way to extract features in 2013. Ma et al. proposed the HCF algorithm [20] to use VGG which integrated the shallow features and deep features in the network into the correlation filters in 2015. It showed a good experimental result, but the algorithm did not process the scale of target and assumed that the target scale is invariant in the tracking process which has far less robustness when tracking targets with large-scale changes. In 2016, Qi et al. used the Hedge algorithm [21] to improve the fusion of the correlation filters trained by each layer of features. Then, Danelljan et al. proposed the C-COT algorithm [22] that combined the deep semantic information and shallow appearance information to obtain a continuous spatial resolution response map by interpolating according to the response of different spatial resolutions and then found the optimal scale and position by iteration [23]. The C-COT algorithm can integrate the feature maps of different resolutions harmoniously. However, the disadvantage is that the amount of data in training is very large, which is easy to lose frames. In 2017, Danelljan et al. proposed the ECO [24] algorithm, which was improved by grouping samples, decomposing convolution factors, and updating strategies. It improved the speed of the algorithm while ensuring the accuracy of the algorithm. In 2018, Bhat et al. proposed the UPDT algorithm [25], which made a distinction between deep features and shallow features and made use of data enhancement and the difference response function to improve the accuracy and robustness of tracking effectively and proposed a quality evaluation method concurrently to self-adaptation and fuse the response map to further optimize the tracking effect. The deep learning model based on pretraining requires less training data that can be used for target detection directly. However, the model is larger, the parameters are more, and the model structure is not flexible which leads to a large amount of calculation.

The methods of target tracking based on the offline training depth model can achieve good tracking results through the end-to-end training features matching the tracking task. Nam and Han et al. proposed the MDNet algorithm [26] in 2016, which learned convolutional features to represent the target by a lightweight small-scale network and used SoftMax classifier [27] to classify the samples that sampled which had good tracking performance, but the speed of tracking ought to be better. The deep learning model based on offline training can achieve higher precision with less parameters, which can speed up the convergence while reducing the number of parameters.

2.2. Siamese Neural Network. Siamese neural network belongs to the deep learning model of offline training. Bertinetto et al. proposed the Siamese-FC algorithm [28] that solved the more general similarity learning problem by training a depth network in the initial offline stage and trained a fully convolutional Siamese network to locate candidate regions in larger search images. This algorithm performed well in real-time, but the accuracy is not as good as the correlation filtering method combined with depth features. Tao et al. made improvements on this foundation and proposed the SINT algorithm [29], generated multiple candidate regions in images, learned the matching function of the candidate regions and the target templates in Siamese neural network, and then selected the candidate region with the smallest difference as template for online tracking which transmuted the tracking problem into a matching problem for the first time. However, the process of processing large number of candidate regions was cumbersome and time-consuming. In 2018, Li et al. used the region proposal network (RPN) [30] based on the Siamese-FC algorithm to replace bounded box regression with multiscale detection for obtaining the bounding box with maximum response which can improve the efficiency and performance of tracking, but the feature extraction capability of the convolutional layer remained to be improved. However, most of the Siamese networks that mentioned above are based on shallow networks, while the deeper networks are prone to position errors due to filling.

2.3. SE-Block. SE-block is a substructure that consists of squeeze and excitation, and it is remarkable that the SE-block does not belong to integrity network structure. SE-block is to learn the feature weights according to the loss of network so that the effective feature weight becomes larger and the little effect or invalid feature weight becomes smaller and enhance the image by effective channels [23] which makes the input image frame enhance the effective features extracted by using the channel correlation while considering the spatial feature information fully in order to make the training model achieve better results.

2.4. Temporal Attention Mechanism. Attention mechanism is an important concept in neural network, which has been used widely in different fields, especially in image recognition, image processing, and NLP [31]. The attention mechanism in deep learning is to focus attention on the key point, obtain the key information, and ignore other useless information. Most attention mechanism models are based on encoder-decoder framework. The framework is shown in Figure 1.

In Figure 1, we give input x, and target y is generated by encoder-decoder framework. The encoder encodes input x and transforms input into the intermediate semantics, which is represented by c through nonlinear transformation. The decoder generates information of target according to semantic representation c of input x and generated historical information previously. So, encoder-decoder framework is regarded as a general framework; encoder and decoder can use various model combinations, such as CNN, RNN, LSTM, and GRU.

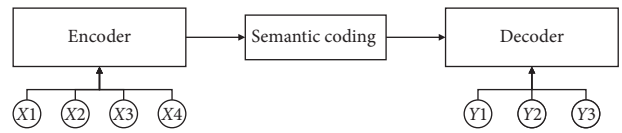


Figure 1: Encoder-decoder framework.

Many approaches are proposed for handing occlusion but have received only limited acclaim [12, 18, 32]. In this paper, temporal attention mechanism is introduced to handle occlusion. In detail, the temporal attention mechanism is used to update the target state by adjusting the weight of loss from training samples at current frame and historical frame.

3. Tracking Network

3.1. Construction of Network. This paper proposes the network that based on the Siamese neural network, which can improve the speed and accuracy and handle occlusion of target tracking. The training of the network is offline through the end-to-end way. The structure of our network is shown in Figure 2.

The structure of our network is composed of two processes, one is the feature extraction operation in Siamese neural network and another is using the positive and negative samples at current and historical frame to update the target state with the help of the temporal attention mechanism. The target is generally the bounding box given by the first frame, we adopt the exemplar images whose size is 127×127 pixels after preprocessing, the search images mean the candidate box search region in the frames to be tracked later, and the size of the search images is 255×255 pixels after preprocessing [23]. SE-block is added after conv5 of the network to form the SE-CNN structure, which can make full use of the channel and spatial information of the image to enhance the effectiveness of the channel features and improve the effect of feature extraction. SE-CNN is used to extract the features and then weighted of exemplar images and search images. The state of the search image with the maximum classification score is used as the estimated target state. Then, we collect the positive and negative training samples at current frame according to the overlap with the estimated target state. The positive training samples at historical frame are also used for updating the target state. Temporal attention mechanism actually reflects the weight of the estimated target state in the total loss when we update the parameters online. The total loss is composed of positive and negative samples at current frame and positive sample of historical frame, and the model is trained using cross-entropy loss.

3.2. Network Improvement. In this paper, the network uses the AlexNet structure [10] that includes five convolution layers and three full connection layers to extract features. The convolution kernel of conv1 is 11×11 pixels, conv2 is 5×5 pixels, and conv3–5 is 3×3 pixels, respectively. Kernel

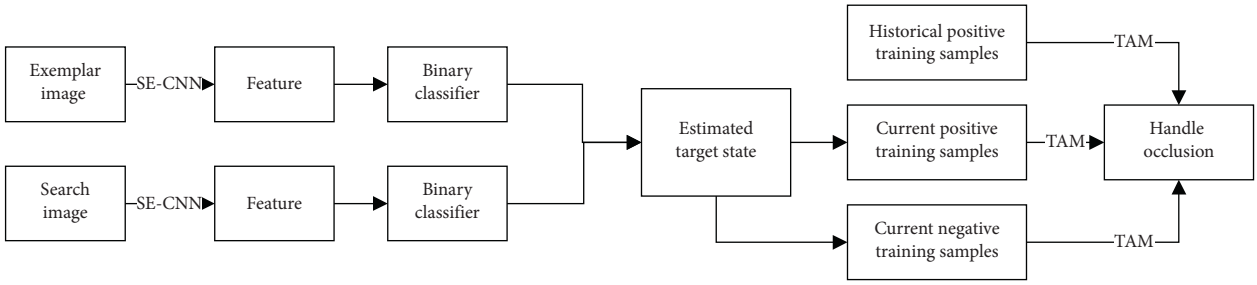


Figure 2: Network architecture.

size is 3, and stride is 2. The network adopts the Max-pooling, and there is a ReLu (rectified linear unit) nonlinear activation function after each convolution layer except the conv5. Adding SE-block after the conv5, the normalization layer prevents the data distribution from changing in order to reduce the risk of overfitting in the training process.

The first measure of our improvements is to embed the SE-block after the conv5 to form SE-CNN module in this paper [23]. The SE-block consists of squeeze and excitation. The squeeze operation reduces the dimension of features, turns each two-dimensional feature channel into a real number, which has a global receptive field to some extent, and matches the output dimension with the number of input feature channels, representing the global distribution of responses on feature channels. The excitation operation generates weights for each channel through the correlation between the feature channels, and the weight means the importance of each feature channel after feature selection. The reweight operation uses multiplication to weight the feature channels to the original features one by one and completes the recalibration of the original features in the channel dimension.

The details of the improvement are shown in Figure 3.

The second improvement is making full of the temporal attention mechanism to pay attention to historical and current samples based on occlusion status. Encoder-decoder framework can give different influences (i.e., weight) to the positive and negative samples of video frames in different time and extract the key frames and their information we contained that may be useful for tracking which make the model be more accurate on judgment of target tracking without increasing the cost of calculation and storage. The historical sample is the reliable and positive sample collected at historical frame, and the sample at current frame reflects the state change of the target.

3.3. Tracking Strategy. In essence, the tracking strategy can be divided into the following four parts roughly:

Feature Extraction. SE-CNN is used as a feature extractor to extract the features of the input images and the search images

Binary Classification. The feature extracted from SE-CNN is input into the binary classifier, and the output indicates the probability of candidate state belonging to the estimated target state, that is the classification score

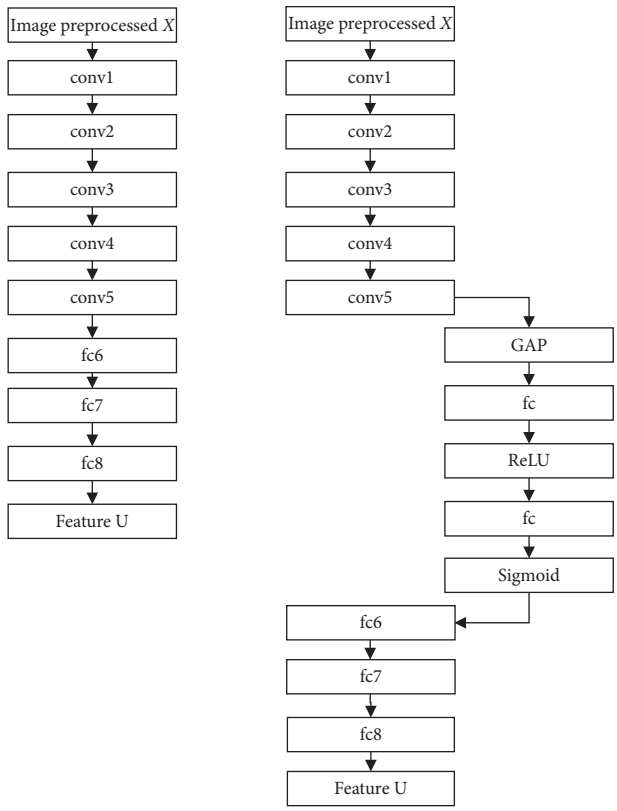


Figure 3: Diagram of improvement detail.

Estimated Target State. After comparing the classification score, the candidate state with the maximum score is selected as the estimated target state

Handle Occlusion. We obtain the training samples from current frame and historical frame. Temporal attention mechanism is to balance the relative importance between current and historical visual cues based on occlusion status

3.4. Algorithm Process. The process of the algorithm in this paper is shown in Figure 4.

3.4.1. Image Preprocessing. The exemplar images and the search images are "modified" to a fixed size. Specifically, it includes padding, cutting, and scaling, and these processes cannot damage the information on the size of the object to

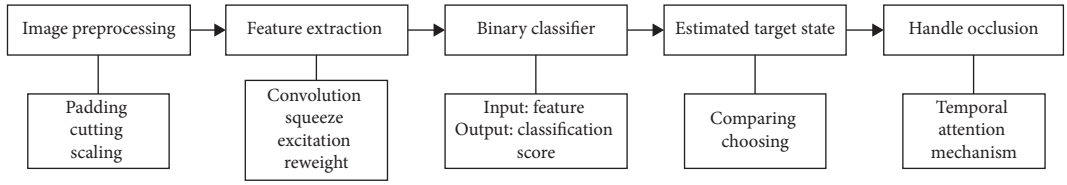


Figure 4: Algorithm process diagram.

make the target which is manually labeled to be at the center of the image [23].

3.4.2. Feature Extraction. The exemplar and search images after preprocessing are input into the convolution layer in pair for convolution operation. Assuming that the input image is $X \in R^{W' \times H' \times C'}$ and the output feature map is $U \in R^{W \times H \times C}$, the formula of the convolution operation is as follows:

$$u_c = v_c * X = \sum_{s=1}^{c} v_c^s * X^s, \tag{1}$$

where v_c means the c – th convolution kernel, X^s means the s – thinput, and u_c means the receptive field of the feature map in the c – th channel.

Then, the feature map is to squeeze operation after GAP (global average pooling), which is written as $F_{\text{sq}}(\cdot)$. In order to express the global information of the feature map, we transform the feature map from the input of $H \times W \times C$ to the output of $1 \times 1 \times C$, as shown in the following:

$$Z_c = F_{\text{sq}}(u_c) = \frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} u_c(i, j). \tag{2}$$

Next, the feature is to the excitation operation, which is denoted as $F_{\text{ex}}(Z, W)$, as shown in the following:

$$s = F_{\text{ex}}(Z, W) = \sigma\left(W_2 \delta\left(W_1 Z\right)\right), \tag{3}$$

where ReLu is a nonlinear activation function, $\sigma(\cdot)$ means the sigmoid function, Z means the result of squeeze operation, W_1 and W_2 mean the parameters of two full connection layers, respectively, the two full connection layers are used to fuse the feature map information of each channel, and s means the weight of feature maps in different channels that is set as $\omega_i (i = 1, 2, 3, 4, 5)$. These weights are learned by the full connection layers and the nonlinear layers, so they can be trained by end to end [23].

Finally, the reweight operation is performed, and the weights that output are recalibrated in the original image, corresponding to the following:

$$X_c = F_{\text{scale}}(u_c, s_c) = s_c \cdot u_c, \tag{4}$$

where$s_c$means the weight and $u_c$is a two-dimensional matrix. We give the different weights to different channels. The network can not only strengthen the effective channels according to their importance but also improve the characterization ability of feature after the above improvements [23].

3.4.3. Binary Classifier. Given the refined feature representation $\Phi_{\text{att}}(X_{t,j}^i)$, the classification score is obtained as follows:

$$p_{t,j}^i = f_{\text{cls}}\left(\Phi_{\text{att}}\left(X_{t,j}^i\right); \omega_{\text{cls}}^i\right), \tag{5}$$

where $p_{t,j}^i \in [0, 1]$ is the output of binary classifier that represents the probability of whether the candidate state $X_{t,j}^i$ is the target T^i and ω_{cls}^i is the parameter of the classifier for target T^i.

3.4.4. Estimated Target State. The initial state of target T^i is estimated by choosing the candidate state with the maximum classification score as follows:

$$\widehat{X}_t^i = \arg\max f_{\text{cls}}\left(\Phi_{\text{att}}\left(X_{t,j}^i\right); \omega_{\text{cls}}^i\right). \tag{6}$$

It is worth noting that the initial estimated state with too small classification score will lead to deviation to the updating of the model. To avoid model degeneration, we set a threshold if the score is lower than the threshold, which represents that the target is not tracked at current frame. Otherwise, the initial state $\widehat{X}_t^i$ will be further refined using the object detection states $D_t = \left\{X_{t,m}^d\right\}_{m=1}^M$. In detail, the nearest detection state for $\widehat{X}_t^i$ is obtained as follows:

$$X_t^{d,i} = \arg\max \text{IoU}\left(\widehat{X}_t^i, X_{t,m}^d\right), \tag{7}$$

where $\text{IoU}(\widehat{X}_t^i, X_{t,m}^d)$ calculates the bounding box IoU overlap ratio between $\widehat{X}_t^i$ and $\widehat{X}_{t,m}^d$. Then, the final state of target T^i is defined as follows:

$$X_t^i = \begin{cases} o_t^i X_t^{d,i} + \left(1 - o_t^i\right) \widehat{X}_t^i, & o_t^i > o_0, \\ \widehat{X}_t^i, & \text{otherwise}, \end{cases} \tag{8}$$

where $o_t^i = \text{IoU}(\widehat{X}_t^i, X_t^{d,i})$and o_0 is a predefined threshold.

3.4.5. Handle Occlusion. The training samples for online updating are obtained from the current frame and historical states. For the target that be tracked, positive samples are sampled at current frame t with scale variations and small displacement around the estimated target state X_t. In addition, the historical states are also used as positive samples. If the target is considered untracked at current frame, we only use the historical states of the target as positive sample. All negative samples are collected at current frame t. The target-specific branches require the ability to discriminating the target that we tracked from other targets and background. Therefore, the estimated status of other tracking

targets and samples sampled randomly from the background are regarded as the negative samples.

For target T^i, given the current positive samples set $\{X_{t,j}^{i+}\}_{j=1}^{N_t^{i+}}$, the current negative samples set $\{X_{t,j}^{i-}\}_{j=1}^{N_t^{i-}}$, and positive samples set from historical set $\{X_{h,j}^{i+}\}_{j=1}^{N_h^{i+}}$, the function of loss for updating corresponding target-specific branch is defined as follows:

$$L_t^i = L_t^{i-} + \left(1 - \alpha_t^i\right) L_t^{i+} + \alpha_t^i L_h^{i+}, \tag{9}$$

$$L_t^{i-} = -\frac{1}{N_t^{i-}} \sum_{j=1}^{N_t^{i-}} \log\left[1 - f_{\text{cls}}\left(\Phi_{\text{att}}\left(X_{t,j}^{i-}\right); \omega_{\text{cls}}^i\right)\right], \tag{10}$$

$$L_t^{i+} = -\frac{1}{N_t^{i+}} \sum_{j=1}^{N_t^{i+}} \log\, f_{\text{cls}}\left(\Phi_{\text{att}}\left(X_{t,j}^{i+}\right); \omega_{\text{cls}}^i\right), \tag{11}$$

$$L_h^{i+} = -\frac{1}{N_h^{i+}} \sum_{j=1}^{N_h^{i+}} \log\, f_{\text{cls}}\left(\Phi_{\text{att}}\left(X_{h,j}^{i+}\right); \omega_{\text{cls}}^i\right), \tag{12}$$

where L_t^{i-} is the loss from negative samples at current frame, L_t^{i+} is the loss from positive samples at current frame, L_h^{i+} is the loss from positive samples in the history, and α_t^i is introduced by the temporal attention mechanism.

In order to alleviate the problem of target occlusion in the process of tracking, we introduce the temporal attention mechanism. The temporal attention of target T^i is defined by feature weighted $U(X_t^i)$ and the overlap statuses with other targets as follows:

$$\alpha_t^i = \sigma\left(\gamma^i s_t^i + \beta^i o_t^i + b^i\right), \tag{13}$$

where γ^i, β^i, and b^i are parameters that are learnable, s_t^i is the mean value of feature weighted $U(X_t^i)$, α_t^i represents the occlusion status of target T^i, the larger the value is, the more seriously the target is occluded and the smaller the weight of positive samples is at current frame, o_t^i is the maximum overlap between T^i and the other targets at current frame t, and $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function. Therefore, we add the temporal attention mechanism to our network that provides a good balance between the current and historical visual cues of the target.

4. Experience

4.1. Experimental Setup. The experiment in this paper is based on the PyTorch framework to build and train the convolutional neural network. In terms of model training, this experiment uses the GeForce RTX 2080ti GPU and the 2.4 GHz CPU to iterate 50 times [23]. The first 20 iterations only train the feature extraction network and the last 30 iterations train the whole network which means whether the location found by object tracker is covered by object detection. In terms of parameter setting, the SGD optimizer [33–35] is used to optimize the loss function of the network and update to the network weights in order to avoid affecting the speed of tracking. Meanwhile, the algorithm parameters are set as follows: the training batch size is set to 16, the warmup learning rate mechanism is adopted, the initial learning rate for the first 20 iterations is 0.001, the learning rate for the last 30 iterations is 0.005, which decreased to 0.0005 (weight decay), the speed of test sequences is 0.5 fps, and the momentum is 0.9. We collect positive with ≥0.7 and negative samples with ≤0.3 IoU overlap ratios with the target state at current frame [36].

4.2. Experimental Process. The experiment process in this paper is shown in Figure 5, which is mainly divided into the following processes.

Firstly, SE-network and temporal attention mechanism are introduced on the framework of the Siamese-FC algorithm to debug the code. And dataset is selected for training and testing, the training data are preprocessed, the code is implemented to improve the algorithm, and the tracking model is trained. Then, the trained model is used to conduct experiments in the dataset, and the results were evaluated. Finally, other advanced tracking algorithms are tested andthe results are compared with that of the algorithm we proposed.

4.3. Qualitative Evaluation. For evaluating the effectiveness of the proposed algorithm, we train and test network on the public available GOT10k benchmark [37] in unconstrained environments. It includes more than 10000 videos in 563 categories. The test video sequences include many interference factors such as rotation, occlusion, light change, direction change, and scale transformation which are helpful to verify the practical value of the algorithm that we proposed. C-COT [22], ECO [24], UPDT [25], MDNet [26], and CFNet [38] are selected to compare with our algorithm on this benchmark. All the compared state-of-the-art algorithms including ours use the same parameters during testing for fair comparison.

This paper shows some experimental results of six algorithms on GOT10k. Boxes with different colors represent the tracking results of different algorithms, and the algorithm that we proposed is represented with red box. Qualitative evaluation of the algorithm is carried out from the following five aspects so as to show the tracking effect better than other algorithms to a certain extent, as shown in Figure 6.

(1) *Target Rotation.* In "000577" and "005501," the direction of the target we tracked has changed dramatically, which makes other five algorithms track failure but our algorithm can track the target accurately.

(2) *Motion Blur.* For video sequences "003867" and "006037," motion blurs due to fast moving of the target or camera shaking, which result in the algorithms that compared have drift, but our algorithm is not affected.

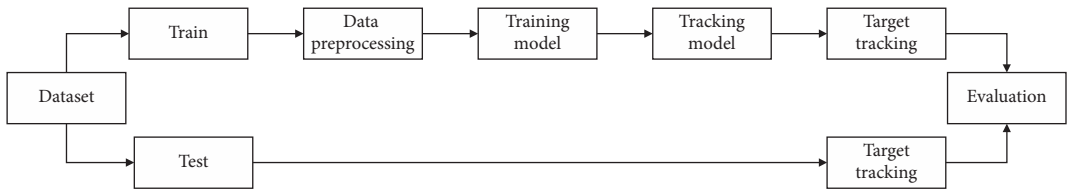


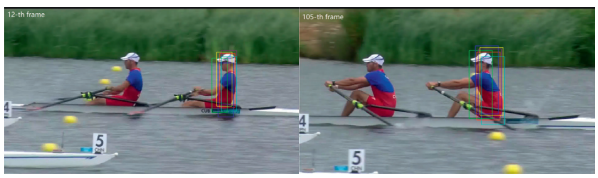
Figure 5: Experimental process diagram.

: Ours : MDNet
: ECO : CFNet
: C-COT : UPDT

(a)

: Ours : MDNet
: ECO : CFNet
: C-COT : UPDT

(b)

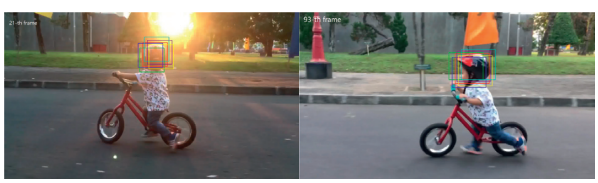


: Ours : MDNet
: ECO : CFNet
: C-COT : UPDT

(c)

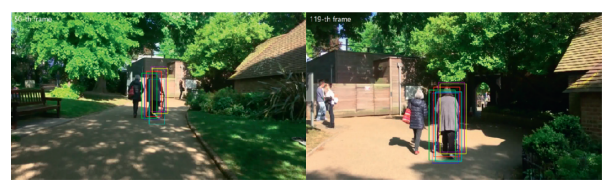
: Ours : MDNet
: ECO : CFNet
: C-COT : UPDT

(d)



: Ours : MDNet
: ECO : CFNet
: C-COT : UPDT

(e)



: Ours : MDNet
: ECO : CFNet
: C-COT : UPDT

(f)

: Ours : MDNet
: ECO : CFNet
: C-COT : UPDT

(g)

: Ours : MDNet
: ECO : CFNet
: C-COT : UPDT

(h)

Figure 6: Continued.

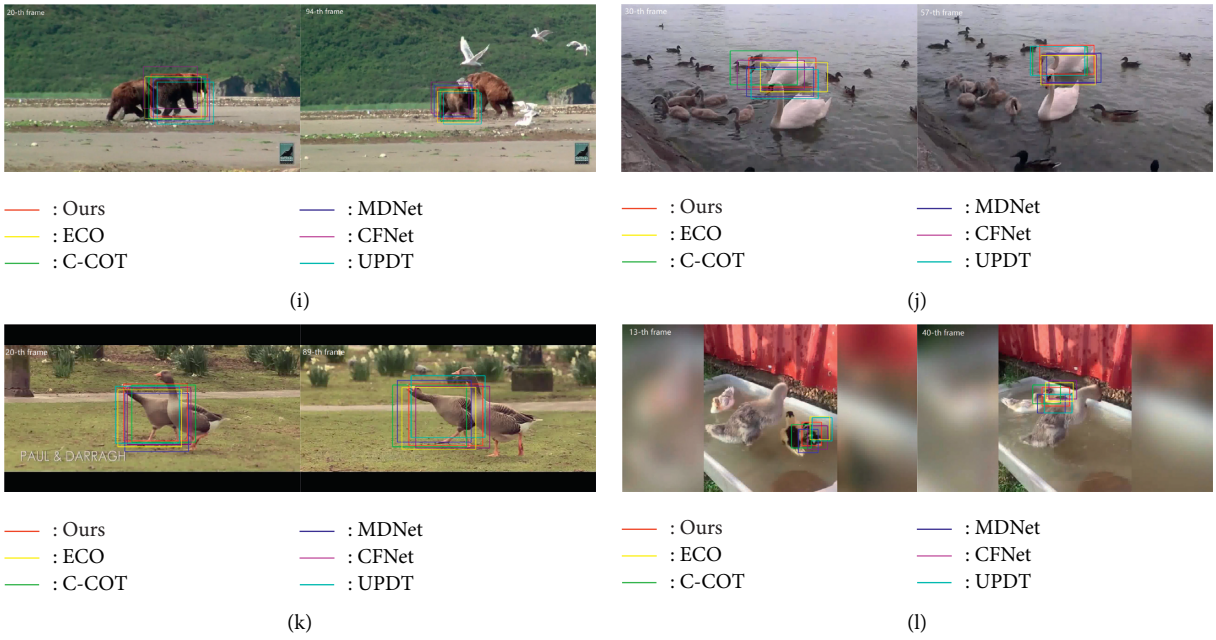


Figure 6: Tracking results of different algorithms on GOT10k: (a) 000577; (b) 005501; (c) 003867; (d) 006037; (e) 000047; (f) 006504; (g) 000492; (h) 000501; (i) 000496; (j) 000505; (k) 000507; (l) 000510.

(3) *Illumination Change*. For video sequences "000047" and "006504," the illumination changes dramatically in the process of tracking, which requires the algorithm to be robust to the influence of illumination. In video sequences "006504," the contrast algorithm fails one after another and only our algorithm can track the target accurately after the 119-th frame when the illumination changes dramatically.

(4) *Complex Background*. For video sequences "000492" and "000501," complex background has great challenge to the tracking accuracy of the algorithms. In addition to our algorithm, the comparison algorithms are interfered by complex background which lead to loss the target in "000492." In "000501," the comparison algorithms have different degrees of drifts except our algorithm from the 10-th frame, and our algorithm can track the target accurately.

(5) *Occlusion*. For video sequences "000496," "000505," "000507," and "000510," the target appears occluded in different degrees in the process of tracking. In "000510," the target is occluded by several animals seriously but only our algorithm can track the target correctly.

4.4. Quantitative Evaluation. For demonstrating the effectiveness of the algorithm objectively and comprehensively, we compare our proposed algorithm with several advanced tracking methods on three challenging benchmarks.

GOT10k [37]: It is a large-scale benchmark including over 10,000 videos. Our algorithm is compared with C-COT [22], ECO [24], MDNet [26], SiamFCv2 [38], CF2 [39], GOTURN [40], and SiamFC [28] choosing AO (average overlap) and SR (success rates) as the evaluation criteria. Results are shown in Table 1.

VOT2018 [41]: It is a benchmark consisting of 60 videos. Our algorithm is compared with UPDT [25], MFT [41], ATOM [42], DiMP [43], DRT [44], RCO [45], and LADCF [46] choosing accuracy (average overlap over successfully tracked frames) and EAO (expected average overlap) [47] as the evaluation criteria. Results are shown in Table 2.

OTB2015 [48]: It is a benchmark that includes over 100 videos. Our algorithm is compared with C-COT [22], UPDT [25], MDNet [26], SiamFC [28], CF2 [43], ADNet [49], and CREST [50] choosing OPE (one pass evaluation) as the evaluation criteria [51]. Results are shown in Figure 7.

Table 1: Results of AO and SR.

	ECO	C-COT	MDNet	SiamFCv2	CF2	GOTURN	SiamFC	Ours
AO	31.6	32.5	29.9	37.4	31.5	34.7	34.8	39.6
SR	30.9	32.8	30.3	40.4	29.7	37.5	35.3	42.9

Table 2: Results of accuracy and EAO.

	UPDT	RCO	DRT	MFT	LADCF	ATOM	DiMP	Ours
Accuracy	0.536	0.507	0.519	0.505	0.503	0.590	0.594	0.599
EAO	0.378	0.376	0.356	0.385	0.389	0.401	0.402	0.407

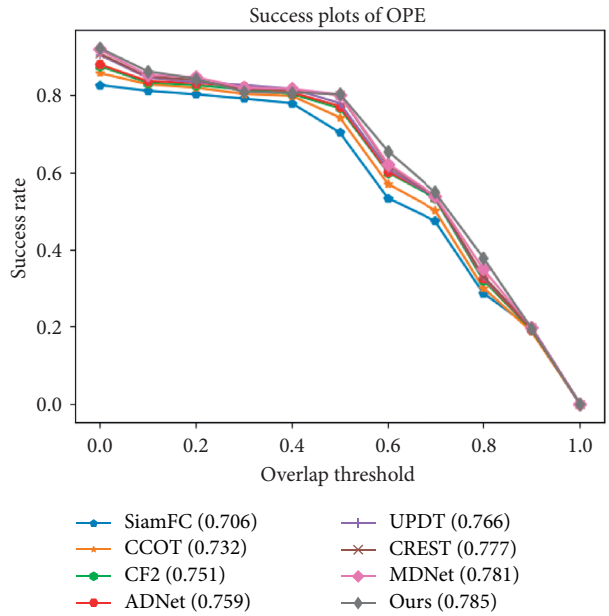


Figure 7: Results of OPE.

5. Conclusions

This paper uses the Siamese neural network as the research framework and adds SE-block and TAM to the network. SE-CNN can make full use of spatial feature information and channel correlation and make the extracted feature weights change according to contribution which is equivalent to a channel attention mechanism. TAM can update the target state by adjusting the weight changes of samples at current frame and historical frames. The experimental results show that the proposed algorithm has good robustness in the application of target tracking, which can satisfy the real-time requirements of tracking and alleviate the problem of occlusion effectively. However, there is still a problem of deviation because the speed is too fast of the target in some video sequences. How to solve this problem is the focus of the next research. We should do further research on this problem.

Data Availability

The data used to support the findings of this study are available at https://http://www.votchallenge.net/vot2018/dataset.html; http://got-10k.aitestunion.com/downloads; and http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the Young Backbone Teacher Project of the Zhengzhou University (2019ZDGGJS029): Research on Moving Object Modeling in Video Tracking; Online Excellent Course Project of Zhengzhou University in 2017: Digital Image Processing; Research and practice project of education and teaching reform in 2019 of Zhengzhou University (2019ZZUJGLX224): Research and practice on the training mode of innovation and the entrepreneurship talents of information security specialty; and Offline Excellent Course Project of Zhengzhou University in 2019 (2019ZZUXXKC023): Digital Image Processing.

References

[1] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan, "Interacting multiple model methods in target tracking: a survey," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, pp. 103–123, 1998.

[2] H. Gao, X. Qin, R. J. D. Barroso et al., "Collaborative learning-based industrial IoT API recommendation for software-defined devices: the implicit knowledge discovery perspective," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, 2020.

[3] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part I. Dynamic models," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.

[4] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part V. Multiple-model methods," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1255–1321, 2005.

[5] G. Bhat, M. Danelljan, L. Van Gool, and T. Radu, "Learning discriminative model prediction for tracking," *IEEE Computer Vision Foundation*, vol. 69, no. 2, pp. 79–91, 2020.

[6] J. Bromley, I. Guyon, Y. LeCun et al., "Signature verification using a "siamese" time delay neural network," in *Proceedings of the Advances in Neural Information Processing Systems ACM*, pp. 737–744, Colorado, USA, December 1994.

[7] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, pp. 539–546, San Diego, CL, USA, June 2005.

[8] Y. Taigman, M. Yang, M. A. Ranzato et al., "Deepface: Closing the Gap to Human-Level Performance in Face verification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701–1708, Columbus, OH, USA, November 2014.

[9] M. Kristan, A. Leonardis, J. Matas et al., "The sixth visual object tracking vot2018 challenge results," *ECCV Workshop*, vol. 2, p. 7, 2018.

[10] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation network," arXiv preprint https://arxiv.org/abs/1709.01507, 2017.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1097–1105, Lake Tahoe, ND, USA, December 2012.

[12] H. Izadinia, I. Saleemi, W. Li, and M. Shah, "Multiple people multiple parts tracker," in *Proceedings of the 12th European Conference on Computer Vision*, Florence, Italy, October 2012.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, https://arxiv.org/abs/1409.1556.

[14] C. Szegedy, V. Vanhoucke, S. Ioffe et al., "Rethinking the Inception Architecture for Computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, Las Vegas, ND, USA, December 2016.

[15] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, ND, USA, July 2016.

[16] G. Huang, Z. Liu, L. Van Der Maaten et al., "Densely connected convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, Honolulu, HI, USA, July 2017.

[17] N. Wang and D. Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 809–817, Lake Tahoe, ND, USA, December 2013.

[18] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, "Part based multiple-person tracking with partial occlusion handling," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, December 2012.

[19] H. Gao, W. Huang, and Y. Duan, "The cloud-edge-based dynamic reconfiguration to service workflow for mobile ecommerce environments," *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1–23, Article ID 6, 2021.

[20] C. Ma, J. B. Huang, X. Yang et al., "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3074–3082, Las Condes, CH, USA, June 2015.

[21] Y. Qi, S. Zhang, L. Qin et al., "Hedged Deep tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4303–4311, Las Vega, ND, USA, June 2016.

[22] M. Danelljan, A. Robinson, F. S. Khan et al., "Beyond correlation filters: learning continuous convolution operators for visual tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 472–488, Amsterdam, Netherlands, August 2016.

[23] H. Pang, X. Qi, M. Xie, C. Liu, and Z. Li, "Target tracking based on siamese convolution neural networks," in *Proceedings of the 2020 International Conference of the IEEE Conference on Computer, Information and Telecommunication Systems (CITS)*, Hangzhou, China, October 2020.

[24] M. Danelljan, G. Bhat, F. Shahbaz Khan et al., "Eco: Efficient Convolution Operators for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6638–6646, Honolulu, HI, USA, July 2017.

[25] G. Bhat, J. Johnander, M. Danelljan et al., "Unveiling the power of deep tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 483–498, Munich, Germany, September 2018.

[26] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4293–4302, Las Vegas, ND, USA, June 2016.

[27] S. Kumarawadu, K. Watanabe, K. Kiguchi et al., "Adaptive output tracking of partly known robotic systems using SoftMax function networks," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, pp. 483–488, Honolulu, HI, USA, July 2002.

[28] L. Bertinetto, J. Valmadre, J. F. Henriques et al., "Fully-convolutional Siamese Networks for Object tracking," in *Proceedings of the European Conference on Computer Vision*, pp. 850–865, Amsterdam, The Netherlands, October 2016.

[29] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1420–1429, Las Vegas, ND, USA, June 2016.

[30] B. Li, J. Yan, W. Wu et al., "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8971–8980, Salt Lake City UT, USA, June 2018.

[31] H. Gao, C. Liu, Y. Li, and X. Yang, "V2VR: reliable hybrid-network-oriented V2V data transmission and routing considering RSUs and connectivity probability," *IEEE Transactions on Intelligent Transportation Systems*, vol. 42, 2020.

[32] S. Tang, M. Andriluka, and B. Schiele, "Detection and tracking of occluded people," *International Journal of Computer Vision*, vol. 110, no. 1, pp. 58–69, 2014.

[33] Y. Jia, E. Shelhamer, J. Donahue et al., "Convolutional architecture for fast feature embedding," 2013, https://arxiv.org/abs/1408.5093.

[34] N. Wang, S. Li, A. Gupta et al., "Transferring rich feature hierarchies for robust visual tracking," 2015, https://arxiv.org/abs/1501.04587.

[35] L. Wang, W. Ouyang, X. Wang et al., "Visual tracking with fully convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3119–3127, Las Condes, CH, USA, December 2015.

[36] H. Gao, L. Kuang, Y. Yin, B. Guo, and K. Dou, "Mining consuming behaviors with temporal evolution for personalized recommendation in mobile marketing apps," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1233–1248, 2020.

[37] L. Huang, X. Zhao, and K. Huang, "Got-10k: a large high-diversity benchmark for generic object tracking in the wild," 2018, https://arxiv.org/abs/1810.11981.

[38] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, H. Philip, and S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proceedings of the 2017 Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, July 2017.

[39] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the 2015 International Conference on Computer Vision*, Las Condes, CH, USA, December 2015.

[40] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," in *Proceedings of the European Conference on Computer Vision*, Amsterdam, The Netherlands, October 2016.

[41] M. Kristan, A. Leonardis, J. Matas et al., "The sixth visual object tracking vot2018 challenge results," in *Proceedings of the European Conference on Computer Vision Workshop*, Munich, Germany, September 2018.

[42] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: accurate tracking by overlap maximization," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Long Beach, CL, USA, September 2019.

[43] G. Bhat, M. Danelljan, L. Van Gool, and T. Radu, "Learning discriminative model prediction for tracking," 2019, https://arxiv.org/pdf/1904.07220.pdf.

[44] C. Sun, D. Wang, H. Lu, and M.-H. Yang, "Correlation tracking via joint discrimination and reliability learning," in *Proceedings of the 2018 Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, June 2018.

[45] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *Proceedings of the European Conference on Computer Vision*, Munich, Germany, August 2018.

[46] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, "Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual tracking," 2018.

[47] X. Yang, S. Zhou, and M. Cao, "An approach to alleviate the sparsity problem of hybrid collaborative filtering based recommendations: the product-attribute perspective from user reviews," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 376–390, 2020.

[48] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence Information for Authors*, vol. 37, no. 6, 2015.

[49] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, June 2017.

[50] Y. Song, C. Ma, L. Gong, J. Zhang, R. Lau, and M.-H. Yang, "Convolutional residual learning for visual tracking," in *Proceedings of the International Conference on Computer Vision*, Venice, Italy, November 2017.

[51] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *The Primary Angioplasty in Myocardial Infarction*, vol. 33, no. 7, pp. 1619–1632, 2011.