Hindawi
Journal of Sensors
Volume 2022, Article ID 2804114, 11 pages
https://doi.org/10.1155/2022/2804114

Research Article

Improved SiamFC Target Tracking Algorithm Based on Anti-Interference Module

Yejin Yan, Wenxiao Huo, Jiayu Ou, Zhifeng Liu, and Tianping Li

College of Physics and Electronic Science, Shandong Normal University, 250300 Shandong, China

Correspondence should be addressed to Tianping Li; sdsdltp@sdnu.edu.cn

Received 16 November 2021; Revised 13 January 2022; Accepted 19 January 2022; Published 10 February 2022

Academic Editor: Chao Wang

The SiamFC target tracking algorithm has attracted extensive attention because of its good balance between speed and performance, but the tracking effect of the SiamFC algorithm is not satisfactory in complex background scenes. When SiamFC algorithm uses deep semantic features for tracking, it has good recognition ability for different types of objects, but it has insufficient discrimination for the same types of objects. Therefore, we propose an effective anti-interference module to improve the discrimination ability of the algorithm. The anti-interference module uses another feature extraction network to extract the features of the candidate target images generated by the SiamFC main network. In addition, we set up the feature vector set to save the feature vectors of the tracking target and the template image. Finally, the tracking target is selected by calculating the minimum cosine distance between the feature vector of the candidate target and the vector in the feature vector set. A large number of experiments show that our anti-interference module can effectively improve the performance of SiamFC algorithm, and the performance of this algorithm can be comparable to the popular algorithms.

1. Introduction

The field of computer vision has advanced rapidly in recent years, and the direction of target tracking has become a research hotspot for many research institutions and universities. Current target tracking is typically based on delimiting the target area in the first frame of the video sequence and then tracking the target in the subsequent frame [1]. Target tracking has a wide range of applications, such as autonomous driving [2], video surveillance, and human-computer interaction [3]. However, many problems still exist in the field of target tracking, such as complex background, target occlusion, and scale change [4].

Current mainstream target tracking algorithms can be divided into two categories. One category is based on the Siamese network [5–13]. Algorithms in this category are designed using the Siamese network structure and have achieved good results. The other category is based on a non-Siamese network [14–18], which is mostly studied using correlation filter (CF) [19–22]; however, because algorithms in this category are constantly improving, their tracking speed and performance based on CF cannot be well balanced. The majority of researchers prefer a Siamese network-based target tracking algorithm, and its classical algorithm SiamFC [5] has become a milestone algorithm. It can effectively balance the speed and accuracy of target tracking and has become the cornerstone of many subsequent improved algorithms. However, these improved algorithms [5–9] cannot effectively solve the intraclass interference problem of target tracking in a complex background because they do not effectively distinguish the tracking target from the interference target. Moreover, we believe that simply relying on an improved network model to improve the anti-interference ability of the target cannot meet the requirements. In some cases, the response value of the interference target in the tracking process exceeds the response value of the tracking target, as shown in Figure 1. However, this is inevitable because a convolutional neural network (CNN) cannot obtain such a high discriminant network model to avoid the overfitting problem in the training process. If we want to further improve the discrimination ability of the target while also considering the universality of the target tracking effect, we must increase the number of training parameters and the training set. These two requirements have significant limitations in terms of current conditions.

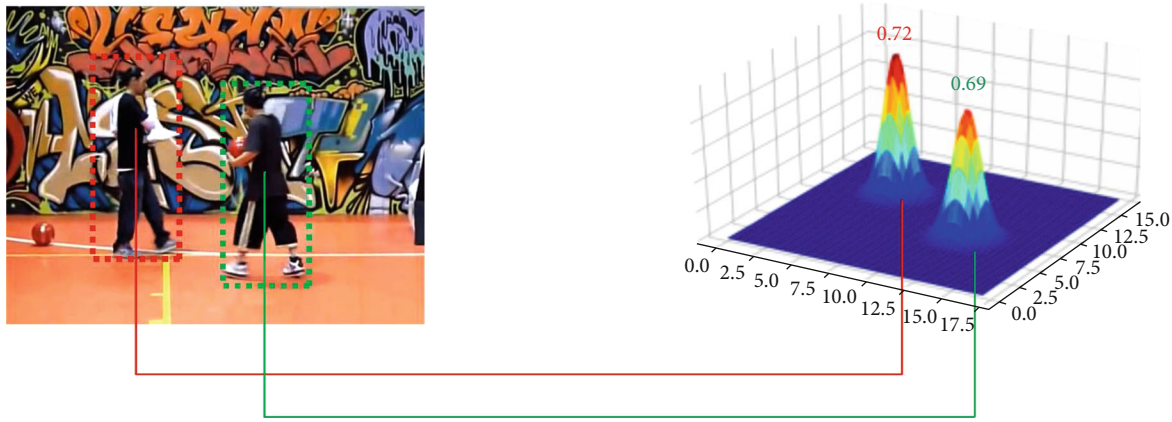

Figure 1: Response scores of SiamFC algorithm. Red box target for interference target, and green box target for tracking target.

The similarities between these algorithms [5–8] are all based on the screening of candidate targets, with the majority of the screening aimed at selecting the target with the highest score in the score map. SiamFC [5] is screened directly based on the score map after cross-correlation operation, SiamRPN [6] is screened directly based on the score map after nonmaximum suppression, and DaSiamRPN [7] is the same as SiamRPN [6]. However, in the complex background, the response value of the target is close to the interference target, and even the response value of the interference target is higher than that of the tracking target, which will inevitably affect the tracking effect.

Based on this background, this study proposes an anti-interference module and designs an appearance feature extraction network. First, it extracts features of the tracking target in recent and initial frames and then extracts features of the candidate target in the current frame. Finally, it judges the best tracking target by calculating the minimum cosine distance of the feature vectors of the two parts and finally realizes the effective judgment of the candidate target frame.

The main contributions of this paper are as follows:

(1) The anti-interference module is designed to improve the robustness of the algorithm to complex background scenes

(2) An appearance feature extraction network which can effectively extract the appearance features of the target is designed. Multiple candidate boxes are extracted on the basis of SiamFC, and the candidate boxes are input into the appearance feature extraction network to finally obtain the correlation vector

(3) The feature vector set is designed, which can save the tracking target feature vector in recent frames and the template image

(4) The cosine distance between the vector in the feature vector set and the feature vector of the candidate target is calculated to determine the tracking target, which solves the disadvantage that only template image features can be used in SiamFC algorithm and improves the performance of the algorithm for long-time tracking

2. Related Works

2.1. Target Tracking Algorithm Based on Deep Learning. In recent years, due to the continuous expansion of available datasets and the improvement in computing power, the advantages of deep learning (DL) methods have gradually become evident. DL methods are far more powerful than traditional algorithms in terms of target tracking. In addition, the great potential of DL direction has piqued the interest of an increasing number of researchers. The advantage of a DL algorithm lies in the strong feature extraction ability and representation ability of its network model. At present, methods based on the DL network model are mainly divided into the following categories: CNN method, recurrent neural network (RNN) method, and generative adversarial network (GAN) method. The most popular DL network model in the field of computer vision is CNNs, and RNNs are more commonly used in natural language processing. Although GANs have some applications in image processing, they are limited to data processing. DL was first applied to target tracking in [23], and a target tracking framework based on offline training and online fine-tuning was proposed. Several subsequent algorithms have been improved on this network framework and have achieved good results.

2.2. Convolutional Neural Network-Based Methods. In recent years, CNNs have swept through the field of DL. From natural language processing to image processing, computer vision has also made great progress through the continuous improvement of CNNs. In 2012, the success of the AlexNet network model on the ImageNet classification dataset sparked a surge in researchers' interest in DL. There are three popular network models: AlexNet [24], VggNet [25], and ResNet [26]. AlexNet [24] has a network structure of only eight layers, of which five layers are convolution layers and the other three layers are fully connected layers. Compared with AlexNet [24], VggNet [25] has more network depth, so the tracking effect is greatly improved. However, with the increase in network depth, grid degradation will occur. At this time, the emergence of ResNet

[26] introduces neural networks in a new direction. ResNet [26] connects network layers through the jump connection, which effectively solves the problem of grid degradation when the network depth is deepened. Finally, ResNet [26] won ImagNet2015 [27]. In recent years, lightweight models have attracted increasing attention. MobileNetV1 [28] uses depthwise (DW) separable convolutions, ignores the pooling layer, and uses convolution with a stripe equal to 2. Compared with V1, MobileNetV2 [29] introduces a residual structure. Before DW, 1×1 convolution is used to increase the feature map channel. After pointwise, a rectified linear unit (ReLU) is abandoned and replaced with a linear activation function to prevent the destruction of features by ReLU. MobileNetV3 [30] integrates the depth separable convolution of V1 and the inverse residual structure of V2 and introduces the h-swish activation function. EffNet [31] decomposes the DW layer of MobileNetV1 into 3×1 and 1×3 convolutions. After the first layer, pooling is adopted to reduce the amount of calculation in the second layer. The smaller the size of the model, the higher the accuracy is obtained. EfficientNet [32] designs a standardized convolution network expansion method to optimize the efficiency and accuracy of the network from the three dimensions of balance resolution, depth, and width. ShuffleNetV1 [33] reduces computation complexity by grouping convolution and enriches channel information by reorganizing channels. ShuffleNetV2 [34] mainly designs and uses more efficient CNN network structure design criteria.

CNNs typically extract the deep semantic features of images through deep neural networks and then use the appropriate classifier to extract the target. At present, the full CNN is the most popular; that is, there is no full connection layer in the entire network model, which greatly reduces the number of network parameters and increases the running speed. In SiamFC [5] tracking algorithm, the network model is improved on the basis of AlexNet [24], removing the final full connection layer and part of the convolution layer. Finally, the target tracking problem is transformed into a similarity matching problem, and the location of the target is judged by a cross-correlation operation. SiamRPN [6] algorithm introduces the RPN [35] network to target detection based on the SiamFC algorithm, significantly improving the accuracy of target tracking through classification and regression. DaSiamRPN [7] optimizes the imbalance of positive and negative samples in the training process based on SiamRPN. SiamRPN + + [8] increases the network depth based on SiamRPN [6] and has achieved good results. CFNet [9] adds the CF layer based on the SiamFC [5] structure to realize the end-to-end training of the network, which proved that this network structure could use fewer convolution layers of the network without degrading accuracy. The main improvement of SiamFC++ [36] is to add a boundary box regression branch and quality estimation branch based on SiamFC [5]. In [37], the authors propose a multilevel similarity model under a Siamese framework for robust TIR object tracking, which solves the problem that only RGB images can be used in the training process. Motivated by the forward-backward tracking consistency of a robust tracker, self-SDCT [38] proposes a multicycle consistency loss as self-supervised information for learning feature extraction network from adjacent video frames. TRBACF [39] proposes a temporal regularization strategy based on the correlation filter, which effectively solves the problem that the model can not adapt to tracking scene changes and improves the robustness and accuracy of the algorithm.

2.3. Image Similarity Judgment. At present, there are several ways to judge the similarity of images. The first method is based on histograms. The histogram method judges the similarity by describing the color distribution in an entire image, but a histogram is too simple to capture the similarity of color information and cannot use more information. The second method is to calculate the mutual information about two images. Although this method is accurate, it has great limitations. It requires that the size of the two images must be the same. If the two images are cut into the same size, it is bound to lose a lot of information, thereby degrading accuracy. The third method is the cosine distance judgment method. Images are represented as vectors, and the cosine distance between these vectors is calculated to determine the similarity. The cosine distance pays more attention to the direction of the vector to avoid the influence of the absolute value of the vector on the similarity judgment. It is very suitable for us to extract the target feature and use the vector for similarity judgment.

3. The Proposed Algorithm

In the classical SiamFC [5] algorithm, an improved network on AlexNet [24] is used as the backbone network of the tracking network. The Siamese network is used to extract the feature of the search and template images, respectively. Finally, the position score map of $17 \times 17 \times 1$ is obtained by a cross-correlation operation, as shown in Figure 2.

However, the resolution of the feature map calculated using the two feature branches in SiamFC [5] is too small. Although it can effectively search the target, it cannot effectively distinguish the target within a class. As shown in Figure 1(b), the interference target even produces a higher thermal value than the tracking target. Inspired by the appearance feature module in DeepSort [40], we consider whether we can construct another special appearance feature extraction network to extract the appearance feature of the target to better distinguish intraclass targets.

Thus, we design a new type of target anti-interference module. The main body of the anti-interference module is composed of a feature extraction network and similarity calculation. Unlike other algorithms for suppressing the interference target, we choose the tracking results of several adjacent frames of the tracking target to measure the tracking target twice.

We will describe the overall framework of the algorithm in Section 3.1. Section 3.2 focuses on the main framework of the benchmark algorithm SiamFC [5]. Section 3.3 describes the main network of our anti-interference module. Section 3.4 mainly describes the working mode of the anti-interference module and how to determine the position of the final target box.

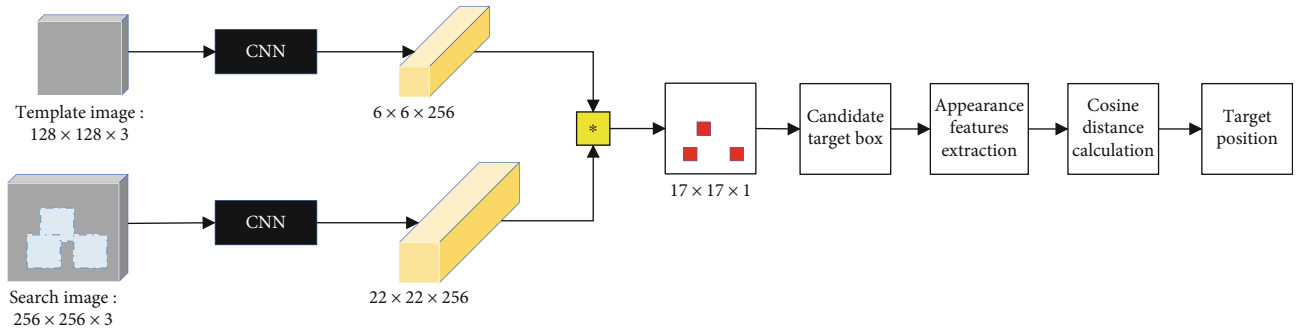

Figure 2: Overall frameworks.

3.1. Overall Framework. The algorithm is mainly composed of two parts. The first part is the main framework of SiamFC [5], as shown in the red box in Figure 2. The main role is to extract features and generate candidate targets. Different from SiamFC [5], where only one target box is generated in SiamFC [5], multiple candidate boxes are selected in our algorithm. The second part is our anti-interference module, such as the green box in Figure 2. The main function is to process the multiple candidate boxes generated in the first part to output the final target position. Figure 2 shows the overall frame diagram.

3.2. The SiamFC Framework. The main framework of SiamFC [5] is divided into two branches: template and search branches. The main network of SiamFC [5] is improved on the basis of AlexNet [24], removing the full connection layer and partial convolution layer. There are only Conv and pooling layers in the entire network structure, and the template branch shares the same network parameters with the search branch, which satisfies the definition of full convolution and twin network. The processing of the original image in SiamFC [5] is as follows: we select the first frame image of the video sequence as the template image and other images as the search image. We use 127×127-pixel template images and 255×255-pixel search images. To facilitate the extraction of appearance features by the anti-interference module, we cut the template image to 128×128 pixels and the search image to 256×256 pixels. The specific feature processing process of SiamFC is as follows. A template image of $128 \times 128 \times 3$ is input into the template branch to obtain the feature map of $6 \times 6 \times 128$. Similarly, a search image of $256 \times 256 \times 3$ is input into the search branch to obtain the feature map of $22 \times 22 \times 128$. A template image feature is used as the convolution kernel, and the two feature maps are cross-correlated; sliding window detection is performed on the features of the search image. Then, we obtain a $17 \times 17 \times 1$ score map about the target location information. The cross-correlation operation formula is as follows:

$$F(Z, X) = \varphi(Z) * \varphi(X) + b_i. \tag{1}$$

$\varphi(Z)$ and $\varphi(X)$ represent the extracted template and search image features, respectively. The symbol $*$ indicates convolution operation, where b_i denotes a signal that takes the value $b \in R$ in every location.

In the actual tracking process, our template branch only needs to be executed once to obtain the features of a template image. In the subsequent tracking process, information about the target position can be obtained by convolution operation between the extracted features of the search image and the features of the template image. The position of the target in the original image is obtained by upsampling according to the score map of $17 \times 17 \times 1$.

3.3. Extract Appearance Features. Figure 3 is our appearance feature extraction network, which is also the main part of the anti-interference module. It is mainly composed of two convolution layers and six residual blocks. The GOT10k [41] dataset is used to train the residual network model offline and output the normalized characteristics. Candidate boxes are reshaped into $128 \times 128 \times 3$ images, which are then input into the feature extraction network, producing 256-dimensional vectors. Finally, the normalization operation is performed to facilitate subsequent calculation.

3.4. Determination of Target Position by Minimum Cosine Distance. First, the network extracts the appearance features of five adjacent target frames and the initial frame and saves them to the feature vector set. Then, the vector of multiple prediction target frames in the current frame is extracted. The best tracking target is judged by calculating the cosine distance between the multiple candidate target features and the feature vector set of the current frame. Then, the feature vector set is updated according to the predicted target. The specific flowchart is shown in Figure 4. There are typically six vectors in the feature vector set, including five adjacent frame vectors and one initial frame vector. The selection of the prediction target box is based on the score map of $17 \times 17 \times 1$ generated by SiamFC [5]. First, we normalize and sort the score map of $17 \times 17 \times 1$ and then select the maximum three values. Take out the candidate target boxes corresponding to the three values. Then, their feature vectors are obtained using the feature extraction network. By calculating the cosine distance between the feature vector set and the feature vectors of the three prediction target boxes, the matrix of 3×6 can be obtained. The formula for calculating the single value of matrix $R(i, j)$ is as follows:

$$R(i, j) = 1 - r_i^T r_j \quad i \in (1, 3), \quad j \in (1, 6). \tag{2}$$

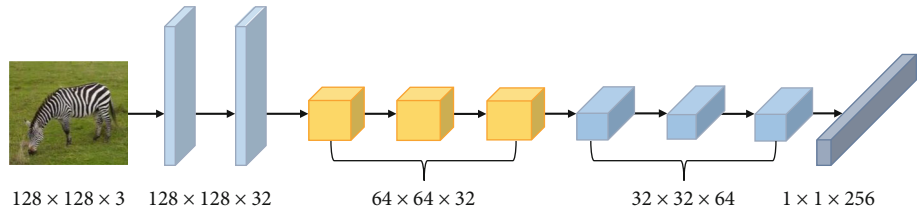

Figure 3: Deep appearance feature extraction network structure diagram.

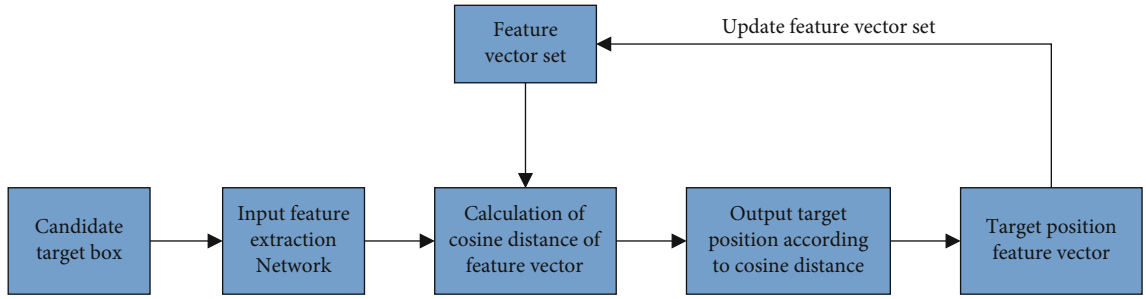

Figure 4: Flowchart of anti-interference module.

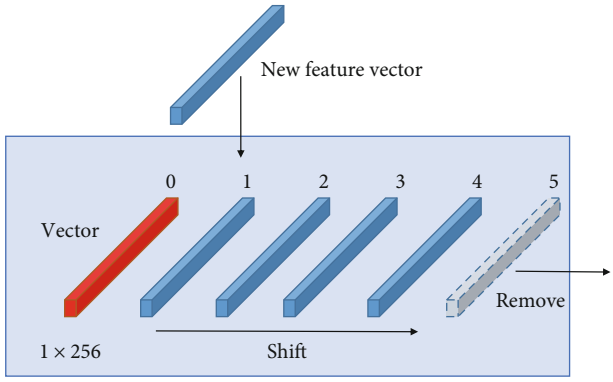

Figure 5: Update feature vector set.

The value of each row of the matrix is calculated by linear weighting. The formula is as follows:

$$R[i] = k1R(i,1) + k2\sum_{j=1}^{6} R(i,j), \quad i \in (1,3). \tag{3}$$

$k1$ and $k2$ are hyperparameters, typically $k1 = 0.35$ and $k2 = 0.65$.

Select the smallest $R[i]$ value as the final target location. The cosine distance can judge the similarity between vectors by calculating the angle between the directions of vectors, which effectively avoids the effect of the difference in absolute values of image pixels on the similarity judgment.

For update feature vector set, as shown in Figure 5, the feature vector set saves the feature vectors of our last five frames (as shown in Figure 5, blue vector) and template pictures (as shown in Figure 5, red vector). When we determine the position of the target in the current frame, we save the appearance feature vector obtained from the corresponding target candidate box to our feature vector set and remove the last feature vector.

4. Experiments

4.1. Experimental Configuration. We conducted experiments on a Linux system, and the experimental code was written in Python language and the PyTorch [42] framework. The experimental system configuration is Inter Core i7-10700 k CPU @ 3.80 GHz ×16, and a single GTX1070ti GPU.

4.2. Training Process. The training CNN part uses ILSVRC15 [27] and GOT10k [41] datasets for training. The appearance feature extraction network is trained using GOT10k [41].

4.3. Test Process. The OTB2015 [43] dataset is used for performance tests, and the VOT2016 [44] and VOT2017 [45] datasets are used to test the universality of the algorithm. To verify the effectiveness of the anti-interference module, we first compare the discrimination ability of the anti-interference module with the original algorithm, and then, we conduct tracking experiments on public datasets OTB2015 [43], VOT2016 [44], and VOT2017 [45] to prove the effectiveness and universality of our algorithm.

4.4. Single Discriminant Ability Experiment. Figure 6(a) is the first frame in the OTB2015 [43] video sequence "Board," where the green frame is the selected tracking target. Figure 6(b) is the SiamFC [5] tracking failure frame, where the red frame is the SiamFC [5] tracking failure position, and the green frame is the ground truth of the tracking target. To verify the effectiveness of our anti-interference

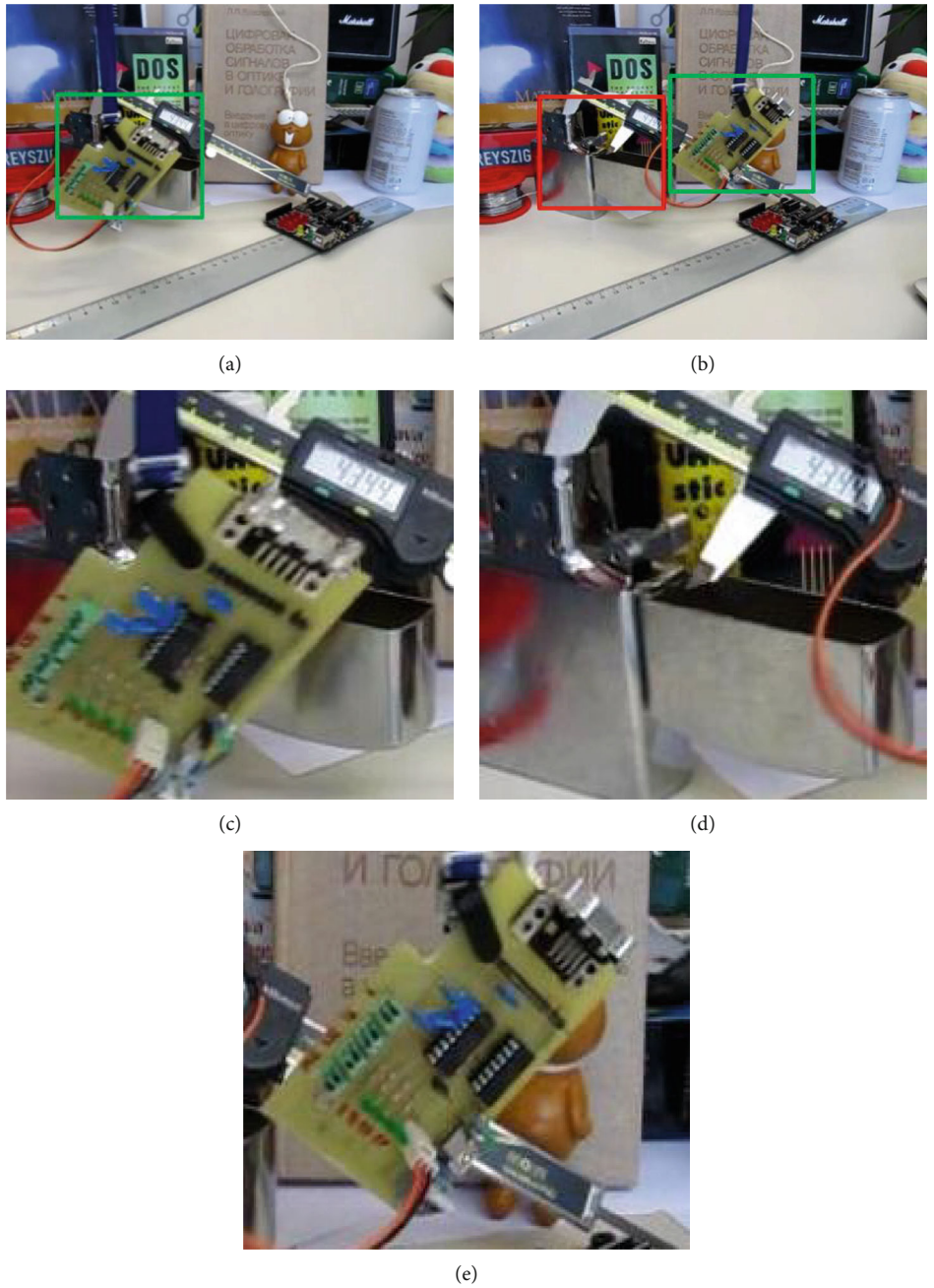

Figure 6: (a, b) The initial frame and SiamFC [5] tracking failure frame images, respectively. (c–e) The input images of the anti-interference module.

module, we first input the initial frame part as Figure 6(c) into our anti-interference module to obtain the feature vector. Then, we extract the tracking failure frame part, as shown in Figure 6(d), and the tracking target of the first frame, as shown in Figure 6(e). After that, we input Figures 6(d) and 6(e) into our anti-interference module to obtain the corresponding vector and then calculate the cosine distance using the feature vector of the initial frame tracking target. Finally, the cosine distance between the failure target and the initial frame is 0.73, and the cosine distance obtained from the ground truth part is 0.92. The higher the similarity, the closer the cosine distance is to 1. Thus, our anti-interference module can effectively judge the intraclass interference target, allowing our algorithm to outperform the baseline algorithm SiamFC [5].

4.5. Experiments in OTB2015. The OTB2015 [43] dataset is the benchmark dataset to test the performance of the target tracking algorithm. The dataset contains 100 manually annotated video sequences. The dataset mainly has two evaluation indexes: success and precision rates. The success rate is determined by whether the overlap rate between the bounding box and ground truth obtained using a frame during the tracking

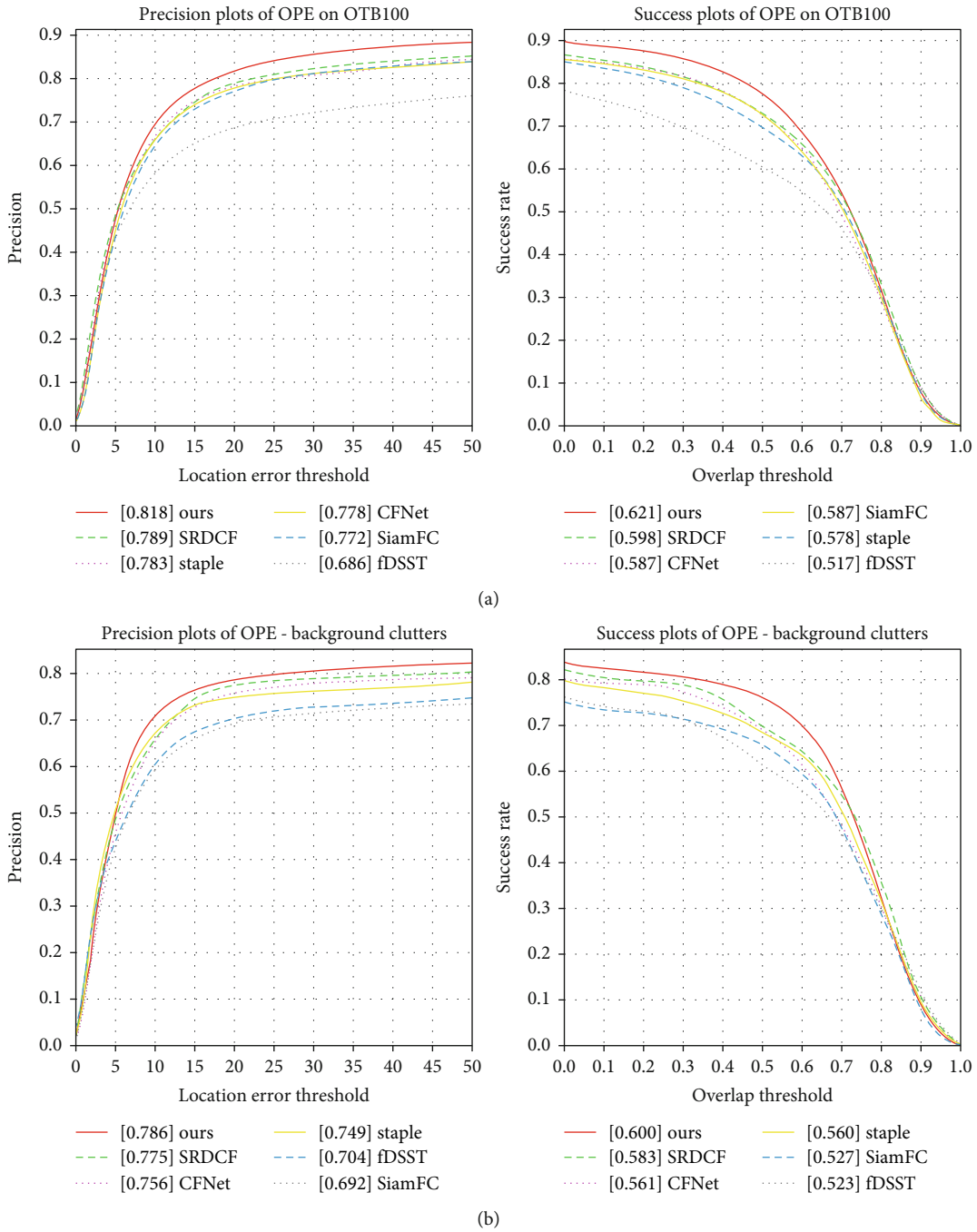

Figure 7: (a) The comparison of our algorithm with other algorithms on the OTB2015 dataset [43]. (b) The comparison of our algorithm with other algorithms on the OTB2015 [43] complex background dataset.

process exceeds a certain threshold; if so, the frame is regarded as a successful tracking frame. The percentage of successful frames in all frames is the success rate. The precision rate is defined as the center point of the target bounding box estimated by the tracking algorithm and the center point of the target manually labeled ground truth, and the distance between the two is less than the percentage of the video frames in a given threshold. Different thresholds have different percentages, and the general threshold is set to 20 pixels.

Figure 7(a) shows the comparison of our algorithm with other popular algorithms and benchmark algorithm SiamFC [5] on the OTB2015 [43] dataset. Other algorithms are SRDCF [46], Staple [47], CFNet [9], and fDSST [48]. Figure 7(b) shows the experimental results on the dataset in the OTB2015 complex background section. Figure 7 shows that the effect of our algorithm on the overall dataset has been compared to several existing popular algorithms. Our algorithm outperforms the benchmark algorithm SiamFC [5] in terms of accuracy and

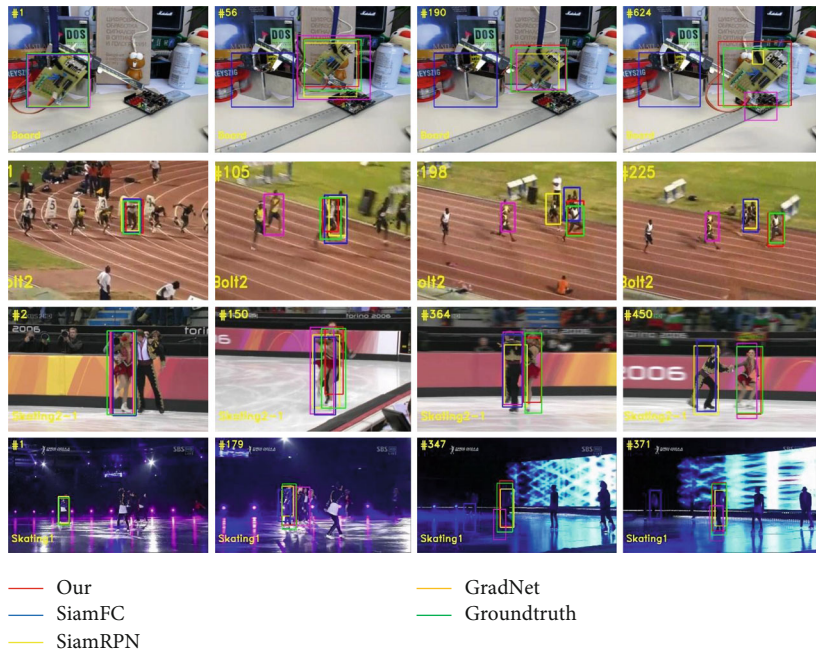

Figure 8: Comparison of tracking effects between our algorithm and other algorithms.

Table 1: Comparison of VOT2016 algorithm results.

Tracker	EAO	A	R	EFO
Our	0.301	0.558	0.286	3.857
SiamFC [5]	0.277	0.549	0.382	5.444
Staple [46]	0.295	0.544	0.378	11.114
CCOT [48]	0.331	0.539	0.238	0.507
TCNN [49]	0.325	0.554	0.268	1.049
DDC [44]	0.293	0.541	0.345	0.198
EBT [50]	0.291	0.465	0.252	3.011
STAPLEp [44]	0.286	0.557	0.368	44.765
DNT [51]	0.278	0.515	0.329	1.127
DeepSRDCF [52]	0.276	0.528	0.380	0.380
MDNet_N [53]	0.257	0.541	0.337	0.534

success rates. In particular, our algorithm has a good tracking effect in the case of complex background, which also shows that our algorithm can effectively distinguish between intraclass targets, reducing the misjudgment rate. In terms of running speed, SiamFC [5] is 80 fps, whereas our algorithm is 56 fps. Although the tracking speed of our algorithm is lower than that of the SiamFC algorithm, it still exceeds 30 fps, meeting the speed requirements of real-time target tracking. Figure 8 shows the comparison of tracking effects between our algorithm and other algorithms.

4.6. Experiments in VOT2016. To verify the universality of the improved algorithm, we also conducted experiments on the VOT2016 [44] dataset. The VOT challenge is one of the most influential competitions in the field of computer vision. The VOT2016 [44] benchmark dataset consists of 60 video sequences, and all are color sequences. There are three main evaluation indicators in VOT2016 [44]: accuracy (A), equivalent filter operations (EFO), robustness (R), and expected average overlap (EAO). Accuracy is the accuracy of the target tracking, that is, the average overlap between the target box and the true value box during successful tracking. Robustness (R) is defined as the number of tracking failures. EAO represents the average value of the intersection and union ratio between the prediction box and the true ground-truth box in the entire video sequence. EFO is used to measure the tracking speed of the tracker.

We compare our algorithm with ten other popular algorithms on the VOT2016 [44] dataset, including the benchmark algorithm SiamFC [1] and nine other popular algorithms: Staple [47], CCOT [49], TCNN [50], DDC [44], EBT [51], STAPLEp [44], DNT [52], DeepSRDCF [53], and MDNet_N [54]. The comparison results are shown in Table 1. The chart shows that CCOT [38] has the best EAO of 0.331, our algorithm has the best accuracy of 0.558, CCOT [48] has the best robustness of 0.238, and STAPLEp [44] has the best tracking speed EFO of 44.745.

From the comparison results, our algorithm outperforms the benchmark algorithm SiamFC [5] in terms of EAO, accuracy, and robustness. In particular, the robustness of our algorithm is greatly improved compared with the original algorithm SiamFC [5]. This is because our anti-interference module effectively reduces the number of tracking failures, thereby improving the robustness of tracking. The accuracy of the algorithm is also improved compared with SiamFC [5], and it is better than other algorithms. This is because the tracking robustness can be increased after screening candidate targets through the anti-interference module. Second, our anti-interference module uses the minimum cosine distance to judge similar targets and processes the tracking target vector

Table 2: Comparison of VOT2017 algorithm results.

Tracker	SiamNAB	SiamFC	ECOHC	KFebT	ASMS	SSKCF	CSRDCF	UCT	MOSSEca
A	0.517	0.502	0.494	0.451	0.489	0.513	0.475	0.490	0.400
R	0.486	0.604	0.571	0.684	0.627	0.656	0.646	0.777	0.810
EAO	0.215	0.182	0.177	0.169	0.168	0.158	0.158	0.145	0.139

of recent frames in a weighted way, which reduces the probability of losing targets in the tracking process and improves the performance of long-time tracking. Therefore, accuracy can be improved. However, the accuracy has not been greatly improved. We believe that this is because the SiamFC regression is not sufficiently accurate. Compared with other algorithms, even though our indicators are not the highest, we do a good job of balancing speed and performance. For example, although the EAO value of CCOT reaches 0.331, its tracking speed is very slow; its EFO is only 0.507, whereas ours reaches 3.857.

4.7. Experiments in VOT2017. In this experiment, we evaluated the proposed algorithm on the VOT2017 [45] benchmark dataset. Then, we compared its accuracy, robustness, and EAO score with SiamFC [5] and the seven popular real-time tracker algorithms in VOT2017 [45]. These trackers are SiamFC, ECOHC [55], KFebT, ASMS, SSKCF, CSRDCF, UCT [56], and MOSSEca. Table 2 presents the experimental results. It can be seen from Table 2 that all indexes of our algorithm are better than other algorithms, and the accuracy is improved by 1.5% compared with the benchmark algorithm SiamFC. Especially in terms of robustness, our algorithm has great advantages over other methods; we believe that this is because the anti-interference module reduces the error rate in complex background. In addition, combined with the characteristics of targets in recent frames, the robustness of the algorithm for long-time tracking is also improved. The above experiments showed that on the VOT2017 [45] dataset, the proposed method is highly competitive with other most advanced trackers.

5. Conclusions

In this study, a new anti-interference module is proposed. Based on the benchmark algorithm SiamFC [5], another feature extraction network is designed, and its intraclass discriminant ability is trained on the GOT10k [41] dataset. The cosine distance is used to select the best tracking target by extracting the vector of the target frame. The experimental results show that, compared with the original benchmark algorithm SiamFC [5], our algorithm can well cope with the effect of intraclass targets on tracking performance in a complex background, thereby improving tracking accuracy; this also proves the effectiveness of the proposed anti-interference module. In the future, we will incorporate the anti-interference module into more advanced algorithms for research.

Data Availability

The test results already exist in the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Hare, S. Golodetz, A. Saffari et al., "Struck: structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096–2109, 2015.

[2] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[3] L. Mao, X. Li, D. Yang, and R. Zhang, "Convolutional feature frequency adaptive fusion object detection network," *Neural Process Letter*, vol. 53, no. 5, pp. 3545–3560, 2021.

[4] S. T. Cheng, C. W. Hsu, G. J. Horng, and S. Y. Chen, "Across-camera object tracking using a conditional random field model," *The Journal of Supercomputing*, pp. 1–28, 2021.

[5] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European conference on computer vision*, pp. 850–865, Springer, Cham, 2016.

[6] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with Siamese region proposal network," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8971–8980, 2018.

[7] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware Siamese networks for visual object tracking," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 101–117, 2018.

[8] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: evolution of Siamese visual tracking with very deep networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4282–4291, 2019.

[9] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2805–2813, 2017.

[10] P. Gao, R. Yuan, F. Wang, L. Xiao, H. Fujita, and Y. Zhang, "Siamese attentional keypoint network for high performance visual tracking," *Knowledge-Based Systems*, vol. 193, p. 105448, 2020.

[11] L. Zheng, Y. Chen, M. Tang, J. Wang, and H. Lu, "Siamese deformable cross-correlation network for real-time visual tracking," *Neurocomputing*, vol. 401, pp. 36–47, 2020.

[12] K. Yang, Z. He, Z. Zhou, and N. Fan, "Siam Att: Siamese attention network for visual tracking," *Knowledge-Based Systems*, vol. 203, p. 106079, 2020.

[13] D. Li, F. Porikli, G. Wen, and Y. Kuai, "When correlation filters meet Siamese networks for real-time complementary tracking," in *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 509–519, 2020.

[14] C. Ma, J. Huang, J. B. Yang, X. Yang, and M. H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, pp. 3074–3082, 2015.

[15] Y. Song, C. Ma, X. Wu et al., "Vital: visual tracking via adversarial learning," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8990–8999, 2018.

[16] H. Fan and H. Ling, "Sanet: structure-aware network for visual tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 42–49, 2017.

[17] M. Zhang, J. Xing, J. Gao, X. Shi, Q. Wang, and W. Hu, "Joint scale-spatial correlation tracking with adaptive rotation estimation," in *in Proceedings of the IEEE international conference on computer vision workshops*, pp. 32–40, 2015.

[18] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. Lau, and M. H. Yang, "Crest: convolutional residual learning for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, pp. 2555–2564, 2017.

[19] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, 2010.

[20] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *European conference on computer vision*, pp. 702–715, Springer, Berlin, Heidelberg, 2012.

[21] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[22] P. Zhang, Q. Guo, and W. Feng, "Fast and object-adaptive spatial regularization for correlation filters based tracking," *Neurocomputing*, vol. 337, no. 14, pp. 129–143, 2019.

[23] N. Y. 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, 2013.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

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

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition.*, pp. 770–778, 2016.

[27] O. Russakovsky, J. Deng, H. Su et al., "Image net large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[28] A. G. Howard, M. Zhu, B. Chen et al., "MobileNets: efficient convolutional neural networks for mobile vision applications," 2017, http://arxiv.org/abs/1704.04861.

[29] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018.

[30] A. Howard, M. Sandler, G. Chu et al., "Searching for MobileNetV3," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. , 20191314–1324, 2019.

[31] I. Freeman, L. Roese-Koerner, and A. Kummert, "Effnet: an efficient structure for convolutional neural networks," in *2018 25th ieee international conference on image processing (icip)*, pp. 6–10, 2018.

[32] M. Tan and Q. Le, "Efficientnet: rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*, pp. 6105–6114, 2019.

[33] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: an extremely efficient convolutional neural network for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6848–6856, 2018.

[34] N. Ma, X. Zhang, H. T. Zheng, and J. Sun, "ShuffleNet V2: practical guidelines for efficient cnn architecture design," in *Proceedings of the European conference on computer vision*, pp. 116–131, 2018.

[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[36] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "Siam FC++: towards robust and accurate visual tracking with target estimation guidelines," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34no. 7, pp. 12549–12556, 2020.

[37] Q. Liu, X. Li, Z. He, N. N. Fan, D. Yuan, and H. P. Wang, "Learning deep multi-level similarity for thermal infrared object tracking," *IEEE Transactions on Multimedia*, pp. 2114–2126, 2020.

[38] D. Yuan, X. Chang, P. Y. Huang, Q. Liu, and Z. He, "Self-supervised deep correlation tracking," *IEEE Transactions on Image Processing*, vol. 30, pp. 976–985, 2020.

[39] D. Yuan, X. Shu, and Z. He, "TRBACF: learning temporal regularized correlation filters for high performance online visual object tracking," *Journal of Visual Communication and Image Representation*, vol. 72, p. 102882, 2020.

[40] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP). IEEE*, pp. 3645–3649, 2017.

[41] L. Huang, X. Zhao, and K. Huang, "GOT-10k: a large high-diversity benchmark for generic object tracking in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1562–1577, 2021.

[42] A. Paszke, S. Gross, S. Chintala, and G. Chanan, "Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration," *PyTorch: Tensors and Dynamic Neural Networks in Python with Strong GPU Acceleration*, vol. 6, no. 3, 2017.

[43] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," in *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pp. 1834–1848, 2015.

[44] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, and R. Pflugfelder, "The visual object tracking VOT2016 challenge results," in *European conference on computer vision*Springer.

[45] M. Kristan, A. Leonardis, J. Matas et al., "The visual object tracking VOT2017 challenge results," in *2017 IEEE International Conference on Computer Vision Workshop (ICCVW)*IEEE.

[46] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, pp. 4310–4318, 2015.

[47] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: complementary learners for real-time tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition.*, pp. 1401–1409, 2016.

[48] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 8, pp. 1561–1575, 2017.

[49] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision*, pp. 472–488, Springer, 2016.

[50] H. Nam, M. Baek, and B. Han, "Modeling and propagating CNNs in a tree structure for visual tracking," 2016, http://arxiv.org/abs/1608.07242.

[51] G. Zhu, F. Porikli, and H. Li, "Beyond local search: tracking objects everywhere with instance-specific proposals," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 943–951, 2016.

[52] Z. Chi, H. Li, H. Lu, and M. H. Yang, "Dual deep network for visual tracking," *IEEE Transactions on Image Processing*, vol. 26, no. 4, pp. 2005–2015, 2017.

[53] M. Danelljan, G. Hager, F. Shahbaz Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proceedings of the IEEE international conference on computer vision workshops*, pp. 58–66, 2015.

[54] 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, 2016.

[55] M. Danelljan, G. Bhat, F. Shahbaz Khan, and M. Felsberg, "Eco: efficient convolution operators for tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6638–6646, 2017.

[56] Z. Zhu, G. Huang, W. Zou, D. Du, and C. Huang, "UCT: learning unified convolutional networks for real-time visual tracking," in *Proceedings of the IEEE international conference on computer vision workshops*, pp. 1973–1982, 2017.