BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 70(3), 2022, Article number: e139961
DOI: 10.24425/bpasts.2021.139961

ARTIFICIAL AND COMPUTATIONAL INTELLIGENCE

# Single target tracking algorithm for lightweight Siamese networks based on global attention

**Zhentao WANG, Xiaowei HE*, and Rao CHENG**

College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua, Zhejiang, 321000, China

**Abstract.** Object tracking based on Siamese networks has achieved great success in recent years, but increasingly advanced trackers are also becoming cumbersome, which will severely limit deployment on resource-constrained devices. To solve the above problems, we designed a network with the same or higher tracking performance as other lightweight models based on the SiamFC lightweight tracking model. At the same time, for the problems that the SiamFC tracking network is poor in processing similar semantic information, deformation, illumination change, and scale change, we propose a global attention module and different scale training and testing strategies to solve them. To verify the effectiveness of the proposed algorithm, this paper has done comparative experiments on the ILSVRC, OTB100, VOT2018 datasets. The experimental results show that the method proposed in this paper can significantly improve the performance of the benchmark algorithm.

**Key words:** target tracking; Siamese network; semantic information; training strategy; feature fusion; deep learning.

## 1. INTRODUCTION

With the development of artificial intelligence, computer vision and other technologies, convolutional neural network has been gradually applied to computer vision tasks [1], such as target detection [2], semantic segmentation [3] and target tracking. Thanks to the powerful representation ability of convolutional neural network features, traditional artificially defined features are gradually replaced with convolutional features. An important direction of the tracking framework based on convolutional neural networks is the Siamese network tracking framework. Among them, the fully convolutional network Siamese-FC was proposed by Bertinetto *et al*. [4]. This end-to-end network has a small number of parameters and has a faster tracking effect during forwarding inference. At the same time, the network can solve part of the work offline, and the algorithm when deployed in embedded devices, only the forward inference part of the search area needs to be calculated, which reduces the amount of calculation for online tracking. Therefore, the lightweight Siamese network has the advantages of fast and low calculation amount in embedded deployment. Although Siamese-FC algorithm has good real-time performance and tracking ability, it may be difficult for feature matching to accurately identify real targets in scenes with many similar targets due to the lack of update mechanism. After Siamese-FC, a series of more excellent tracking models emerged successively, such as GradNet [5], DaSiamRPN[6]and SiamRPN++ [7]. These models have higher performance than the models of the Siamese-FC series, but these network structures are more complex, with a large number of network parameters, and require a large number of training sets or a series of improved data enhancement strategies. For example, the latest SiamRPN ++ trackers have 7.1 G FLOPS and 11.2M parameters, which makes it difficult to deploy such a model to resource-constrained devices, while earlier Siamese-FC used only 2.7G FLOPS and 2.3M parameters. There are two broad approaches to bridging the gap between an academic model and an industrial deployment of tracking. One is model compression, and the other is the manual design of compact models. Existing compression techniques, such as pruning and quantizing, can reduce the complexity of the model to varying degrees, but inevitably lead to a great reduction in the performance of the tracking model [8, 9], so we choose to design a compact model by hand.

The contributions of this paper are as follows: In this paper, we propose a full convolutional connected network method based on global attention, which can generate a discriminant template that is robust to tracking target changes. In this paper, only one ILSVRC dataset is needed as the training set, rather than a large number of data sets like DaSiamRPN. The method used can achieve faster convergence with fewer iterations. The starting point of this paper is to improve the simple tracking models, such as Siamese-FC and CFNet [10], to better cope with the challenges of deformation, similar semantic information, or illumination changes in the tracking process. Compared with GradNet and DaSiamRPN, these models only require more common devices to achieve similar performance, to verify the algorithm proposed in this paper Effectiveness. This paper conducts comparative experiments on the ILSVRC, OTB100, VOT2018 datasets. The experimental results show that the method proposed in this paper can effectively improve the discriminative ability and robustness of the online tracker while ensuring a higher tracking speed. Through a series of improvements to lightweight SiamFC, the improved trace model can be deployed to resource-constrained platforms in real-time. The organizational structure of this paper is as follows: introduction, related work, method, experiment, and conclusion.

*e-mail: jhhxw@zjnu.edu.cn

Manuscript submitted 2021-07-15, revised 2021-10-09, initially accepted for publication 2021-10-13, published in June 2022.

## 2. RELATED WORKS

The design of traditional target tracking algorithm is based on the method of correlation filter [11], such as the mobile target tracking method using improved particle filter [12], the target tracking method based on color correlation histogram and particle filter [13], and Kalman filter, a combined adaptive window anti-occlusion moving target tracking algorithm [14], and the use of background attention correlation filtering [15] and three-dimensional Gabor filtering [16] for tracker design. This method uses training and learning filters to separate the tracked object from the background to achieve video tracking. However, these methods can only work well when the background is simple, and the object deformation is small, and high-quality online update strategy is required. The extensive application of deep learning has found new directions for the development of target tracking, which are mainly reflected in the following two aspects: On the one hand, the use of deep features in the correlation filter can improve the accuracy of the algorithm. The literature [17] extracts deep features and uses the integrated idea to obtain a stronger tracking model. The literature [18] introduces continuous convolution operators to solve the learning problem of continuous space; On the other hand, directly using deep learning algorithms for target tracking, such as multi-domain convolutional neural network structure system network is applied to target tracking [19], and literature [20] proposed an online visual tracking algorithm based on a tree structure to manage multiple target appearance models, with better results. Studies have shown that the direct use of deep learning algorithms for end-to-end target tracking is more suitable for development needs. SO-DLT [21] follows the tracking algorithm DLT [22], which uses the deep model for single-target tracking tasks, adds online fine-tuning strategies to the tracking data pre-training, and designs a targeted network structure, and solves the updated Sensitive problems, taking multiple values for specific parameters for smoothing, but it is difficult to meet real-time requirements. literature [23] regards target tracking as a regression problem and uses CNN to directly return the location of the tracking object. The Siamese candidate region generation network [24] is the target tracking problem is regarded as a target detection problem, and the idea of regional regression in the fast regional convolutional neural network [25] is used to locate the position of the tracked object. The fully convolutional Siamese neural network (Siamese-FC) [26] regards tracking as similarity learning, learning a discriminative model to locate the position of the object's center point through a deep convolutional neural network. Although the Siamese-RPN [27] and GOTURN [28] algorithms have high tracking capabilities, the former requires more parameters to be set, so a large amount of training data will be added, and the latter cannot meet the real-time requirements.

Literature [29] proposed the use of Siamese neural networks for similarity measurement to solve the problem of face recognition. The two sub-networks of the Siamese network obtain two identical sub-networks through weight sharing and then judge the similarity of the two inputs. The Siamese network has the characteristics of naturally increasing training data and provides a way to solve problems in tracking areas with less training data. Literature [30] first proposed a target tracking algorithm based on a Siamese network. The algorithm treats target tracking as a matching problem through a neural network, but it needs to process a large number of candidate frames each time, so it takes a long time. Literature [31] proposed a real-time Siamese-FC algorithm, which is also a target tracking algorithm based on the Siamese network, which can select and locate candidate images in a larger search image. The network training is carried out in a full convolution method. Finally, the two input relationships are represented by return values. A high score indicates that the two input objects are the same; a low score indicates that the two input objects are different. The network structure of Siamese-FC is shown in Fig. 1, where input x often represents a target with a given first frame. This sub-network is defined as a template branch, the other sub-network is defined as a detection branch and receives the input of the current frame, usually expressed by $z$. Both branches of Siamese-FC use deep convolution to perform feature extraction transformation, denoted by $\varphi$, and then $f(x,z) = g(\varphi(x), \varphi(z))$ to calculate the correlation between the two inputs, where $g$ represents the convolution operation, $\varphi(x)$, $\varphi(z)$ represents the convolution kernel.

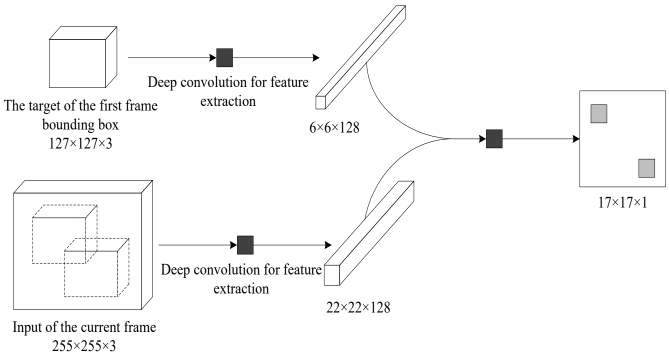
Deep convolution for feature extraction
The target of the first frame bounding box
127×127×3
6×6×128
17×17×1
Deep convolution for feature extraction
22×22×128
Input of the current frame
255×255×3

**Fig. 1.** Siamese-FC structure

The literature [32] presents LightTrack, which uses neural architecture search (NAS) to design more lightweight and efficient object trackers. Comprehensive experiments show that their LightTrack is effective. It can find trackers that achieve superior performance compared to handcrafted SOTA trackers, such as SiamRPN++ and Ocean while using much fewer model Flops and parameters. However, their starting point is still the SiamRPN++ series model with a high number of parameters, and NAS has high equipment requirements and a long search time, which is obviously inconsistent with our original design.

## 3. THE METHOD OF THIS PAPER

The main goal of this paper is to optimize the feature extraction of the network by introducing mechanisms such as attention so that more stable features can be extracted, and the problem that the previous algorithms are easily affected by external conditions can be solved. This article is based on the Siamese-FC

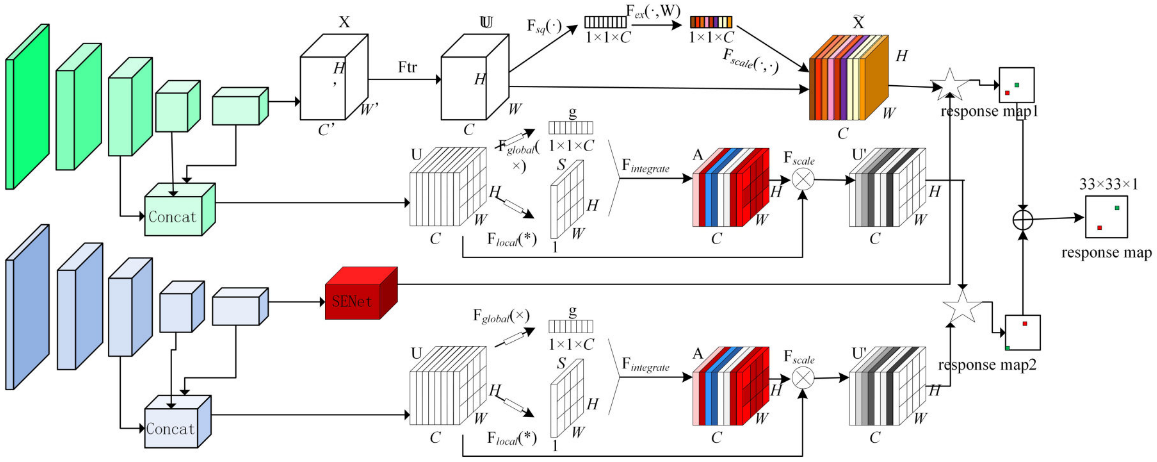

**Fig. 2.** The global attention target tracking network

algorithm. The algorithm's deep learning framework ensures real-time performance. The network is increased into two parallel subnets, one of which integrates and optimizes features, and adds a global attention module, so that the extracted features have more effective information such as space and semantics, thereby improving the tracking accuracy of the algorithm; The other subnet improves the recognition ability of target features by adding self-attention mechanism, which enables the algorithm to distinguish between target and local interference.

The overall structure of this paper is shown in Fig. 2, which mainly includes the basic framework of the Siamese-FC network, a feature fusion module, a global attention module that integrates spatial attention and channel attention, and a self-attention module. The algorithm is composed of two parts: the subnetwork of feature fusion and global attention module and the self-attention subnetwork. The response graph obtained by the infrared image through the two sub-networks is then fused by the average method to obtain the final response graph.

### 3.1. Self-attention model

In the self-attention subnetwork, we use the compression excitation network mechanism and use the global information to explicitly model the dynamic and nonlinear dependence between channels, which can simplify the learning process and significantly enhance the representation ability of the network. The compression excitation network first performs the Squeeze operation. The operation object is the intermediate result obtained after the traditional convolution operation. Its purpose is to compress the spatial dimension and change the channel information of each dimension into a real number with a global receptive field. The output dimension is the same as the number of input feature channels; secondly, an Excitation operation similar to the gate mechanism in the recurrent network is performed, and the correlation between the feature channels is explicitly modeled according to the channel parameter w; finally, the Excitation output the weight of is used as a representative of the importance of each channel after feature selection. Reweight performs item-by-item multiplication and weighting on the original features to complete the recalibration of the original features in the channel dimension. SENet has a simple structure and can be used as an independent module to be embedded in different network structures. Its structure is shown in Fig. 3.

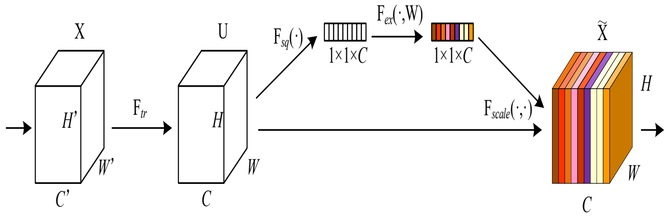

**Fig. 3.** SENet structure

### 3.2. Multi-layer feature fusion

The previous Siamese networks [33–35] used the features obtained in the last layer of the network as output features, which is enough to represent the tracking target. Since the deep feature receptive field is large, it has discriminative semantic feature information, and the resolution is low, while the shallow feature receptive field is small, it has strong spatial structure information, and the resolution is high, so it is not enough to use the last layer of features alone. Feature fusion is an optimization operation for feature maps. The output features of multiple convolutional layers are cascaded at a uniform resolution to obtain a new fusion feature map with more channels, that is, more image characterization information [36]. When selecting feature maps for fusion, considering that although the first two layers of features contain rich spatial information, they are large in size, and the unification process will lose too much feature information. Starting from the third layer, using the last three layers of features to perform Fusion not only solves the problem of excessive information loss, but also contains rich spatial and semantic information. Therefore, the algorithm in this paper fuses the last three features of the Siamese-FC feature extraction network. Of course, we also confirmed the effectiveness of convolutional feature fusion through ablation experiments. The shallow information is used to locate the target location, and the deep information is used to distinguish different objects. To effectively fuse the multi-layer features with different resolutions together, the shallow features are pooled to the maximum to obtain the same resolution as the deep features, and after normalization to balance the effects of the three-layer features, the connection is obtained Fusion feature map, considering that the dimension of the feature map is too large, after the connection, $1\times 1$ convolution is used to reduce the dimensionality of the fusion feature map to reduce the training time. The fusion process can be expressed as:

$$f_{\text{fusion}} = \text{concat}_{i=3,4,5}\left\{bn\left[mp\left(f_{\text{conv}i}\right)\right]\right\}, \tag{1}$$

$$f_{\text{final}} = \text{conv}\left(f_{\text{fusion}}\right). \tag{2}$$

In the above formula: $f_{\text{conv3}}$, $f_{\text{conv4}}$, $f_{\text{conv5}}$, $f_{\text{fusion}}$, and $f_{\text{final}}$ represent the output features of the corresponding convolutional layer, the fused features, and the final features after dimensionality reduction. The mp represents the maximum pooling operation. The bn represents batch normalization. The concat represents features Connect and perform feature concatenation operation. The size of the fused feature map fusion obtained in this step is $49\times 49\times 832$; conv represents the convolution of $1\times 1$, and the final feature map final size obtained is $49\times 49\times 256$.

### 3.3. Global attention module

After obtaining the fused multi-layer features through (1) and (2), it already contains both spatial information and semantic information. On this basis, we hope that the network model can have global awareness, that is, it can deal with the target deformation, rotation, and other changes while assigning corresponding weights to characteristic channels to obtain the importance of each channel so that the algorithm shows robustness to both space and channel information. In response to this problem, this article established a global attention module, as shown in Fig. 4.

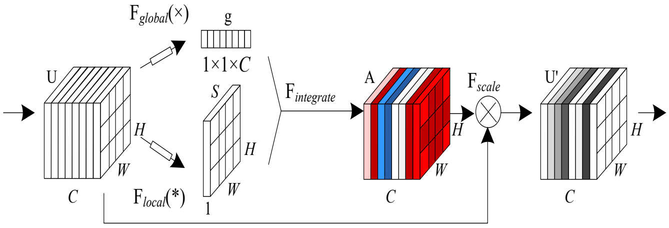

**Fig. 4.** Global attention module structure

As an independent module that can be inserted into the convolutional neural network, the global attention module takes the feature map as the input and outputs it in the original size after affine transformation, to optimize the feature map.

The Fig. 4 here shows a global attention module applied to the feature activity from a convolutional layer in a deep network model, $U$. Information in the diagram flows from left to right, and $U$ is processed with separate local ($F_{\text{local}}$) and global ($F_{\text{global}}$) operators to derive the local- and global-attention masks, which is integrated into the attention activity $A$. Attention is applied to the original activity, with elementwise multiplication ($F_{\text{scale}}$), to yield $U'$.

The global attention module learns to combine local saliency with global contextual information to guide attention towards image regions that are diagnostic for object detection. Spatial attention and channel attention, which are used to capture pixel-level pair relationships and channel dependencies, respectively. Fusing them together gives better performance than their separate implementations, which inevitably increases computational overhead. The effect of the global attention module will be displayed in the heat map of the experimental part.

The global attention module modulates an input layer activity $U$ with an attention mask $A$ of the same dimension as the input, which captures a combination of global and local forms of attention. Here, the spatial height, width and the number of the features are represented as $H$, $W$, $C$ respectively s.t. $U, A \in R^{H\times W\times C}$ . The global attention module proposed in this paper is based on the SE module (Hu *et al.*, 2017 [37]). Here global attention is represented by $F_{\text{global}}$ in this model and yields the global feature attention vector $g \in R^{1\times 1\times}C$. Its generation process can be roughly decomposed into two steps. Firstly, channel-by-channel statistics. Secondly, the data of channel-by-channel statistics are nonlinear transformed by multi-layer perceptron (MLP). Summary statistics are computed with a global average pooling applied to the feature maps. Here, $U = [u_k]_{k=1,\ldots,C}$ can yield vector $P = (P_k)_{k=1,\ldots,C}$, $P_k = \dfrac{1}{WH}\sum\limits_{x=1}^{W}\sum\limits_{y=1}^{H} u_{kxy}$. Next, two layers of MLP are used to reduce and extend the dimensions of $P$. An intervening nonlinearity enables the module to learn complex dependencies between channels. The reducing of MLP is applied to vector $P$ by the operator $w_{\text{reduce}} \in R^{\frac{c}{r}\times C}$, leading to a reduction in the dimension of $P$. This is followed by an expansion operation $w_{\text{expand}} \in R^{C\times\frac{c}{r}}$, return to original, dimensional space.

$$g = w_{\text{expand}}\left(\sigma\left(w_{\text{reduce}}(P)\right)\right). \tag{3}$$

Here $\sigma$ is a corrected linear function (Relu), the "reduction ratio" $r$ is a super parameter, and it is better to set $r$ to 4 in the experimental test.

After the global (channel) attention module, we introduce the spatial attention module to focus on where features make sense. The module consists of three sub-modules, namely positioning network, grid generator, and sampler, as shown in the spatial transformation section in Fig. 4. Firstly, the location network receives the input feature map $U$ and obtains the change parameters after passing through the hidden layer $\theta$. In getting $\theta$, after that, the matrix operation is further carried out, and the coordinates of the original feature graph $U$ are obtained by taking the coordinates of the pixel in the target feature graph $S$ (Fig. 5) as the independent variables, to obtain the pixel value of the point in $S$.

$$S = v_{\text{collapse}} * \left(\sigma\left(v_{\text{reduce}} * U\right)\right). \tag{4}$$

Here, convolution is represented as $*$, $v_{\text{collapse}} \in R^{1\times 1\times\frac{c}{r}\times 1}$, and $v_{\text{reduce}} \in R^{1\times 1\times C\times\frac{c}{r}}$. The $F_{\text{integrate}}$ operation consolidates the output of the local and global path to generate attention volume $A \in R^{H\times W\times C}$. A lot of times, it wasn't clear whether the model would benefit more from $F_{\text{local}}$ or $F_{\text{global}}$ attention, or whether tasks would perform better if they were combined or multiplied, so we use both addition and multiplication for both types of attention, which have equal weight. To combine attention activities $g$ and $S$, they are first tiled to produce $G^*$, $S^* \in R^{H\times W\times C}$. Finally, we calculate the attention activities of a global attention module as:

$$A_{h,w,c} = \varsigma\left((G^*_{h,w,c} + S^*_{h,w,c}) + (G^*_{h,w,c} \cdot S^*_{h,w,c})\right). \tag{5}$$

The $\varsigma$ of equation (5) is set to the tanh function, which squashes activities in the range [–1, 1]. Finally, attention is applied by $F_{\text{scale}}$ as:

$$U' = U \odot A. \tag{6}$$

### 3.4. Loss function

We used the Logistic loss function to optimize the tracker we proposed.

$$\ell(y,v) = \log(1+\exp(-yv)), \tag{7}$$

$$L(y,v) = \frac{1}{|D|}\sum_{u\in D} \ell\left(y[u],\, v[u]\right). \tag{8}$$

Here, $v$ is tracking a single response value of the network output, $y$ is the actual value, and $y \in \{-1,+1\}$, $D$ has generated heatmap, $u$ for a certain value in $D$, $|D|$ for the size of the heatmap. The ground-truth in the heatmap is marked according to the following formula:

$$y[u] = \begin{cases} +1, & \text{if } k\|u-c\| \le R, \\ -1, & \text{otherwise.} \end{cases} \tag{9}$$

Formula 9c for objects in the center of the heatmap, $u$ for any point in the heatmap, $||u-c||$ is the distance between $u$ and $c$, here distance calculation using Euclidean distance, $R$ for distance threshold, $k$ for heatmap multiples of narrow after network. $k$ is the multiple of the heatmap narrowed after passing through the network.

The convolution parameter $\theta$ is obtained by using SGD method to minimize the loss function:

$$\arg\min_{\theta} E_{z,x,y} L\left(y, f\left(z, x : \theta\right)\right). \tag{10}$$

## 4. EXPERIMENTS

First of all, we provide experimental details. Second, our methods are compared with state-of-the-art trackers. Meanwhile, we carry out ablation studies to analyze the modules and framework of the model.

### 4.1. Implementation details

The network structure: the backbone of the Siamese network is AlexNet. To be fairer in comparison with the original benchmark tracker, our tracker was also trained using ILSVRC2015 datasets. The dataset in this article is the video data set of the ImageNet Visual Recognition Challenge (ILSVRC). As part of the new object detection in the video challenge, it provides about 4500 (divided into training set and validation set) containing 30 different images of animals and vehicles. Video information, more than one million frames with annotations. The advantage of the ILSVRC dataset in video tracking is that it not only contains more data but the scenes and objects it depicts are different from those in the classic tracking benchmarks. Therefore, this paper chooses the ILSVRC data set to train the deep tracking model to have more generalization ability. In the training process, the data frames used for training are extracted from the 4417 frames of video of Imagenet, and there are more than 2 million labeled bounding boxes in total. The data set is randomly divided into three parts: 70% of the training set, 10% of the validation set, and 20% of the test set. Two performance indicators are used to evaluate the tracking effect: accuracy (accuracy) and frame rate (FPS). The former is calculated in the form of average IoU, and the latter is the number of pictures that can be processed per second.

The proposed global attention network is implemented in Python with Pytorch on RTX3090. The software configuration: Ubuntu18.0, CUDA11.2, cuDNN11.2, Pycharm.

### 4.2. Evaluation of single-size test data

We know that in the tracking models of SiamFC and CFNet, the size of the target box of the first frame image is fixed at $127\times127\times3$, and the size of the search image is $255\times255\times3$. We try to explore whether there are drawbacks in the method of fixing the size of training and testing data. The template image of each frame is extracted offline, and the detection image size is set as follows: For a fixed $S_1 \in [S_{1\min}, S_{1\max}]$, $S_2 \in [S_{2\min}, S_{2\max}]$, the tracking performance of the independent network model obtained in Table 1 is evaluated, and the results are listed in Table 3. $S_1$ represents the clipping size of the template branch in the training dataset, $S_2$ represents the clipping size of the search branch image in the training dataset, $Q_1$ represents the clipping size of the template branch in the test dataset, and $Q_2$ represents the clipping size of search branch image in the test dataset.

It can be seen from Table 1 that the multi-scale training network performs better in the size fluctuation test (compared with the fixed training size). That is to say, when we input not only multi-scale training data but also multi-scale test data, our model will get better performance. The last group of multi-scale network training has the best performance, the scale fluctuation performance is better than the fixed minimum edge performance. That is, the cutting size of the template frame in the training set changes in the interval [127, 255], and the cutting size of the search frame changes in the interval [255, 383]. In addition to retaining the cutting size of the template frame before the test, the cutting size of the template frame is $127\times127$, and the search frame is $255\times255$. A test policy with test template frame size $191\times191$, search frame size $319\times319$, and test template frame size $255\times255$, search frame size $383\times383$ have also been added. The accuracy rate of the optimal single-network model in this paper is 0.610 on the verification set and 0.651 on the test set.

**Table 1**
Network performance of multi-scale

| Picture minimum size | | Accuracy | Speed (frame·s$^{-1}$) |
|---|---|---|---|
| train ($S_1, S_2$) | test ($Q_1, Q_2$) | | |
| 127, 255 | (111, 239),<br>(127, 255),<br>(143, 271) | 0.530 | 158 |
| 127, 255 | (111, 239),<br>(127, 255),<br>(143, 271) | 0.531 | 157 |
| 255, 383 | (239, 267),<br>(255, 383),<br>(271, 399) | 0.532 | |
| [127, 255],<br>[255, 383] | (127, 255),<br>(191, 319),<br>(255, 383) | 0.545 | |
| 127, 255 | (111, 239),<br>(127, 255),<br>(143, 271) | 0.563 | 157 |
| 255, 383 | (239, 267),<br>(255, 383),<br>(271, 399) | 0.574 | |
| [127, 255],<br>[255, 383] | (127, 255),<br>(191, 319),<br>(255, 383) | 0.580 | |
| 127, 255 | (111, 239),<br>(127, 255),<br>(143, 271) | 0.586 | 156 |
| 255, 383 | (239, 267),<br>(255, 383),<br>(271, 399) | 0.617 | |
| [127, 255],<br>[255, 383] | (127, 255),<br>(191, 319),<br>(255, 383) | 0.629 | |
| 127, 255 | (111, 239),<br>(127, 255),<br>(143, 271) | 0.637 | 157 |
| 255, 383 | (239, 267),<br>(255, 383),<br>(271, 399) | 0.643 | |
| [127, 255],<br>[255, 383] | (127, 255),<br>(191, 319),<br>(255, 383) | **0.651** | 157 |

**Table 2**
Comparison of test results of different tracking algorithms

| Method | Accuracy | Speed (frame·s$^{-1}$) |
|---|---|---|
| DAT [38] | 0.4720 | 115 |
| SENet [37] | 0.5399 | 152 |
| Siamese-FC-3s [4] | 0.5950 | 187 |
| Siamese-FC [4] | 0.6270 | 158 |
| Siam-SEFC (Ours) | **0.6511** | 157 |

Table 2 compares the test results of different tracking algorithms. It can be seen from Table 3 that both the accuracy and the speed of processing pictures are not as good as the Ours algorithm in terms of accuracy and processing speed. Compared with the Siamese-FC algorithm, our algorithm has higher accuracy and accuracy. The performance can reach 0.6511 without loss of real-time performance, and the image processing speed FPS can still reach 157 frames/s. Therefore, the Siam-SEFC tracking model has good tracking performance.

To sum up, this module aims at the Siamese-FC algorithm without a strategy for learning object size information changes. By introducing the SENet module into the Siamese-FC backbone network. Our algorithm uses the characteristics of the SENet module to learn the central information features of the object while adding the learning of spatial information features and uses multi-scale data for training and testing, which further adds information about the size of the object on the object scale. Through single-size, multi-size test experiments and comparison experiments with other tracking methods, the effect of Siam-SEFC in video tracking is evaluated. The results show that the Siam-SEFC model not only has good accuracy on the ILSVRC15 dataset but also meets real-time performance.

### 4.3. Comparison with the state of the arts

Extensive experiments are conducted to evaluate the proposed tracker against other state-of-the-arts VOT2018, OTB100 benchmarks. All the experiments were carried out through the official toolkits.

**OTB Benchmarks.** OTB100 contains 100 different video sequences and is often used to evaluate the performance of trackers. The evaluation of OTB follows the standard protocols and uses two metrics to rank trackers i.e., precision plot and success plot.

The proposed tracking algorithm is compared with other real-time tracking algorithms based on the OTB evaluation benchmark. Real-time tracking algorithms Siam-Fc, DaSiamRPN, CFNet, and GradNet are evaluated once (OPE). The accuracy and success rate are shown in Fig. 5 and Fig. 6.

Figure 7 shows the success rate of the improved model under motion blur. Figure 8 shows the precision rate of the improved model under motion blur.

As shown in Table 3, our tracker shows a good performance. In a more compact architecture, its performance is very close to GradNet in OTB100. The baseline indicates that the network shuts down the global attention module and the self-attention module, and only adds the strategy of feature fusion.

**Table 3**
Contrast results of the tracker

| Tracker name | Success | Norm Precision | Precision |
|---|---|---|---|
| DaSiamRPN[6] | **0.658** | 0.000 | **0.880** |
| Ours | 0.639 | 0.000 | 0.845 |
| GradNet[5] | 0.639 | 0.000 | 0.861 |
| Baseline | 0.601 | 0.000 | 0.794 |
| CFNet[10] | 0.587 | 0.000 | 0.778 |
| SiamFC[4] | 0.587 | 0.000 | 0.772 |

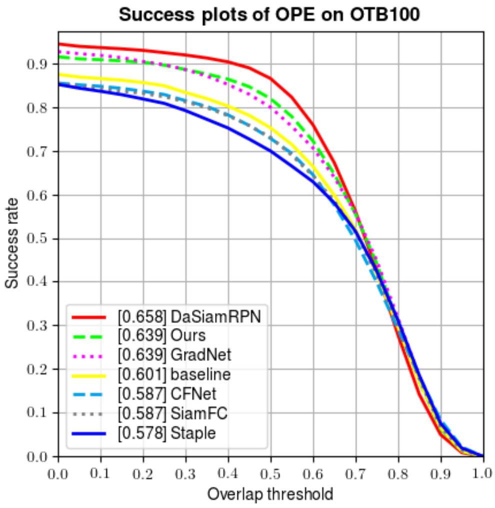

**Fig. 5.** Graph of success

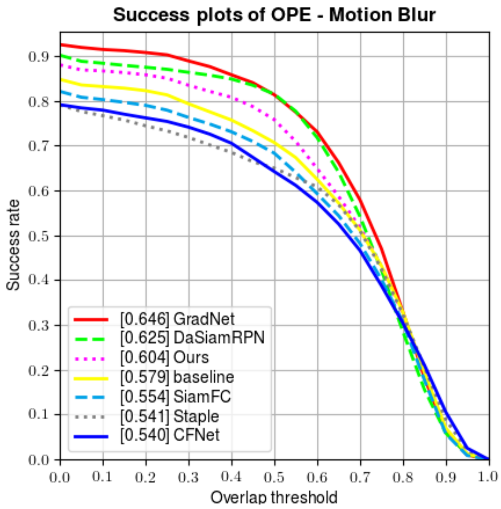

**Fig. 7.** Graph of success under motion blur

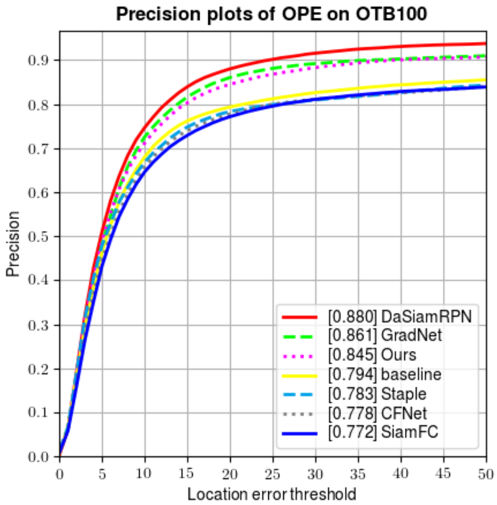

**Fig. 6.** Graph of precision

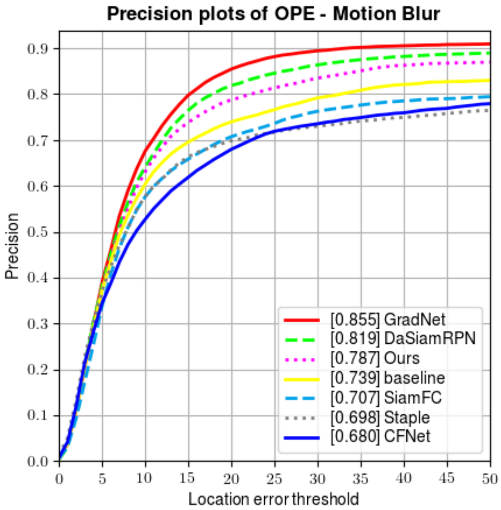

**Fig. 8.** Graph of precision under motion blur

Bold data indicates the best results, and underlined data indicates the second best. The baseline indicates that the network shuts down the global awareness module and the self-attention module, and only adds the strategy of feature fusion

It can be seen that compared with CFNet and SiamFC, it has some improvement effect. On Suc score and Pre. score, the baseline tracker has an improvement of 1.4% and 1.6% compared to CFNet. When the self-attention module and the global awareness module are turned on, the proposed tracking algorithm (Ours) Suc and Pre reach 0.640 and 0.845 respectively, which are 5.2% and 6.7% higher than that of CFNet and 3.8% and 5.1% higher than that of the baseline. Judging from the dataset OTB100, the tracking algorithm we proposed has a certain improvement effect compared with the original model. Although it is regrettable that there is a certain gap between the performance of the tracker we proposed and that of DaSiamRPN in OTB100, after all, in order to make a more fair comparison with the benchmark (Siamese-FC) in this paper, we only used ILSVRC2015-VID for the training dataset. It is important to note that the original intention of this work is to use the lightest possible tracking model, which is as close as possible to the performance of some of the best recent tracking algorithms while using only one training dataset. In addition to ILSVRC2015-vid and YouTube-BB data sets, DaSiamRPN also introduced static images from ILSVRC2015 and COCO datasets and expanded the types of positive sample pairs through a series of enhancement methods (translation, resizing, grayscale, etc.) to improve the discrimination ability of the tracking network. So, it makes perfect sense that our tracker is slightly inferior to DaSiamRPN.

In Fig. 9, in the Soccer and Bird 1 video sequences, both the benchmark networks SiamFc and CFNet in this paper failed to track the target when they encountered interference with similar semantic information while tracking, and our proposed tracker (BoundingBox in red) was able to meet this challenge well. In the MotorRolling video sequence, the benchmark network is also unable to successfully deal with the challenge of rotation, and the tracker we proposed can still deal with it well. The good performance of the proposed tracker benefits from the complementarity of more details extracted from the self attention module and the global attention module. Of course, multi-layer feature fusion also has some effects.

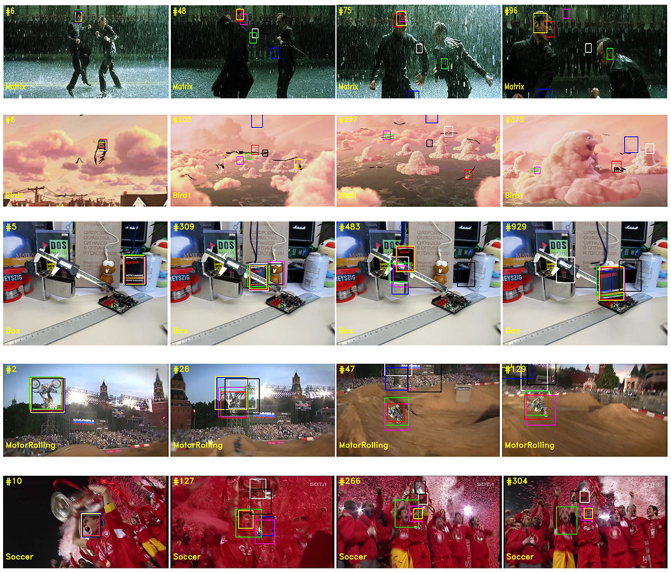

**Fig. 9.** Different tracking algorithms can be compared and viewed

To better analyze the tracking performance of the proposed tracker, we used actual experiments to challenge the tracker with different attributes, such as fast movement, in-plane rotation, scale transformation, illumination change, and beyond the field of view Fig. 10 shows the qualitative results, which show that the proposed tracker can complete the tracking challenge task under the condition of guaranteed verification time.

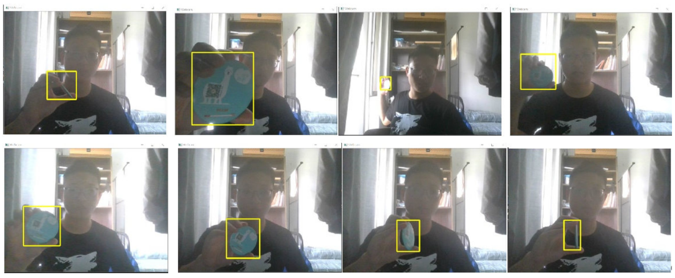

**Fig. 10.** Multi-attribute target tracking test in real scenario

In Fig. 10, the experimenter holds the target (circular mirror) to be tracked and performs a series of rapid movements, plane rotation, scale scaling, and exposure processing on the target (circular mirror). It can be seen that the tracking algorithm proposed in this paper can complete the tracking task well under these challenges.

### 4.4. Ablation experiments

In order to verify the effectiveness of the multi-layer feature fusion and global awareness module, we conducted an evaluation experiment on the dataset VOT2018. In order to more intuitively reflect the effect of these modules, we conducted heatmap visualization on the tracked video frames, as shown in Fig. 11.

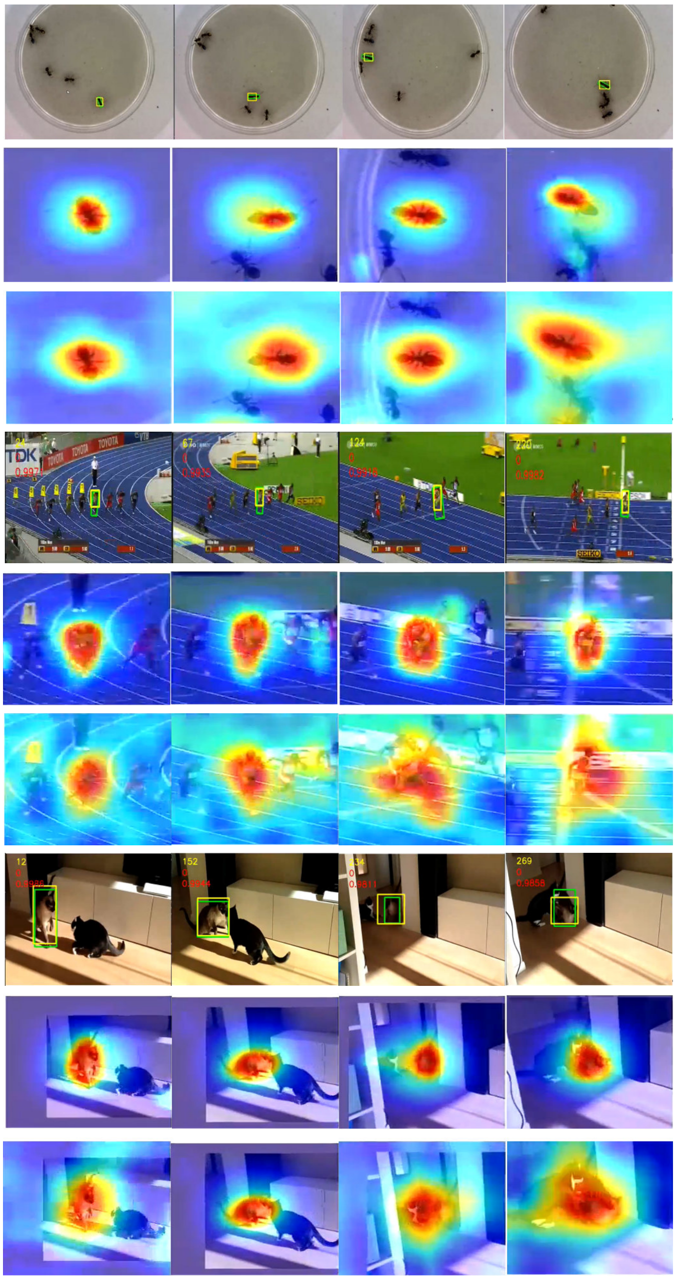

**Fig. 11.** Comparison of thermal map effects of the modified module ablation experiment on the VOT2018 dataset

**VOT Benchmarks.** Our proposed tracker is evaluated in VOT2018 which contain 60 various challenges sequences to evaluate the performance. The VOT provides expected average overlap (EAO), accuracy ($A$), and robustness ($R$) as metrics. Table 4 reports the EAO on VOT2018, which shows the competitive results with real-time speed. On the EAO of VOT2018, the proposed tracking algorithm has an improvement of 6% compared to SiamFc.

**Table 4**
Performance comparison on Vot2018 dataset

| Traker | VOT2018 | | |
|---|---|---|---|
| | EAO | *A* | *R* |
| CCASiam [39] | 0.287 | 0.540 | 0.380 |
| Ours+MLFF+GAM | 0.266 | 0.491 | 0.435 |
| Ours+MLFF-GAM | 0.234 | 0.489 | 0.458 |
| Ours-MLFF-GAM | 0.218 | 0.480 | 0.502 |
| SA-Siam [40] | 0.236 | 0.500 | 0.459 |
| SiamFC [4] | 0.206 | 0.511 | 0.627 |

The ablation experiment of this part is based on the SiamFC algorithm.

When the multi-layer feature fusion is not used and the global attention module is removed, the proposed model still has a 1.2% EAO improvement compared to the benchmark algorithm. When the multi-layer feature fusion is enabled, the proposed algorithm still has a 1.6% EAO improvement. When the global attention module (GAM) is enabled, our algorithm continues to have a 3.2% EAO improvement. It can be seen from the ablation experiment that the improvement strategies proposed in this paper are all effective, and the improvement effect brought by the global attention module is the most obvious.

Figure 11 in each group of the first row of the video sequence of video images said VOT2018 test video sequences, the second-row heat maps for open global attention module and feature fusion tracking network module performance, after the third global attention module behavior was off online tracking performance, it can be seen that each group of the third row compared the heat effect of a drop in the second video sequence.

Red font is best, green font is second. MIFF represents multilayer feature fusion, GAM represents globally attention module, and plus and minus signs indicate whether or not these modules are turned on

## 5. CONCLUSIONS

The tracking algorithm based on Siamese networks is easy to fail in the face of deformation, similar semantic information interference and scale change. To solve this problem, a full-convolution Siamese lightweight network method based on global awareness is proposed in this paper, which can be updated online in real time. The SiamFC algorithm, which has no change in the size information of strategy learning objects, is targeted in this paper. By introducing the SeNet module into the SiamFC backbone network, the features of the SeNet module are used to learn the central information features of the object, and the multi-scale data are used for training and testing, and the size information of the object is further added to the object scale. In addition, features were extracted by the AlexNet L3 layer, L4 layer, and L5 layer, and feature fusion was performed by a multi-scale feature fusion module. In this way, improved networks can be simpler and better able to learn effective features. And we also propose a global attention module. As an independent module that can be inserted into the convolutional neural network, the global attention module takes the feature map as the input and outputs it in the original size after affine transformation, to optimize the feature map. In order to verify the effectiveness of the proposed algorithm, this paper has done comparative experiments on the ILSVRC, OTB100, Vot2018 datasets. The experimental results show that the method proposed in this paper can significantly improve the performance of the benchmark algorithm. Although the algorithm we proposed has significantly improved its performance compared with the benchmark network, there are still big differences compared with some better tracking algorithms at present. The main reason for the difference is that our work uses a SiamFC lightweight framework rather than a complex network model like SiamRPN, and we use only one training dataset rather than multiple datasets like other tracking algorithms. Our original intention is to allow the lightweight SIAMFC with low equipment requirements to meet or even exceed some excellent tracking algorithms with complex structures and extensive training, thus closing the gap between the academic model and the industrial deployment of target tracking.

There is still the problem of tracking failure in some video sequences of VOT2018, which may be the challenge to be completed in the future. Perhaps improvements to better frameworks such as SiamRPN ++ will overcome these trace failures.

## ACKNOWLEDGEMENTS

The authors acknowledge the financial support from the National Natural Science Foundation of China (NSFC) (61572023, 61672467) and the Natural Science Foundation of Zhejiang Province (Z22F023843).

## REFERENCES

[1] E. Kot, Z. Krawczyk, K. Siwek, L. Krolicki, and P. Czarnowski, "Deep learning based framework for tumour detection and semantic segmentation," *Bull. Pol. Acad. Sci. Tech. Sci.*, p. e136750, 2021, doi: 10.24425/bpasts.2021.136750.

[2] A.M. Osowska-Kurczab, T. Markiewicz, M. Dziekiewicz and M. Lorent, "Combining texture analysis and deep learning in renal tumour classification task," *2020 IEEE 21st International Conference on Computational Problems of Electrical Engineering (CPEE)*, 2020, pp. 1–4, doi: 10.1109/CPEE50798.2020.9238757.

[3] Z. Krawczyk and J. Starzyński, "Segmentation of bone structures with the use of deep learning techniques," *Bull. Pol. Acad. Sci. Tech. Sci.*, vol. 69, p. e136751, 2021, doi: 10.24425/bpasts.2021.136751.

[4] L. Bertinetto *et al*, "Fully-Convolutional Siamese Networks for Object Tracking," *Computer Vision – ECCV 2016 Workshops. ECCV 2016. Lecture Notes in Computer Science*, vol. 9914, doi: 10.1007/978-3-319-48881-3_56.

[5] P. Li, B. Chen, W. Ouyang, D. Wang, X. Yang, and H. Lu, "GradNet: Gradient-Guided Network for Visual Object Tracking," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 6161–6170, doi: 10.1109/ICCV.2019.00626.

[6] Z. Zhu, *et al*, "Distractor-aware Siamese Networks for Visual Object Tracking," *Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science*, vol. 11213, pp. 103–119, 2018, doi: 10.1007/978-3-030-01240-3_7.

[7] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "Siam RPN++: Evolution of Siamese Visual Tracking With Very Deep Networks," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 4277–4286, doi: 10.1109/CVPR.2019.00441.

[8] S. Han, H. Mao, and W.J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *4th International Conference on Learning Representations, ICLR 2016*, 2016

[9] Y. Liu, X. Dong, W. Wang, and J. Shen, "Teacher-students knowledge distillation for siamese trackers," Available: https://arxiv.org/abs/1907.10586.

[10] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P.H.S. Torr, "End-to-End Representation Learning for Correlation Filter Based Tracking," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5000–5008, doi: 10.1109/CVPR.2017.531.

[11] T. Wang, "DenseNet Siamese network target tracking with global context feature module," *J. Electron. Inf. Technol.*, vol. 43, no. 1, pp. 179–186, 2021, doi: 10.11999/JEIT190788.

[12] X. Yang *et al.*, "An improved target tracking algorithm based on spatio-temporal context under occlusions," *Multidimension. Syst. Signal. Process.*, vol. 31, no. 2, pp. 329–344, 2020, doi: 10.1007/s11045-019-00664-5.

[13] Y.F. Zhang *et al*, "A high robust real-time single-target ship tracking method based on siamese network," Available: http://en.cnki.com.cn/Article_en/CJFDTotal-JCKX2019230 22.html.

[14] Z. Sha and H. Yuqing, "UAV designated target tracking based on Siamese area candidate network," *IEEE Comput. Appl. Power*, vol. 41, no. 2, pp. 523–529, 2021.

[15] W. Xiang, Z.G. Yuxuan, Y. Qiqi, and L. Xiaomao, "Scale-adaptive sea surface target tracking algorithm based on deep learning," *J. Underwater Unmanned Syst.*, vol. 28, no. 6, pp. 618–625, 2020.

[16] Z. Xingchen *et al.*, "DSiamMFT: An RGB-T fusion tracking method via dynamic Siamese networks using multi-layer feature fusion," *Signal Process. Image Commun.*, vol. 84, p. 115756, 2020, doi: 10.1016/j.image.2019.115756.

[17] G. Bhat *et al.*, "Learning Discriminative Model Prediction for Tracking," *2019 ICCV*, pp. 6182–6191.

[18] Y. Fan and X. Song , "Siamese Progressive Attention-Guided Fusion Network for Object Tracking," *Comput-Aided Des. Comput. Graphics*, vol. 33, no. 2, pp. 199–206, doi: 10.3724/SP.J.1089.2021.18392.

[19] L. Enhan, Z. Rui, Z. Shuo, and W. Ru, "An infrared pedestrian target tracking method based on video prediction," *J. Harbin Inst. Technol.*, vol. 52, no. 10, pp. 192–200, 2020.

[20] Ch. Lei, W. Yue, and T. Chunna, "A visual target tracking algorithm with residual attention mechanism," *J. Xidian Univ.*, vol. 47, no. 6, pp. 148–157+163, 2020.

[21] K. Jie, S. Yang, and S. Junge, "Siamese network target tracking based on difficult sample mining," *Comput. Appl. Res.*, vol. 38, no. 4, pp. 1216–1219+1223, 2021.

[22] C. Xi, H. Yifeng, Y. Yunfeng, Q. Donglian, and S. Jianxin, "Target intelligent tracking and segmentation fusion algorithm and its application in substation video surveillance," *2020 Electr. Eng. Conf.*, vol. 40, no. 23, pp. 7578–7587, 2020.

[23] W. Guishan, L. Shubin, Z. Jianghua, and Y. Wenyuan, "Siamese network target tracking based on regional loss function," *Int. J. Syst.*, vol. 15, no. 4, pp. 722–731, 2020.

[24] C. Zhiwang Z. Zhongxin, S. Juan, L. Hongfu, and P. Yong, "Siamese network tracking algorithm based on target attention feature selection," *Acta Optics*, vol. 40, no. 9, pp. 110–126, 2020.

[25] P. Lei, F. Xinxi, H. Zhiqiang, Y. Wangsheng, and M. Sugang, "Siamese network visual tracking algorithm based on cascaded attention mechanism," *J. Aeronautics and Astronautics*, vol. 46, no. 12, pp. 2302–2310, 2020.

[26] Y. Zhichao and Z. Ruihong, "Improved Siamese network tracking algorithm combined with deep contour features," *J.Xidian Univ.*, vol. 47, no. 3, pp. 40–49, 2020.

[27] B. Li, J. Yan, W. Wu, Z. Zhu and X. Hu, "High Performance Visual Tracking with Siamese Region Proposal Network," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8971–8980, doi: 10.1109/CVPR.2018.00935.

[28] Held, David, S. Thrun, and Silvio Savarese, "Learning to track at 100 fps with deep regression networks," *Computer Vision – ECCV 2016. ECCV 2016. Lecture Notes in Computer Science*, vol. 9905, pp. 749–765, doi: 10.1007/978-3-319-46448-0_45.

[29] Z. Hongwei, L. Xiaoxia, Z. Bin, and M. Qi, "Target tracking error correction method based on multi-scale suggestion frame," *Comput. Eng. Appl.*, vol. 56, no. 19, pp. 132–138, 2020.

[30] W. Junling and W. Shuohao, "Deep learning target tracking algorithm based on Siamese network," *Comput. Eng. Des.*, vol. 40, no. 10, pp. 3014–3019, 2019.

[31] Q. Zhuling, Z. Yufei, Z. Peng, and W. Min, "Visual tracking algorithm based on Siamese neural network online discriminant features," *Acta Optica Sinica*, vol. 39, no. 9, pp. 253–261, 2019.

[32] Yan, Bin, *et al*, "LightTrack: Finding Lightweight Neural Networks for Object Tracking via One-Shot Architecture Search," *2021CVP*, 2021, pp.15180-15189.

[33] Z. Tengfei, Z. Shuren, and P. Jian, "Adaptive selection tracking system based on dual Siamese network," *Comput. Eng.*, vol. 46, no. 6, pp. 103–107, 2020.

[34] Y. Kang, S. Huihui, and Z. Kaihua, "Real-time visual tracking based on dual attention Siamese network," *Comput. Appl.*, vol. 39, no. 6, pp. 1652–1656, 2019.

[35] C. Xu and M. Zhaohui, "Overview of target video tracking algorithms based on deep learning," *Comput. Syst. Appl.*, vol. 28, no. 1, pp. 1–9, 2019.

[36] S. Lulu, Z. Suofi, and W. Xiaofu, "Target tracking based on Tiny Darknet fully convolutional Siamese network," *J. Nanjing Univ. Posts and Telecommun.*, vol. 38, no. 4, pp. 89–95, 2018.

[37] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, 1 Aug. 2020, doi: 10.1109/TPAMI.2019.2913372.

[38] Pu, Shi, *et al*, "Deep attentive tracking via reciprocative learning," Available: https://arxiv.org/abs/1810.03851.

[39] Z. Dawei *et al*, "Learning Fine-Grained Similarity Matching Networks for Visual Tracking," *2020 Int. Multimedia Retrieval Conf.*, 2020, pp. 296–300, doi: 10.1145/3372278.3390729.

[40] H. Anfeng *et al*, "A twofold siamese network for real-time object tracking," 2018CVPR, 2018, pp. 4834–4843.