EURASIP Journal on Image
and Video Processing

RESEARCH

Open Access

Multi-layer features template update object tracking algorithm based on SiamFC++

Xiaofeng Lu[1], Xuan Wang[1*], Zhengyang Wang[1] and Xinhong Hei[1]

*Correspondence:
2191220016@stu.xaut.edu.cn

[1] Department of Computer Science and Technology, Xi'an University of Technology, Xi'an, Shaanxi, China

Abstract

SiamFC++ only extracts the object feature of the first frame as a tracking template, and only uses the highest level feature maps in both the classification branch and the regression branch, so that the respective characteristics of the two branches are not fully utilized. In view of this, the present paper proposes an object tracking algorithm based on SiamFC++. The algorithm uses the multi-layer features of the Siamese network to update template. First, FPN is used to extract feature maps from different layers of Backbone for classification branch and regression branch. Second, 3D convolution is used to update the tracking template of the object tracking algorithm. Next, a template update judgment condition is proposed based on mutual information. Finally, AlexNet is used as the backbone and GOT-10K as training set. Compared with SiamFC++, our algorithm obtains improved results on OTB100, VOT2016, VOT2018 and GOT-10k data sets, and the tracking process is real time.

Keywords: Object tracking, Fully convolutional Siamese networks, Template update, Mutual information, FPN

1 Introduction

Object tracking is a research hotspot in the field of computer vision. Object tracking technology is widely used in fields, such as intelligent video surveillance, human-computer interaction, robot visual navigation, virtual reality, and medical diagnosis. The object tracking field is mainly divided into single object tracking [1–12] and multiple object tracking [13, 14].

The algorithm in this paper is a single object tracking algorithm. Early object algorithms such as optical flow method [1], Kalman filter [2], and kernel method [3] are all processed in the time domain. Their calculation involves complex matrix inversion, the heavy computational load of which leads to poor real-time performance. The Mosse algorithm [4] creatively introduced correlation filtering into the object tracking field. By converting the calculation from the time domain to the frequency domain, it utilizes the nature of the circulant matrix that can be diagonalized in the frequency domain, thus greatly reducing the amount of calculation and increasing its speed. Moreover, KCF [5] introduced a cyclic matrix on the basis of correlation filtering to increase the number of negative samples and improve the quality of classifier training. In the meanwhile, the

Gaussian kernel, which can transform nonlinear problems into a high-dimensional linear space, was added to the ridge regression, thus simplifying the calculation.

With the maturity of deep learning technology, researchers began to apply it to object tracking. However, the increase in the number of convolutional layers and the complexity of the training network have led to poor real-time performance of the algorithms, failing to realize the real-time tracking of fast-moving objects. Nevertheless, the tracker based on the Siamese network [7–12] could obtain image features through the Siamese network, and then perform matching operations to achieve a balance between accuracy and real-time performance. Therefore, it has become another key research direction after correlation filtering.

SINT [6] applied the Siamese network to the object tracking field first, thus pioneering the conversion of the object tracking problem into a patch matching problem. Through obtaining the patch block from the training data and acquiring the similarity function by training the network, it uses the trained network to track objects (match by patch block) and gets the tracking results. The SiamFC [7] algorithm proposes an end-to-end object tracking network based on the Siamese network, and thus obtains a very high tracking speed and is capable of analyzing the influence of padding in the network. However, compared with the algorithm that combines correlation filtering and depth feature, it possesses poor robustness and accuracy. On the basis of SiamFC, SiamRPN [8] applies the RPN module in object detection to the tracking task, thereby improving both the accuracy and the speed. It is also verified to be able to exert a better tracking effect on a larger data set. In addition, the main idea of SiamMask [9] is to add a mask branch on the basis of SiamRPN to improve the tracking effect. Using a network, the bounding box and mask of the tracking object are obtained at the same time. However, given that the baseline networks of these algorithms are all AlexNet networks, the tracking effects of deep networks such as ResNet and Inception tend to be worse. Noteworthily, SiamDW [10] and SiamRPN++ [11] have solved the problem of poor tracking effect using deep network from different perspectives. SiamDW [10] analyzed the padding of the network and found that the padding in the convolution will affect the position of the feature on the feature map. Therefore, the Cropping Inside Residual (CIR) module is proposed to eliminate the influence of the padding of the convolutional neural network. Combining the CIR module with ResNet proposes a 22-layer convolutional neural network CIRResNet. After using CIRResNet to replace Backbone in SiamFC and SiamRPN, the tracking results are significantly improved. Different from the SiamDW algorithm to design a new Backbone, SiamRPN++ [11] successfully uses the deep residual network ResNet to improve the performance of the Siamese network-based tracker by modifying the center weight of the tracking image. SiamRPN++ shifts the center of the positive sample in the image, and randomly shifts the center of the positive sample by 16–64 pixels, thereby increasing the attention range of the network. At the same time, SiamRPN is used in multiple layers to filter out more discriminative sample blocks to improve network performance.

Tracking algorithms using RPN networks (such as SiamRPN, SiamMASK, SiamRPN++) can achieve accurate and efficient object state estimation. However, setting predefined anchors not only produces imprecise similarity scores, but also seriously affects the robustness of object trackers. To solve these problems, SiamFC++ [12]

proposed an anchor-free tracking algorithm to improve the robustness and accuracy of the algorithm. SiamFC++ eliminates the pre-defined anchor setting, thus disambiguating the matching and prior knowledge about the object scale.

The object template of the aforementioned object tracking algorithms is always the same during the tracking process, but object change associated with such factors as illumination, scale change, rotation, partial occlusion, or full occlusion during the tracking process is inevitable. To effectively improve the accuracy of the tracking algorithm, the tracking template should be updated in time according to the changes of the object to reflect its current state more accurately. At the same time, in multi-branch tracking algorithms (such as SiamRPN and SiamFC++ [12]), both the classification branch and the regression branch use the last layer feature of backbone. In the past two years, the backbone network of the object tracking algorithm has become increasingly complex, such as ResNet and GoogleNet. Such a complex network requires a lot of computing power in the tracking process, and it will also reduce the tracking speed.

Thus, the present paper proposes the following optimizations to the network structure on the SiamFC++ [12] framework. First, use FPN to extract feature maps from different layers of backbone for classification branch and regression branch, and make full use of the respective functions of the two branches; second, use 3D convolution to update the tracking template of the object tracking algorithm, and propose a template update judgment condition based on mutual information. It is verified that the object tracking algorithm based on Siamese network needs template update during the update process, and that mutual information can be used as a judgment condition for template update. Finally, use AlexNet as the backbone and GOT-10k as the training set to reduce training costs. Through template update, AlexNet can reach the tracking level of GoogLeNet. After experimental verification, the new tracker has improved on OTB2015[15], VOT2016[16], VOT2018[17], and GOT-10k[18].

Our contributions are as follows:

1. The algorithm proposed in this paper uses a small training dataset and a small backbone network. While ensuring real-time performance, its tracking results are close to those of SiamFC++using GoogLeNet.
2. The update condition judgment method based on mutual information proposed in this paper can judge the template update time more accurately and improve the update efficiency. The experiment also verified that using advanced convolutional feature maps in the classification branch and using low-level convolutional feature maps in the regression branch can effectively improve tracking performance.

2 Related work

2.1 Object tracking based on Siamese network

Since Ran Tao [6] proposed the first object tracking algorithm SINT based on Siamese network, tracking algorithms based on Siamese network have gradually become a hot spot in the field of object tracking. From the perspective of the tracking network structure, the Siamese object tracking algorithms can be divided into single-branch object tracking and multi-branch object tracking.

Early tracking algorithms based on Siamese networks, such as SiamFC [7] and CFNet [19], are all single-branch algorithms. The main feature is that the template image and search image are passed through the Siamese network to obtain two feature maps and directly perform cross-correlation operations to obtain the heat map. The bounding box is determined by the maximum response value of the heat map. These algorithms possess fast tracking speed, but they are not ideal for object tracking with frequent changes in size, especially for small objects.

Later, SiamRPN introduced the RPN network into the tracking model, and proposed a dual-branch tracking model with classification branch and anchor-based regression branch. Subsequent tracking models based on this dual-branch model such as SiamMask have also been proposed. SiamFC++ designed an anchor-free multi-branch object tracking model. These algorithms all obtain tracking results through the joint action of classification branch and regression branch. The use of regression branch optimizes the detection of object edges in the object tracking process. But the classification branch and regression branch of these algorithms use the feature map at the highest level of backbone. Therefore, in view of the different characteristics of the two branches, this paper optimizes the network framework of SiamFC++ using different features of backbone for object tracking in classification branch and regression branch.

2.2 FPN

FPN (feature pyramid networks) was proposed by Tsung-Yi Lin [20] in 2017. It mainly overcame the shortcomings of object detection in dealing with multi-scale changes. Many algorithms use a single high-level feature. Faster R-CNN [21] is a good case in point which uses a four-fold down-sampling convolutional layer-Conv4 for subsequent object classification and bounding box regression. However, an obvious defect in doing so lies in that the small object itself has less pixel information and is easily lost during the down-sampling process. To deal with the detection problem with very obvious object size differences, the current paper proposes a feature pyramid network structure which can handle the multi-scale change problem in object detection with a very small increase in the amount of calculation.

SiamFC++ uses the highest-level feature of backbone in both the classification branch and regression branch. However, the focus of the two branches in the tracking process is different. The classification branch is mainly used to distinguish the object and the background, and the determination of the center position of the object requires higher semantic information. The regression branch mainly determines the boundary size of the object, which requires higher position information. It can be seen from FPN that the low-level features contain less semantic information, but the object location is accurate. By contrast, the high-level features contain richer semantic information, but the object location is relatively rough. Therefore, this paper uses FPN to obtain the multi-layer feature maps of the backbone, with the classification branch using the high-level feature map, and the regression branch using the low-level feature map.

2.3 Tracking template update

During the tracking process, SiamFC++ is the same as SiamFC, the tracking template is unchanged which is always the first frame’s feature map of the video, but the object will be deformed or occluded by other objects. Therefore, the algorithm should introduce a template update mechanism to improve the update effect. During the tracking process, the tracked target undergoes changes over time. Therefore, this paper also employs a method that integrates temporal sequence features to update the tracking template. In addition to this approach, if there is no template update judgment condition, the template will be contaminated, resulting in poor tracking results for subsequent frames. And updating every frame will seriously reduce the tracking speed. Thus, the algorithm should add a template update judgment mechanism. After the template update condition is established, the image that is input to the Siamese network will be updated. The object area of the new image is generated by the Siamese network to produce a new object template for the update of subsequent video frames.

3 Methods

To make full use of respective characteristics of classification branch and regression branch?designed a new tracker model. The proposed tracker was optimized based on SiamFC++. This part mainly describes in detail the structure of the template update object tracker based on the multi-layer feature maps of the fully convolutional Siamese network.

The model framework is shown in Fig. 1. In Template Frames, z-p is the object area of the first frame of the video, z-n is the object area of the current frame, and z-q is the object area of the picture frame used in the previous update. In the initial state, all 3three frames are were the object area of the first frame. Detection Frame refers to the current detection frame of the video. Input the three pictures in template frames were input into Siam-FPN to get their respective high-level feature map (z-h) and low-level feature map (z-l). At the same time, input the Detection Frame was input into the same Siam-FPN to get the high-level feature map (x-h) and low-level feature map (x-l) of the detection frame, and copy the feature was copied twice. 3D-cls was used for feature fusion of high-level feature maps of Z and X, and 3D-reg was used for feature fusion of low-level feature maps. 3D-cls and 3D-reg are both convolution nuclei of 3×3×3 for multi-frame feature maps fusion and branch classification. Then Z and X of the two branches are were cross-correlated and down sampled to obtain the heat maps of classification branch and regression branch. Next, cls head and reg head of SiamFC++ are were used

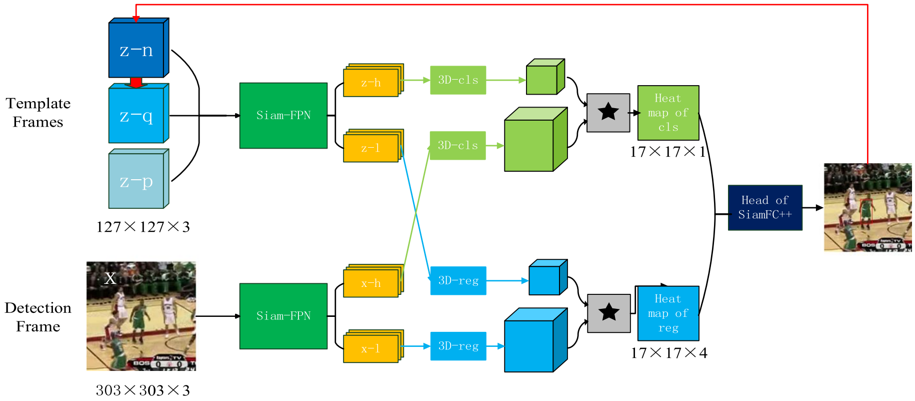

Fig. 1 The structure of the template update object tracker

to determine object positions in the search graph and generate bounding box of the current frame. Whether the template update conditions are were met was then checked. If so, z-n was assigned to z-q, and the images in the current frame bounding box are were assigned to z-n to update the template, and the new template was used to track the objects of subsequent frames.

3.1 Siam-FPN

In this paper, Siam-FPN [8] was the backbone, mainly composed of two parts, AlexNet and FPN. The structure is shown in Fig. 2. First, AlexNet was used to obtain the convolutional feature map of each layer. Next, up-sampling was used to magnify the small feature map to the same size as the feature map of the upper layer, and then the two were added together and transmitted to the next layer until the last layer was reached. The up-sampling was realized by nearest neighbor interpolation. By this method, the semantic information of the feature map could be retained to the maximum extent in the up-sampling process, and then it could be fused with the feature map with rich spatial information in the bottom-up process. In this way, the feature map with both good spatial information and semantic information could be obtained. Finally, P5 was selected as the high-level feature, and P3 as the low-level feature.

3.2 Classification head and regression head based on SiamFC++

After the response map was obtained, the relevant head operation was performed. This paper used the classification head and regression head proposed by SiamFC++ [12]. As shown in Fig. 3, the classification head took the heat map of cls branch as input, and the classification of each point in the feature map was the classification of corresponding patches on the original map. To balance the relationship between the background and

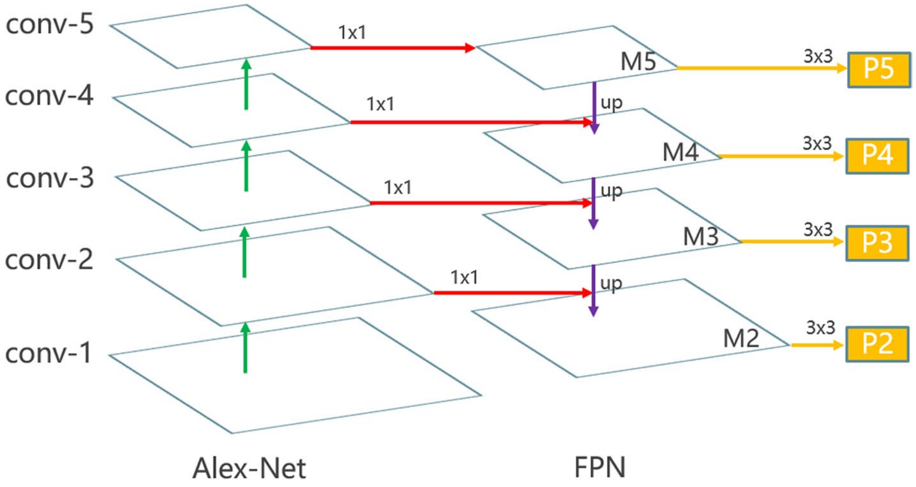

Fig. 2 The fifth layer's feature map is initially subjected to a 1×1 convolutional operation to modify the channel dimension (set to 256 in this chapter), resulting in a new feature map termed M5. M5 is subsequently upsampled using the nearest-neighbor interpolation method to match the dimensions of the fourth layer's feature map. The fourth layer's feature map, altered to match the channel dimension, is then element-wise added to M5 at corresponding positions, yielding a novel fourth layer feature map denoted as M4. This was repeated twice to get M3 and M2. The M layer feature map then went through 3×3 convolution (to reduce the aliasing effect caused by the nearest neighbor interpolation, and the surrounding numbers were all the same). Finally, P2, P3, P4, and P5 layers were obtained

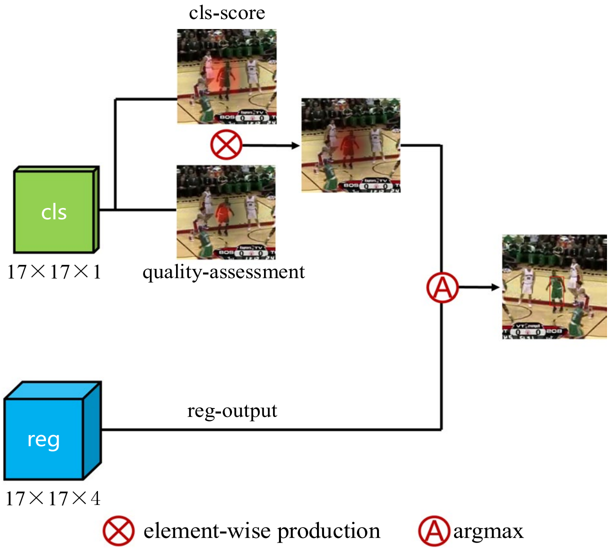

Fig. 3 Classification head and regression head based on SiamFC++

object position, the quality score was introduced, and the score of the final selection box was obtained by multiplying the quality score and the predicted score. The regression head took the heat map of the reg branch as input and output an offset (17×17×4) to optimize border position.

For cls-score, pixel points on the heat map were mainly categorized into positive sample points and negative sample points through the ground truth input of the image. If the corresponding position($\lfloor \frac{s}{2} \rfloor + xs$, $\lfloor \frac{s}{2} \rfloor + ys$) of (x, y)on CLS was within the ground truth of the original image,(x, y) would be regarded as positive sample, and the rest as negative samples (s is AlexNet's stride, which is 8).

For quality-assessment, if the classification branch was directly used to select the final prediction bounding box, the positioning accuracy might be reduced. Given the lack of good correlation between classification confidence and positioning accuracy, it was assumed that feature pixels around the object center had better estimated quality than other pixels. A 1×1 convolutional layer was added in parallel to the classification head for quality assessment. This output was used to estimate Prior Spatial Score (PSS), as defined below:

$$PSS^* = \sqrt{\frac{min(l^*, r^*)}{max(l^*, r^*)} \cdot \frac{min(t^*, b^*)}{max(t^*, b^*)}} \tag{1}$$

wherel^*, t^*, r^*, b^* are, respectively, the distance between the corresponding positions of the final output prediction (x, y) on the original map and the left, top, right, and bottom frames of the ground truth. The predicted PSS^* was multiplied by the corresponding classification score to calculate the final score. The classification branch was then obtained.

For reg-out, if the corresponding point (x, y) on the classification branch feature map of the original image was($\lfloor \frac{s}{2} \rfloor + xs$, $\lfloor \frac{s}{2} \rfloor + ys$), the regression branch would output the predicted value of ground truth at this point, expressed as a four-dimensional vector

$t = (l^*, t^*, r^*, b^*)$. The calculation process of each ground truth component is described as

$$\begin{aligned} l^* &= \left(\left\lfloor \frac{s}{2} \right\rfloor + xs\right) - x_0,\ t^* = \left(\left\lfloor \frac{s}{2} \right\rfloor + ys\right) - y_0 \\ r* &= x_1 - \left(\left\lfloor \frac{s}{2} \right\rfloor + xs\right),\ b^* = y_1 - \left(\left\lfloor \frac{s}{2} \right\rfloor + ys\right) \end{aligned} \tag{2}$$

where (x_0, y_0) and(x_1, y_1) represent the corner points at the upper left and lower right corners of ground truth, respectively.

Finally, the result of classification branch was combined with that of regression branch to get the tracking result of the current frame.

3.3 Template update judgment condition based on mutual information

This paper used 3D convolution fusion time series feature of frames to update the classification branch and regression branch template. However, the object may be blocked by other objects or intersect with other object images during the object tracking process. If no update judgment condition is added, the model updating will not stop. The template will also be polluted, thus greatly affecting the tracking of subsequent frames. And if there is no template update judging mechanism, updating every frame will seriously affect the tracking speed. Therefore, an update judgment mechanism must be added. This paper proposed a template update judgment method based on mutual information [22]. Mutual information represents the amount of information contained in one random variable about another random variable. The larger the mutual information value, the greater the amount of the same information contained in the two variables. Assuming that the heat map obtained by performing correlation operations on the classification branch of the first frame of the image is the most ideal and the heat map of the classification branch of the subsequent detection frame is calculated for mutual information with it, the larger the mutual information value is, the more suitable the current frame will be for updating the template. To facilitate the calculation, this paper converted the obtained mutual information value into a normalized mutual information value.

The judgment process based on mutual information is shown in Fig. 3. In the tracking process, the first frame of the video was used as the Template Frame. Moreover, the first frame was used as the Detection Frame to obtain the heat map of the classification branch as the 1-heat map, and the classification branch of the subsequent frames was in its classification branch. The t-heat map and 1-heat map were used as two variables to get their mutual information value. The mutual information calculation formula is shown as follows:

$$I(X, Y) = \sum_{x \subset X} \sum_{y \subset Y} log \frac{p(x, y)}{p(x)p(y)} \tag{3}$$

where X and Y are the heat maps of the first frame and the heat map of the current detection frame, respectively, p(x) and p(y) are the marginal distributions of X and Y, respectively, and p(x, y) is the joint distribution of X and Y.The obtained mutual information value was converted into normalized mutual information [23] value, as shown in Formula 4:

$$U(X,Y) = 2\frac{I(X,Y)}{H(X)+H(Y)} \tag{4}$$

where H(X) and H(Y) are the entropy of X and Y, respectively. When the obtained normalized mutual information was greater than the $V_{threshold}$ selected in this paper, the object area of the current frame could be used for template update. Otherwise, the template update was not performed, and the tracking process of the next frame was directly started after the tracking result of the current frame was obtained.

To make the mutual information judgment more accurate, the paper used dynamic thresholds. The larger the mutual information value, the more the same information. Therefore, the dynamic threshold was set as a local maximum in this paper. The threshold dynamic update formula is as follows:

$$\begin{cases} I(t) > V_{threshold}, \\ I'(t) = 0 \\ I'(t) > 0 \end{cases} \tag{5}$$

where t represents the current detection frame, $I(t)$ represents the normalized mutual information value of the current detection frame classification branch heat map and 1-heat map and$I'(t) = 0$ and $I''(t) > 0$ indicate the local maximum point of mutual information. Because the mutual information value of the heat map and 1-heat map of each search frame was discrete, Formula 5 can be expressed as

$$\begin{cases} I(t) > V_{threshold}, \\ I(t) - I(t-2) \approx 0 \\ I(t) + I(t-2) - 2I(t-1) > 0 \end{cases} \tag{6}$$

where $I(t-1)$ represents the standard mutual information of the previous frame of t?and $I(t-2)$ represents the standard mutual information of the previous frame of t-1. Because the algorithm required the value of the mutual information of the three-frame search image, but the first frame and the second frame did not meet the conditions required by the formula when searching, the thresholds of the first frame and the second frame were set separately. In addition, the object area obtained from the first search image was generally not much different from that of the first frame template image of the video, so it could be used for direct update. At the same time, because there were very few videos that would be occluded in the second frame, the $V_{threshold}$ of the first and second detection images was set to a fixed value.

4 Experiment analysis

4.1 Training process

This paper used the same one-shot learning [24] in SiamFC++ [7] for training, and randomly selected a PAIR in each video. Each PAIR contained four video frames. Specifically, the first video frame was the first frame of the video, and the next three frames were randomly selected from the video. The interval between the second and the third video frames did not exceed 15 frames in the original video, and the interval between the third and the fourth video frames did not exceed 100 frames. The first three video frames were used as Template Frames, and the last video frame was used as Detection

Frame. In Detection Frame, the three pictures imported into the 3D convolutional network were the same, all being the last pictures of PAIR. The loss function was the same as that of the three branches of SiamFC++. Furthermore, cls-score used focal loss, quality-assessment used IoU loss, and reg-out used BCE loss. The total loss function is shown in Formula 7:

$$L(\{p_{x,y}\}, q_{x,y}, \{t_{x,y}\}) = \frac{1}{N_{pos}} \sum_{x,y} L_{cls}(p_{x,y}, c^*_{x,y}) \\ + \frac{\lambda}{N_{pos}} \sum_{x,y} 1_{c^*_{x,y}\ 0}\, L_{quality}(q_{x,y}, q^*_{x,y}) + \frac{\lambda}{N_{pos}} \sum_{x,y} 1_{c^*_{x,y}>0}\, L_{reg}(t_{x,y}, t^*_{x,y}) \tag{7}$$

where 1{.} is 1 if the point is a positive sample, and 0 if otherwise.$c_(x,y)^*$ represents the label of cls-score and is 1, and if (x, y) is a positive sample, and 0 if is a negative sample. $q^*_{x,y}$ represents the label of quality-assessment, and some pixels similar to Gaussian distribution was sprinkled in the feature map within the range is [0, 1]; $t^*_{x,y}$ represents the label of reg-out, and the feature map areas of the four channels were assigned to $(x_0, y_0, x_1, y_1)$of the detection frame.

4.2 Experimental environment and parameters

The experimental platform of the paper was Python 3.8.3, the CPU of the server was Intel(R) Core(TM) i7–4790 3.6GHz 8 G, and the GPU was NVIDIA GeForce RTX 2080Ti. During the experiment, the parameters of SiamFC++ remained unchanged. The GOT-10k [18] was employed as the training set, and the GOT-10k toolkit in Python was used as the test tool.

4.3 Regression branch uses different layer feature maps

When using FPN to extract feature maps from different layers, four feature maps (P2, P3, P4, P5) were obtained, but which feature map could achieve the best performance in the regression branch? This part mainly verifies the influence of the regression branch using feature maps of different layers on the tracking results. To ensure the validity of the experimental results, all parameters, training times, and image preprocessing during the experiment were the same, and only the regression branch used the feature maps of different layers of FPN. The experimental results are shown in Figs. 4 and 5. When

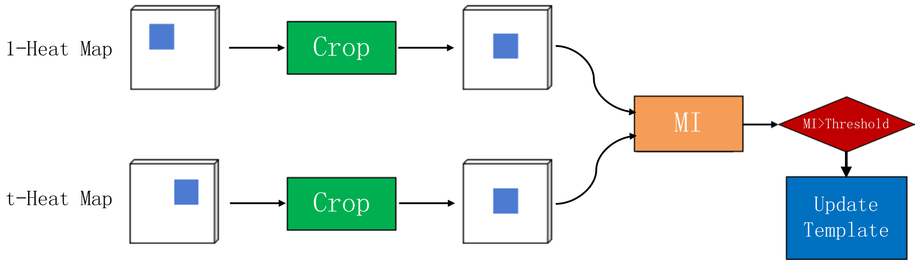

Fig. 4 1-heat map and t-heat map were cropped to the same size with the maximum value as the center, and the part beyond the map boundary was filled with the average value. Two new heat maps were obtained, and the normalized mutual information value of the two heat maps was calculated. If the normalized mutual information value was greater than the predetermined threshold, the update condition would be established

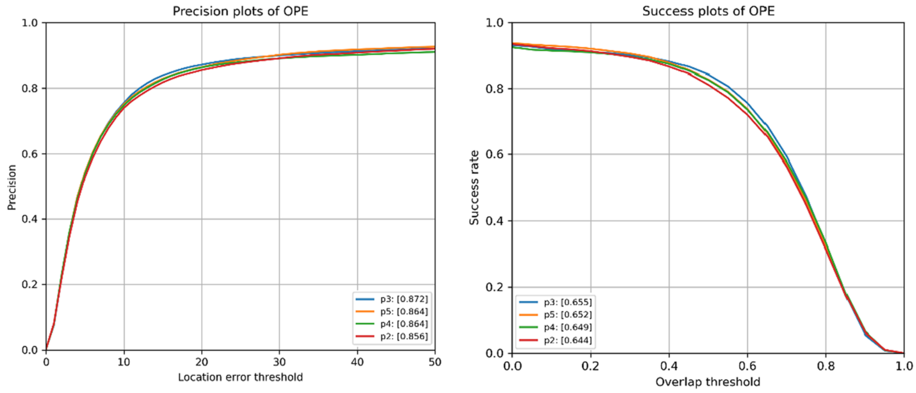

Fig. 5 Experimental results of the regression branch using feature maps of different layers

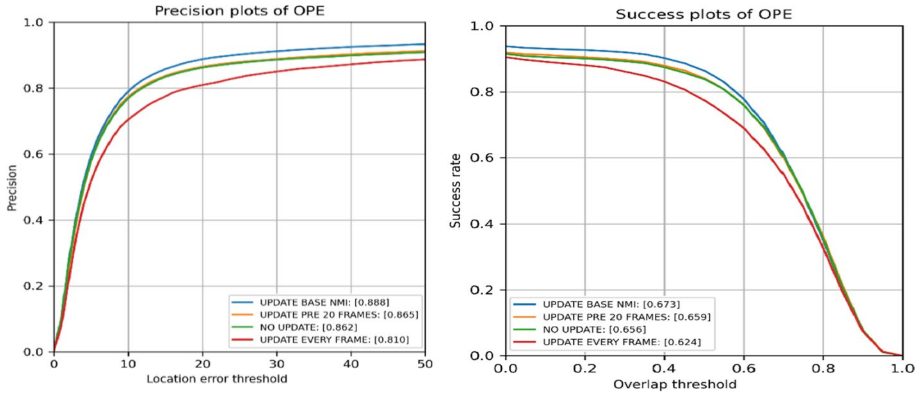

Fig. 6 Results of the template update judgment comparison experiment

the regression branch utilizes the features from P3, the model achieves the best tracking performance. Consequently, the regression branch uses the features from the P3 layer as the low-level feature map.

4.4 Template update judgment

When to update the template during the tracking process is a very important question. This part verifies the superiority of the judgment mechanism based on NMI through comparative experiment. To ensure the validity of the experimental results, all parameters, training times, and image preprocessing during the experiment were the same. The tracking results are shown in Fig. 6. It is clear that the use of mutual information algorithm as a judgment condition could more effectively update the tracking template and improve the tracking effect.

4.5 Threshold of mutual information

After calculating the NMI between the current detection frame and the first frame of the video, how to set the threshold becomes a key issue. This paper used the local maximum of the mutual information value of the video frame as the template update point, and the fixed threshold was 0.75. The experimental results are shown in Fig. 7. Clearly, using the

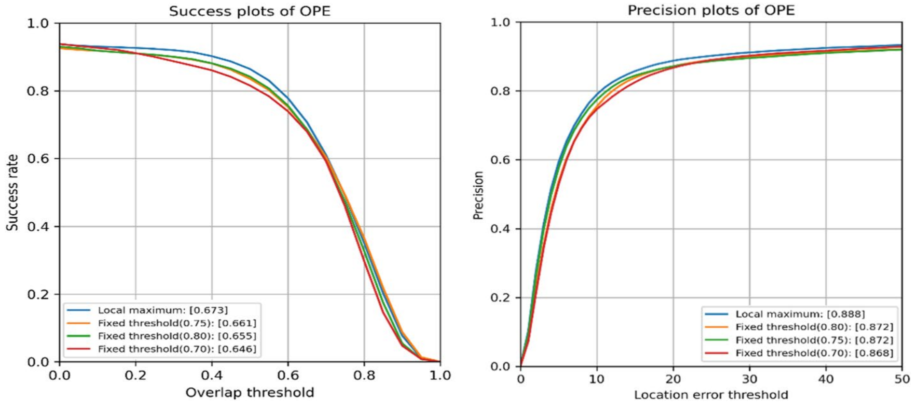

Fig. 7 Results of different thresholds of NMI

Table 1 Evaluate on OTB

	Accuracy	Precision	FPS
SiamFC[7]	0.581	0.771	86
SiamRPN[8]	0.637	0.851	160
ECO[26]	0.691	0.910	8
ROAM[27]	0.680	0.908	13
LK-Siam RPN [28]	0.657	0.882	62
MFST [29]	0.647	0.831	39
AFSN [30]	0.644	0.857	29
SiamFC++(AlexNet)[12]]	0.655	0.872	170
SiamFC++(GoogLeNet)[12]	0.683	–	90
OURS	0.673	0.888	169S

local maximum value as the judgment condition for template update could improve the tracking effect.

5 Results and discussion

5.1 Evaluation on OTB

There are one test set, namely OTB-15 [15], in the OTB dataset. Its evaluation follows the evaluation criteria proposed in [15]. The paper mainly evaluated the precision and Area under Curve (AUC) on OTB-15. As shown in Table 1, the AUC of our algorithm on OTB-15 was 0.673, and the precision was 0.888. Compared with SiamFC++ using Alex Net, our algorithm increased the AUC by 3% and precision by 2%.

5.2 Evaluation on VOT

VOT (visual object tracking) [16, 17] is a testing platform for single object tracking. It has different evaluation indicators from OTB, and has been continuously improved since its inception. The evaluation algorithm in this paper mainly used the three most important indicators of VOT: accuracy (A), robustness (R), and EAO. Because the contents of the VOT2017 and VOT2018 datasets were the same, the evaluation was carried out on VOT2016 [16] and VOT2018 [17]. The evaluation results are shown in Table 2.

Table 2 Evaluate on VOT

		VOT2016			Vot2018	
	A	**R**	**EAO**	**A**	**R**	**EAO**
SiamFC[7]	0.532	0.461	0.235	0.503	0.585	0.188
SiamRPN[8]	0.560	0.260	0.344	0.490	0.460	0.244
ECO[26]	0.550	0.200	0.375	0.480	0.270	0.280
ROAM[27]	0.441	0.599	0.174	0.380	0.543	0.195
LK-Siam[28]	0289	0.531	0.350	0.214	0.533	0.543
MFST[29]	–	–	–	0.200	0.497	0.428
SiamFC++(AlexNet)[12]	0.584	0.342	0.308	0.556	0.183	0.400
SiamFC++(GoogLeNet)[12]	–	–	–	0.587	0.183	0.426
OURS	0.615	0.270	0.346	0.578	0.383	0.271

Table 3 Evaluation on GOT-10k

	AO	**SR0.50**	**SR0.75**
SiamFC[7]	0.348	0.353	0.098
SiamRPN++[11]	0.518	0.618	0.325
ECO[26]	0.316	0.309	0.111
ROAM[27]	0.465	0.532	0.236
LK-Siam RPN[28]	0.520	0.619	0.329
AFSN[30]	0.558	0.605	0.413
SiamFC++(AlexNet)[12]	0.493	0.577	0.323
SiamFC++(GoogLeNet)[12]	0.595	0.695	0.479
OURS	0.527	0.627	0.343

On VOT2016, the accuracy of our algorithm was improved by 0.031 compared to that of SiamFC++ (AlexNet), the EAO was improved by 0.034, and the robustness was improved by 0.062. On VOT2018, the accuracy of our algorithm was improved by 0.022 compared to that of SiamFC++ (AlexNet), the EAO was improved by 0.013, and the robustness was improved by 0.016. The algorithm's performance on VOT dataset has also been improved, but there is still a gap compared to SiamFC++ using GoogLeNet.

GOT-10k [18] is an object tracking dataset released by the Chinese Academy of Sciences. Its evaluation indices include average overlap (AO), the accuracy of successful tracking at AO of 0.5 (SR0.5), and the accuracy of successful tracking at AO of 0.75 (SR0.75). The test results are shown in Table 3. Compared with SiamFC++ (AlexNet), our algorithm increased the AO by 0.034, SR0.50 by 0.05, and SR0.75 by 0.02.

5.3 Discussion

Based on the above experiments, our algorithm demonstrates improvements on all three datasets. The use of template updating mechanism enhances the tracker's performance in tracking occluded objects. Additionally, employing different convolutional layers in the classification and regression branches with deep features increases the tracking accuracy of the tracker. As shown in Fig. 8, the bounding box of our algorithm in this chapter is closer to the Ground Truth on all four sides compared to SiamFC++(AlexNet).

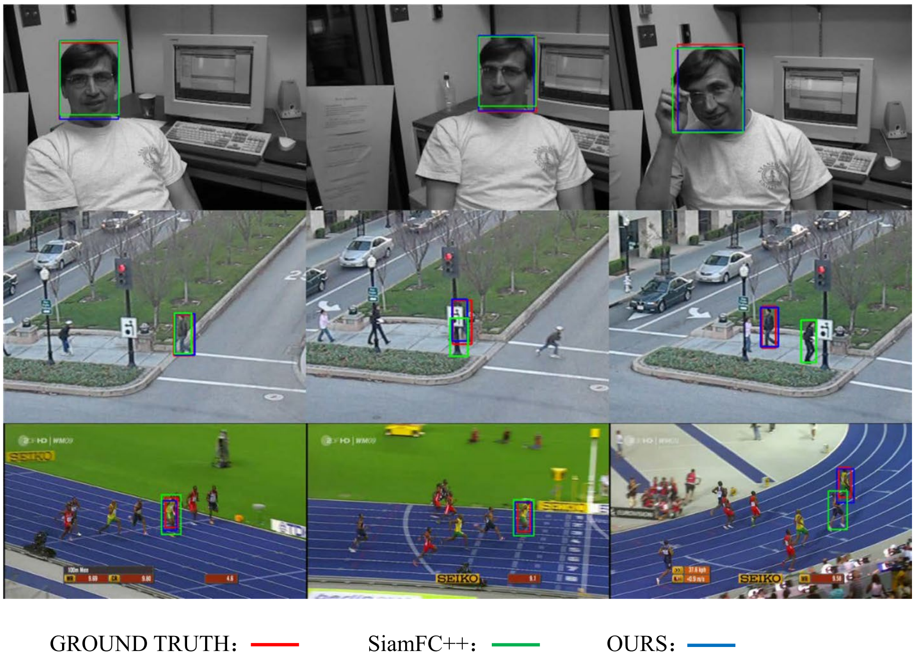

Fig. 8 Comparison of tracking results

Moreover, the second row of Fig. 8 illustrates that our algorithm exhibits superior discrimination results for similar objects.

Our algorithm separates feature extraction and feature fusion. It first individually extracts features from the three images, and during template updates, only features from the new image need to be extracted, while the features from the other two images are not extracted. This optimization leads to the improvement in tracking results without significant speed reduction. The tracking speed of our algorithm is maintained at 169FPS, almost unchanged from the original 170FPS. Our algorithm's results are comparable to other algorithms based on ResNet50 and better utilize the capabilities of AlexNet.

6 Conclusion

To address the limitations of the SiamFC++ algorithm, which solely extracts the object features from the first frame for tracking and employs only the highest-level features in both the classification and regression branches, thus underutilizing the unique characteristics of each branch, this paper presents a template update-based object tracking algorithm using a fully convolutional Siamese network with multi-layer features. In the algorithm, a feature pyramid network is initially employed to extract feature maps from various layers of the backbone network, which are subsequently utilized for the classification and regression branches. Additionally, a 3D convolutional approach is utilized to update the tracking template of the object tracking algorithm. A template update determination criterion based on mutual information is introduced. Lastly, the algorithm uses a small training dataset and a small backbone network. While ensuring real-time performance, its tracking results are close to those of SiamFC++using GoogLeNet. Besides, utilizing 3D convolution to fuse temporal features from multiple frames during the object tracking process, resulting in a new tracking template, proves to be an effective template update approach. This approach holds significant potential for widespread applications in the field of video processing.

The Transformer[25] has achieved remarkable success in the field of Natural Language Proce-ssing. Attention mechanisms [25] have also found extensive applications in computer vision. The Transformer architecture has revolutionized how sequences are modeled. In comparison to RNNs, Transformer models based on attention mechanisms can execute parallel operations. Exploring the use of attention mechanisms for processing video sequences presents a new research direction in the field of object tracking.

Acknowledgements
This work was supported by the National Natural Science Foundation of China(Grant No.62076201, U1934222, 62376213).

Author contributions
XL: Conceptualization; Funding acquisition; Supervision; Writing - original draft. XW: Data curation; Writing—original draft; Methodology; Software; Validation; Formal analysis; Writing-review & editing. ZW: Validation; Formal analysis; Investigation. XH: Conceptualization; Resources; Project administration.

Funding
National Natural Science Foundation of China, Grant/Award Number: 62076201; National Natural Science Foundation of China, Grant/Award Number: U1934222; National Natural Science Foundation of China, Grant/Award Number: 62376213.

Availibility of data and materials
The data that support the findings of this study are openly available in "GOT-10k" at http://got-10k.aitestunion.com/. The data that support the findings of this study are openly available in "OTB" at http://cvlab.hanyang.ac.kr/trackerbe-nch-mark/datasets.html. The data that support the findings of this study are openly available in "VOT2016" at https://www.votchallenge.net-/vot2016/.

Declarations

Ethics approval and consent to participate
Consent for publication

Consent for publication
Not application

Competing interests
We have no competing interests.

Received: 25 January 2022 Accepted: 5 December 2023
Published online: 04 January 2024

References

1. J. Shin, S. Kim, S. Kang, S. Lee, J.K. Paik, B.R. Abidi, M.A. Abidi, Optical flow-based real-time object tracking using non-prior training active feature model. Real Time Imaging **11**(3), 204–218 (2005). https://doi.org/10.1016/j.rti.2005.03.006
2. S. Spors, R. Rabenstein, A real-time face tracker for color video. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2001, 7-11 May, 2001, Salt Palace Convention Center, Salt Lake City, Utah, USA, Proceedings, pp. 1493–1496 (2001). https://doi.org/10.1109/ICASSP.2001.941214
3. A. Doucet, S.J. Godsill, C. Andrieu, On sequential monte Carlo sampling methods for Bayesian filtering. Stat. Comput. **10**(3), 197–208 (2000). https://doi.org/10.1023/A:1008935410038
4. D.S. Bolme, B.A. Draper, J.R. Beveridge, Average of synthetic exact filters. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA, pp. 2105–2112 (2009). https://doi.org/10.1109/CVPR.2009.5206701
5. J.F. Henriques, R. Caseiro, P. Martins, J.P. Batista, Exploiting the circulant structure of tracking-by-detection with kernels. In: Fitzgibbon, A.W., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV. Lecture Notes in Computer Science, vol. 7575, pp. 702–715 (2012). https://doi.org/10.1007/978-3-642-33765-9_50
6. R. Tao, E. Gavves, A.W.M. Smeulders, Siamese instance search for tracking. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pp. 1420–1429 (2016). https://doi.org/10.1109/CVPR.2016.158
7. L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, P.H.S. Torr, Fully-convolutional siamese networks for object tracking. In: Hua, G., Jégou, H. (eds.) Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9914, pp. 850–865 (2016). https://doi.org/10.1007/978-3-319-48881-3_56

8. B. Li, J. Yan, W.Wu, Z. Zhu, , X. Hu, High performance visual tracking with siamese region proposal network. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp. 8971–8980 (2018). https://doi.org/10.1109/CVPR.2018.00935
9. Q. Wang, L. Zhang, L. Bertinetto, W. Hu, P.H.S. Torr, Fast online object tracking and segmentation: a unifying approach. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 1328–1338 (2019). https://doi.org/10.1109/CVPR.2019.00142
10. Z. Zhang, H.Peng, Deeper and wider siamese networks for real-time visual tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 4591–4600 (2019). https://doi.org/10.1109/CVPR.2019.00472
11. B. Li , W. Wu, Q. Wang, F. Zhang, J. Xing, J. Yan, Siamrpn++: evolution of siamese visual tracking with very deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 4282–4291 (2019). https://doi.org/10.1109/CVPR.2019.00441
12. Y. Xu, Z. Wang, Z., Li, Y. Ye, G. Yu, Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pp. 12549–12556 (2020)
13. Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, N. Yu, Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pp. 4846–4855 (2017). https://doi.org/10.1109/ICCV.2017.518
14. A. Sadeghian, A. Alahi, S. Savarese, Tracking the untrackable: learning to track multiple cues with long-term dependencies. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pp. 300–311 (2017). https://doi.org/10.1109/ICCV.2017.41
15. Y. Wu, J. Lim, M. Yang, Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015). https://doi.org/10.1109/TPAMI.2014.2388226
16. M. Kristan, A. Leonardis, J. Matas, The visual object tracking VOT2016 challenge results. In: Hua, G., Jégou, H. (eds.) Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II. Lecture Notes in Computer Science, vol. 9914, pp. 777–823 (2016). https://doi.org/10.1007/978-3-319-48881-3_54
17. M. Kristan, A. Leonardis, J. Matas, The sixth visual object tracking VOT2018 challenge results. In: Leal-Taixé, L., Roth, S. (eds.) Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part I. Lecture Notes in Computer Science, vol. 11129, pp. 3–53 (2018). https://doi.org/10.1007/978-3-030-11009-3_1
18. L. Huang, X. Zhao, K. Huang, Got-10k: A large high-diversity benchmark for generic object tracking in the wild. CoRR arXiv: abs/1810.11981 (2018)
19. J. Valmadre, L. Bertinetto, J.F. Henriques, A. Vedaldi, P.H.S. Torr, End-to-end representation learning for correlation filter based tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pp. 5000–5008 (2017). https://doi.org/10.1109/CVPR.2017.531
20. T. Lin, P. Dollár, R.B. Girshick, K. He, B. Hariharan, S.J. Belongie, Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pp. 936–944 (2017). https://doi.org/10.1109/CVPR.2017.106
21. S. Ren, K. He, R.B. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2017). https://doi.org/10.1109/TPAMI.2016.2577031
22. A.K Mirabadi, S. Rini, The information & mutual information ratio for counting image features and their matches. CoRR arXiv:abs/2005.06739 (2020)
23. L.T. Vinh, S. Lee, Y. Park, B.J. d'Auriol, A novel feature selection method based on normalized mutual information. Appl. Intell. **37**(1), 100–120 (2012). https://doi.org/10.1007/s10489-011-0315-y
24. N.D. Binh, Online multiple tasks one-shot learning of object categories and vision. In: Taniar, D., Pardede, E., Nguyen, H., Rahayu, J.W., Khalil, I. (eds.) MoMM'2011 - The Nineth International Conference on Advances in Mobile Computing and Multimedia, 5-7 December 2011, Ho Chi Minh City, Vietnam, pp. 131–138 (2011). https://doi.org/10.1145/2095697.2095722
25. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need. Adv. Neural. Inf. Process. Syst. (2017). https://doi.org/10.48550/arXiv.1706.03762
26. M. Danelljan, G. Bhat, F.S Khan, M. Felsberg, ECO: efficient convolution operators for tracking. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pp. 6931–6939 (2017). https://doi.org/10.1109/CVPR.2017.733
27. T. Yang, P. Xu, R. Hu, H.Chai, A.B. Chan, Roam: Recurrently optimizing tracking model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6718–6727 (2020)
28. S. Yao, X. Han, H. Zhang, X. Wang, X. Cao, Learning deep Lucas-Kanade Siamese network for visual tracking. IEEE Trans. Image Process. **30**, 4814–4827 (2021)
29. Z. Li, G.-A. Bilodeau, W. Bouachir, Multiple convolutional features in Siamese networks for object tracking. Mach. Vis. Appl. **32**, 1–11 (2021)
30. T. Yuan, W. Yang, Q. Li, Y. Wang, An anchor-free Siamese network with multi-template update for object tracking. Electronics **10**(9), 1067 (2021)

Publisher's note

Xiaofeng Lu received the B.S. and the M.S. degrees in computer and application from Xi'an University of Technology, Xi'an, China, in 2001 and 2006, respectively, and the Ph.D. degree in computer science from the Nihon University, Tokyo, Japan, 2014. He is currently an associate professor and the associate dean with

Department of Computer Science and Technology, Xi'an University of Technology.

Xuan Wang received his B.S. degree in Information and Computing Science from Changsha University of Science & Technology, Hunan, in 2017. Currently, he is a Master student in Computer Science and Technology, Xi'an University of Technology, Xi'an, China. His research interests include Object Tracking and Machine Learning.

ZhengYang Wang received his B.S. degree in Water Conservancy and Hydropower Engineering from Xi'an University of Technology, Xi'an, in 2020. Currently, he is a Master student in Computer Science and Technology, Xi'an University of Technology, Xi'an. His research interests include Object Tracking and Deep Learning.

Xinhong Hei received the B.S. and the M.S. degrees in computer and application from Xi'an University of Technology, Xi'an, China, in 1998 and 2003, respectively, and the Ph.D. degree in computer science from the Nihon University, Tokyo, Japan, 2008. He is currently a professor and the dean of academic affairs with Xi'an University of Technology.