IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 10, 2019, accepted December 23, 2019, date of publication December 26, 2019, date of current version January 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962388

Siamese Visual Tracking With Deep Features and Robust Feature Fusion

DAQUN LI[1,2], XIZE WANG[3], AND YI YU[1]

[1]Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]Technion–Israel Institute of Technology, Haifa 3200003, Israel

Corresponding author: Yi Yu (yuyi_ciomp@163.com)

This work was supported by the General Program of National Natural Science Foundation of China under Grant 51675506.

ABSTRACT Trackers based on fully-convolutional Siamese networks regard tracking as a process of learning a similarity function. By utilizing shallow networks and off-line training, Siamese trackers can achieve high tracking speed and perform well in some simple scenes. However, due to the less semantic information and the invariant template, Siamese trackers still have a gap compared with the state-of-the-art methods in complex scenes and other challenging problems (Occlusion, Deformation, etc.). In this paper, we propose a Siamese tracking algorithm with deep features and robust feature fusion (SiamDF). The improved ResNet-18 network is utilized to replace the traditional shallow network and extract the deep features with more semantic information. For eliminating the negative effect of padding and making better use of the deep network, the proposed algorithm adopts the spatial aware sampling strategy to overcome the strict translation invariance. Meanwhile, a final response map with high quality can be obtained by using the multi-layer feature fusion. Thus, the tracker can significantly reduce the impact of the distractors in complex scenes. In addition, an adaptive feature information fusion is adopted to update the template, so that the algorithm can adapt to various changes of the target appearance. Objective evaluation on the OTB100 dataset shows that the precision and the overlap success can reach 0.852 and 0.658 respectively. Moreover, the EAO value evaluated on the VOT2016 database can reach 0.336. These results demonstrate that our algorithm can effectively improve the tracking performance and perform favorably in both robustness and accuracy.

INDEX TERMS Visual tracking, Siamese networks, deep features, feature fusion.

I. INTRODUCTION

As an important direction in the field of computer vision, visual tracking has been highly concerned by researchers all the time. Moreover, it is widely used in video surveillance, augmented reality and human-computer interaction. Although the performance of the tracking algorithm has been improved in recent research [1]–[7], due to the existence of complex scenes such as illumination variation, occlusion and the similar interference, the task of visual tracking is still full of challenges.

In recent years, the tracking algorithms have been greatly improved by the entrance of convolutional neural networks (CNNs) [8]–[10]. The high-dimensional features with rich semantic information extracted from the network are good at distinguishing different categories of objects. However, due to the complexity of deep CNNs and the high dimension of the convolution features, the end-to-end training and online fine-tuning of the CNNs become extremely complicated. At the same time, the great increase of the computation in CNNs will decrease the tracking speed. Even if the algorithm meets the requirement of accuracy, it's impossible to meet the requirement of real-time performance. To solve these problems, visual tracking algorithms based on Siamese networks have made great progress. The algorithm of the fully-convolutional Siamese networks (SiamFC) is first proposed by Bertinetto *et al.* [11]. This algorithm learns a high-performance similarity function by offline training, so as to achieve the real-time and accurate tracking effect. Moreover, the fully-convolutional structure can achieve end-to-end training where the search branch is no longer limited by size.

Many following-up studies have been carried out after SiamFC [12]–[16]. The SINT tracker [12] decreases the usage of the max-pooling layer and utilizes the region of

The associate editor coordinating the review of this manuscript and approving it for publication was Juan A. Lara.

interest pooling (ROI pooling) to optimize the process of the feature extraction. The GOTURN tracker [13] adds the regression method into the Siamese network. Valmadre *et al.* [14] incorporates the correlation filter into the template branches, which makes the Siamese network efficient while using the shallow structure. The SiamRPN tracker [15] combines the region proposal network (RPN) with the Siamese network. The DaSiamRPN [16] tracker introduces a distractor-aware module and promotes the tracking performance of the model in complex scenes.

Although the above Siamese trackers can make full use of the convolutional features to improve the accuracy while ensuring the real-time performance, Siamese trackers still have a gap compared with the state-of-the-art methods on tracking benchmarks like OTB [17], [18]. We observe that the networks of all the above Siamese trackers are shallow structures. On the one hand, features extracted from the shallow network are only the general representation of the target. Though these general features can perform well in some simple scenes, they cannot distinguish the distractors or the background in complex scenes. On the other hand, using modern deep structure directly will break the strict translation invariance [19] and increase the amount of computation. Even if the SiamRPN++ tracker [19] successfully introduces the deep architecture into the algorithm, the feature information of the multi-layer network has not been fused reasonably. In addition, most Siamese trackers use the constant template to search the target. The constant template will not only make the tracker unable to adapt to various changes of the target appearance, but also lead to model drifting. Thus, it is especially important to update the template adaptively.

In order to overcome these problems and promote the tracking performance of the Siamese tracker, we propose a Siamese tracking algorithm with deep features and robust feature fusion (SiamDF). The improved ResNet-18 network is utilized to replace the traditional shallow network and extract deep features. Meanwhile, for eliminating the negative effect of padding and making better use of the deep network, the proposed algorithm adopts the spatial aware sampling strategy to overcome the strict translation invariance, and a final response map with high quality can be obtained by using the multi-layer feature fusion. Thus, the tracker can significantly reduce the impact of the distractors in complex scenes. In addition, an adaptive feature information fusion is adopted to update the template, so that the algorithm can adapt to various changes of the target appearance.

In summary, the main contributions of this paper are listed below in threefold:

- We utilize the improved ResNet-18 network to replace the traditional shallow network and extract features with more semantic information (information can represent the attribute of the target).
- We propose a novel strategy of multi-layer feature fusion to obtain a final response map with high quality. Thus, the tracker can significantly reduce the impact of the distractors in complex scenes.
- We adopt an adaptive feature information fusion to update the constant template in Siamese network. This makes the tracker more adaptive to various changes of the target appearance.

The rest of the paper is organized as follows: In Section II, we introduce the related works in details. The proposed method is presented in Section III. In Section IV, the experimental details and the objective evaluation of our algorithm will be discussed. In Section V, we reach the conclusions of the paper.

II. RELATED WORK

A. VISUAL TRACKING BASED ON SIAMESE NETWORK

The Siamese network consists of two convolutional neural networks with the same parameters. The network can extract the features of the template branch and the search region branch respectively. Meanwhile, the features can be fused with a specific tensor to obtain a single output. The Siamese network is usually utilized to achieve the similarity learning, which needs no or little online training.

Because of the well-balanced accuracy and tracking speed, Siamese network based trackers gain a lot of attention in recent years. The algorithm of the fully-convolutional Siamese networks (SiamFC) is first proposed by Bertinetto [11]. By utilizing the fully-convolutional structure, SiamFC can achieve end-to-end training and the search branch is no longer limited by size. The SINT tracker [12] uses Siamese network to extract the features of the template and the search region. To increase the tracking accuracy, the algorithm decreases the usage of the max-pooling layer and utilizes the region of interest pooling (ROI pooling) to optimize the process of the feature extraction. Simultaneously, the optical flow method is used to filter the candidates and improve the tracking performance. The GOTURN tracker [13] adopts the Siamese network as the extractor of the features. In the tracking process, the algorithm uses the regression method and the information of the previous frames to locate the target. Valmadre *et al.* [14] incorporates the correlation filter into the template branches, which makes the Siamese network efficient while using the shallow structure. The algorithm not only improves the tracking performance, but also combines the deep features with the correlation filter perfectly. For improving the speed of computation while ensuring the superior performance of the tracking algorithm, the SiamRPN tracker [15] combines the region proposal network (RPN) with the Siamese network. Benefited from the combination, traditional online fine-tuning and the multi-scale test can be removed. The DaSiamRPN [16] tracker introduces a distractor-aware module and promotes the tracking performance of the model in complex scenes. Meanwhile, an effective sampling strategy is used to settle the imbalanced distribution of the training data, which can make the learned features more discriminative. The SiamRPN++ [19] tracker analyzes the impact of the strict translation invariance on Siamese tracker for the first time. In the meantime, the tracker successfully introduces the deep architecture into the

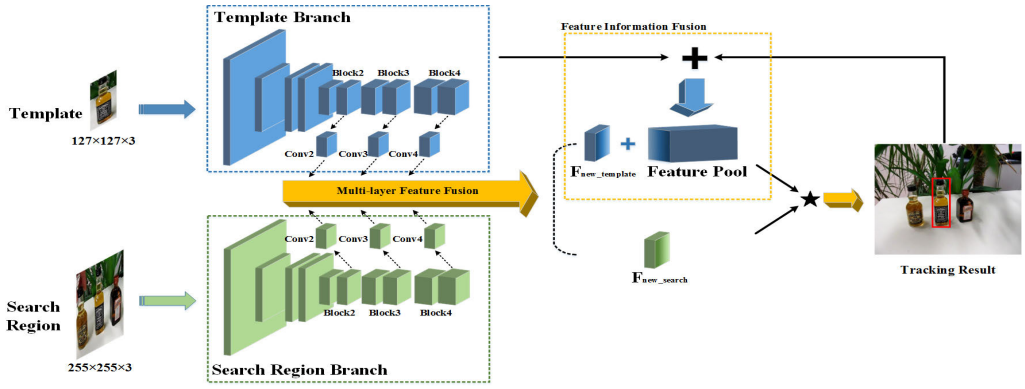


FIGURE 1. The architecture of the proposed tracker SiamDF. The inputs of the tracker consist of two branches: the template branch and the search region branch. The multi-layer feature fusion is utilized to adaptively fuse the features extracted from the selected blocks. The feature information fusion is used to update the features of the template branch. $F_{new\_template}$ means the fused output of the template branch and F_{new_search} means the fused output of the search region branch. Feature pool is used to store the credible features of the target.

algorithm which make the tracking performance significantly improved.

On the basis of these excellent methods, we successfully use the deep architecture in the proposed algorithm. Meanwhile, we adopt the multi-layer feature fusion to obtain a response map with high quality, which can significantly reduce the impact of the distractors in complex scenes. Furthermore, an adaptive feature information fusion is used to update the template, so that the algorithm can adapt to various changes of the target appearance.

B. SOPHISTICATED DEEP ARCHITECTURES

With the proposal of modern convolutional neural network AlexNet [20] in 2012, the research of the network architecture is developing rapidly and many sophisticated deep architectures are presented, such as VGGNet [21], ResNet [22], GoogleNet [23] and MobileNet [24], to name a few. These deep architectures not only provide more excellent ideas for the design of the neural network, but also promote the development of computer vision tasks such as object detection [25], image segmentation [26] and human posture estimation [27], etc.

Although the deep networks are used in many challenging tasks, the shallow networks are still the main choices for the Siamese tracker. In [19], the strict translation invariance is identified as the reason of this restriction and analyzed for the first time. Meanwhile, a spatial aware sampling strategy is proposed to break the restriction and the deep network can be successfully used in the algorithm. On the basis of [19], we introduce the improved ResNet-18 network into the proposed tracker, which makes the tracking performance significantly improved.

C. MULTI-LAYER FEATURE FUSION

In 2013, Zeiler *et al.* [28] used the deconvolution to visualize the features of AlexNet and found many features of the convolution network, one of which is the hierarchy. With the deepening of the network, more semantic information will be obtained. In the field of object detection, Kong *et al.* [29] design a network called HyperNet to fuse the features of the last layer as well as the middle two layers, so that the hierarchy of the CNN features can be fully utilized. In the field of image segmentation, FCN [30] achieves pixel-level segmentation through full convolution network and up-sampling fusion of multi-layer features. In 2015, Ma *et al.* [31] proposed to use multi-layer convolution features to learn and express the tracking target. The comparative study of the CNN features for target tracking shows that the shallow features have more spatial information, which is conducive to accurate positioning. Moreover, the deep features contain more semantic information, which has better invariance to occlusion, deformation and other challenging scenes. At the same time, the response maps of different layers are used to locate the target. Firstly, the area of the target is found by the high-level features of conv5, and then the target is located accurately by using the lower-level features step by step.

Inspired by the above studies, we propose a novel method to achieve multi-layer feature fusion, which can obtain a response map with high quality and significantly reduce the impact of the distractors in complex scenes.

III. METHOD

In this section, we will introduce the proposed tracking framework in detail which is abbreviated into SiamDF. The architecture of the proposed tracker is presented in Fig.1. We first introduce the architecture of the improved ResNet-18 network. Secondly, the multi-layer feature fusion will be described. Finally, a template update strategy with adaptive feature information fusion will be introduced.

A. ARCHITECTURE OF THE IMPROVED RESNET-18

In order to extract the deep features and make the tracking performance significantly improved, the improved

TABLE 1. Layer structure of the improved ResNet-18.

Layer Name	Output Size (template)	Output Size (search region)	Layer Structure
Conv1	62×62	126×126	7×7, 64,2,1
Maxpool	31×31	63×63	3×3,2,1
Block1	31×31	63×63	$\begin{pmatrix}3\times3,64\\3\times3,64\end{pmatrix}\times2$,1,1
Block2	16×16	32×32	$\begin{pmatrix}3\times3,128\\3\times3,128\end{pmatrix}\times2$,2,1
Conv2	8×8	24×24	1×1,128,1,0
Block3	16×16	32×32	$\begin{pmatrix}3\times3,256\\3\times3,256\end{pmatrix}\times2$,1,1
Conv3	8×8	24×24	1×1,128,1,0
Block4	16×16	32×32	$\begin{pmatrix}3\times3,512\\3\times3,512\end{pmatrix}\times2$,1,1
Conv4	8×8	24×24	1×1,128,1,0

ResNet-18 network is introduced into SiamDF. The architecture of the network is shown in Fig.1 (template branch and search region branch), and the layer structure is presented in Table 1.

In Table 1, the layer structure of the convolutional layer and convolutional block (Conv* and Block*) consist of the convolutional kernel size, the number of channels, the step size and the padding size. The layer structure of the Max-pooling layer consists of the kernel size, the step size and the padding size. As can be seen from the Table, we mainly modify the ResNet-18 network from two aspects:

(1) Adjust the steps of the Block3-Block4 to 1. Different from the object detection task, the incremental step size will reduce the accuracy of the tracking algorithm. Meanwhile, the multi-layer feature fusion needs the features to have the same size. Thus, it is convenient to fuse the features into a specific tensor and obtain a single output by utilizing cross-correlation.

(2) Three convolution kernels with 1 × 1 size are introduced into the network. These three convolution kernels (Conv2, Conv3 and Conv4) are used to fuse the feature tensors which are extracted from the Block2, Block3 and Block4. Moreover, these kernels are the basis of the multi-layer feature fusion and their weights can be obtained through the training of the network.

In order to break the restriction of the strict translation invariance, we adopt the spatial aware sampling strategy which is first proposed in [19]. In the training stage of the network, targets are placed in the center with the random translations within ±64 pixels. Thus, the network will not just assign large weights to the image center and the break of the translation invariance caused by the padding will be effectively alleviated.

The loss function for the training stage is described as follows:

$$L(h, r) = \frac{1}{|M|} \sum_{m \in M} l(h[m], r[m]) \tag{1}$$

where M denotes the response map after cross-correlation, $h[m] \in \{+1, -1\}$ and $r[m]$ indicate the label and the score, $m \in M$ is the position in response map M. l is the logistic loss:

$$l(h, r) = log(1 + exp(-hr)) \tag{2}$$

B. MULTI-LAYER FEATURE FUSION

In order to obtain a response map with high quality and significantly reduce the impact of the distractors in complex scenes, we use the multi-layer feature fusion to fuse the feature tensors which are extracted from the Block2, Block3 and Block4.

As can be seen from the Table 1, three convolution kernels (Conv2, Conv3 and Conv4) with 1×1 size are introduced into the template branch and search region branch. These kernels are used to process the outputs of the Block2, Block3 and Block4 respectively. For reducing the heavy computational burden on the cross-correlation, the output channel of each convolution kernel is adjusted to 128, and the output size is cropped to retain the central regions (8×8 and 24×24), which can still capture the feature of the input.

In the fusion stage, we calculate the cross correlation of the output of each convolution kernel (Conv2, Conv3 and Conv4) in the template branch and the search region branch. Thus, we can obtain three response maps corresponding to Block2, Block3 and Block4 respectively. Meanwhile, the average peak-to-correlation energy (APCE) value [32] of each response map can be calculated:

$$APCE = \frac{|M_{\max} - M_{\min}|^2}{mean(\sum_{w,h}(M_{w,h} - M_{\min})^2)} \tag{3}$$

where $M_{\max}$ means the maximum value, $M_{\min}$ means the minimum value, $M_{w,h}$ means the value of row w and column h. After calculating the APCE value of each response map, feature tensors obtained by using Conv2, Conv3 and Conv4 can be fused by utilizing normalized weighted fusion:

$$F_{new} = \sum_{i=2}^{4} \alpha_i * OutConv_i \tag{4}$$

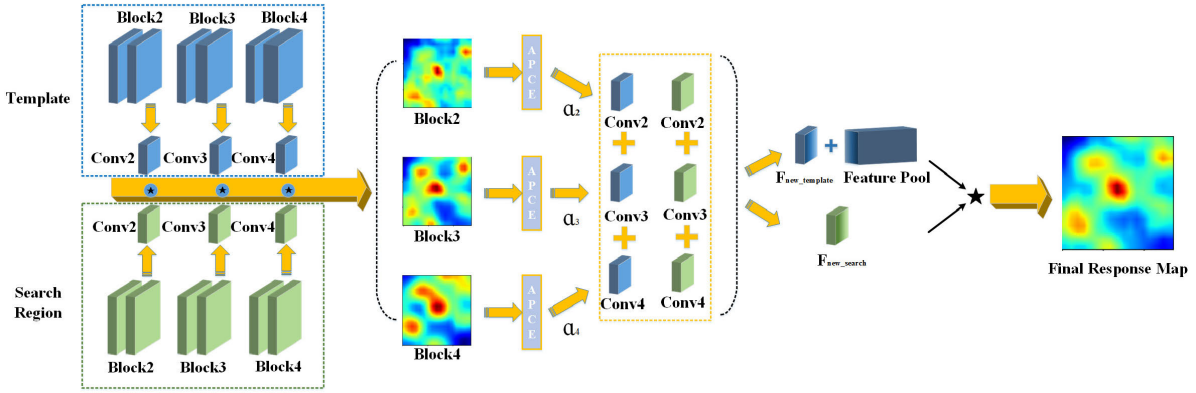


FIGURE 2. The diagram of the multi-layer feature fusion. α_2, α_3 and α_4 represent the normalized weight. $F_{new\_template}$ means the fused output of the template branch and F_{new_search} means the fused output of the search region branch. Feature pool is used to store the credible features of the target.

where $OutConv_i$ means the output feature tensor of Conv(i), F_{new} means the fused output, $\alpha_i = \frac{APCE_i}{\sum_{i=2}^{4} APCE_i}$ means the normalized weight. Thus, multi-layer feature tensors can be adaptively fused by utilizing our novel method.

The diagram of the multi-layer feature fusion can be seen in Fig.2. After fusing the feature tensors extracted from the Block2, Block3 and Block4, the final response map can be obtained. As can be seen from the figure, the response map corresponding to the each Block does not have a high quality. The low peak value and the obvious distractors will have a great impact on tracking performance, sometimes even lead to tracking failure. However, after processing by the multi-layer feature fusion, the peak value can be improved and the distractors are obviously restrained in the final response map. Thus, the tracker can significantly reduce the impact of the distractors and improve the tracking performance in complex scenes. The ''Feature Pool'' in Fig.2 will be introduced in the next section.

C. TEMPLATE UPDATING

In most Siamese network based trackers, the constant template is usually used to search the target. The constant template will not only make the tracker unable to adapt to various changes of the target appearance, but also lead to model drifting. Thus, it is especially important to update the template adaptively.

In order to update the template, we adopt an adaptive feature information fusion to accomplish the task. In the initialization stage, we set up a feature pool to store the credible features of the target, which are extracted from the evaluated tracking results. In our tracker, we still use the APCE value to evaluate the tracking result. If the APCE value of the final response map is larger than a certain threshold T_{APCE}, we will put the tracking result into the template branch (parameters are the same as the current frame) and regard the extracted features as the credible features. When the number of the credible features is less than the upper limit N, the fusion process can be expressed as the following formula:

$$F_{final} = \sum_{k=1}^{n} \omega_k F_k + \eta * F_{new\_result} \quad n \leq N \tag{5}$$

$$\begin{cases} \omega_1 = (1-\eta)^{C-1} & k = 1 \\ \omega_k = (1-\eta)^{C-k} * \eta & 1 < k \leq n \end{cases} \tag{6}$$

where η denotes the update rate, ω_k means the weight of the feature k, n means the number of the existing features in the feature pool, F_k means the feature in the feature pool, C means the update times, F_{new_result} means the fused output of the evaluated tracking result, and the fused output are extracted from the template branch. Thus, we can use the F_{final} as the updated feature of the template to search the target in the next frame. At the same time, the fused output F_{new_result} will be directly added into the feature pool after evaluating and extracting.

However, when the number of the credible features is more than the upper limit N, the feature pool need to be updated. Considering the burden of computation, we adopt a simple strategy to achieve the updating. We assume that the weight and the mean value of the fused output F_{new_result} are ω_{N+1} and μ_{N+1} respectively. There are three possible cases need to be considered:

(1) One of the credible features is invalid. In this case, we define a weight threshold (ω_{min}) of the features in feature pool. If the weight of the feature is lower than the threshold, the feature will be replaced by the new one.

(2) The minimum distance between the new feature and the existing features is shorter than that between any two features in feature pool. In this case, we merge the new feature with the nearest feature in feature pool.

(3) The minimum distance between the new feature and the existing features is longer than that between any two features in feature pool. In this case, we merge

the two closest features in feature pool and put the new feature into the free slot.

In case (2) and case (3), the distance can be calculated by using Euclidean distance and the merging process can be modelled as the following formula:

$$\omega_{update} = \omega_a + \omega_b \tag{7}$$

$$F_{update} = \frac{\omega_a * F_a + \omega_b * F_b}{\omega_a + \omega_b} \tag{8}$$

where a and b denote the two features needed to be merged. Thus, the feature pool can be adaptively updated and the fusion process can keep its high confidence. Moreover, the formula (5) will be constructed as:

$$F_{final} = (1-\eta) * \sum_{k=1}^{n} \omega_k F_k + \eta * F_{new\_result} \quad n > N \tag{9}$$

where ω_k is updated by using formula (7).

IV. EXPERIMENTS

A. IMPLEMENTATION DETAILS

Environment: The proposed tracker is implemented by using pytorch-0.4.1. We perform the experiments on a PC with Intel i7-9800X CPU (3.80GHz), 64GB RAM and a single NVDIA TITAN RTX GPU. The average testing speed is 40 fps.

Training: Inspired by the state-of-the-art Siamese network based trackers. The improved ResNet-18 network of our proposed tracker is pre-trained on ImageNet [33]. We train the network on the training sets of COCO [34] and ImageNet VID. The initial values of the parameters in each layer follow a Gaussian distribution. The weight decay of the SGD is 0.0005. Both template branch and search region branch are trained for 50 epochs with learning rate 0.001. After each epoch, the learning rate is multiplied by a fixed factor until reaching 0.00001 in the final epoch. The momentum is 0.9 and the training batch size is 64. In both training and testing stage, single scale patches with 127×127 pixels are used for the template branch, and patches with 255×255 pixels are used for the search region branch.

Other settings: For the feature information fusion in the template updating, we set the update rate $\eta = 0.4$. The upper limit N is set to $N = 30$ and the minimum template weight is set to $\omega_{\min} = 0.0036$. For evaluating the final response map, we set the certain threshold $T_{APCE} = 8$. All the parameter values which represent the best performance of the proposed tracker are chosen by extensive experiments.

B. ABLATION STUDY

1) MULTI-LAYER FEATURE FUSION

In order to verify the contribution of the multi-layer feature fusion, we conduct the comparative experiment on the OTB100 [18] dataset. Moreover, the influence of different combinations of the fused layers is also illustrated. Fig.3 presents the results and four feature fusion strategies are included: without fusion (SiamDF_OF4), fusion with the outputs of Block2 and Block4 (SiamDF_WF2_4), fusion with the outputs of Block3 and Block4 (SiamDF_WF3_4), and fusion with the outputs of Block2-Block4 (SiamDF_WF2-4). Furthermore, SiamRPN tracker is selected as the baseline tracker to achieve the comparison.

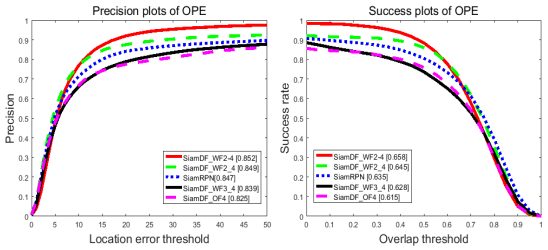


FIGURE 3. Comparison of precision and overlap success on the OTB100 dataset with different fusion strategies.

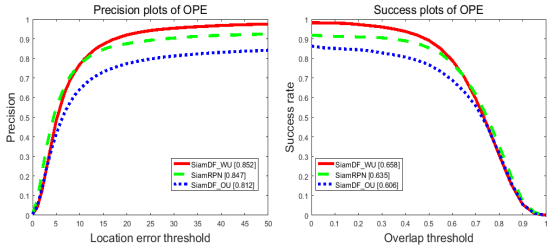


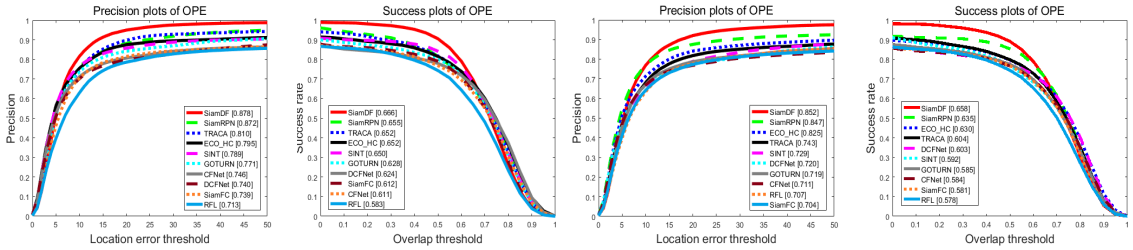
FIGURE 4. Comparison of precision and overlap success on the OTB100 dataset with or without updating strategy.

As can be seen from the Fig.3, the results can only reach 0.825 and 0.615 which are worse than those of the baseline tracker when the fusion is removed. However, when the fusion strategy is added into the tracker, the tracking performance can be obviously improved. Compared with SiamDF_OF4, the results of SiamDF_WF3_4 can reach 0.839 and 0.628, respectively. Moreover, the performances of SiamDF_WF2_4 can exceed the baseline tracker. This shows that the features of the shallow layers are more useful than those of the deep layers in accurate positioning. The best performance is achieved by the SiamDF_WF2-4 which fuses the output features of Block2-Block4.

2) TEMPLATE UPDATING

In order to verify the contribution of the template updating, we also conduct the comparative experiment on the OTB100 dataset. The trackers adopt the same fusion strategy (SiamDF_WF2-4) and two update strategies are included: without updating (SiamDF_OU) and updating with adaptive feature information fusion (SiamDF_WU). SiamRPN tracker is also selected as the baseline tracker to achieve the comparison.

As can be seen from the Fig.4, the results are only 0.812 and 0.605 which are worse than those of the SiamRPN tracker. Compared with the SiamDF_OU, the performances of the SiamDF_WU can reach 0.852 and 0.658, respectively. This shows that the template updating can obviously improve the tracking performance of the tracker.



(a) OTB2013 (b) OTB100

FIGURE 5. The precision plots and the success plots of OPE for 10 trackers. Each tracker is ranked by the performance score. In the precision plot, the score is at error threshold of 20 pixels. In the success plot, the score is the AUC value.

TABLE 2. Detail information about the upper limit *N*.

Value of N	EAO	EFO
10	0.2975	22.46
20	0.3164	21.79
30	0.3360	20.16
40	0.3365	18.37
50	0.3422	15.92
60	0.3428	10.95

At the same time, the upper limit N which represents the limit number of the credible features can also affect the tracking performance. In order to verify its contribution, we utilize the VOT2016 [39] toolkit to conduct the experiment and the detail information can be seen in Table 2. The overall performance of the different value N can be evaluated by using Expected Average Overlap (EAO), and the speed can be evaluated with a normalized speed (EFO).

As can be seen from the Table 2, when the value of N is less than 30, the EAO of the tracker can maintain a significant growth rate. However, when the value of N is larger than 30, the EAO will grow in a slow rate. Moreover, we also find that the EFO will decrease significantly with the increase of N. Thus, we select $N = 30$ as the optimum upper limit.

C. OTB DATASET

1) OVERALL PERFORMANCE COMPARISON

In order to evaluate the performance of our proposed tracker SiamDF, we adopt the widely used OTB [17], [18] sequences to implement the state-of-the-art comparison. We select Siamese network based trackers (SiamFC [11], SINT [12], GOTRUN [13], CFNet [14], DCFNet [35], SiamRPN [15]), correlation filter based trackers (ECO_HC [36], TRACA [37]) and recurrent neural network (RNN) based tracker RFL [38] to achieve the comparison. Meanwhile, the overlap ratio of the bounding box and the error of the center location in the one-pass evaluation (OPE) are employed as the main evaluation mechanism. Moreover, we utilize the success plots and the precision plots to present the evaluating results, the results are shown in Fig.5.

As can be seen from Fig.5 (a), SiamDF can produce leading results in both precision and overlap success. Compared with SiamRPN, SiamDF improves 0.6% in precision and 1.1% in overlap on OTB2013 dataset. It is shown in Fig.5 (b) that the precision of SiamDF is 0.852 and the AUC is 0.658, which improves 0.5% and 2.3%, compared with SiamRPN on OTB100 dataset, respectively.

Compared with other Siamese network based trackers, the superiority of the SiamDF is more obvious. The SiamDF utilizes the improved ResNet-18 network to replace the shallow one. Benefited from the deep network, features extracted from the deep ConvNets are less sensitive to the illumination variation and the background clutter. The rich semantic information can also greatly improve the discriminating ability of the tracker. At the same time, the template used in these trackers is labeled in the first frame, and no longer updates in the tracking process. The constant template will not only make the tracker unable to adapt to various changes of the target appearance, but also can lead to model drifting. SiamDF adopts an adaptive feature information fusion to update the template, so that the tracker can overcome the restriction and obtain an excellent tracking performance.

Compared with the correlation filter based trackers (ECO_HC and TRACA), SiamDF still achieves better performances. By using the multi-layer feature fusion, the features in different ConvNets can be fully utilized and the final response map with a high quality is the key to compete with these trackers in complex scenes.

2) ATTRIBUTE COMPARISON

We compare the performances of the trackers by using 11 annotated attributes in OTB100 dataset. The precision plots and the success plots are shown in Fig.6 and Fig.7 respectively. The results demonstrate that SiamDF performs well in the attributes of background clutter, deformation, illumination variation, in-plane rotation, occlusion, out-of-view, and out-of-plane rotation.

Although SiamDF can achieve an excellent tracking performance in most challenges, it cannot perform well in the attributes of scale variation and low resolution. Compared with the other trackers with higher scores like SiamRPN, SiamDF does not improve the strategy to deal with the scale variation. It only utilizes limited scale factors to overcome the problem. In addition, the application of the deep network may result in the loss of the target information, which only contains no more than 400 pixels.

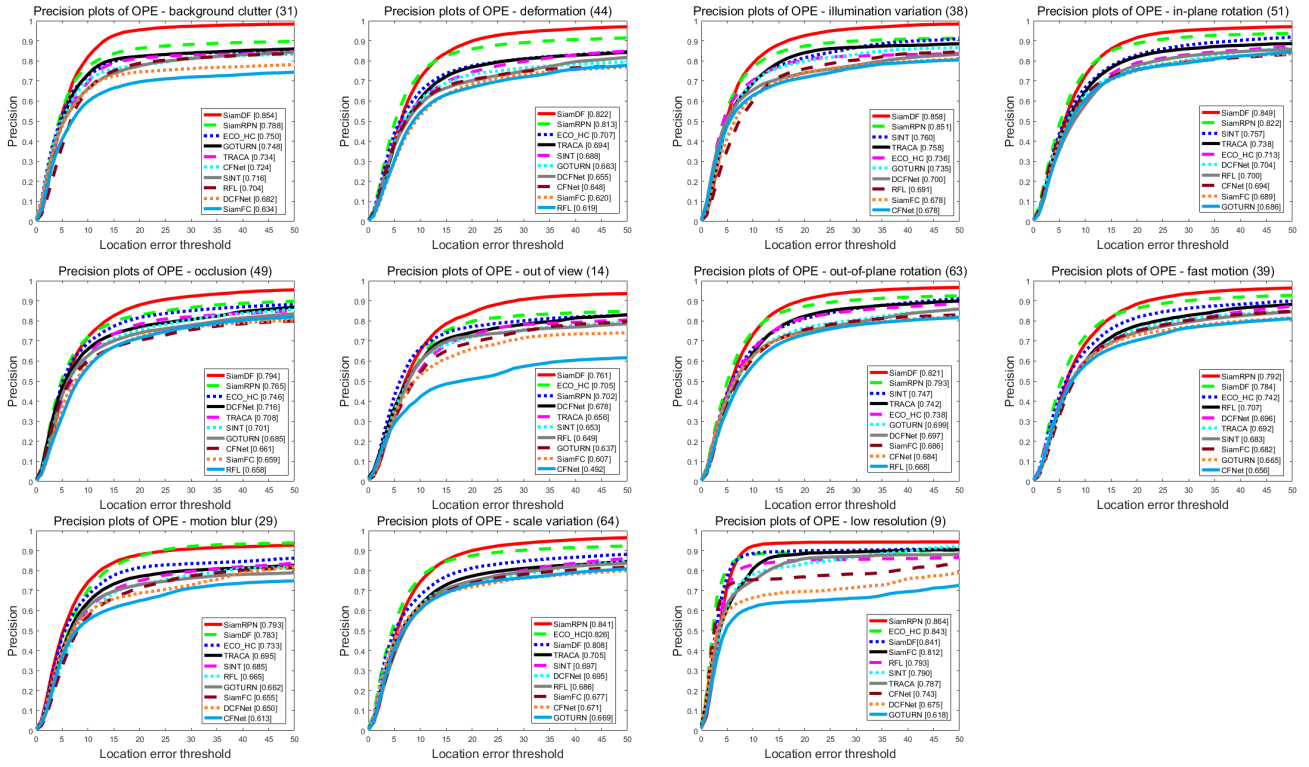


FIGURE 6. Attribute-based precision plots on the OTB100 dataset. The later digits of the title denote the number of the sequences with that attribute.

3) QUALITATIVE COMPARISON

To better analyze the tracking performance of SiamDF, the tracker is compared with other trackers in some challenging video sequences. At the same time, in order to show the tracking results clearly, only six excellent tracking algorithms are shown in Fig.8.

As can be seen from the Fig.8, SiamDF performs well in most complex scenes. In order to further validate the superiority of the SiamDF, we will choose some typical results to analyze.

a: FREEMAN4

The test sequence mainly shows a man freely walk through the classroom. The main challenge of this sequence is the occlusion. The target is nearly completely occluded and the SiamFC tracker directly loses the target in the 42-*th* frame. Both SiamRPN and TRACA are also confused by the occlusion in the subsequent sequence. However, SiamDF can perform well and achieve stable results in the whole tracking process, since the adaptive feature information fusion is added to update the template.

b: FOOTBALL

The test sequence mainly shows a fierce football game. The main challenges of this seuquence are the background clutter and the occlusion. SiamFC loses the target in the 107-*th* frame, and other Siamese network based trackers meet the drifting problem in the following frames because of their constant templates. Although some trackers can redetect the target, the tracking performance has been affected. SiamDF can perform better than other trackers in the whole tracking process. The deep features with rich semantic information can make the tracker less sensitive to the background clutter. The multi-layer feature fusion and the adaptive template are also the keys to achieve the stable results.

c: BOLT2

The test sequence mainly shows a running competition. The main challenges are the deformation and the similar object in the background. In the 133-*th* frame, the SINT tracker fail to track the target and cannot redetect the target in the following frames. Other trackers meet the drifting problem from the 185-*th* frame, and some trackers even track the similar objects. Compared with these trackers, SiamDF can continuously follow the target and achieve a stable tracking performance. The results can illustrate that our proposed tracker can deal with the distractors in the complex scenes.

d: SKATING1

The main challenges of this sequence are the illumination variation, the occlusion and the background clutter. The tracking performance of the trackers begin to differ in the 175-*th* frame: SiamRPN, TRACA and SINT gradually get out of the target. In the 194-*th* frame, trackers excluding SiamDF and TRACA have lost the true target and track the similar objects in the background. In the 280-*th* frame, the illumination has

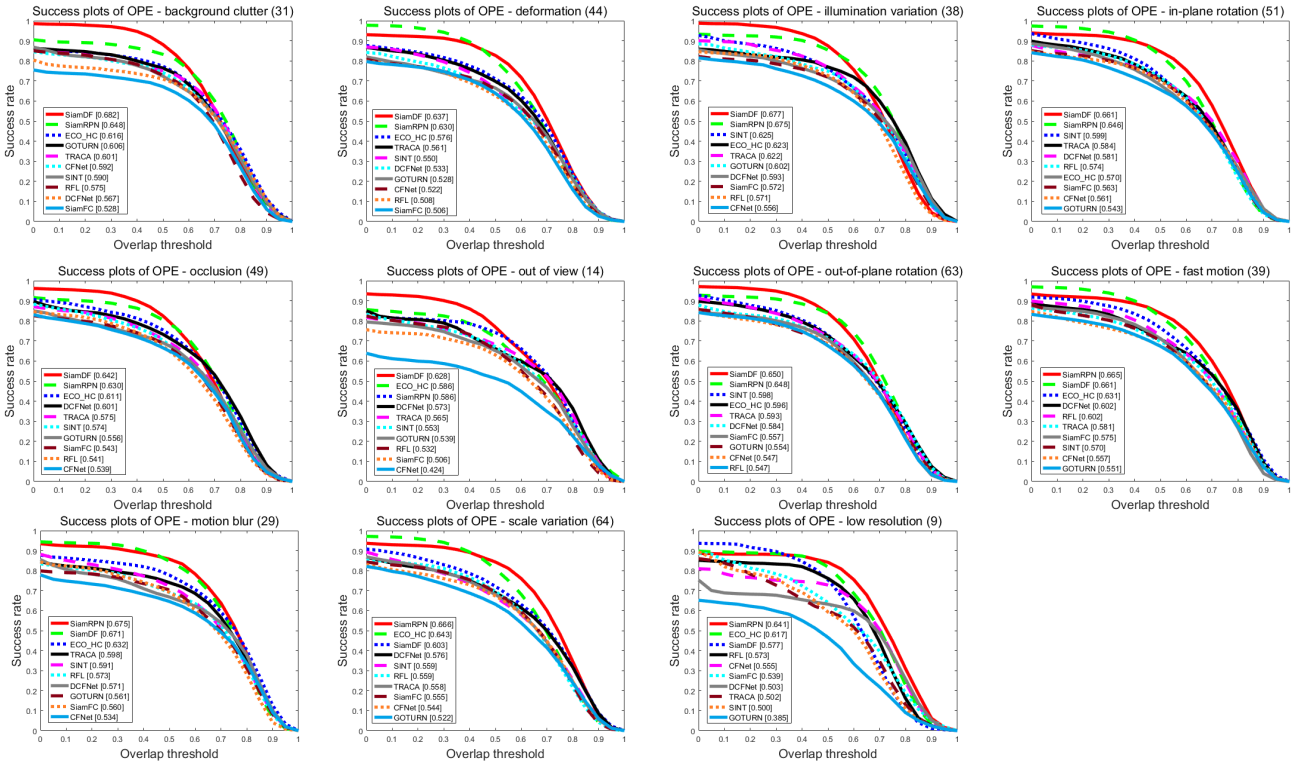


FIGURE 7. Attribute-based success plots on the OTB100 dataset. The later digits of the title denote the number of the sequences with that attribute.

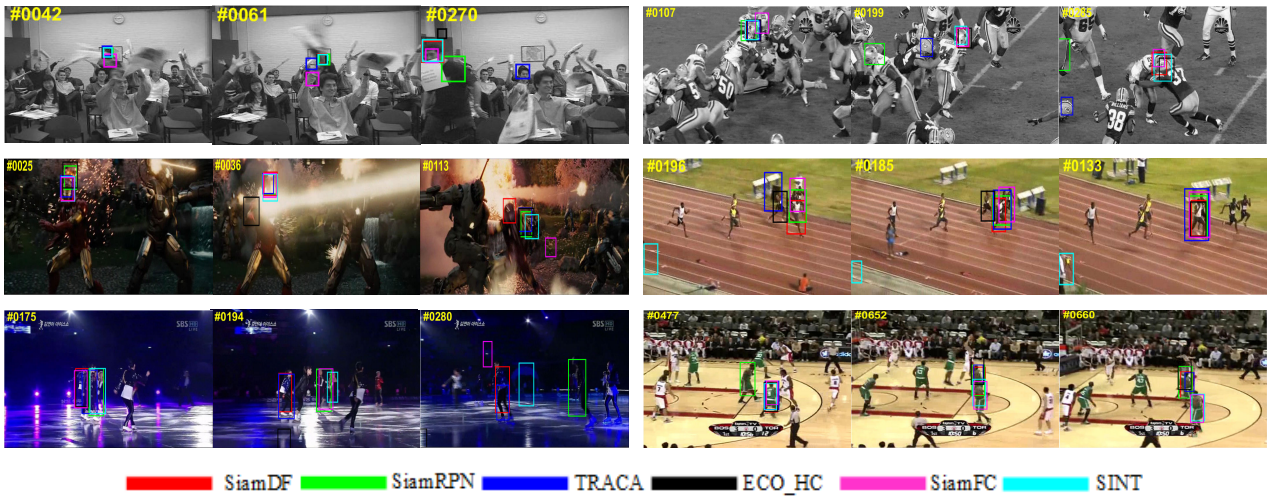


FIGURE 8. Sample tracking results on challenging OTB sequences (from top to down are Freeman4, Football, Ironman, Bolt2, Skating1 and Basketball). For the sake of clarity, six trackers are shown in the result.

degraded and trackers excluding SiamDF fail to track. The results can illustrate that our tracker can not only improve the tracking performance in the complex scenes, but also can adapt to the illumination variation in the tracking process.

4) FAILURE ANALYSIS

Although our proposed tracker SiamDF can achieve an excellent performance in the qualitative comparison, it cannot deal with all the challenges in the OTB sequences. We show sample tracking failures in Fig.9. In the case of low resolution and scale variation (Skiing and CarScale), SiamDF cannot perform well. By analyzing the results and the structure of SiamDF, we find that the direct application of the deep network in SiamDF may result in the loss of the target information, which is the main reason to influence the tracking performance in the attribute of low resolution. Moreover, the

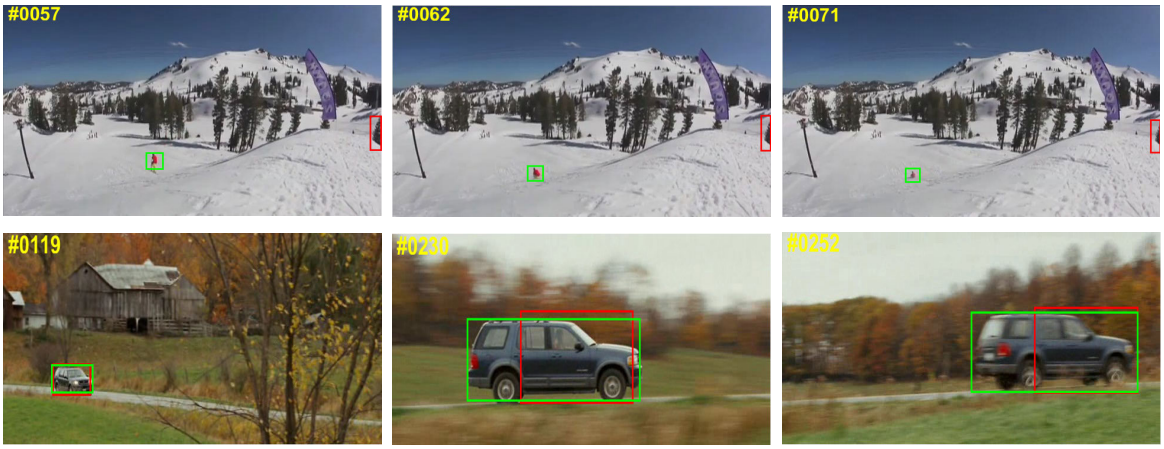


FIGURE 9. Failure cases on Skiing and CarScale sequences. The Green boxes are the ground truth and the red ones are results of SiamDF.

TABLE 3. Detail information about the trackers.

Tracker	EAO	Accuracy	Failure
TCNN [40]	0.3249	0.55	17.94
ECO-HC [36]	0.2665	0.51	21.40
SSAT [39]	0.3207	0.56	19.27
MLDF [39]	0.3106	0.49	15.04
DDC [39]	0.2929	0.53	20.98
SiamRN [39]	0.2766	0.55	24.00
MDNet_N [8]	0.2572	0.54	21.08
SiamAN [39]	0.2352	0.53	29.80
Staple [41]	0.2952	0.54	23.89
C-COT [42]	0.3310	0.53	16.58
TRACA [37]	0.1599	0.46	37.95
DCFNet [35]	0.2071	0.49	24.94
SiamFC [11]	0.2039	0.51	24.67
SiamRPN [15]	0.3979	0.60	14.46
SiamDF	0.3360	0.57	12.18

limited scale factors are used to deal with the scale variation. When faced the great variation, this strategy may be invalid. Overall, how to improve the deep network and adopt an adaptive strategy to deal with the scale variation still need to be further solved.

D. VOT2016

VOT2016 [39] is one of the most widely used databases in the visual tracking field. The VOT2016 database consists of 60 sequences and the performances of the trackers can be evaluated by utilizing accuracy (average overlap while tracking successfully) and robustness (failure times). Moreover,

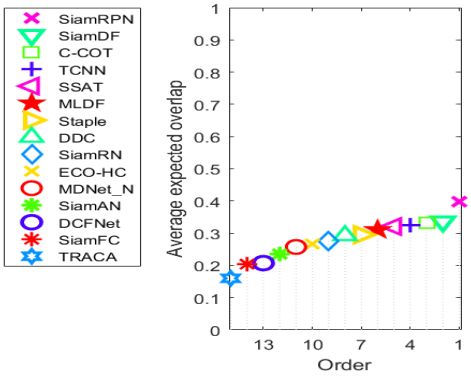


FIGURE 10. Expected average overlap scores of 11 trackers in VOT2016 challenge.

the overall performance of the algorithm is evaluated by using Expected Average Overlap (EAO). In this paper, we compare our proposed tracker SiamDF with 14 excellent trackers. Fig.10 illustrates the EAO ranking and Table 3 shows the detail information about the trackers.

In the Table 3, red, blue and green colors mean the 1st, 2nd and 3rd rank of the performance. It can be seen from the table, though SiamDF only achieves the 2nd place in Accuracy and EAO, it can outperform the other trackers in Failure. The evaluation shows that the overall performance of our tracker can be effectively improved by integrating the deep features and the robust feature fusion.

V. CONCLUSION

In this paper, we propose a Siamese network based tracker with deep features and robust feature fusion. The improved ResNet-18 network is utilized to replace the traditional shallow network and extract deep features. Meanwhile, for eliminating the negative effect of padding and making better use of the deep network, the proposed tracker adopts the spatial aware sampling strategy to overcome the strict translation invariance, and a final response map with high quality can

be obtained by using the multi-layer feature fusion. Thus, the tracker can significantly reduce the impact of the distractors in complex scenes. In addition, an adaptive feature information fusion is adopted to update the template, so that the algorithm can adapt to various changes of the target appearance. Sufficient and reliable experiments show that our tracker can effectively improve the tracking performance and performs favorably in both robustness and accuracy. In order to achieve a better tracking performance, the next step of our work is to improve the deep network and adopt an adaptive strategy to deal with the scale variation.

ACKNOWLEDGMENT

Daqun Li would like to thank Yi Yu and Xize Wang for the support. The authors wish to thank the anonymous reviewers for their valuable suggestions.

REFERENCES

[1] W. Kang, X. Li, S. Li, and G. Liu, "Corrected continuous correlation filter for long–term tracking," *IEEE Access*, vol. 6, pp. 11959–11969, 2018.

[2] A. Koubaa and B. Qureshi, "DroneTrack: Cloud–based real–time object tracking using unmanned aerial vehicles over the Internet," *IEEE Access*, vol. 6, pp. 13810–13824, 2018.

[3] R. J. Mstafa, K. M. Elleithy, and E. Abdelfattah, "A robust and secure video steganography method in DWT–DCT domains based on multiple object tracking and ECC," *IEEE Access*, vol. 5, pp. 5354–5365, 2017.

[4] A. Toshev and C. Szegedy, "DeepPose: Human pose estimation via deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1653–1660.

[5] Y. Ioannou, D. Robertson, R. Cipolla, and A. Criminisi, "Deep Roots: Improving CNN efficiency with hierarchical filter groups," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5977–5986.

[6] X. Sheng, Y. Liu, H. Liang, F. Li, and Y. Man, "Robust visual tracking via an improved background aware correlation filter," *IEEE Access*, vol. 7, pp. 24877–24888, 2019.

[7] D.-H. Lee, "One–shot scale and angle estimation for fast visual object tracking," *IEEE Access*, vol. 7, pp. 55477–55484, 2019.

[8] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4293–4302.

[9] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2016, pp. 472–488.

[10] B. Han, J. Sim, and H. Adam, "BranchOut: Regularization for online ensemble tracking with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 521–530.

[11] L. Bertinetto, J. Valmadre, O. F. Jo Henriques, A. Vedaldi, and H. S. Philip Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 850–865.

[12] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1420–1429.

[13] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 749–765.

[14] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5000–5008.

[15] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.

[16] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 103–109.

[17] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2411–2418.

[18] Y. Wu, J. Lim, and M. H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, Sep. 2015.

[19] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of siamese visual tracking with very deep networks," Dec. 2018, *arXiv:1812.11703*. [Online]. Available: https://arxiv.org/abs/1812.11703

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[21] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Apr. 2015, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[24] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," Apr. 2017, *arXiv:1704.04861*. [Online]. Available: https://arxiv.org/abs/1704.04861

[25] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, and J. Sun, "MegDet: A large mini–batch object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6181–6189.

[26] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 833–851.

[27] W. Tang, P. Yu, and Y. Wu, "Deeply learned compositional models for human pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 197–214.

[28] M. D. Zeiler, G. W. Taylor, and R. Fergus, "Adaptive deconvolutional networks for mid and high level feature learning," in *Proc. Int. Conf. Comput. Vis.*, Nov. 201, pp. 2018–2025.

[29] T. Kong, A. Yao, Y. Chen, and F. Sun, "HyperNet: Towards accurate region proposal generation and joint object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 845–853.

[30] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Nov. 2014.

[31] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3074–3082.

[32] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4800–4808.

[33] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[34] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 740–755.

[35] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu, "DCFNet: Discriminant correlation filters network for visual tracking," Apr. 2017, *arXiv:1704.04057*. [Online]. Available: https://arxiv.org/abs/1704.04057

[36] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6931–6939.

[37] J. Choi, H. J. Chang, T. Fischer, S. Yun, K. Lee, J. Jeong, Y. Demiris, and J. Y. Choi, "Context–aware deep feature compression for high–speed visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 479–488.

[38] T. Yang and A. B. Chan, "Recurrent filter learning for visual tracking," Aug. 2017, *arXiv:1708.03874*. [Online]. Available: https://arxiv.org/abs/1708.03874

[39] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin, T. Vojír, G. Häger, A. Lukežič, G. Fernández, A. Gupta, A. Petrosino, A. Memarmoghadam, A. Garcia-Martin, and A. S. Montero, "The visual object tracking vot2016 challenge results," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct.2016, pp. 777–823.

[40] H. Nam, M. Baek, and B. Han, "Modeling and propagating CNNs in a tree structure for visual tracking," Aug. 2016, *arXiv:1608.07242*. [Online]. Available: https://arxiv.org/abs/1608.07242
[41] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real–time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1401–1409.
[42] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 621–629.

DAQUN LI received the B.E. degree in optoelectronic information and the M.E. degree in optical engineering from the Beijing Institute of Technology, Beijing, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree in optical engineering with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun, China. His research interests include visual tracking, object detection, and deep learning.

XIZE WANG is currently pursuing the degree in mechanical engineering with the Technion–Israel Institute of Technology, Israel. His research interests include robot design, object tracking, computer vision, and machine learning.

YI YU received the Ph.D. degree in optical engineering from the Changchun Institute of Technology, Changchun, China, in 2016. He is currently a Professor with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun. His research interests include optoelectronic measurement, image/video processing, computer vision, and machine learning.