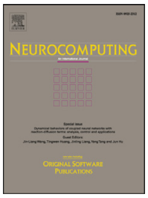
Neurocomputing 401 (2020) 36–47

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Siamese Deformable Cross-Correlation Network for Real-Time Visual Tracking

Linyu Zheng[a,b,c], Yingying Chen[a,b,c,d,*], Ming Tang[a,b,c], Jinqiao Wang[a,b,c], Hanqing Lu[a,b,c]

[a] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, No.95, Zhongguancun East Road, Beijing 100190, China
[b] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China
[c] University of Chinese Academy of Sciences, Beijing 100049, China
[d] The Key Laboratory of Rich-Media Knowledge Organization and Service of Digital Publishing Content, Institute of Scientific and Technical Information of China, Beijing 100038, China

ARTICLE INFO

Article history:
Received 15 July 2019
Revised 18 December 2019
Accepted 20 February 2020
Available online 24 February 2020

Communicated by Prof. Zhou Xiuzhuang

Keywords:
Visual Tracking
Convolutional Neural Networks
Siamese network
Deformable Convolutional Network

ABSTRACT

In recent years, SiamFC-based trackers have received much attention because of their great potentials in balancing tracking accuracy and speed. However, the robustness of most such trackers is greatly affected by the large deformations of targets. We argue that in the cross-correlation operation which is widely used by modern SiamFC-based trackers, the static correlation between the template kernel and the feature maps of test sample is difficult to adapt to the large deformation of the target object. In this paper, we propose a Siamese deformable cross-correlation network (SiamDCN), which introduces the deformable cross-correlation operation into SiamFC in an online self-adaptive way, for robust visual tracking. Compared to the previous SiamFC-based trackers, our SiamDCN is more robust to the large deformations of targets through dynamically and adaptively adjusting the location of correlation calculation for each element of the template kernel in the cross-correlation operation. Moreover, we build a twofold Siamese network, named SiamDCN+, which consists of a SiamDCN branch and a SiamFC branch, for accurate and real-time visual tracking after observing that the features learned in SiamFC are static and discriminative, whereas those in SiamDCN are dynamic and robust, and they complement each other. Extensive experiments on three public benchmarks, OTB2015, VOT2016, and VOT2017, show that the proposed SiamDCN achieves superior localization accuracy than its baseline tracker SiamFC and the proposed SiamDCN+ achieves competitive performance compared to state-of-the-art real-time trackers, while running beyond 40 FPS.

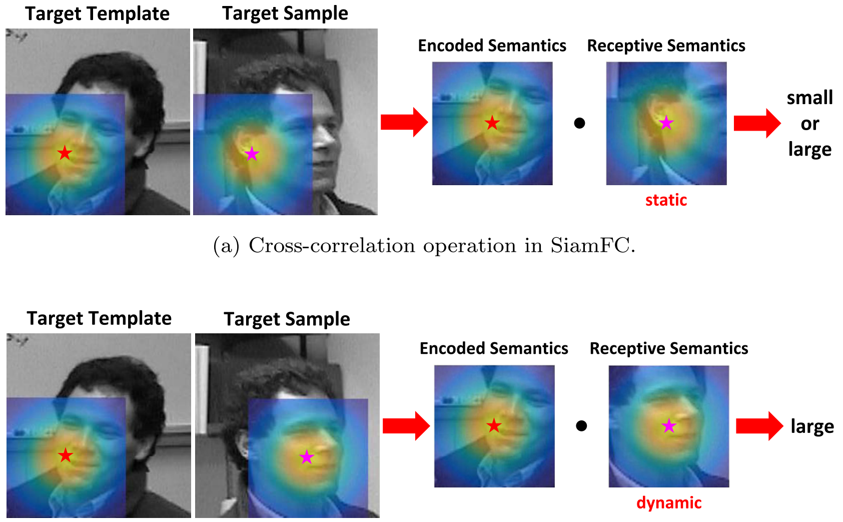
1. Introduction

Visual object tracking is one of the most fundamental problems in computer vision with many applications. Given the initial state (e.g., position and size) of a target object in the first frame, the goal of visual tracking is to estimate the states of the target in the subsequent frames [1]. While visual object tracking finds numerous applications in surveillance, autonomous systems, and augmented reality, it is challenging in the large appearance changes of target objects caused by deformation, view angle, and fast motion. Besides, the running speed is also important in practical applications [2–4].

From a technical standpoint, most modern trackers can be roughly divided into two branches. The first branch is based on correlation filters [5–8], which trains a regressor online by exploiting the properties of circular correlation and transforming the convolution operation from time-domain to frequency-domain so that it can largely improve the computation efficiency and speed. However, most recent correlation filters-based trackers suffer from low frame-per-second (FPS) in pursuit of higher localization accuracy by relaxing the boundary effect [9–11] or exploiting high-dimensional convolutional neural networks features [12–14]. Another branch is based on Siamese convolutional neural networks [15–19], which treats tracking as a problem of similarity learning and is trained offline. Typically, SiamFC [16] employs a fully-convolutional Siamese network to extract the feature maps of the target template and search region, then uses a simple cross-correlation operation to perform dense and efficient evaluations in the search region, with beyond real-time speed. However, its robustness and discrimination power are not sufficient for accurate tracking [20–22] even though vast training datas are employed.

[*] Corresponding author. Tel.: +86-10-82544774.
E-mail address: yingying.chen@nlpr.ia.ac.cn (Y. Chen).

https://doi.org/10.1016/j.neucom.2020.02.080
0925-2312/© 2020 Elsevier B.V. All rights reserved.



(a) Cross-correlation operation in SiamFC.

(b) Deformable cross-correlation operation in the proposed SiamDCN.

Fig. 1. Illustration of the cross-correlation operation in SiamFC and the deformable cross-correlation operation in SiamDCN, where the target experiences a large deformation. The colorful pentagram denotes the center of the receptive field of one element in the template kernel or the feature maps of target sample. The red pentagram receives the semantic information of the pink pentagram by correlation calculation. In (a), although we expect the correlation value between the two pentagrams is large, this is hard to be guaranteed because they cover different semantic informations. Whereas in (b), this is easy to be because the red pentagram is learned to compute correlation value with the shifted pink pentagram and they cover similar semantic informations.

Currently, state-of-the-art SiamFC-based trackers can be roughly fallen into two categories. One category is to formulate tracking as a problem of one-shot detection and regress rough localization results accurately by utilizing region proposal networks [23]. Despite such trackers [24,25] achieve high performance on multiple challenging benchmarks [26–28], they do not aim to improve the robustness or discrimination power of SiamFC. Another category is to improve the discrimination power of SiamFC by various techniques such as learning attentions [20], fusing complementary features [21], introducing graph convolution networks [29], and improving the sampling strategy of offline training [30]. However, they are hard to be trained and their robustness struggles when the target experiences large deformations. We argue that the main reason for this problem is that in the cross-correlation operation which is widely used by modern SiamFC-based trackers, the static correlation between the template kernel (*i.e.* the feature maps of target template) and the feature maps of test sample is difficult to adapt to the large deformation of the target object. Intuitively, this is because for one element of the template kernel that has a fixed receptive field location and encodes specific semantic information, it is hard for it to generate a large correlation value with the features of the same receptive field location in the test sample of target when the target experiences a large deformation and leads to the semantic informations of the same receptive field location in the target template and the test sample of target are obviously different (see Fig. 1a). However, this ability is necessary to ensure the robustness of SiamFC-based trackers especially when online updates are not performed.

To solve the above problem, in this paper, we propose a Siamese deformable cross-correlation network (SiamDCN), which exploits the deformable cross-correlation operation [31] to advance the robustness of SiamFC, for robust visual tracking. As shown in Fig. 2, instead of the cross-correlation operation, our SiamDCN introduces the deformable cross-correlation operation into SiamFC in an online self-adaptive way to perform dense and efficient evaluations in a search region. Specifically, two extra branches (the red convolutional layer and the orange one) are trained to learn to generate the offsets map used in the deformable cross-correlation operation in the similar way that SiamRPN [24] generates the regression values of bounding boxes. Compared to SiamFC, SiamDCN is more robust to the large deformations of targets through dynamically and adaptively adjusting the location of correlation calculation for each element of the template kernel in the cross-correlation operation (Fig. 1b). Therefore, it is easier to be trained and more robust than the previous SiamFC-based trackers.

Moreover, we build a twofold Siamese network, named SiamDCN+, for accurate and real-time visual tracking. Our SiamDCN+ consists of two separately trained branches. One branch is SiamFC whose learned features are considered as static because the locations of correlation calculations for all elements of the template kernel in its cross-correlation operation are always static and invariant. Another branch is SiamDCN whose learned features are considered as dynamic because the locations of correlation calculations for all elements of the template kernel in its deformable cross-correlation operation are determined by the offsets map generated in an online self-adaptive way and change as the appearance changes of the target, thus they are dynamic and variant. After observing that the static features learned in SiamFC are discriminative, whereas the dynamic features learned in SiamDCN are robust, and they complement each other, we combine these two branches in the similar way that SA-Siam [21] combines the semantic branch and the appearance branch, for accurate visual tracking.

To the best of our knowledge, the proposed SiamDCN is the first tracker which introduces the deformable cross-correlation operation into SiamFC in an online self-adaptive way, and the proposed SiamDCN+ is the first tracker which combines the static and discriminative features and the dynamic and robust features to achieve accurate visual tracking. Extensive experiments are performed on three public benchmarks, OTB2015 [1], VOT2016 [26], and VOT2017 [27]. Our SiamDCN achieves superior localization accuracy than its baseline tracker SiamFC, and our SiamDCN+ achieves competitive performance compared to state-of-the-art real-time trackers, while running beyond 40 FPS.

2. Related Work

Since one of the main contributions of this paper is the SiamDCN which exploits the deformable cross-correlation operation to advance the robustness of SiamFC in an online

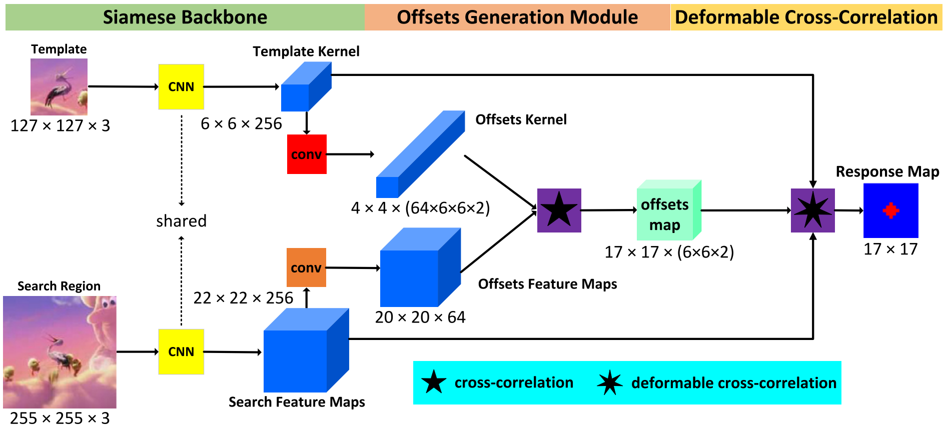


Fig. 2. Main framework of the proposed SiamDCN. Left side is a fully-convolutional Siamese network for features extraction. Right side is an online adaptive deformable cross-correlation module for score evaluation of each candidate in the search region. The red convolutional layer and the orange convolutional layer are trained to learn to generate the kernel and feature maps of the offsets used in the deformable cross-correlation operation, respectively. Best viewed in color. See Sec. 3 for details.

adaptive way and another is the SiamDCN+ which fuses SiamFC and SiamDCN to achieve accurate visual tracking, in this section, we give a brief review on the following four aspects related to our work: Siamese network based trackers, ensemble trackers, deformable convolutional networks, and template matching based trackers, then state the key differences between our approach and others.

2.1. Siamese Network Based Trackers

For the first time, SINT [15] models visual object tracking as a problem of similarity learning and applies the Siamese network to tracking. By comparing the target image patch with the candidate patches in a search region, it can track the target object to the location where the highest similarity score is obtained. A notable advantage of SINT is that it does not need online learning and update. However, SINT can only run at 4 FPS because its fully-connected layers are very time-consuming. To achieve real-time tracking speed, SiamFC [16] employs a fully-convolutional Siamese network to extract the feature maps of target template and search region, then uses a simple cross-correlation layer to perform dense and efficient sliding-window evaluations in the search region. As a result, each patch in the search region with the same size as the target template gets a similarity score, and the one with the highest score is identified as the target object. SiamFC achieves beyond real-time tracking speed benefitting from the efficient fully-convolutional network and no need for online learning and update. However, its robustness and discrimination power are not sufficient for accurate visual tracking [20–22] even though vast training datas are employed.

To improve the localization accuracy of SiamFC, SiamRPN [24] introduces an elegant region proposal subnetwork [23] to SiamFC, and achieves high performance on multiple challenging datasets [26–28]. To improve the discrimination power of SiamFC, SA-Siam [21] implements a two-branch Siamese network with one branch for semantic and the other for appearance, RASNet [20] introduces three different attention mechanisms for selecting the most discriminative features of a given target object in an online self-adaptive and weighted way, DaSiamRPN [30] designs a novel sampling strategy for offline training, SiamImp [32] proposes a novel triplet loss instead of pairwise loss to train SiamFC, and GCNT [29] introduces graph convolutional networks to SiamFC. However, all above SiamFC-based trackers pay little attention to improving the robustness of SiamFC especially when the target experiences large deformations. Therefore, their robustness struggles, and the main cause of this problem has already been analyzed in Sec. 1 and Fig. 1a in detail.

All previous SiamFC-based trackers perform dense evaluations through the cross-correlation operation where the locations of correlation calculations for all elements of the template kernel are always static and invariant (Fig. 1a), resulting in weak robustness when the target experiences large deformations. Different from them, to improve the robustness, our SiamDCN introduces the deformable cross-correlation operation into SiamFC in an online self-adaptive way. This improvement enables the locations of correlation calculations for all elements of the template kernel to change as the appearance changes of the target object dynamically and adaptively (Fig. 1b). Therefore, its robustness is stronger than the previous SiamFC-based trackers.

2.2. Ensemble Trackers

Our proposed SiamDCN+ is composed of two separately trained branches focusing on different types of convolutional neural networks (CNNs) features. It shares some insights and design principles with the following ensemble trackers.

TCNN [33] maintains multiple CNNs in a tree structure to learn ensemble models and estimate target states. BranchOut [34] employs the CNNs for target representation, which has a few common convolutional layers but multiple branches of fully-connected layers. It allows each branch to have a different number of layers so as to maintain variable abstraction levels of the target appearance. PTAV [35] keeps two classifiers, one acting as a tracker and the other acting as a verifier. The combination of an efficient tracker which runs for all frames and a powerful verifier which only runs when necessary strikes a good balance between tracking accuracy and speed.

The most related ensemble tracker with our SiamDCN+ is SA-Siam [21] which is also a SimaFC-based tracker. The main difference between SiamDCN+ and SA-Siam lies in the different CNNs features selected for fusion. Specifically, SA-Siam proposes a twofold Siamese network comprised of a semantic branch and an appearance branch after observing that the semantic features learned in an image classification task and the appearance features learned in SiamFC complement each other, while our SiamDCN+ is comprised of a dynamic branch and a static branch after observing that the dynamic features learned in SiamDCN are robust and the static features learned in SiamFC are discriminative, and they complement each other. Therefore, the basic observation and motivation of SA-Siam and SiamDCN+ are different.

2.3. Deformable Convolutional Networks

It is well-known that deformable convolution [31] can enhance the performance of image recognition, object detection, and semantic segmentation, with trivial extra computation. Deformable convolutional networks [31] develop the deformable convolution by adding 2D offsets on the regular sampling grid in the regular convolution, where the offsets are learned from the preceding feature maps. To our best knowledge, our SiamDCN is the first tracker which introduces the deformable convolution into SiamFC, and it is called the deformable cross-correlation in this paper for a more precise representation. Different from other tasks which exploit the deformable convolution to improve their performance, SiamDCN introduces it in an online self-adaptive way [24], where the 2D offsets are adaptively determined by the given and constant target template along with the ever-changing search region in an image sequence together.

2.4. Template Matching Based Trackers

Best-Buddies Similarity (BBS) [36,37] proposes a novel method for template matching in unconstrained environments based on properties of the Nearest-Neighbor match between the features of target and the features of template. It relies only on a subset of the points that are best buddies in the template, thus is robust against complex geometric deformations and high levels of outliers. Based on this, BBT [38] applies BBS to tracking and proposes a modification to BBS so that it can handle point sets of different size. By combining the modified BBS and particle filter based framework, BBT obtains good tracking results. Moreover, TM^3 [39] proposes Mutual Buddies Similarity (MBS) to evaluate the similarity between two image regions for tracking, where every image region is split into a set of non-overlapped small image patches and only the patches within the reciprocal nearest neighbor relationship are considered so as to the similarity computation relies on a subset of reliable pairs. As a result, TM^3 is more robust to significant outliers and appearance variations than BBT.

After observing that a global similarity metric is often suboptimal for template matching based trackers when the target object experiences large appearance variations or occlusions, SWSF [40] employs a part-based model as the object representation, and jointly learns the local similarity metric and spatially regularized weights in a coherent process, such that the total matching accuracy between the target and candidates can be effectively enhanced. Beyond the linear cross-correlator, KCC [41] proposes a kernel cross-correlator by defining the correlator in frequency domain directly. It breaks the traditional limitation and assumption of KCF [5] and does not impose any constraints on the training data, thus is applicable for affine transform prediction (e.g., translation, rotation, and scale) and more robust to signal noises and distortions. By utilizing neural networks, DML [42] learns a set of hierarchical nonlinear transformations to project both the template and particles (positive samples and negative samples) into the same feature space where the intra-class variations of positive training pairs are minimized and the interclass variations of negative training pairs are maximized simultaneously. Our baseline tracker SiamFC [16] can also be regarded as a template matching based tracker where dot product between the target template and the test sample is used to measure the similarity between them. Compared to the above template matching based trackers, the disadvantage which SiamFC and DML share is that the locations of each pair of matched points in their template matching processes are fixed and inflexible, thus they are not robust to the large deformations of targets.

On the one hand, our proposed SiamDCN makes up for the shortcoming of SiamFC and DML by introducing the deformable cross-correlation in an online adaptively way. On the other hand, different from the above template matching based trackers except for SiamFC and DML, SiamDCN utilizes the deformable cross-correlation operation to model the target deformation and automatically learns the adaptive and robust template matching method from a large number of offline training datas with convolutional neural networks, avoiding the tedious manual design for matching criteria where more human interventions are needed. As a result, our SiamDCN is more robust to the large deformations of targets than the previous template matching based trackers.

3. Siamese Deformable Cross-Correlation Network

As shown in Fig. 2, the proposed Siamese deformable cross-correlation network (SiamDCN) consists of three parts: Siamese backbone for features extraction, adaptive offsets generation module, and deformable cross-correlation layer. We will introduce them in detail in this section, respectively.

Here, the input image, feature maps, and convolution are 3D. The deformable cross-correlation operates on the 2D spatial domain, and this operation remains the same across the channel dimension. For simplicity, all of them are described in 2D. Extension to 3D is straightforward.

3.1. Siamese Backbone

In Siamese backbone, we adopt a fully-convolutional neural network without padding to satisfy the definition of fully convolution with stride k:

$$h(\mathbf{I}((x_0, y_0), (x_1, y_1))) = h(\mathbf{I})\left(\left(\frac{x_0}{k}, \frac{y_0}{k}\right), \left(\frac{x_1}{k}, \frac{y_1}{k}\right)\right), \tag{1}$$

where $\mathbf{I}$ is the input image matrix, $\mathbf{I}((x_0, y_0), (x_1, y_1))$ is the submatrix of $\mathbf{I}$ with (x_0, y_0) and (x_1, y_1) as its top-left and down-right corner subscripts, and $h(\cdot)$ is the function for features extraction. Here, we use the modified AlexNet [43] where the groups from conv-2 and conv-4 are removed.

The Siamese backbone for features extraction consists of two branches (the two yellow CNNs in Fig. 2). One is called template branch which receives the target image patch in the historical frame as input and outputs the target feature maps which is called *template kernel* for one-shot detection in this paper. The other is called search branch which receives the search image patch in the current frame as input and outputs the search feature maps. These two branches share parameters in CNNs so that their inputs are implicitly encoded by the same transformation which is suitable for the subsequent tasks. For convenience, we denote the template kernel and the search feature maps as $\mathbf{w}$ and $\mathbf{x}$, respectively, and they will be often used in the subsequent sections.

3.2. Adaptive Offsets Generation Module

The adaptive offsets generation module consists of two convolutional layers (the red convolutional layer and the orange convolutional layer in Fig. 2) and one cross-correlation layer. The red convolutional layer receives the template kernel $\mathbf{w}$ as input and outputs the offsets kernel. The orange convolutional layer receives the search feature maps $\mathbf{x}$ as input and outputs the offsets feature maps. Here, all convolutions are 3×3 without padding to satisfy the definition of fully convolution in Sec. 3.1, and their numbers of input and output channels are marked in Fig. 2. As a result, the offsets kernel and the offsets feature maps perform cross-correlation operation to obtain the offsets map with $6 \times 6 \times 2$ channels. In the offsets map, every two adjacent channels correspond to the offsets of x and y directions for one element in the template kernel, respectively, and the template kernel has 6×6 elements on the 2D spatial domain. For convenience, we denote the offsets map as $\mathbf{v}$, and it will be used in the subsequent deformable cross-correlation operation.

3.3. Deformable Cross-Correlation Layer

The cross-correlation layer in SiamFC [16] receives the output of Sec. 3.1, *i.e.* the template kernel $\mathbf{w}$ and the search feature maps $\mathbf{x}$, as inputs, and outputs the response map $\mathbf{y}$ that indicates the similarity score between the target patch and each candidate patch with the same size as the target in the search region. The 2D cross-correlation operation consists of two steps: 1) sampling using a regular grid $\Re$ over the search feature maps $\mathbf{x}$; 2) summation of the sampled values weighted by the template kernel $\mathbf{w}$. The grid $\Re$ defines the kernel size, dilation and padding of correlation calculation. For example in SiamFC,

$$\Re = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N\} = \{(0,0), (0,1), ..., (5,4), (5,5)\} \tag{2}$$

defines a 6 × 6 kernel with dilation 1 and padding 0, where $N = |\Re|$ and $\mathbf{p} = \left(p_x, p_y\right)$ denotes a 2D spatial location. Specifically, in the cross-correlation operation, for each location $\mathbf{p}$ on the output response map $\mathbf{y}$, it is calculated as

$$\mathbf{y}(\mathbf{p}) = \sum_{\mathbf{p}_n \in \Re} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p} + \mathbf{p}_n), \tag{3}$$

where $\Re$ enumerates all 2D spatial locations of $\mathbf{w}$.

Different from the cross-correlation operation in SiamFC, the deformable cross-correlation operation in our SimaDCN receives the outputs of both Sec. 3.1 and Sec. 3.2, *i.e.* the template kernel $\mathbf{w}$, the search feature maps $\mathbf{x}$, and the offsets map $\mathbf{v}$, as inputs, and outputs the response map $\mathbf{y}$ after augmenting the regular grid $\Re$ in Eq. 3 with a offsets set $\{\Delta\mathbf{p}_n \mid n = 1, ..., N\}$. Specifically, in the deformable cross-correlation operation, for each location $\mathbf{p}$ on the output response map $\mathbf{y}$, it is calculated as

$$\mathbf{y}(\mathbf{p}) = \sum_{\mathbf{p}_n \in \Re} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p} + \mathbf{p}_n + \Delta\mathbf{p}_n), \tag{4}$$

where the values of all $\Delta\mathbf{p}_n$s come from the location $\mathbf{p}$ of the input offsets map $\mathbf{v}$ across the channel dimension.

In Eq. 4, the sampling is on the irregular and offset location $\mathbf{p}_n + \Delta\mathbf{p}_n$. As the offset $\Delta\mathbf{p}_n$ is typically fractional, Eq. 4 is implemented via bilinear interpolation

$$\mathbf{x}(\mathbf{p}^*) = \sum_{\mathbf{q}} G(\mathbf{q}, \mathbf{p}^*) \cdot \mathbf{x}(\mathbf{q}), \tag{5}$$

where $\mathbf{p}^*$ denotes an arbitrary fractional location ($\mathbf{p}^* = \mathbf{p} + \mathbf{p}_n + \Delta\mathbf{p}_n$ for Eq. 4), $\mathbf{q}$ enumerates all integral spatial locations in the search feature maps $\mathbf{x}$, and $G(\cdot, \cdot)$ is the bilinear interpolation kernel. Here, G is a two dimensional kernel, and it can be separated into two one dimensional kernels as

$$G(\mathbf{p}^*, \mathbf{q}) = g(p_x^*, q_x) \cdot g\left(p_y^*, q_y\right), \tag{6}$$

where $g(a, b) = \max(0, 1 - |a - b|)$. Note that in Eq. 5, for each $\mathbf{p}^*$, $G(\mathbf{q}, \mathbf{p}^*)$ is non-zero only for four different $\mathbf{q}$s at most.

Last but not least, in order to satisfy the definition of fully convolution in Sec. 3.1, we limit the minimum value and maximum value of $\mathbf{p}_n + \Delta\mathbf{p}_n$ (denoted by $\hat{\mathbf{p}}$) in Eq. 4 as

$$\begin{aligned} \hat{p}_x &= \max\left(\hat{p}_x, 0\right), \quad & \hat{p}_x &= \min\left(\hat{p}_x, \sqrt{N} - 1\right), \\ \hat{p}_y &= \max\left(\hat{p}_y, 0\right), \quad & \hat{p}_y &= \min\left(\hat{p}_y, \sqrt{N} - 1\right). \end{aligned} \tag{7}$$

4. SiamDCN+ Network

The network architecture of our proposed twofold SiamDCN+ is depicted in Fig. 3. It is composed of a SiamFC branch and a SiamDCN branch indicated by the red block and the orange block in the figure, respectively. Both the SiamFC branch and the SiamDCN branch receive a pair of image patches cropped from the first frame (target image patch) and the current frame (search image patch) as inputs. The output of each branch is a response map which indicates the similarity score between the target patch and each candidate patch with the same size as the target in the search region. Finally, the two response maps are weighted and fused to locate the target.

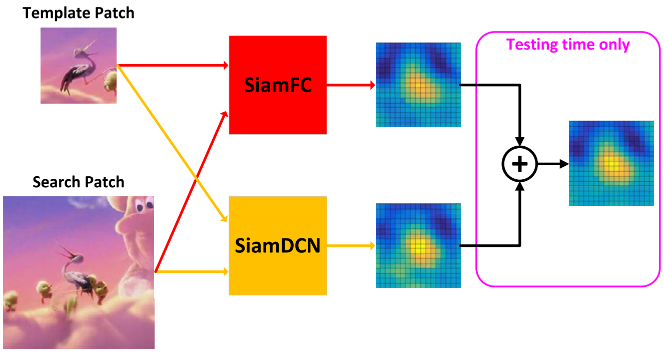


Fig. 3. Network architecture of the proposed twofold SiamDCN+. It is comprised of a SiamFC branch and a SiamDCN branch. These two branches are separately trained and combined during test.

The fundamental idea behind fusing the response maps of the above two branches is completely different from that of general model fusion method which is to fuse different models of the same algorithm. The insight of our twofold SiamDCN+ comes from an important observation that the static features learned in SiamFC with strong discrimination power and the dynamic features learned in SiamDCN with strong robustness complement each other, and therefore should be jointly considered for accurate visual tracking. Additionally, an important design choice in SiamDCN+ is to separately train the two branches to keep the heterogeneity of the two types of features.

SiamFC branch. The backbone of SiamFC branch for features extraction is the same as that in Sec. 3.1. SiamFC branch takes (x, X) as input where x and X denote the target image patch and the search image patch, respectively. Its output $h_{fc}(x, X)$, *i.e.* the response map $\mathbf{y}_{fc}$, can be written as:

$$h_{fc}(x, X) = \mathrm{corr}(f(x), f(X)), \tag{8}$$

where $\mathrm{corr}(\cdot, \cdot)$ is the cross-correlation operation and $f(\cdot)$ is the function for CNNs features extraction. In the offline training, each element of the ground-truth response map $\mathbf{y}_{gt}$ can be expressed as:

$$\mathbf{y}_{gt}(\mathbf{p}) = \begin{cases} +1 & \text{if } \|\mathbf{p} - \mathbf{c}\| \leqslant 2 \\ -1 & \text{otherwise} \end{cases} \tag{9}$$

where $\mathbf{c}$ is the center of $\mathbf{y}_{gt}$, and $\mathbf{y}_{gt}$ has been showed in Fig. 2. With abundant pairs (x_i, X_i) from training videos, SiamFC branch is optimized by minimizing the logistic loss function $L(\cdot)$ as follows:

$$\underset{\theta_{fc}}{\mathrm{argmin}} \frac{1}{N} \sum_{i=1}^{N} L\left(h_{fc}\left(x_i, X_i; \theta_{fc}\right), \mathbf{y}_{gt}\right), \tag{10}$$

where θ_{fc} denotes the CNNs parameters in $f(\cdot)$ and N is the number of training sample pairs in a mini-batch.

We consider the features learned in SiamFC branch as static because the locations of correlation calculations for all elements of the template kernel in its cross-correlation operation are always static and invariant. We find that these features are more discriminative but less robust. Specifically, in this branch, the similarity scores to most background interferences are usually small and to small deformations of the target are usually large, however, they are difficult to keep large when the target experiences large deformations.

SiamDCN branch. The network architecture of SiamDCN branch has been introduced in Sec. 3 in detail. SiamDCN branch takes (*x*, *X*) as input, which is the same as that in the SiamFC branch. Its output $h_{dcn}(x, X)$, *i.e.* the response map $\mathbf{y}_{dcn}$, can be written as:

$$h_{dcn}(x, X) = \text{def-corr}(f(x), f(X), \text{corr}(g_1(f(x)), g_2(f(X)))) \quad (11)$$

where def-corr(· , · , ·) is the deformable cross-correlation operation, *f*(·) is the function used to extract CNNs features for matching, and $g_1(\cdot)$ and $g_2(\cdot)$ are two functions used to transform the features for matching into the features for generating offsets map of the deformable cross-correlation operation. In the offline training, its ground-truth response map and loss function are the same as those of the SiamFC branch.

We consider the features learned in SiamDCN branch as dynamic because the locations of correlation calculations for all elements of the template kernel in its deformable cross-correlation operation are determined by the offsets map generated in an online self-adaptive way and change as the appearance changes of the target. We find that these features are more robust but less discriminative. Specifically, in this branch, the similarity score to the target is usually large even though the target experiences a large deformation, however, these features are not good at distinguishing multiple similar objects effectively because a disturbing object which is similar to the target can be regarded as a deformed target object and it also has a large matching score with the target template. In addition, these features are not suitable for locating the target accurately because the strong robustness often makes most candidate patches around the true target position to have large similarity scores.

During testing time, the overall output *h*(*x*, *X*), *i.e.* the response map **y**, is computed as a weighted average of the response maps from the above two branches, and this process can be expressed as:

$$h(x, X) = \lambda h_{fc}(x, X) + (1 - \lambda) h_{dcn}(x, X) \quad (12)$$

where λ is the weighting parameter to balance the importance of the two branches. Finally, the candidate patch with the highest score is identified as the target.

5. Experiments

We first provide implementation details, and then carry out ablative studies to analyse and compare the proposed SiamDCN and its baseline tracker SiamFC for both offline training process and online tracking performance. Extensive experiments are conducted to evaluate the proposed SiamDCN+ and compare its tracking performance against plenty of state-of-the-art real-time trackers on three public benchmarks: OTB2015 [1], VOT2016 [26], and VOT2017 [27]. Code will be made available to facilitate the future research.

5.1. Implementation Details

Platform. We implement SiamDCN+ (including the SiamFC branch and the SiamDCN branch) in Python with PyTorch [44] framework. Our experiments are performed on Linux with a Intel E5-2630 CPU @2.20GHz and a single TITAN X(Pascal) GPU. The running speeds of SiamDCN and SiamDCN+ are 70+ FPS and 40+ FPS, respectively.

Training Data Preparation. Both the SiamFC branch and the SiamDCN branch of SiamDCN+ are pre-trained offline from scratch on two public visual object tracking datasets, LaSOT [45] and GOT-10K [46], which contain 1400 videos and 10000 videos, respectively. In each video snippet of an object, we collect the pair of training frames within the nearest 100 frames. In the template frame, if the center and size of the target's bounding box are denoted as (*x*, *y*) and (*w*, *h*), respectively, we crop the template image patch centering on (*x*, *y*) with size *A* × *A* which is defined as

$$A^2 = \left(w + \frac{w+h}{2}\right) \times \left(h + \frac{w+h}{2}\right). \quad (13)$$

It is resized to 127 × 127 afterwards. In the same way, the search image patch is cropped on the search frame with double the size of the template image patch, and then resized to 255 × 255.

Training Setting. For both the SiamFC branch and the SiamDCN branch, we apply stochastic gradient descent (SGD) with momentum of 0.9 to train their whole network models from scratch and set the weight decay to 0.0001. The learning rate exponentially decays from 10^{-2} to 10^{-5}. The models are trained for 50 epochs with a mini-batch size of 32.

Scale Estimation. To adapt to the scale variations of target object, we search on three scales of the current search image patch with scale factors {0.9638, 1.0, 1.0375}. The current target scale is determined by a linear interpolation with a factor of 0.6 on the newly predicted scale for a smooth tracking.

Hyperparameters. We set the λ in Eq. 12 to 0.5.

5.2. Evaluation Metrics

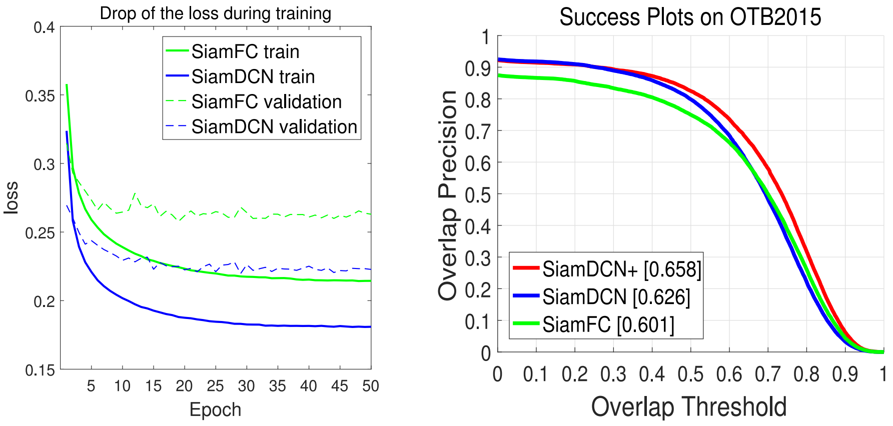
In OTB2015 experiments, following the standard benchmark protocols of OTB2015 [1], all trackers are quantitatively evaluated by five metrics: (i) Distance Precision (DP), which is the percentage of frames where the objects are located within the center errors of 0 to 20 pixels in a sequence; (ii) Precision Plot, which is simply a curve of the distance precisions with the center errors changing from 0 to 50 pixels; (iii) Overlap Precision (OP), which is the percentage of frames where the Intersection-over-Union between the estimated bounding box and the ground truth exceeding 0.5 in a sequence; (iv) Success Plot, which is simply a curve of overlap precisions with the Intersection-over-Union changing from 0 to 1; (v) Area-Under-the-Curve (AUC), which is the area under the success plot.

In VOT2016 and VOT2017 experiments, we follow the VOT challenge protocol [26] to compare tracking algorithms, where a tracker is re-initialized whenever its failure is observed. VOT2016 and VOT2017 mainly report the accuracy, robustness, and expected average overlap (EAO) score which is a result of combining accuracy and robustness, and rank trackers based on EAO score since it can give an insight to the comprehensive performance of a tracker.

5.3. Ablation Studies

We first conduct the study to compare the offline training processes of our proposed SiamDCN and its baseline tracker SiamFC [16]. Here, SiamDCN and SiamFC are both trained on LaSOT and GOT-10K, and validated on the validation set of GOT-10K, employing the same backbone. Fig. 4a shows the training and validation losses of SiamDCN and SiamFC changing with the number of epochs. It is seen that from the beginning of training to the end, the training and validation losses of SiamDCN are consistently smaller than those of SiamFC. We believe that this is because compared to the cross-correlation operation in SiamFC, the adaptive deformable cross-correlation operation in SiamDCN makes the whole network easier to train and more generalizable especially when the target experiences large deformations.

Second, we compare the tracking performance of SiamDCN and SiamFC [16] on OTB2015 benchmark, and only report their success plots and AUC scores for the sake of simplicity. Fig. 4b shows the results. It is seen from the success plots that the overlap precision of SiamDCN is higher than that of SiamFC when the over-



(a) Offline training.

(b) Online evaluation.

Fig. 4. Comparison between the proposed SiamDCN and its baseline tracker SiamFC for both offline training process (a) and online tracking performance (b) where AUC scores are reported in the legend. Best viewed in color.

Table 1
The mean overlap precisions of the proposed SiamDCN+ and other seven state-of-the-art real-time trackers on OTB2015. SiamDCN+ outperforms other trackers by large margins. The best two results are shown in bold.

Tracker	SiamDCN+	MKCFup	ECOHC	PTAV	BACF	CSRDCF	Staple	LCT
where	this paper	CVPR2018	CVPR2017	ICCV2017	ICCV2017	CVPR2017	CVPR2016	CVPR2015
mean OP	**0.823**	0.689	**0.777**	0.776	0.776	0.695	0.691	0.630

lap threshold is less than 0.65, and the opposite conclusion can be drawn when the overlap threshold is larger than 0.65. These results verify our claim that the features learned in SiamFC are more discriminative but less robust, whereas the features learned in SiamDCN are more robust but less discriminative. Taken together, SiamDCN achieves better localization accuracy than SiamFC since the AUC score of SiamDCN exceeds that of SiamFC with a large margin. Here, note that the AUC score of SiamFC on OTB2015 benchmark reported in this paper is higher than that reported in its original paper (0.601 vs. 0.582). This is mainly because the original implementation trains SiamFC on the video object detection dataset ILSVRC15 [47], however, our implementation trains it on the visual object tracking datasets, LaSOT and GOT-10K.

Last, it is seen from Fig. 4b that the overlap precision and the AUC score of our proposed SiamDCN+ exceed those of SiamFC and SiamDCN with large margins. This result verifies our important claim that the features learned in SiamFC and the features learned in SiamDCN complement each other, and fusing them can effectively improve the tracking accuracy.

5.4. Evaluation on OTB2015

Overall performance. We evaluate the proposed SiamDCN+ on OTB2015 benchmark, and divide state-of-the-art real-time trackers into two groups for a thorough comparison. The first group consists of seven trackers which are not SiamFC-based. They are ECOHC [14], PTAV [35], BACF [9], CSRDCF [48], Staple [49], LCT [50], and MKCFup [51]. The second group is formed by other seven trackers which are SiamFC-based. They are CFNet [52], DaSiamRPN [30], SiamRPN [24], SASiam [21], RASNet [20], StructSiam [22], and Siamlmp [32]. Here, when comparing with the second group trackers, all trackers are quantitatively evaluated only by the AUC metric because they all report AUC scores in their original papers and there is no way to obtain the detailed tracking results of some of them to evaluate them with other metrics.

Fig. 5 and Table 1 show the comparisons of our SiamDCN+ with the first group trackers. As shown in Fig. 5, SiamDCN+ obtains the mean DP and AUC scores of 86.0% and 65.8% on OTB2015 benchmark, respectively, leading the second best trackers with a significant gain of 1.2% and 1.1%, respectively. In addition, as shown in Table 1, SiamDCN+ obtains the mean OP of 82.3% on OTB2015 benchmark, leading the second best tracker with a significant gain of 4.6% and a significant relative gain of 5.9%.

Table 2 shows the comparison of our SiamDCN+ with the second group trackers. It is seen that SiamDCN+ achieves competitive performance compared to DaSiamRPN and SASiam, and outperforms other trackers by large margins. It is worth noting that both DaSiamRPN and SiamRPN employ significantly more training datas from Youtube and COCO than other trackers, and the semantic branch of SASiam must be trained with large amounts of classification data from ImageNet [53].

Attribute-based evaluation. The videos in OTB2015 benchmark are annotated with 11 attributes to describe the different challenges in the tracking problem. These attributes are useful for analyzing the performance of trackers in different aspects. Following [12], we compare our SiamDCN+ with seven state-of-the-art real-time trackers, including DaSiamRPN, PTAV, ECOHC, SiamFC, BACF, MKCFup, and CFNet, on eight main challenging attributes of OTB2015 benchmark, then explain in more detail the advantages and disadvantages of SiamDCN+ over the other trackers. Fig. 6 shows the results. It is seen that: (1) In terms of deformation challenge, SiamDCN+ outperforms other trackers (including DaSiamRPN) with large margins. This confirms that our contribution in introducing the deformable cross-correlation to SiamFC to improve its robustness to the large deformations of targets is effective; (2) In terms of in plane rotation, we can get the same conclusion as above. Although the localization accuracy of SiamDCN+ is slightly worse than that of DaSiamRPN in terms of out of plane rotation, we believe that the main reason for this result lies in the main difference between in plane rotation and out of plane rotation, that is in plane rotation preserves the semantics of target components, whereas out of plane rotation does not, resulting in the deformable cross-correlation operation in SiamDCN+ cannot work well under the out of plane rotation challenge; (3) SiamDCN+ performs worse than DaSiamRPN in background clutter challenge. We believe that this is because the distractor-aware module in DaSiamRPN is effec-

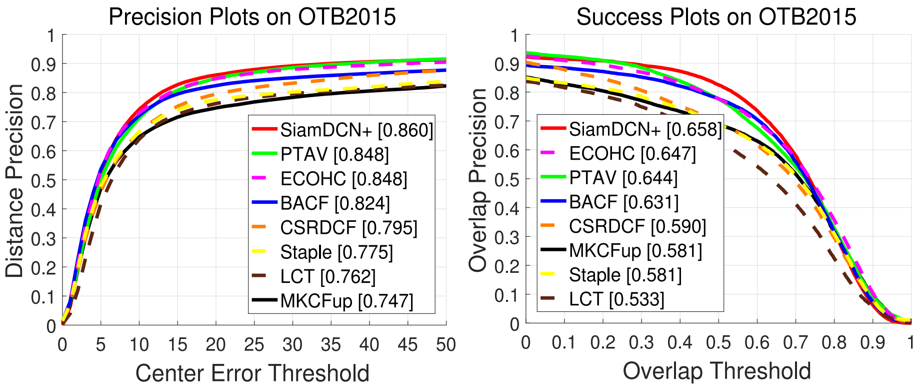


Fig. 5. The mean precision and success plots of the proposed SiamDCN+ and other seven state-of-the-art real-time trackers on OTB2015. The mean distance precisions and AUC scores are reported in the legends. SiamDCN+ outperforms other trackers by large margins.

Table 2
The mean AUC scores of the proposed SiamDCN+ and other seven state-of-the-art real-time trackers on OTB-2015. SiamDCN+ achieves competitive performance compared to DaSiamRPN and SASiam, and outperforms other trackers by large margins. The best three results are shown in bold.

Tracker	SiamDCN+	CFNet	DaSiamRPN	SiamRPN	SASiam	RASNet	StructSiam	SiamImp
where	this paper	CVPR2017	ECCV2018	CVPR2018	CVPR2018	CVPR2018	ECCV2018	ECCV2018
AUC	**0.658**	0.592	**0.658**	0.637	**0.657**	0.642	0.621	0.629

Table 3
The accuracy, robustness, and EAO scores of the proposed SiamDCN+ and SiamRPN along with the top-10 real-time trackers on VOT2016 challenge and the top-10 trackers on VOT2017 real-time challenge. The best two results in each line are shown in bold.

Tracker	SiamDCN+	SiamRPN	Staple	STAPLE+	SSKCF	DPT	SiamFC-R	SiamFC-A	CCCT	GCF	NSAMF	ColorKCF
A ↑	0.546	**0.56**	0.544	**0.557**	0.547	0.492	0.549	0.532	0.442	0.520	0.502	0.503
R ↓	**0.338**	**0.26**	0.378	0.368	0.373	0.489	0.382	0.461	0.461	0.485	0.438	0.443
EAO ↑	**0.298**	**0.344**	0.295	0.286	0.277	0.236	0.277	0.235	0.223	0.218	0.227	0.226

(a) Comparison with SiamRPN along with the top-10 real-time trackers on VOT2016 challenge.

Tracker	SiamDCN+	SiamRPN	CSRDCF+	SiamFC	ECOHC	Staple	KFebT	ASMS	SSKCF	CSRDCFf	UCT	MOSSEca
A ↑	0.496	0.49	0.459	0.502	0.494	**0.530**	0.451	0.489	**0.530**	0.475	0.490	0.400
R ↓	0.517	**0.46**	**0.398**	0.604	0.571	0.688	0.684	0.627	0.656	0.646	0.777	0.810
EAO ↑	**0.229**	**0.243**	0.212	0.182	0.177	0.170	0.169	0.168	0.164	0.158	0.145	0.139

(b) Comparison with SiamRPN along with the top-10 trackers on VOT2017 real-time challenge.

tive in improving its robustness to background clutters. It is worth mentioning that both PTAV and ECOHC perform well on this challenge. This verifies that online discriminative training is effective in improving the robustness of trackers to background clutters as stated in [54]; (4) SiamDCN+ achieves leading performance (top-2) on most attributes, and always outperforms SiamFC with large margins.

5.5. Evaluation on VOT2016

We evaluate the proposed SiamDCN+ on VOT2016 benchmark, then compare its accuracy, robustness, and EAO score with SiamRPN [24] along with the top-10 real-time trackers on VOT2016 challenge [26]. These trackers are Staple, STAPLE+, SSKCF, DPT, SiamFC-R, SiamFC-A, CCCT, GCF, NSAMF, and ColorKCF. The results are shown in Table 3a. SiamDCN+ achieves competitive robustness and EAO scores compared to Staple and outperforms other state-of-the-art real-time trackers except for SiamRPN by large margins. It is worth noting that SiamDCN+ outperforms its baseline tracker SiamFC-A on all accuracy, robustness, and EAO scores by large margins.

5.6. Evaluation on VOT2017

We evaluate the proposed SiamDCN+ on VOT2017 benchmark, then compare its accuracy, robustness, and EAO scores with SiamRPN along with the top-10 trackers on VOT2017 real-time sub-challenge [27] where if a tracker fails to process the result in real-time speed, the evaluator will use the bounding box of the last frame as the result of current frame. These trackers are CSRDCF++, SiamFC, ECOHC, Staple, KFebT, ASMS, SSKCF, CSRDCFf, UCT, and MOSSEca. The results are shown in Table 3b. In terms of EAO score that shows the comprehensive performance of a tracker, SiamDCN+ outperforms other state-of-the-art trackers except for SiamRPN by large margins.

According to the above experiments, it is not difficult to find that the localization performance of our SiamDCN+ is superior to SiamRPN on OTB2015 benchmark but inferior to it on VOT benchmarks. We explain these experimental results as follows. The basic locator, which is only used to determine the position of the target but not responsible for estimating its scale, of SiamRPN is essentially the same as that of SiamFC. SiamRPN mainly focuses on improving the accuracy of the target's scale estimation of SiamFC. Specifically, different from SiamFC which employs the scale pyramid based method to estimate the target's scale, SiamRPN proposes to achieve this more accurately by training a bounding box regression branch. This is the main reason of that SiamRPN can achieve high performance on VOT benchmarks where the dramatic change in target's scale is one of the main challenges. Similar to CFNet [52], SASiam [21], RASNet [20], StructSiam [22], and SiamImp [32], our SiamDCN+ mainly focuses on improving the robustness and discrimination power of the locator of SiamFC, therefore, it is more reasonable to compare SiamDCN+ with them, and the localization performance of SiamDCN+ outperforms theirs. In

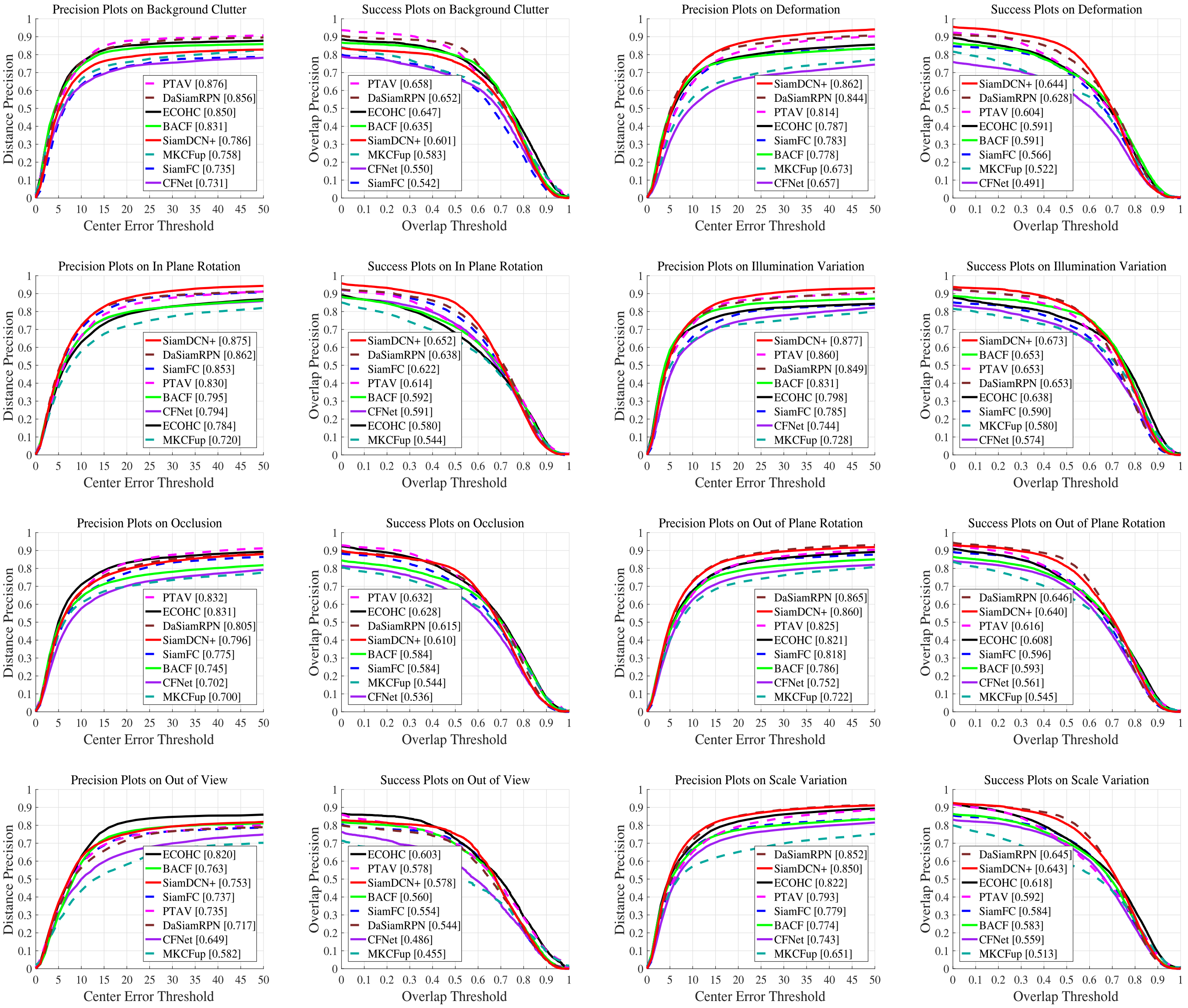


Fig. 6. The mean precision and success plots of the proposed SiamDCN+ and other seven state-of-the-art real-time trackers on eight main attributes of OTB2015. The mean distance precisions and AUC scores are reported in the legends. SiamDCN+ performs favorably against other trackers.

conclusion, SiamDCN+ and SiamRPN have different goals for improvements, and they can easily take advantage of each other.

Last but not least, we argue that the accuracy and robustness scores in VOT challenges may not accurately represent the accuracy and robustness of a tracker. According to the above, it is obvious that SiamRPN can effectively improve the accuracy of SiamFC. However, the accuracy score of SiamRPN [1] is lower than that of SiamFC on the VOT2017 challenge. In addition, SiamFC employs CNNs features which is more robust than the hand-crafted features used in ECOHC. However, the robustness score of SiamFC is lower than that of ECOHC on the VOT2017 challenge. Therefore, the EAO score is more convincing.

5.7. Qualitative Results

Successful Cases. Fig. 7 illustrates the qualitative results of the proposed SiamDCN+ and seven representative state-of-the-art real-time trackers on four hard sequences, where most trackers are hard to track successfully, of OTB2015 benchmark. Owing to the strong robustness of SiamDCN branch and the strong discrimination power of SiamFC branch, SiamDCN+ is able to track the targets accurately and robustly in these hard cases. It is worth noting that in MotorRolling and Skiing sequences where the targets experience large deformations, SiamFC fails, whereas SiamDCN+ can track the targets successfully.

Failure Cases. We observe the following three main types of failures in the proposed SiamDCN+, as shown in Fig. 8. (1) In Girl2 and Suv sequences, when the target is heavily occluded, the tracking box drift and will never stick on the target again. This is because there are no occlusion prediction and target re-detection module in SiamDCN+. (2) In Coupon sequence, there is an object that is very similar to the initial appearance of the target object around the target, which confuses SiamDCN+. This is because, like most SiamFC-based trackers, there is no online update module in SiamDCN+. (3) In Jump and Trans sequences, the aspect ratio of the target changes dramatically, which makes the tracking box inaccurate. This is because the scale estimation of target is based on the assumption that the aspect ratio of the target is constant in SiamDCN+. Interestingly, although Jump and Trans sequences de-

[1] This result is obatined from the Table 5 of [55] and Table 4 of [56]

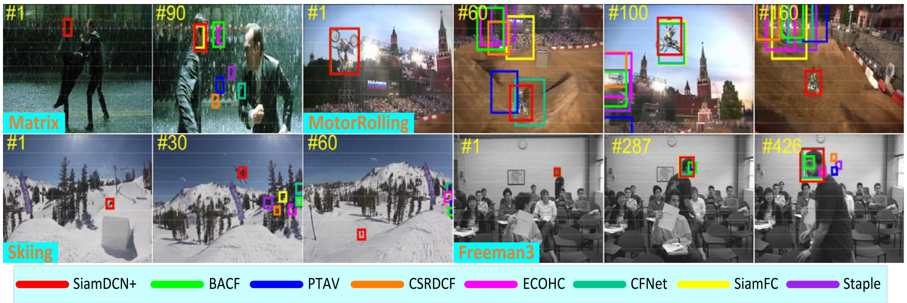


Fig. 7. Qualitative results for the proposed SiamDCN+, compared with plenty of state-of-the-art real-time trackers on four hard sequences, Matrix, MotorRolling, Skiing, and Freeman3. Our SiamDCN+ can track the targets accurately and robustly in these hard cases where most trackers fail. Best viewed on a high-resolution screen.

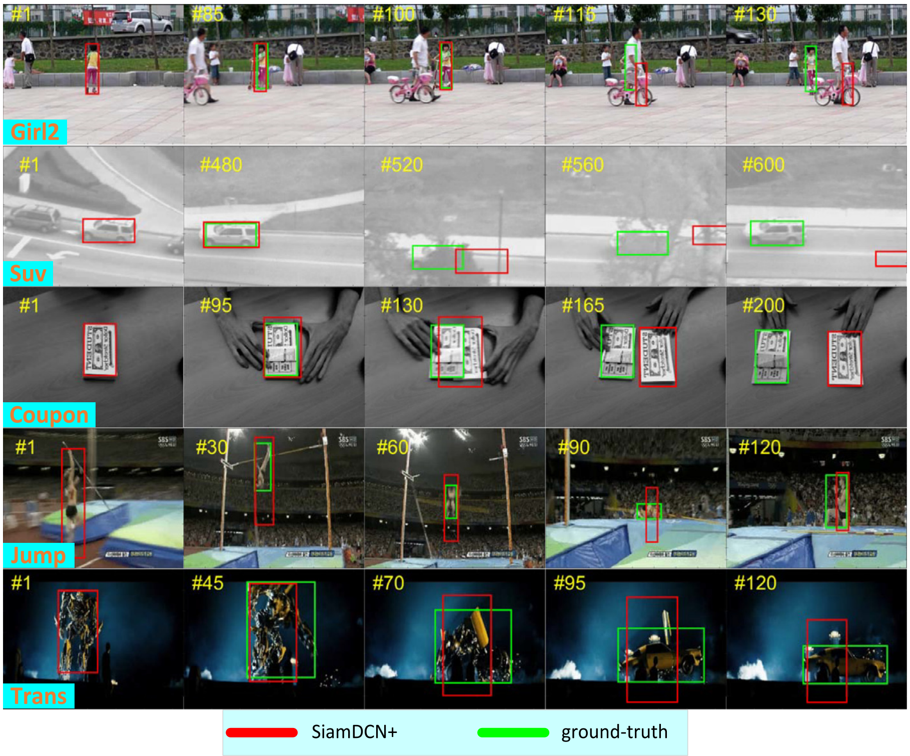


Fig. 8. Visualization of failure cases. In Girl2 and Suv sequences, the targets are heavily occluded. In Coupon sequence, there is an object that is very similar to the initial appearance of the target object around the target. In Jump and Trans sequences, the aspect ratios of the targets change dramatically. These three reasons cause the tracking failure.

viate from the above assumption seriously and SiamDCN+ cannot track the targets accurately, SiamDCN+ dose not lose the targets completely. This confirms the robustness of SiamDCN+ to some extent.

6. Conclusion and Future work

In this paper, a novel tracker SiamDCN is proposed by introducing the deformable cross-correlation operation into SiamFC in an online self-adaptive way. Compared to SiamFC, SiamDCN can deal with the large deformations of targets robustly through dynamically and adaptively adjusting the location of correlation calculation for each element of the template kernel in the cross-correlation operation. Further, an accurate and real-time tracker SiamDCN+ is proposed after observing that the dynamic features learned in SiamDCN are robust, while the static features learned in SiamFC are discriminative, and they complement each other. As a result, the proposed SiamDCN achieves superior localization accuracy than its baseline tracker SiamFC and the proposed SiamDCN+ achieves competitive performance compared to the state-of-the-art real-time trackers on three public benchmarks, OTB2015, VOT2016, and VOT2017, while running beyond 40 FPS.

A possible future work to extend our SiamDCN+ is to combine the recent novel idea of multi-target tracking with deep reinforcement learning in C-DRL [57] to achieve efficient and robust multi-target tracking.

Declaration of Competing Interest

The authors declared that they have no conflicts of interest to this work.

CRediT authorship contribution statement

Linyu Zheng: Conceptualization, Methodology, Software, Validation, Writing - original draft. **Yingying Chen:** Data curation, Visualization, Writing - review & editing. **Ming Tang:** Writing - review & editing. **Jinqiao Wang:** Writing - review & editing. **Hanqing Lu:** Writing - review & editing.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grants 61976210, 61876086, 61806200, and 61772527. This work was also supported by the fund of the key laboratory of rich-media knowledge organization and service of digital publishing content (ZD2019-10/01), and the Research and Development Projects in the Key Areas of Guangdong Province (No.2019B010153001).

References

[1] Y. Wu, J. Lim, M.-H. Yang, Object tracking benchmark, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (9) (2015) 1834–1848.

[2] L. Zheng, M. Tang, Y. Chen, J. Wang, H. Lu, Fast-deepkcf without boundary effect, in: The IEEE International Conference on Computer Vision (ICCV), 2019.

[3] D. Held, S. Thrun, S. Savarese, Learning to track at 100 fps with deep regression networks, in: European Conference on Computer Vision, Springer, 2016, pp. 749–765.

[4] M. Xin, J. Zheng, B. Li, G. Niu, M. Zhang, Real-time object tracking via self--adaptive appearance modeling, Neurocomputing 349 (2019) 21–30.

[5] J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters, IEEE transactions on pattern analysis and machine intelligence 37 (3) (2014) 583–596.

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

[7] S. Zhai, P. Shao, X. Liang, X. Wang, Fast rgb-t tracking via cross-modal correlation filters, Neurocomputing 334 (2019) 172–181.

[8] P. Zhang, Q. Guo, W. Feng, Fast and object-adaptive spatial regularization for correlation filters based tracking, Neurocomputing 337 (2019) 129–143.

[9] H. Kiani Galoogahi, A. Fagg, S. Lucey, Learning background-aware correlation filters for visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1135–1143.

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

[11] L. Zheng, M. Tang, J. Wang, Learning robust gaussian process regression for visual tracking., in: IJCAI, 2018, pp. 1219–1225.

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

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

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

[15] R. Tao, E. Gavves, A.W. Smeulders, Siamese instance search for tracking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 1420–1429.

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

[17] G. Pan, G. Chen, W. Kang, J. Hou, Correlation filter tracker with siamese: A robust and real-time object tracking framework, Neurocomputing 358 (2019) 33–43.

[18] H. Zhang, W. Ni, W. Yan, J. Wu, H. Bian, D. Xiang, Visual tracking using siamese convolutional neural network with region proposal and domain specific updating, Neurocomputing 275 (2018) 2645–2655.

[19] C. Jiang, J. Xiao, Y. Xie, T. Tillo, K. Huang, Siamese network ensemble for visual tracking, Neurocomputing 275 (2018) 2892–2903.

[20] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, S. Maybank, Learning attentions: residual attentional siamese network for high performance online visual tracking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4854–4863.

[21] A. He, C. Luo, X. Tian, W. Zeng, A twofold siamese network for real-time object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4834–4843.

[22] Y. Zhang, L. Wang, J. Qi, D. Wang, M. Feng, H. Lu, Structured siamese network for real-time visual tracking, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 351–366.

[23] 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) (2017) 1137–1149, doi:10.1109/TPAMI.2016.2577031.

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

[25] H. Fan, H. Ling, Siamese cascaded region proposal networks for real-time visual tracking, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[26] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Čehovin Zajc, T. Vojir, G. Häger, A. Lukežič, G. Fernandez, The visual object tracking vot2016 challenge results, 2016, (Springer).

[27] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Cehovin Zajc, T. Vojir, G. Hager, A. Lukezic, A. Eldesokey, et al., The visual object tracking vot2017 challenge results, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1949–1972.

[28] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pfugfelder, L.C. Zajc, T. Vojir, G. Bhat, A. Lukezic, A. Eldesokey, G. Fernandez, et al., The visual object tracking vot2018 challenge results, (2018).

[29] J. Gao, T. Zhang, C. Xu, Graph convolutional tracking, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

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

[31] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 764–773.

[32] X. Dong, J. Shen, Triplet loss in siamese network for object tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 459–474.

[33] H. Nam, M. Baek, B. Han, Modeling and propagating cnns in a tree structure for visual tracking, CoRR abs/1608.07242 (2016).

[34] B. Han, J. Sim, H. Adam, Branchout: Regularization for online ensemble tracking with convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3356–3365.

[35] H. Fan, H. Ling, Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5486–5494.

[36] T. Dekel, S. Oron, M. Rubinstein, S. Avidan, W.T. Freeman, Best-buddies similarity for robust template matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2021–2029.

[37] S. Oron, T. Dekel, T. Xue, W.T. Freeman, S. Avidan, Best-buddies similarityârobust template matching using mutual nearest neighbors, IEEE transactions on pattern analysis and machine intelligence 40 (8) (2017) 1799–1813.

[38] S. Oron, D. Suhanov, S. Avidan, Best-buddies tracking, arXiv preprint arXiv:1611.00148 (2016).

[39] F. Liu, C. Gong, X. Huang, T. Zhou, J. Yang, D. Tao, Robust visual tracking revisited: From correlation filter to template matching, IEEE Transactions on Image Processing 27 (6) (2018) 2777–2790.

[40] X. Zhou, Q. Huo, Y. Shang, M. Xu, H. Ding, Learning spatially regularized similarity for robust visual tracking, Image and Vision Computing 60 (2017) 134–141.

[41] C. Wang, L. Zhang, L. Xie, J. Yuan, Kernel cross-correlator, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[42] J. Hu, J. Lu, Y.-P. Tan, Deep metric learning for visual tracking, IEEE Transactions on Circuits and Systems for Video Technology 26 (11) (2015) 2056–2068.

[43] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[44] A. Paszke, S. Gross, S. Chintala, G. Chanan, Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration, (2017).

[45] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, H. Ling, Lasot: A high-quality benchmark for large-scale single object tracking, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5374–5383.

[46] L. Huang, X. Zhao, K. Huang, Got-10k: A large high-diversity benchmark for generic object tracking in the wild, CoRR abs/1810.11981 (2018).

[47] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., Imagenet large scale visual recognition challenge, International journal of computer vision 115 (3) (2015) 211–252.

[48] A. Lukezic, T. Vojir, L. Ëhovin Zajc, J. Matas, M. Kristan, Discriminative correlation filter with channel and spatial reliability, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6309–6318.

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

[50] C. Ma, X. Yang, C. Zhang, M.-H. Yang, Long-term correlation tracking, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 5388–5396.

[51] M. Tang, B. Yu, F. Zhang, J. Wang, High-speed tracking with multi-kernel correlation filters, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4874–4883.

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

[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee, 2009, pp. 248–255.

[54] L. Zheng, M. Tang, H. Lu, et al., Learning features with differentiable closed–form solver for tracking, arXiv preprint arXiv:1906.10414 (2019).

[55] Z. Zhang, H. Peng, Deeper and wider siamese networks for real-time visual tracking, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[56] G. Wang, C. Luo, Z. Xiong, W. Zeng, Spm-tracker: Series-parallel matching for real-time visual object tracking, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[57] L. Ren, J. Lu, Z. Wang, Q. Tian, J. Zhou, Collaborative deep reinforcement learning for multi-object tracking, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 586–602.

Linyu Zheng received his B.E. degree in 2016 from University of Electronic Science and Technology, China. He is currently working toward the Ph.D. degree in pattern recognition and intelligence systems from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences. His current research interests include pattern recognition and machine learning, image and video processing, and visual tracking.

Yingying Chen received her B.S. degree from Communication University of China, Beijing, China, in 2013. She is currently working toward the Ph.D. degree in pattern recognition and intelligence systems from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences. Her current research interests include pattern recognition and machine learning, image and video processing, and intelligent video surveillance.

Ming Tang received the B.S. degree in computer science and engineering and M.S. degree in artificial intelligence from Zhejiang University, Hangzhou, China, in 1984 and 1987, respectively, and the Ph.D. degree in pattern recognition and intelligent system from the Chinese Academy of Sciences, Beijing, China, in 2002. He is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His current research interests include computer vision and machine learning.

Jinqiao Wang received the B.E. degree in 2001 from Hebei University of Technology, China, and the M.S. degree in 2004 from Tianjin University, China. He received the Ph.D. degree in pattern recognition and intelligence systems from the National Laboratory of Pattern Recognition, Chinese Academy of Sciences, in 2008. He is currently a Professor with Chinese Academy of Sciences. His research interests include pattern recognition and machine learning, image and video processing, mobile multimedia, and intelligent video surveillance.

Hanqing Lu received his B.E. degree in 1982 and his M.E. degree in 1985 from Harbin Institute of Technology, and Ph.D. degree in 1992 from Huazhong University of Sciences and Technology. Currently, he is a deputy director of National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include image and video analysis, medical image processing, object recognition, etc.