IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 16, 2019, accepted October 11, 2019, date of publication October 15, 2019, date of current version November 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2947472

Auto-Selecting Receptive Field Network for Visual Tracking

JUNFEI ZHUANG[1], YUAN DONG[1], HONGLIANG BAI[2], PEILIANG ZUO[1], AND JIANMING CHENG[1]

[1]Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Beijing Faceall Technology Company Ltd., Beijing 10081, China

Corresponding author: Junfei Zhuang (zjf1@bupt.edu.cn)

This work was supported by the Chinese National Natural Science Foundation under Grant 61532018.

ABSTRACT Recently, Convolutional Neural Networks (CNNs) have shown tremendous potential in the visual tracking community. It is well-known that the receptive field is a critical factor for CNN affecting performance. However, standard CNNs based tracking methods design the receptive fields of artificial neurons in each layer that have the same size. We identify the main bottleneck of affecting the tracking accuracy as regular receptive fields. To settle the problem, we propose an Auto-Selecting Receptive Field Network (ASRF) to select receptive field information and effective clues dynamically. In particular, a Selective Receptive Field Block (SRFB) is designed to adaptively adjust receptive field size for each neuron according to multiple scales of input information. Additionally, we develop a Multi-Scale Receptive Field module (MSRF) that marks a further step in selecting effective clues from different scale receptive fields. The proposed ASRF method performs favorably against state-of-the-art trackers on five benchmarks, including OTB-2013, OTB-2015, UAV-123, VOT-2015, and VOT-2017 while running beyond real-time tracking speed.

INDEX TERMS Visual tracking, deep learning, Siamese network, receptive field.

I. INTRODUCTION

Visual object tracking remains a fundamental research topic in computer vision with many applications, including human-computer interaction, automated surveillance, autonomous driving, and vehicle navigation [51]. The core task for visual tracking is to estimate the trajectory of an arbitrary target in a video. However, visual tracking still remains challenging due to some practical factors like background clutter, scale variation, occlusions, fast motion, deformation, and other varieties.

Inspired by the great success of CNNs in various vision tasks, researchers have made substantial efforts to utilize CNNs power to improve tracking accuracy. Some trackers [8], [10], [13], [35], [40], [46] have integrated the expressive power of CNN features into conventional correlation filters tracking approaches. Despite high performance, these trackers cannot train a deep architecture from end to end leading to insufficient data-driven utilization and low efficiency. Some trackers [11], [34], [45] follow tracking-by-detection framework. They directly employ CNNs as classifiers and take full advantage of end-to-end training. However, these trackers suffer from expensive computations due to a high volume of CNN features and the online model update application.

Based on the above considerations, the Siamese architecture is designed to balance accuracy and speed. [2] proposes a fully off-line convolutional network (SiamFC) without a model update to improve the speed of the tracking process. In spite of the promising result, a considerable gap between the state-of-the-art performance still exists. The reason is that SiamFC ignores the importance of receptive field properties in designing CNNs. Object targets always have different heights, widths, and aspect ratios in visual tracking tasks. It reveals a fixed local receptive field is not suitable for locating arbitrary objects. Besides, the fix local receptive field cannot collect different scale spatial information in the same processing stage.

To address the above issue, we propose an Auto-Selecting Receptive Field Network (ASRF) based on a Siamese structure that contains two branches. One is the template branch, and the other is the exemplar branch. Meanwhile, the following two modules are included in the proposed ASRF to adaptively select and fully explore the different

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

different scale receptive field clues. (1) Selective Receptive Field Block (SRFB) adaptively combines multiple scales of input information that have different receptive field sizes. Firstly, SRFB uses different size kernels to generate various clues that correspond to different receptive field information. Next, ASRF aggregates the information from different clues and calculates the selection weights for each clue. Finally, ASRF combines all clues with their corresponding selection weights as the final output features. SRFB is computationally lightweight and imposes only a slight increase in parameter and computational cost. (2) Multi-Scale Receptive Field (MSRF) module manually designs four scale kernels that are employed on template feature maps and exemplar feature maps. Then we obtain four scale sub-feature maps on the template branch and the exemplar branch, respectively. Finally, four score maps are collected by computing the cross-correlation of two branch sub-feature maps. During the training phase, these four score maps are separately supervised. During the tracking phase, sum the four score maps up to calculate the final score map, which is applied to locate the target. The MSRF further integrates multi-scale information.

To evaluate our proposed ASRF, we carry out extensive experiments on UAV-123, VOT-2015, VOT-2017, OTB-2013, and OTB-2015 benchmarks. On the popular benchmarks OTB-2013, OTB-2015 and UAV-123, the proposed tracker achieves the fairly good performance in the area under curve (AUC). On the VOT-2015, our tracker ranks second place with an expected average overlap (EAO) score of 0.342 while running faster 68 times than the best tracker MDNet [34]. On the real-time VOT-2017 benchmark, ASRF achieves first place with an EAO of 0.231.

To summarize, the main contributions of this work are three-fold.

- We propose a Selective Receptive Field Block (SRFB) to automatically calculate selection weights and combine multiple receptive field information using selection weights. SRFB is a computationally lightweight architecture.
- We propose a Multi-Scale Receptive Field (MSRF) module which further integrates multi-scale receptive field information by separate supervision.
- We perform the proposed ASRF on multiple benchmarks and demonstrate outstanding performance beyond real-time tracking speed.

The rest of the paper contains four sections. The most related work is discussed in Section II. Section III clarifies our main contribution including SRFB which adaptively adjusts receptive field information and MSRF that further selects effective clues from different scale receptive fields. We also present the baseline tracker of the proposed algorithm in this section. In Section IV, we provide experimental results on several benchmarks. Finally, we perform the summarized conclusion of this paper in Section V.

II. RELATED WORK

In recent decades, a lot of research has focused on the visual tracking field. We discuss the related work in the following part.

A. DEEP LEARNING IN VISUAL TRACKING

CNNs showed great success in various computer vision tasks [2], [24], [27]. In the visual tracking field, some trackers [7], [8], [12], [31] introduced deep CNN features into the traditional Correlation Filter (CF) tracking model. Despite excellent performances, these trackers cannot train an end to end model, which implies they can hardly use the power of data-driven. Besides, MDNet [34] proposed a light architecture with the multi-domain branch. The model was used to learn generic features during off-line training and special features during online tracking. Real-time MDNet [21] improved tracking efficiency in [34] using the ROIAlign technique. SANet [11] introduced Recurrent Neural Networks (RNN) to learn multi-level directional features, leading to more powerful features for object tracking. VITAL [38] used adversarial learning to overcome the class imbalance problem in visual tracking. DRL-IS [36] proposed an iterative shift method with deep reinforcement learning. The main idea of this method is predicting the iterative shifts of the object bounding boxes. ACT [4] built the Actor-Critic framework that aims to infer the optimal choice in a continuous action space. ATOM [3] proposed a deep model, consisting of dedicated target estimation and classification components, to guarantee high discriminative power and predict more accurate estimated bounding boxes. Although these trackers had achieved excellent results, they perform visual tracking at low efficiency due to the on-line updating strategy.

B. SIAMESE NETWORK TRACKING

Siamese network based trackers [2], [17], [25], [44], [48], [53] contain two branches. One is an exemplar branch for selecting target patches. The other is a template branch for providing template patch. The goal of Siamese trackers is to predict the trajectory of the target object in videos. GOTURN [17] learned to regress the target bounding box using exemplar and template of consecutive frames. SINT [44] formulated visual tracking as a verification problem. SINT trained a deep CNN model to learn a matching metric for template and exemplar. Despite high performance, the SINT perform tracking with only 2 fps. SiamFC [2] put forward a novel fully-convolutional Siamese network that measures the feature similarity between the template and candidates. The SiamFC gained competitive accuracy and satisfactory efficiency. RASNet [48] introduced multiple attention mechanisms into [2] to learn more powerful deep features for visual tracking. SiamRPN [26] combined Siamese network with Region proposal Network (RPN), providing more accurate bounding boxes for target location. UDT [47] proposed a novel Siamese correlation filter network, which is trained using raw videos without labels. In spite of great

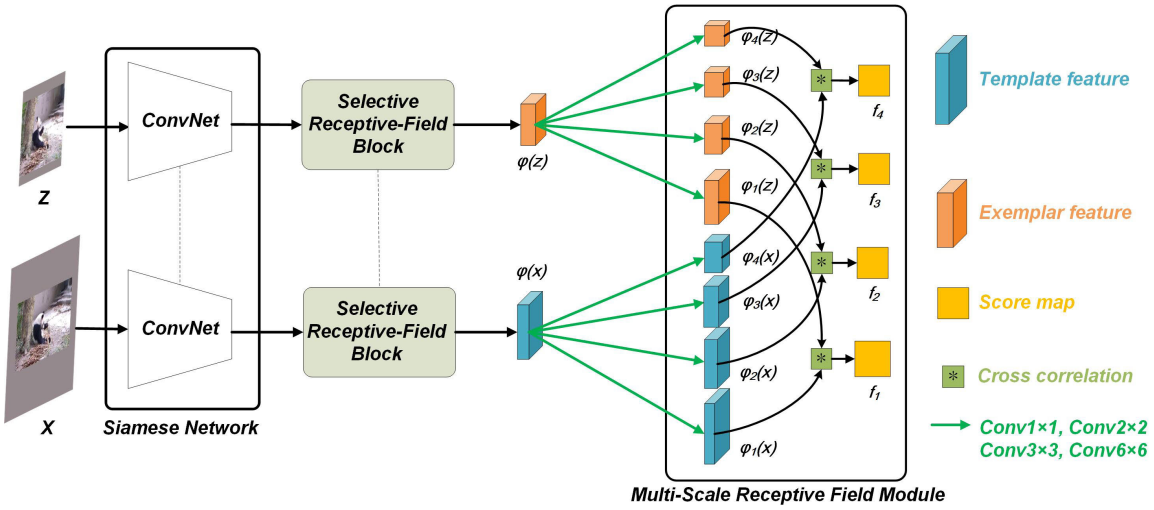


FIGURE 1. A pipeline of the proposed tracking method. ASRF is composed of a *Siamese network* for feature extraction, the *Selective Receptive Field Block* for dynamic selecting receptive field information, and the *Multi-Scale Receptive Field* module for effective aggregating clues from different scale receptive fields. The ASRF contains two branches: a template branch x and an exemplar branch z. We extract feature maps from x and z using Siamese Network. Then these feature maps are flowed into SRFB to distribute selection weights for different scale receptive field information automatically. Afterward, we combine different scale receptive field information with corresponding selection weights to obtain feature maps φ_x and φ_z. The sub-feature maps (e.g., $\varphi_1(x)$, $\varphi_1(z)$) are generated by using different scale kernels to φ_x and φ_z. Finally, the sub-score maps (e.g., $f1$) can be calculated by combining the corresponding sub-feature maps applying a cross-correlation layer. The figure is best viewed in color.

achievements, these trackers ignore the importance of receptive field properties of cortical neurons in designing CNNs, leading to a gap with a more robust visual tracking method.

C. MULTI-SCALE RECEPTIVE FIELD IN CNNS

We aim to improve accuracy and speed simultaneously. Thus, our proposed technique must be low-computational without incurring too much computational burden. Therefore, changing receptive field (RF), instead of applying very deep backbones, is our best choice to enhance tracking accuracy with a lightweight model-based feature representation. However, all the previous studies about RFs in CNN remain narrow in focus dealing only with the detection and segmentation, the RF study in visual tracking tasks remains relatively few.

Inception family [41]–[43] adopted multiple branches with different kernel sizes to aggregate multi-scale information for each convolutional layer. ASPP [5] proposed a novel atrous convolution. Atrous convolution adjusts the filter's field-of-view and controls the resolution of features. Deformable CNN [6] designed deformable convolution to augment the spatial information and learn the offsets from target detection tasks, without additional supervision. RFBNet [28] learned the relationship between the size and eccentricity of RFs, and proposed novel RF Block (RFB) module to enhance the feature discriminability and robustness.

In visual tracking community, HCF [31] investigated the effect of features from different deep layers and used these features to improve tracking accuracy and robustness. HDT [35] proposed a tracking framework which takes advantage of features from different CNN layers and uses an adaptive Hedge method to hedge several CNN trackers into a stronger one. DSiam [15] presented elementwise multi-layer fusion to integrate the network outputs using multi-level deep features adaptively. StructSiam [52] proposed a local structure learning method, which simultaneously considers the local patterns of the target and their structural relationships for more accurate target tracking. SA-Siam [16] built a tracking method which is composed of a semantic branch and an appearance branch. Each branch was separately trained to keep the heterogeneity of the two types of features.

III. THE PROPOSED ASRF

In this section, we detail the Auto-Selecting Receptive Field Network (ASRF), as shown in Figure 1. In contrast to the basic framework (SiamFC [2]), we have designed two novel modules, Selective Receptive Field Block (SRFB) and Multi-Scale Receptive Field Module (MSRF), to aggregate multi-scale receptive field information automatically. In the rest of this section, we will show the baseline tracker in Section III-A. Then, SRFB and MSRF will be presented in Section III-B and III-C.

A. BASELINE TRACKER-SIAMFC

In consideration of the balance between speed and accuracy, we choose the SiamFC as the basic block of our algorithm. SiamFC employs a Siamese structure to learn the general matching function. The detailed network architecture can be referred to [2]. Figure. 1 shows the SiamFC network in the left part. Inputs of the Siamese network are an 127×127 exemplar image z within a larger 255×255 template image x. SiamFC applies a transformation φ to both z and x and aims at computing the similarity of their representations with

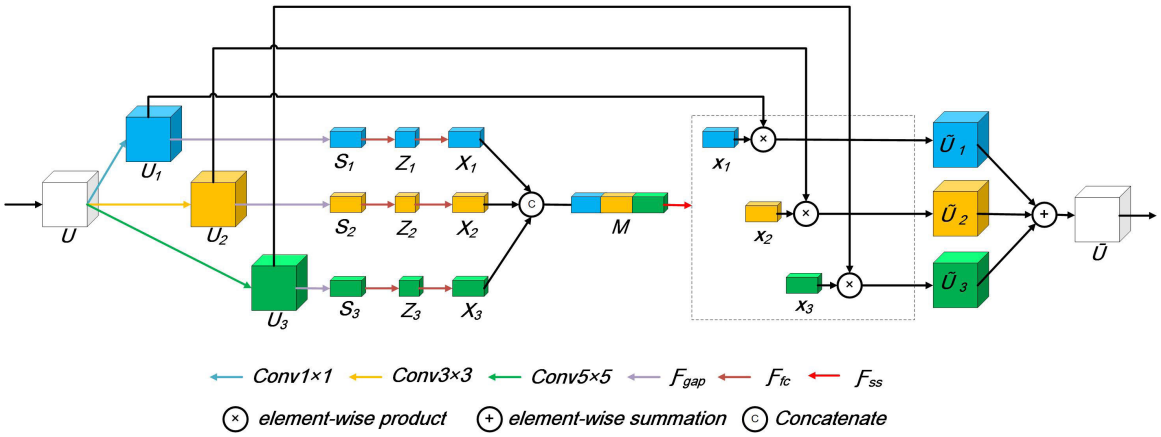


FIGURE 2. Illustrations of the proposed SRFB. The figure is best viewed in color.

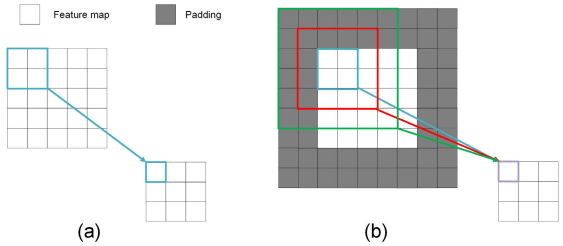


FIGURE 3. Comparison between traditional convolution and our Selective Receptive Field convolution.

a fully-convolutional operation

$$f(z, x) = \varphi(z) * \varphi(x) + b \tag{1}$$

where φ is an identical transformation generated by the two branches; $b \in \mathbb{R}$ denotes the bias for each location; More details about the Siamese network can be found in [2].

B. SELECTIVE RECEPTIVE FIELD BLOCK

To enable the last layer neurons to adjust their receptive field sizes adaptively among multiple kernels, we propose a Selective Receptive Field Block. As shown in Figure. 3, sub-figure (a) represents traditional convolution, which has a fixed receptive field. Sub-figure (b) illustrates our Selective Receptive Field convolution, which combines information from different receptive fields.

The flowchart of SRFB is illustrated in Figure. 2. Our SRFB employs gates to control the weights of different scales information flows from multiple branches carrying into neurons in the next layer. The specific steps are described as follows.

For any given feature map $U \in \mathbb{R}^{\mathbb{H}\times\mathbb{W}\times\mathbb{C}}$. We conduct three transformations using different scale kernels ($Conv1\times1$, $Conv3\times3$, and $Conv5\times5$), respectively. Batch Normalization [20] and ReLU [33] follows these kernels. We replace $Conv5\times5$ with the dilated convolution $Conv3\times3$ and dilation size 2. After the above operation, we obtain three feature maps U_1, U_2, and U_3.

$$\begin{aligned} U_1 &= Conv1\times1(U) \\ U_2 &= Conv3\times3(U) \\ U_3 &= Conv5\times5(U) \end{aligned} \tag{2}$$

Then we use global average pooling F_{gap} to calculate channel-wise statistics as $S_i \in \mathbb{R}^C$. Specifically, the S_i element is calculated by shrinking U_i through spatial dimensions $H \times W$:

$$S_i = F_{gap}(U_i) = \frac{1}{H \times W} \sum_{k=1}^{H} \sum_{j=1}^{W} U_c^i(k, j) \tag{3}$$

To achieve a better efficiency, we reduce the dimensions of S_i by using a simple fully connected layer F_{fc} and obtain Z_i:

$$Z_i = F_{fc}(S_i) \tag{4}$$

The dimensions of Z_i is controlled by a compression ratio r

$$d = ceil(C/r) \tag{5}$$

where d denotes the dimensions of Z_i; r is a constant value $r = 2$ in our algorithm. To guarantee our selection weights has the same dimensions with U_i, we expand Z_i to X_i by a fully connected layer F_{fc}:

$$X_i = F_{fc}(Z_i) \tag{6}$$

A Batch Normalization layer and a ReLU unit follows the connected layers. Then, a concatenate operation is applied on X_1, X_2, and X_3 to obtain the merged features M. A softmax and split operation F_{ss} is used to adaptively select different spatial scale information

$$x_i = \frac{e^{X_i}}{e^{X_1} + e^{X_2} + e^{X_3}} \tag{7}$$

where x_i is the selection weights and $x_1 + x_2 + x_3 = 1$. To calculate weighted features $\widetilde{U}_i$, we conduct a element-wise product between x_i and U_i.

$$\widetilde{U}_i = x_i \times U_i s \tag{8}$$

Finally, we easily perform an element-wise summation between all weighted features $\widetilde{U}_i$ to obtain the final output feature map $\widetilde{U}$.

$$\widetilde{U} = \widetilde{U}_1 + \widetilde{U}_2 + \widetilde{U}_3 \tag{9}$$

C. MULTI-SCALE RCEPTIVE FIELD MOUDLE

To learn effective clues from different scale receptive fields further, MSRF manually designs four scale kernels which are employed on template feature maps and exemplar feature maps to obtain four scale sub-feature maps.

1) TRACKING PHASE

As shown in Figure. 1, f_i represents the predicted confidence score map that highlights the 17×17 target region; and $i \in \{1, 2, 3, 6\}$ denotes the scale of kernel size corresponding to different receptive fields. We obtain $\varphi(z)$ and $\varphi(x)$ by throwing images (x and z) into Siamese Network and SRFB gradually. Then, we employ convolutional layers with different scale kernel size to transfer $\varphi(z)$, $\varphi(x)$ to $\varphi_i(z)$ and $\varphi_i(x)$, respectively. We take the $\varphi(z)$ as an example.

$$\begin{aligned}
\varphi_1(z) &= Conv1 \times 1(\varphi(z)) \\
\varphi_2(z) &= Conv2 \times 2(\varphi(z)) \\
\varphi_3(z) &= Conv3 \times 3(\varphi(z)) \\
\varphi_4(z) &= Conv6 \times 6(\varphi(z))
\end{aligned} \tag{10}$$

Next, we compute the similarity of their representations with a fully-convolutional operation referring to SiamFC [2]

$$f_i(z, x) = \varphi_i(z) * \varphi_i(x) + b_i \tag{11}$$

Each $f_i(z, x)$ is independently supervised by the same ground-truth label $y \in \{+1, -1\}$ as [2]. The final score map is the sum of these four independent score maps

$$f_{out} = \sum_{i}^{\{1,2,3,4\}} (f_i + b_i) \tag{12}$$

The location of the target is determined by the distance d between the maximum score position and the center of the final output score map f_{out}. Then we multiply d by the stride of the network and give the displacement of the target from frame to frame. Multiple scales are searched in a single forward-pass by assembling a mini-batch of scaled images.

2) TRAINING PHASE

The loss of each branch is defined as

$$L_i(y, v) = \frac{1}{|D|} \sum_{u \in D} l_i(y, v) \tag{13}$$

where $L_i(y, v)$ is the ith branch loss; $D \rightarrow R$ denotes the map of scores; and $l_i(y, v)$ represents the logistic loss of each position defined as

$$l_i(y, v) = log(1 + exp(-yv)) \tag{14}$$

where v is the score of a single exemplar-candidate pair and y is its ground-truth label. The final loss L is a combination of the loss from four branches

$$L = \sum_{i}^{\{1,2,3,4\}} \lambda_i * L_i \tag{15}$$

where λ_i the weight parameter is a constant value and equals to 0.25 in our algorithm.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. IMPLEMENTATION DETAILS

ASRF is implemented using PyTorch on a PC with an Intel(R) Xeon(R) 2.60GHz CPU and a single Nvidia GTX1080Ti with 12GB memory. To avoid over-fitting, we choose the video object detection dataset of ImageNet Large Scale Visual Recognition Challenge (ILSVRC15) [37] as our training data. The backbone Siamese Network adopts the modified AlexNet [2]. The parameters of all convolution layers are randomly generated. The stochastic gradient descent (SGD) is applied to our train model with the momentum of 0.9 to train the network, and the weight decay is set to 0.0005. We set the learning from 10^{-2} to 10^{-5} by a rate of exponentially decay. The model is trained for 50 epochs with a mini-batch size of 32. To find the scale of targets, we set three scales variation $1.025^{\{-1,0,1\}}$ for the object.

B. STATE-OF-THE-ART COMPARISON

1) OTB BENCHMARK

OTB is a popular public benchmark which is widely used for assessing trackers. The OTB-2013 and OTB-2015 benchmark respectively include 50 and 100 sequences tagged with 11 attributes, and all sequences are fully annotated. We evaluate the proposed ASRF with comparisons to several state-of-the-art trackers, including DSST [9], SiamFC [2], CNNSVM [19], CF2 [31], SRDCFdecon [30], CREST [39]. All the trackers were initialized in the first frame with their corresponding ground-truth. Average success plots and precision plots were reported. The OPE criteria for OTB is applied to evaluate our ASRF. The success plots in AUC is the main criteria, and it is defined as the Intersection-over-Union (IOU) ratio between the predicted bounding box and the ground truth. According to Figure. 4 and Figure. 5, our ASRF ranks the first place among the state-of-the-art trackers on both datasets. The success AUC is 0.673 on OTB-2013 and 0.641 on OTB-2015, respectively.

2) ATTRIBUTE-BASED EVALUATION

To show the detailed performance comparison, an attribute-based analysis on the OTB-100 dataset is illustrated in Figure 10. We adopt the success AUC plot as the main criteria to compare our ASRF with other trackers which keep the same as **OTB benchmark** part. All videos in this benchmark are annotated with nine different attributes: scale variation, fast motion, deformation, illumination variation, in-plane rotation, out-of-plane rotation, out-of-view,

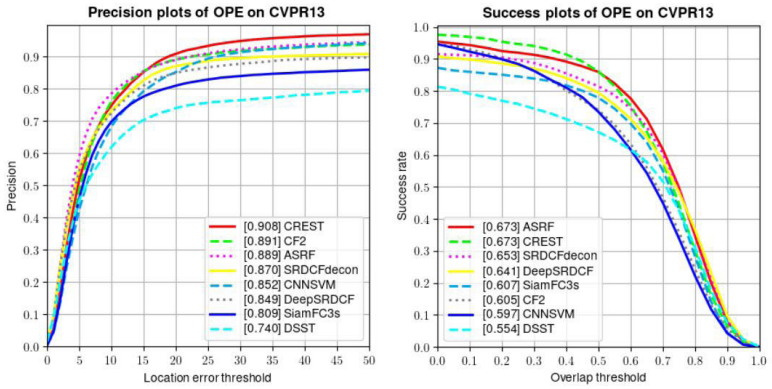


FIGURE 4. Precision and success plots with AUC for OPE on the OTB-2013 benchmark [49].

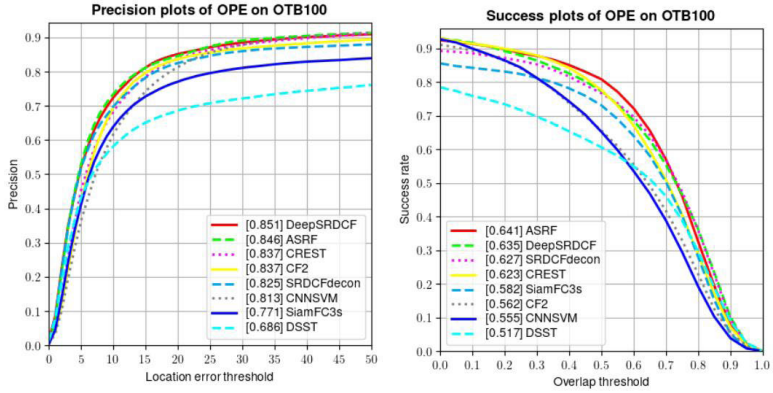


FIGURE 5. Precision and success plots with AUC for OPE on the OTB-2015 benchmark [50].

occlusion, motion blur, and background clutter. As shown in Figure. 10, our proposed tracker ranks first place on eight attributes and achieves consistent superior performance compared to the baseline tracker SiamFC all nine attributes.

3) COMPARISON OF COMPUTATIONAL EFFICIENCY

To compare speed and accuracy simultaneously, we compare several state-of-the-art real-times trackers, including PTAV [29], fDSST [9], SiamFC [2], Staple [1], GOTURN [17], KCF [18], Re3 [14], LMCF [46] with our proposed ASRF on the OTB-2013 benchmark. As illustrated in Figure. 9, our ASRF achieves the fairly good performance considering speed and accuracy simultaneously. It is worth mentioning that our tracker can run at fast speed over 68 fps, which is far exceeding real-time speed.

4) UAV-123 BENCHMARK

We also evaluate our tracker on the aerial video benchmark, UAV-123 [32]. The characteristics of UAV-123 inherently differ from above datasets such as OTB-2013 and OTB-2015. Figure. 6 illustrates the precision and success plots of the trackers. Our ASRF achieves the fairly good performance among other trackers with success AUC 0.517 and precision AUC 0.730.

5) VOT BENCHMARK

The VOT2015 [23] dataset contains 60 sequences, aiming at assessing the short-term performance of trackers. The benchmark applies a reset-based methodology to evaluate trackers. The Expected Average Overlap (EAO), which takes account of both accuracy and robustness, is used to assess the overall performance.

Figure. 7 illustrates the EAO score of our proposed method and 62 other state-of-the-art trackers evaluated on VOT2015. Although our ASRF ranks second in terms of EAO score, ASRF can conduct at 68 fps, which is more than 68 times of MDNet (first rank).

Compared with VOT2015, VOT2017 [22] provides a new real-time experiment. The real-time experiment requires trackers to deal with real-time video stream at least 25 fps if the tracker fails to submit the tracking result in 40ms, the bounding box of the last frame will be reused as the result in the current frame.

Figure. 8 reports the EAO score of ASRF against 51 other state-of-the-art trackers, and our ASRF achieves the first rank

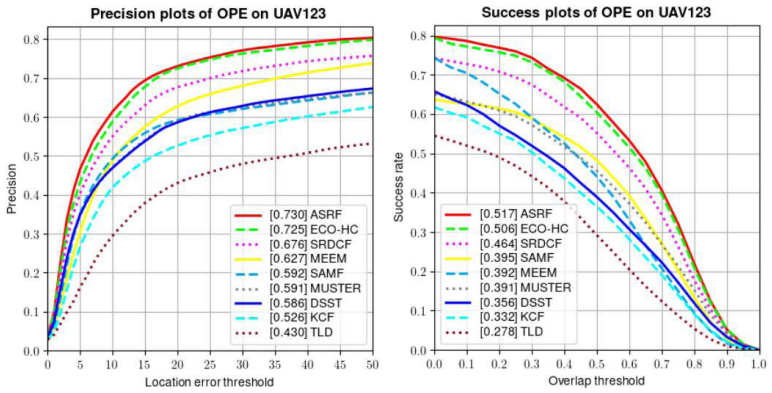


FIGURE 6. Precision and success plots with AUC for OPE on the UAV123 benchmark [32].

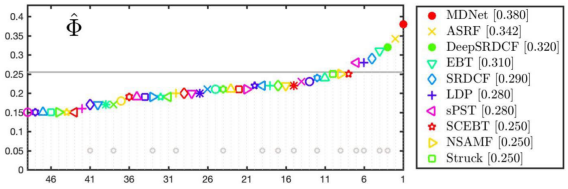


FIGURE 7. An illustration of the expected average overlap plot on the VOT2015 [23] challenge.

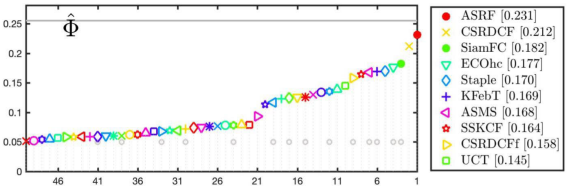


FIGURE 8. The EAO scores for the real-time experiment on VOT2017 [22] challenge.

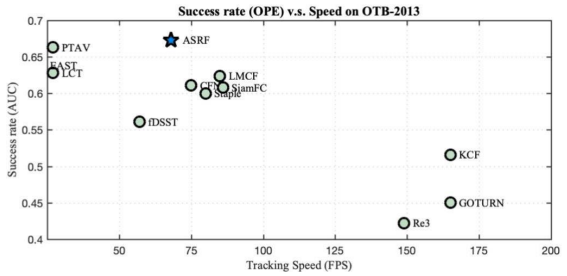


FIGURE 9. Relationship between speed and success AUC on OTB-2013 [49] challenge.

according to EAO score. Specifically, ASRF surpasses the baseline SiamFC by 26%.

C. COMPARISON BETWEEN ASRF AND BASELINE SIAMESEFC

To intuitively exhibit the improvement of our tracker compared with baseline SiameseFC, we visualize the predicted bounding boxes of these two trackers and the ground-truth.

TABLE 1. Ablation study of our proposed method on the OTB benchmark.

	SRFB	MSRF	OTB-2013	OTB-2015
Baseline			0.607	0.582
Variation1	✓		0.623	0.605
Variation2		✓	0.654	0.619
Ours	✓	✓	**0.673**	**0.641**

TABLE 2. Comparison with other multi-scale feature trackers on the OTB-2013 benchmark [49].Red, Green and Blue fonts indicate the top-3 trackers, respectively.

tracker	ASRF	HCF	HDT	DSiam	StructSiam	SA-Siam
AUC	0.673	0.605	0.603	0.656	0.638	0.676
Speed	68	11	10	45	46	50

Figure. 11 illustrates the tracking snapshots of these two trackers on four typical sequences (*Basketball, CarScale, Singer2, and Ironman*) from the OTB-100 dataset. We can conclude that ASRF learns more effective deep features to discriminate the target with similar distractors from the *Basketball* sequence (first column). The *Ironman* sequence (second column) shows ASRF has better adaptability to fast motion challenge compared with the SiameseFC. From the *CarScale* sequence (third column), it is observed that ASRF can better localize the target in the presence of large scale changes. Finally, we analyze the *Singer2* sequence (fourth column), our ASRF and SiameseFC drift to the background while tracking the target (#190). Our ASRF can re-track the target due to larger receptive fields while SiameseFC fails. Overall, our proposed tracker enlarges the discrimination between targets and semantic backgrounds, has a better re-tracking ability, and provides a more accurate location. Thus our ASRF can track the target effectively in all given sequences.

D. COMPARISON WITH OTHER MULTI-SCALE FEATURE TRACKERS

In this section, we compare our ASRF with other multi-scale feature trackers including HCF [31], HDT [35], DSiam [15], StructSiam [52], and SA-Siam [16], on the OTB-2013 bench-

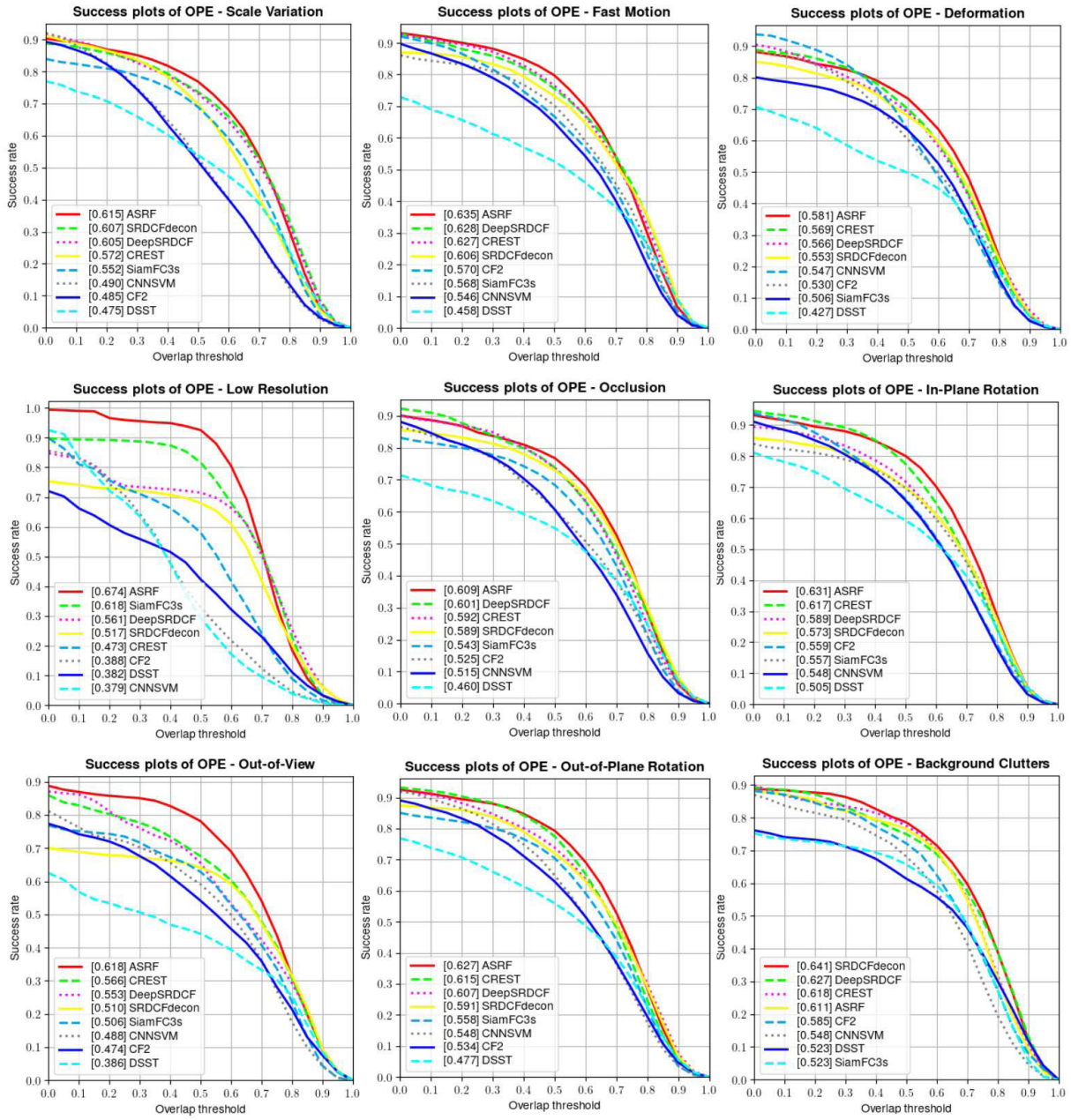


FIGURE 10. Overlap success plots of OPE with AUC for 10 tracking challenges on OTB-100 [50].

mark. Table 2 shows that our ASRF achieves the second best performance among the other multi-scale feature trackers with the first rank speed 68 fps. It is worth mentioning that our method shows almost the same performance with first rank tracker SA-Siam, but 18 fps faster than SA-Siam.

E. ABLATION STUDY

To show the impacts of different components, we perform three variations (*Variation*1 $=$ *SiamFC* $+$ *SRFB*, *Variation*2 $=$ *SiamFC* $+$ *MSRF*, *Ours* $=$ *ASRF*) of our tracker and evaluate them on the OTB benchmark by the success overlap. In this section, all training parameters are the same for the variations. We first test the impact of the SRFB on the quality of our tracking algorithm. Table 1 summarizes that *Variation*1 outperforms the baseline tracker by 2.6% on OTB-2013 and 3.9% on OTB-2015. This proves the fact that the SRFB improves tracking performance. Secondly, we investigate the impact of MSRF (*Variation*2). Table 1 shows that *Variation*2 makes an extraordinary progress on the success overlap 7.7% and 6.3%. *Ours* represents our proposed ASRF, we can observe that ASRF outperforms the baseline tracker by 10.9% on OTB-2013 and 10.1% on OTB-2015. Overall, all the experiment results clearly indicate that all results consistently support that each component of our improved methods makes a meaningful contribution to tracking performance improvement.

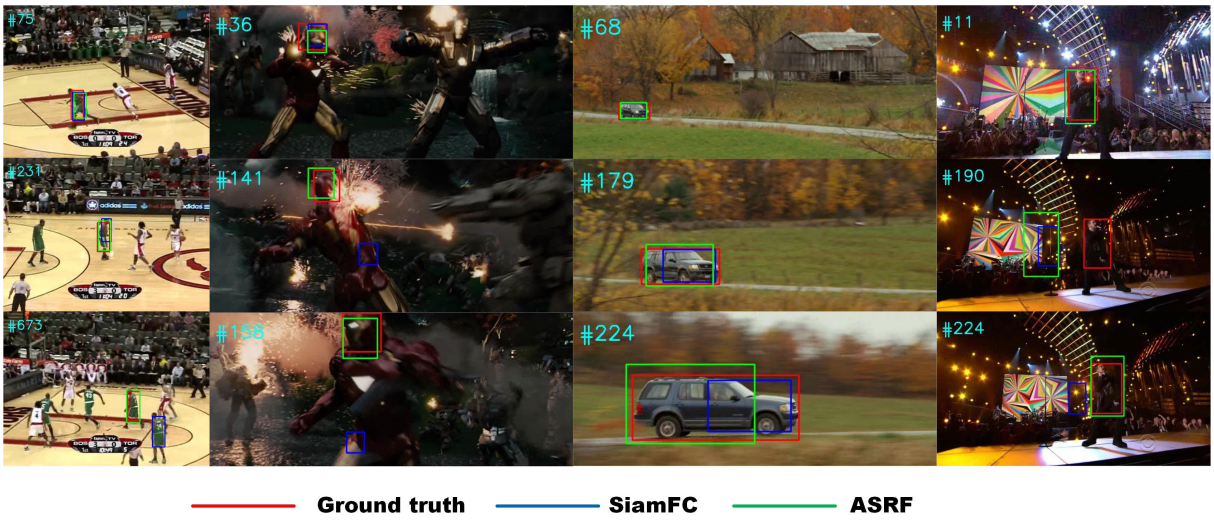


FIGURE 11. Tracking snapshots of SiamFC and ASRF on four challenging sequences selected from the OTB-100 [50].

V. CONCLUSION

In this paper, we design a novel Auto-Selecting Receptive Field network (ASRF) for visual tracking. The proposed ASRF includes two novel modules: (1) SRFB provides a way to adaptively adjust receptive field size for each neuron based on multiple scales of input information. (2) MSRF selects effective clues from different scale receptive fields further. Compared with the state-of-the-art scheme, the proposed ASRF demonstrates more robust performance in handling complex backgrounds such as similar distractors, model drift, and fast motion. Besides, ASRF provides a more accurate target location. We evaluate our tracker on five public benchmarks and have validated the advantages of tracking robustness and efficiency of the proposed method, and our model runs beyond real-time speed.

REFERENCES

[1] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1401–1409.

[2] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 850–865.

[3] G. Bhat, "Accurate tracking by overlap maximization," Tech. Rep., 2019.

[4] B. Chen, D. Wang, P. Li, S. Wang, and H. Lu, "Real-time'actor-critic'tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 318–334.

[5] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*. [Online]. Available: https://arxiv.org/abs/1706.05587

[6] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 764–773.

[7] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *Proc. ECCV*, 2016, pp. 472–488.

[8] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 6638–6646, Jul. 2017.

[9] M. Danelljan and G. Häger, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2016.

[10] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 58–66.

[11] H. Fan and H. Ling, "SANet: Structure-aware network for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, pp. 42–49, Jul. 2017.

[12] J. Fan, Y. Wu, and S. Dai, "Discriminative spatial attention for robust tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2010, pp. 480–493.

[13] J. Gao, T. Zhang, X. Yang, and C. Xu, "Deep relative tracking," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 1845–1858, Apr. 2017.

[14] D. Gordon, A. Farhadi, and D. Fox, "Re^3: Real-time recurrent regression networks for object tracking of generic objects," 2017, *arXiv:1705.06368*. [Online]. Available: https://arxiv.org/abs/1705.06368

[15] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic siamese network for visual object tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1763–1771.

[16] A. He, C. Luo, X. Tian, and W. Zeng, "A twofold siamese network for real-time object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4834–4843.

[17] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 749–765.

[18] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[19] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2015, pp. 597–606.

[20] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: https://arxiv.org/abs/1502.03167?context=cs

[21] I. Jung, J. Son, M. Baek, and B. Han, "Real-time MDNet," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 83–98.

[22] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. C. Zajc, T. Vojir, G. Hager, A. Lukezic, and A. Eldesokey, "The visual object tracking VOT2017 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1949–1972.

[23] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder, "The visual object tracking VOT2015 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Dec. 2015, pp. 1–23.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

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

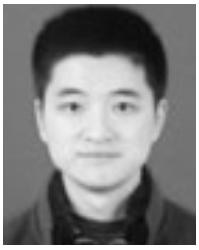
[26] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.
[27] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988.
[28] S. Liu, D. Huang, and Y. Wang, "Receptive field block net for accurate and fast object detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 385–400.
[29] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 4902–4912.
[30] A. Lukežic, T. Vojír, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4847–4856.
[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.*, Dec. 2015, pp. 3074–3082.
[32] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 445–461.
[33] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, Jun. 2010, pp. 807–814.
[34] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4293–4302.
[35] Y. Qi, S. Zhang, L. Qin, H. Yao, Q. Huang, J. Lim, and M.-H. Yang, "Hedged deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4303–4311.
[36] L. Ren, X. C. J. Lu, M. Yang, and J. Zhou, "Deep reinforcement learning with iterative shift for visual tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 684–700.
[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, C. Alexander Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[38] Y. Song, M. Chao, X. Wu, L. Gong, and M. H. Yang, "VITAL: Visual tracking via adversarial learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8990–8999.
[39] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. Lau, and M.-H. Yang, "Crest: Convolutional residual learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2574–2583.
[40] C. Sun, D. Wang, H. Lu, and M.-H. Yang, "Learning spatial-aware regressions for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8962–8970.
[41] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 1–7.
[42] 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.*, Jun. 2015, pp. 1–9.
[43] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2818–2826.
[44] R. Tao, E. Gavves, and A. W. Smeulders, "Siamese instance search for tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1420–1429.
[45] L. Wang, W. Ouyang, X. Wang, and H. Lu, "STCT: Sequentially training convolutional networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1373–1381.
[46] M. Wang, Y. Liu, and Z. Huang, "Large margin object tracking with circulant feature maps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4021–4029.
[47] N. Wang, Y. Song, C. Ma, W. Zhou, W. Liu, and H. Li, "Unsupervised deep tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1308–1317.
[48] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. Maybank, "Learning attentions: Residual attentional siamese network for high performance online visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4854–4863.
[49] 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.
[50] 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.
[51] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: A survey," *ACM Comput. Surv.*, vol. 38, no. 4, p. 13, Dec. 2006.
[52] Y. Zhang, L. Wang, J. Qi, D. Wang, M. Feng, and H. Lu, "Structured siamese network for real-time visual tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 351–366.
[53] 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. 101–117.

JUNFEI ZHUANG received the B.Eng. and M.Eng. degrees in safety engineering from the China University of Petroleum, Beijing, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree with the Beijing University of Posts and Telecommunications. His current research interests include computer vision and visual tracking.

YUAN DONG was born in 1970. He received the B.S. and M.S. degrees in communication and information system from Xidian University, China, in 1996, and the Ph.D. degree in communication and information system from Shanghai Jiao Tong University, China, in 1999. From 1999 to 2001, he was a Researcher with Nokia Research and Development Center, China, and as a Research Scientist. From 2001 to 2003, he was a Postdoctoral Research Staff with the HTK Speech Research Group, University of Cambridge, U.K. Since 2003, he has been a Professor with the Communication Engineering, Beijing University of Posts and Telecommunications, China. He has published more than 50 articles. His current research interests include video classification and clustering, and computer vision. He is a Reviewer and an Expert in many academic journals and academic conferences.

HONGLIANG BAI was born in Xinmin, China, in April 1977. He received the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, in 2006. He is currently the Founder and the CEO of Beijing Faceall Technology Company Ltd.. He has published more than 40 articles and applied for more than 90 patents. His main research interests include in machine learning, computer vision, and image retrieval.

PEILIANG ZUO received the B.S. degree from Xidian University, Xi'an, China, in 2013, and the M.S. degree from the Beijing Electronics Science and Technology Institute, Beijing, China, in 2016. He is currently pursuing the Ph.D. degree with the Key Laboratory of Universal Wireless Communication, Ministry of Education, Beijing University of Posts and Telecommunications (BUPT). His research interests include wireless networking, cognitive radio networks, network planning/optimizing, software defined radio (SDR), compressive sensing, and source localization.



JIANMING CHENG received the M.E. degree from the Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2016, where he is currently pursuing the Ph.D. degree and fourth year student of Graduate School of Information and Communication Engineering. His research interests include cross-layer design and ad-hoc networks.