IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 3, 2021, accepted January 13, 2021, date of publication January 18, 2021, date of current version January 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052511

Channel Positive and Negative Feedback Network for Target Tracking

YUE CHEN, XIAOWEI HE, (Member, IEEE), ZHONGLONG ZHENG, (Member, IEEE), PENGCHENG BIAN, AND YI LI
College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321000, China

Corresponding author: Xiaowei He (jhhxw@zjnu.edu.cn)

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 61572023 and Grant 61672467.

ABSTRACT Aiming at alleviate the detrimental effect of similar object interferences and target state changes in SiamRPN tracker, a Channel Positive and Negative Feedback Network (CPFN) is proposed, in which the Gaussian score map is generated by the feature channels selected by a Gaussian kernel, and the map is combined with the classification branches of SiamRPN. In this way, the feature channels are divided into positive feedback channels and interference channels, and these feature channels are effectively utilized. In addition, a channel weight update strategy is proposed to enhance the robustness of the tracker and avoid template pollution caused by inadequate template update. Extensive experiments on tracking benchmarks including VOT2016, VOT2018, VOT2019, OTB100, UAV123, LaSOT and GOT-10k show that the proposed CPFN outperforms the state-of-the-art methods based on small backbone network in terms of accuracy and achieves high-speed tracking.

INDEX TERMS Target tracking, Siamese network, Gaussian kernel, feature combination.

I. INTRODUCTION

Target tracking is one of the most challenging visual processing tasks for tracking uncertain targets. Given the bounding box of the target in the first frame, the location of it in subsequent frames needs to be predicted. Due to the deformation, scale variation, fast motion, occlusions, background clutter in scene, it is hard to accurately calibrate the target, which is the main challenge of target tracking. In order to apply the target tracking algorithm to actual scenarios, e.g., autonomous driving, security monitoring, and human-computer interaction, a more accurate and efficient target tracking algorithm is required.

There are currently two main research directions in target tracking: (1) Traditional tracking algorithm based on correlation filter; (2) Tracking model based on deep convolutional network. As one of two research directions, correlation filter plays an important role in target tracking. Corresponding template is calculated from the current frame to determine the target position in the next frame. However, previous hand-crafted feature of the correlation filter is too simple, and it cannot handle the complex circumstances of target state changes and the interference of similar objects, resulting in the tracker fails to track the target accurately [1], [3], [13]. With the development of deep learning, deep features have shown excellent effects in image processing and are widely used in image classification, face recognition, object segmentation [11], [28], [32], [35]. Therefore, deep features instead of hand-crafted feature are introduced into correlation filter methods [6], [8], [9], [14], [16], which greatly improved the tracking accuracy. In addition, channel feature combination, Bayesian and domain adaptation [20], [21] schemes are combined with target tracking, thus some new methods are proposed [15], [22], [23], [39].

Although early deep learning models have shown desirable tracking accuracy in the tracking task, the tracking speed is very slow and cannot meet the requirement of real-time tracking [14], [30], [33], [38]. In recent years, Siamese network architectures have shown desirable accuracy and speed for tracking, which have drawn widespread attention. A series of Siamese models convert the target tracking problem into a template matching problem, and achieve real-time tracking combining offline training with online tracking [2], [18], [24], [25], [34], [42]. One of the most prominent model is the SiamRPN [18], which uses region proposals to solve the problem of tracking rate drop caused by multi-scale feature extraction, and pushes the tracking rate up to 160fps.

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

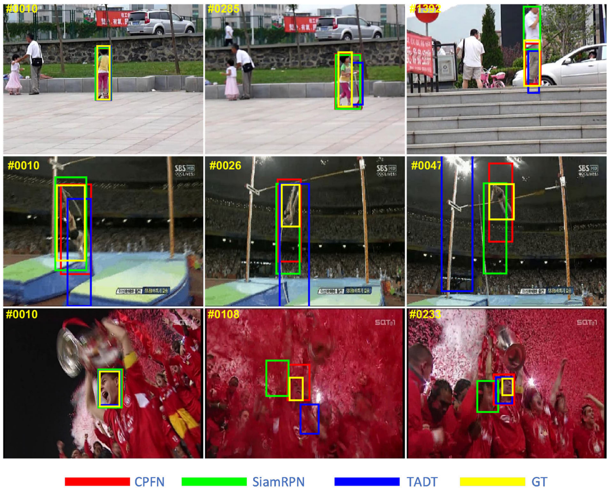

FIGURE 1. Comparison of CPFN, SiamRPN and TADT on three challenging video sequences. Our CPFN can better deal with similar object interference and pose variation.

Although the SiamRPN model shows good accuracy, it still exists two problems: (1) SiamRPN has a higher response value for similar objects, and target response value decreases when the target state changes. (2) In the tracking process, SiamRPN takes the target in the first frame as the fixed template feature, lacking update strategy. These problems will lead to the target drift during the tracking process. As shown in Figure 1, the SiamRPN is more susceptible to interference from target state changes and similar objects, and is easier to deviate from the target.

In this paper, based on SiamRPN, a novel Channel Positive and Negative Feedback Network is proposed, named as CPFN. In the network, a feature channel selection module based on Gaussian kernel is added, by comparing the normalized target location features with background features, it searches positive feedback feature channels that distinguish the target from surrounding objects and interference feature channels that interfere with the judgment of the target. The Gaussian score map generated by combining the two types of feature channels is used to judge the target. In Figure 1, CPFN shows better robustness and can determine the target position more accurately.

Extensive experiments are conducted with the proposed CPFN model on 6 benchmark datasets, including VOT2016, VOT2018, VOT2019, OTB100, UAV123, LaSOT, GOT-10k. In summary, the main contributions of this paper are as follows:

1. According to the feature of tracking task, we design a fast feature channel selection scheme based on Gaussian kernel, which can quickly select the positive feedback feature channel. The positive feedback feature channel can better highlight the feature difference between target and background in tracking, and it is conducive to determine the target.
2. Different from previous methods that discard of interference feature channels, e.g, TADT only uses some feature channels that can distinguish target instances, and a large amount of other feature information is discarded. By analyzing the characteristics of the interference feature channel, CPFN uses the interference feature channel for negative feedback excitation to suppress the background and improve recognition performance of target recognition, moreover, the model can make use of depth features effectively.
3. A new update strategy is proposed to deal with the changes of environment and target posture during the tracking process, and it avoids the template pollution caused by the simple template update scheme, hence the robustness and accuracy of tracker can be enhanced.

II. RELATED WORKS

A. USE CNN AS FEATURE EXTRACTION

In recent years, due to the wide use of deep features in image processing, some correlation filter trackers begin to use Convolution Neural Network (CNN) to extract the deep feature of images, so as to improve the tracking performance. The most typical models are DeepSRDCF [8] and CF2 [29], which use different levels of pre-trained CNN features combined with correlation to improve the robustness and accuracy of tracking. Traditional correlation filter models such as KCF [13] and fDSST [7] greatly improve tracking accuracy and robustness by replacing hand-crafted feature with deep feature. C-COT [9] and ECO [6] use continuous convolution operators to enhance the feature extraction of the tracker, which can achieve the most advanced tracking accuracy in terms of tracking performance, however, when the size of input image is large, tracking rate cannot support real-time tracking. CNN-SVM [14] regards the tracking problem as a classification problem, and uses CNN to extract features, and SVM for division. The FCNT model [37] uses a regression framework for feature selection, looking for feature channels that reflect the difference between the target and the background in the deep feature. The advantage of these methods is the deep feature is utilized to improve performance. Although these models all need the deep feature, they all use pre-trained classification feature extractors and do not perform offline training for tracking tasks. Tracking speed decreases if online training or updating of deep features is required, which limits the richness of tracking model.

B. SIAMESE NETWORK FOR TRACKING

Bertinetto *et al.* proposed the Siamese-FC model based on template matching [2], which is one of the most representative deep learning models of target tracking. Utilizing a backbone network with shared parameters, deep features of the template and the search area are extracted. In the previous online tracking process, the target in the first frame is used as a template to extract features, but these features are not updated in subsequent tracking, and the search area is directly matched. Without online updating, the speed of the tracker can exceed the requirements of real-time tracking. Both Siamese-FC and

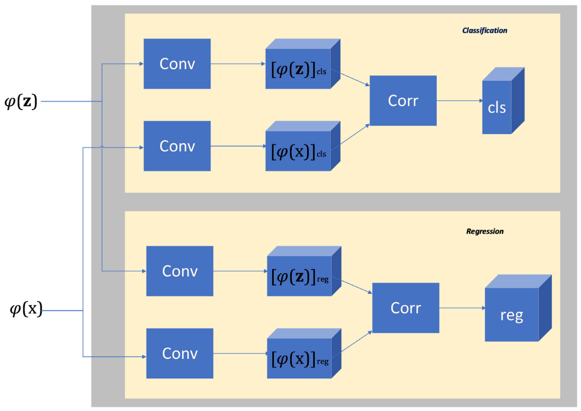

FIGURE 2. RPN module architecture, including classification branch and regression branch, $\varphi(\cdot)$ is feature extractor.

DCF [7], [26] use a multi-scale method to estimate a rough target size, which requires a lot of additional calculations. Bo Li *et al.* proposed the Siamese-RPN model based on Region Proposal Network (RPN), which introduced the prior information on the anchor box to quickly estimate the change of target scale, and improving the tracking rate to 160 FPS. Fig.2 shows the RPN module structure, including classification branches and regression branches. By analyzing the essential difference between tracking and classification, Xin Li *et al.* proposed to integrate target-aware features into Siamese architecture, and established a TADT model [25] for target tracking tasks. They analyzed the difference between instance distinction and class distinction in the target tracking process, and judged that only a few channels is distinguishing for instance distinction, so they explore channels with distinguishing power. In addition, some channels, are more sensitive to scale changes of target. The TADT model uses pre-trained classification features, and the filtered channel features are directly used for tracking tasks without additional offline training. In this paper, a feature selection module is designed to explore distinctive feature channels online by Gaussian kernels, and combine the negative feedback incentive of interference channels to generate Gaussian score maps. CPFN improves target tracking ability by integrating Gaussian score map and classification branch.

III. PROPOSED METHOD

The defects of SiamRPN are as follows: (1) Similar objects and target have high response values. (2) The change of target state will cause a decreased response value of the result. Therefore, in the tracking process, it is easy to happen target drift phenomenon. The distribution of the ideal labelled state is similar to the Gaussian distribution. Therefore, in order to get an ideal response result, feature channels that similar to Gaussian kernel should be selected from the feature map. In this paper, we combine these selected feature channels to generate a Gaussian score map, which acts as a supplementary of SiamRPN results to improve the robustness of the model. Then the selected feature channels are divided into positive feedback feature channels and interference feature channels. The positive feedback channel is defined as a feature channel that can effectively distinguish the target and the background, and the interference feature channel is defined as a feature channel that can interfere with target and background recognition. Different from TADT, the proposed method not only uses the positive feedback feature channel, in which the feature is similar to Gaussian distribution, to generate Gaussian score map, but also uses the interference feature channel. By analyzing the effect of the interference feature channel on target tracking, we propose a negative feedback scheme to suppress the response value of similar objects and improve the accuracy of target tracking. Figure 3 shows the architecture of CPFN.

In Figure 3, firstly, the deep feature of the target and search area is obtained by feature extraction, then the fused features are obtained by feature matching with convolution.

$$f(z,x)_i = \phi(z)_i \star \phi(x)_i \tag{1}$$

where $\phi(\cdot)$ is feature extractor, $\star$ is convolution operation, z is template, x is search area, $i \in 0, 1, \ldots N-1$ is the channel number, N is the total number of channels. The fused feature map contains the spatial and semantic information of target and background after template matching.

A. POSITIVE FEEDBACK FEATURE CHANNEL

In the process of tracking, the ideal state of tracking results is that the target area is with high response and the non target area is with low response, which is similar to Gaussian kernel. Through the lower response value, we can clearly judge that the object at its location is not the target object. Considering this precondition, we set a similar hinge loss function $H(\cdot)$ to eliminate the interference of low response value. The $H(\cdot)$ is defined as:

$$H(x) = max(0, x-\varepsilon) + \varepsilon * (\lceil x-\varepsilon \rceil), \quad \varepsilon \in (0,1) \tag{2}$$

We define the target state as follows:

$$y = H(e^{-\alpha\frac{(s-s_0)^2+(l-l_0)^2}{2*\beta^2}}) \tag{3}$$

where (s_0, l_0) is target center position, (s, l) is map location. Inspired by TADT, it is important to find the distinguishing feature channels directly. Therefore, it is necessary to find positive feedback feature channels that similar to the Gaussian kernel, as shown in Figure 5 (e). We define the method of using positive feedback channels to help target location as channel positive feedback. By normalizing $f(z,x)_i$, the relative activation response of the target and background features in the search area to the template features can be compared. If the relative value of the region position is higher, the region feature and the template feature are more similar on the current channel, and the feature matching degree is better than other regions. Otherwise, the area feature does not match the target feature well. In order to select feature channels similar

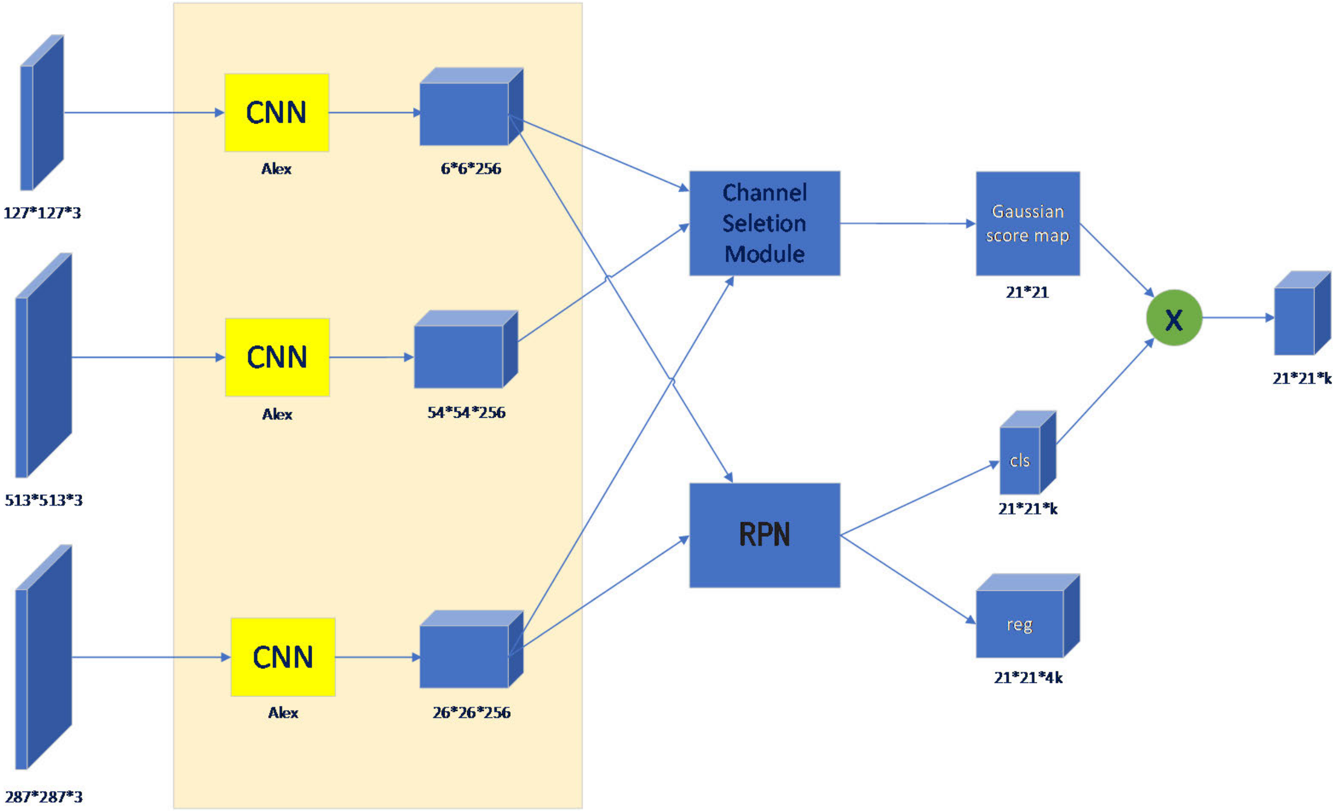

FIGURE 3. CPFN network architecture. It consists of an Alex backbone with shared parameters, a channel selection module, and a RPN module. Details of RPN module could be found in Figure 2, details of channel selection module could be found in Figure 4.

to the target state, the normalized $f(z,x)_i$ and y are calculated:

$$s_i = \|H(\frac{f(z,x)_i - min(f(z,x)_i)}{max(f(z,x)_i - min(f(z,x)_i))}) - y\|_{L_1} \tag{4}$$

The s_i is smaller, $f(z,x)_i$ is closer to the target state y, and the i-th feature channel can better distinguish the target and the background, thus a higher weight should be assigned for $f(z,x)_i$. The weight calculation formula is defined as follows:

$$w_i = e^{-\frac{v*j}{N}+b}, \quad j = sort(s_i),\ j \subseteq [0,1,2,\ldots N-1] \tag{5}$$

where v is the scaling factor, b is the offset, $sort(\cdot)$ is sort function. The positive feedback feature channel can be selected through w_i and used for the generation of the score map. Figure 5 (b) shows the score map generated by positive feedback feature channel.

B. NEGATIVE FEEDBACK OF INTERFERENCE FEATURE CHANNEL

The difference between the tracking task and the classification task is that the former pays more attention to the difference between the examples, and the latter pays more attention to the distinction of categories. According to the TADT and GradNet [24], only a few feature channels have distinguishing effects on target instances, and the features of these channels can represent the difference between instances. In TADT, lots of channel information are not used fully, which leads to low utilization of deep features. In addition, certain feature channels that are significantly different from the Gaussian kernel will interfere with the recognition of targets and backgrounds, these feature channels are interference feature channel, as shown in Figure 5 (f).

Although distinguishing features are very important in the recognition of target and background, there are a few such feature channels. In a large number of interference channel features, the matching degree between the background and the template may be the same or higher than the one between the target and the template. A larger s_i indicates that there are many high corresponding values in $f(z,x)_i$. Multiple non-target areas have high response values, indicating that the channel feature highlights the non-target features. During the tracking process, if there is a high activation response value in these interference feature channels, the possibility that the activation area is not a target is greater than the area with a low response value. Based on this judgment, we give a negative incentive to the feature channels with higher s_i, which is defined as the negative feedback of the interference channel. By subtracting the score map generated by the interference channel, the background can be suppressed. Since not all interference feature channels have obvious negative incentive, and the interference features channels are quite different

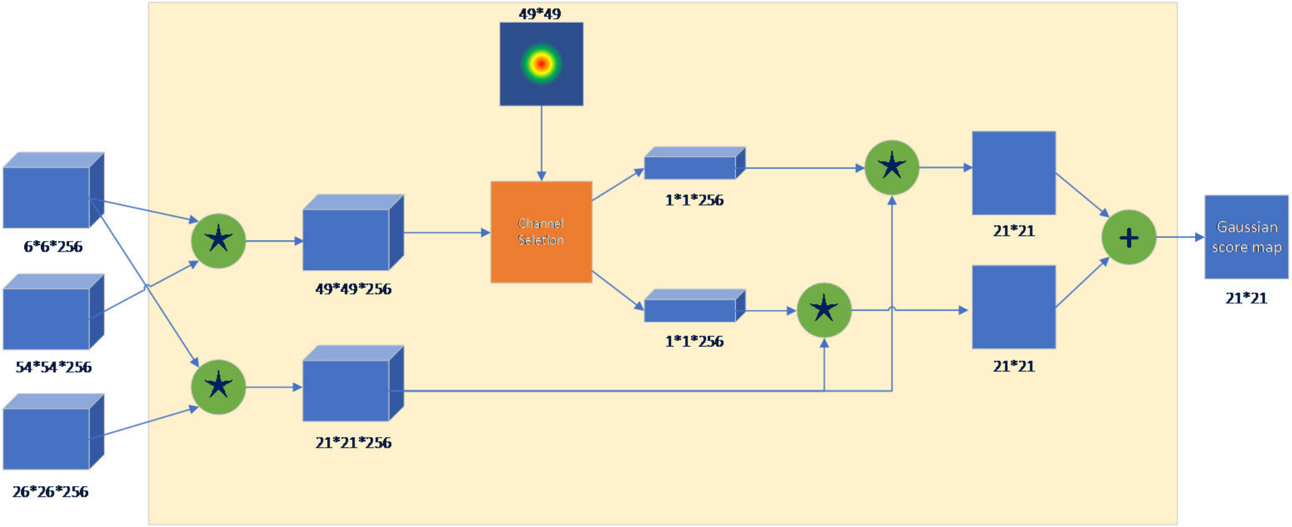

FIGURE 4. Selection of the positive feedback channel and the interference channel based on the Gaussian kernel, and different weights are combined to generate a Gaussian score map.

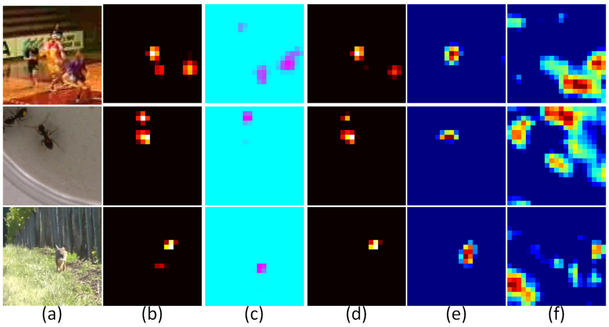

FIGURE 5. (a) is the search area, (b) is the score map generated by positive feedback feature channel, (c) is the score map generated by interference feature channel, (d) is the Gaussian score map generated by the combination of (b) and (c), (e) is the feature map selected from positive feature channel, (f) is the feature map selected from interference feature channel. For convenience of observation, only high response values are shown in the figure, and the higher the brightness, the greater the response value, and the higher the brightness, the greater the response value.

from the target state, these interference channels are given equal weights:

$$w_{neg_i} = \begin{cases} 1 & w_i < \theta \\ 0 & else \end{cases} \tag{6}$$

Figure 4 shows the division and combination framework of feature channels. Figure 5 (c) shows the score map generated by interference feature channel, and Figure 5 (d) shows the Gaussian score map after subtracting Figure 5 (b) from Figure 5 (c).

IV. TRACKING PROCEDURE

The offline pre-trained SiamRPN model is used as overall framework, the proposed channel selection scheme is embedded for online tracking.

A. INITIALIZATION

Given the first frame, the cropping template z_1 and the search area x_1 are sent to the feature extractor $\phi(\cdot)$, and the feature $f(z_1, x_1)$ is obtained by equation (1). Using the center of the target position, y_1 is obtained by equation (3).

B. TRACKING

w and w_{neg} are obtained by equations (4) (5). Given the initial target z_1 and the search area x_t in the current frame, we get $f(z_1, x_t)$, w, w_{neg} and combine them to get the Gaussian score map:

$$p_{pos_t} = \sum_{i=0}^{N-1} w_i \star f(z_1, x_t)_i, \quad p_{neg_t} = \sum_{i=0}^{N-1} w_{neg_i} \star f(z_1, x_t)_i \tag{7}$$

$$p_t = G((1-\lambda) * G(p_{pos_t}) - \lambda * G(p_{neg_t})) \tag{8}$$

where $G(\cdot)$ is the 0-1 normalization function.

p_t is then combined with the classification branch result of RPN cls_t to get the target position of the t-th frame:

$$\hat{loc_t} = \arg\max_{loc}(cls_t * p_t) \tag{9}$$

C. ONLINE UPDATE

Different from the way of updating the template, this paper uses the strategy of updating w and w_{neg} to avoid inaccuracy of template updates caused by the introduction of contaminated samples. When the target position of the t-th frame is determined, the target state y_t is generated by equations (2), (3). $f(z_1, x_t)$ is obtained with the convolution operation of the template z_1 and the search area x_t, and the w^t is obtained by equations (4), (5). The cosine loss is used to calculate the similarity between w and w^t. For reducing the sudden transition from low-quality channels to high-quality channels, this paper uses a weight judgment to filter out

TABLE 1. Results on the VOT2016 and VOT2018 dataset.

Model	Year	VOT2016			VOT2018			FPS
		EAO	A	R	EAO	A	R	
Ours (CPFN)	2020	0.428	**0.620**	0.191	**0.361**	**0.586**	0.281	130
SPM [36]	2019	**0.434**	**0.620**	0.210	0.338	0.580	0.300	120
ASRCF [4]	2019	0.391	0.563	**0.187**	0.328	0.494	0.234	28
C-RPN [10]	2019	0.363	0.594	-	0.273	-	-	23
SiamDW [42]	2019	0.303	0.535	0.303	0.270	0.538	0.398	70
TADT [25]	2019	0.301	0.551	0.326	-	-	-	33
LSSiam [27]	2019	0.294	0.530	0.320	0.229	-	-	100
GradNet [24]	2019	-	-	-	0.247	0.375	0.507	80
SiamRPN [18]	2018	0.393	0.618	0.238	0.352	0.576	0.290	160
DeepSTRCF [19]	2018	0.313	0.550	0.920	0.345	0.523	**0.215**	50
SA-Siam [12]	2018	0.291	0.540	1.080	0.236	0.500	0.459	50
ECO [6]	2017	0.375	0.550	0.200	0.280	0.480	0.270	8
SiamFC [2]	2016	0.235	0.530	0.460	0.188	0.500	0.590	86

Algorithm 1 Tracking Process

```
Initialize: input parameters $z_1, x_1$
  $f(z_1, x_1)$ according to Eqn. (1)
  $y_0$ according to Eqn. (3)
  $s$ according to Eqn. (4)
  $w$ according to Eqn. (5)
  $w_{neg}$ according to Eqn. (6)
Tracking: input parameters $x_t$
  $f(z_1, x_t)$ according to Eqn. (1)
  $p_t$ according to Eqn. (7)(8)
  $cls_t$ according to RPN
  $\hat{loc}_t$ according to Eqn. (9)
  if $t \bmod T == 0$ then
    $w^t$ according to Eqn. (4)(5)
    $\eta$ according to Eqn. (11)
    if $\eta > \sigma$ then
      $w$ according to Eqn. (12)
      $w_{neg}$ according to Eqn. (6)
    end if
  end if
  return
```

channels that have changed too much. The process can be formulated as follows:

$$c_i = \begin{cases} 1 & |w_i - w_i^t| < \gamma \\ 0 & else \end{cases} \tag{10}$$

$$\eta = \frac{(c \cdot w) * w^t}{|c \cdot w||c \cdot w^t|} \tag{11}$$

Only when $\eta > \sigma$, the update operation is performed:

$$w = w * (1 - \tau) + \tau * (w^t \cdot c), \quad \{\eta > \sigma\} \tag{12}$$

According to equation (12) and equation (6), new w_{neg_i} is obtained. Algorithm 1 can be obtained by integrating the above elements together.

V. EXPERIMENTS

The proposed CPFN is implemented on pytorch platform and all the experiments are conducted on a PC with Intel(R) Core(TM) i7-9700 CPU@3.00GHz and a Nvidia GTX 2080TI GPU. The input size of the template and search area is 127 pixels and 287 pixels, respectively. During the tracking process, parameters are set as follows: $\varepsilon = 0.7$, $v = 7$, $\lambda = 0.2$, $\gamma = 0.5$, $\tau = 0.3$. In VOT2016, VOT2018, UAV123 and GOT-10k, we set $b = 0.3$, referring to the TADT, and set θ to s_i corresponding to the 60th w_i. In VOT2019, we set $b = 0.25$ and θ to s_i corresponding to the 50th w_i. In OTB100, we set $b = 0.4$ and θ to s_i corresponding to the 40th w_i. In LaSOT, we set $b = 0.3$ and θ to s_i corresponding to the 50th w_i. We refer to the parameter settings of TADT and SiamRPN, and fine-tune them manually.

Our CPFN is composed of Alex backbone, RPN module and Gaussian selection module, which is a high-speed tracker of small backbone. Extensive experiments are conducted to evaluate the CPFN tracker against plenty of high-speed trackers of small backbone on VOT2016, VOT2018, VOT2019, OTB100, UAV123, LaSOT and GOT-10k benchmarks. The evaluation indicators on different testsets show the tracker's ability to cope with target drift issue, such as Expected Average Overlap (EAO), the center location error (precision plot), overlap between the predicted and field bounding boxes (success plot). For fair comparison, the compared trackers selected in the experiment are small backbone high-speed models, such as: C-RPN [10], SiamRPN, SPM [36], and SiamDW [42] etc.

A. RESULTS ON VOT

VOT challenge is composed of 60 different video sequences, each video has different challenging factors. The VOT dataset mainly includes three evaluation indicators: Expected Average Overlap (EAO), Accuracy (A), and Robustness (R). We use these three indicators to test model tracking performance. The proposed CPFN is evaluated on VOT2016, VOT2018, and VOT2019. Table 1 show the comparison results of different tracking models in VOT2016 and VOT2018. As shown in Table 1, the proposed CPFN ranks first in VOT2018, and ranks second in VOT2016. Compared with the most advanced high-speed tracker SPM, the EAO of CPFN can increase by 2.3% on VOT2018. Although the EAO

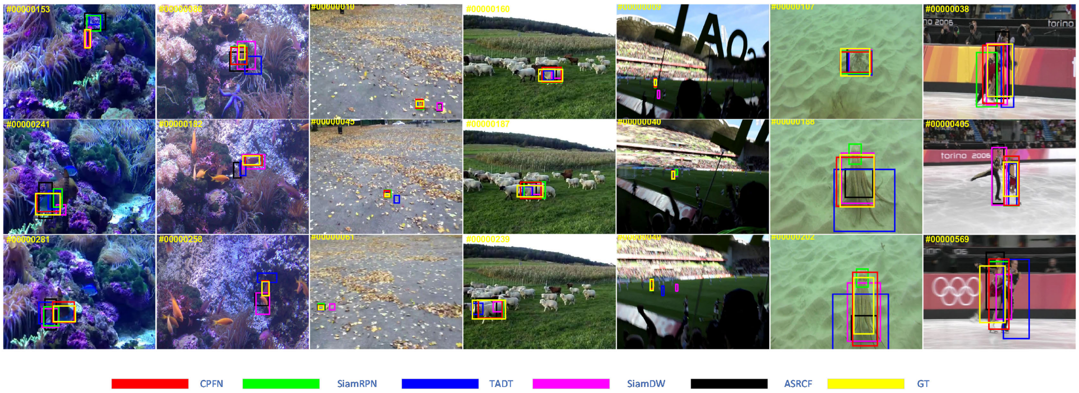

FIGURE 6. Representative visual results of different tracking algorithms on the VOT2016 dataset. GT is ground truth, CPFN is our proposed tracker.

of CPFN on VOT2016 is very close to SPM, the CPFN has better robustness and higher speed under the same accuracy. Compared with SiamRPN, EAO of CPFN has increased by 3.5%, 0.9% in VOT2016 and VOT2018. Compared with the C-RPN, TADT, GradNet, and SiamDW models, CPFN model has far better results on the VOT2016 and VOT2018. Figure 6 shows a partial visualization of CPFN and other trackers on VOT2016. It can be seen from Figure 6 that CPFN can better deal with the problem of target drift.

TABLE 2. Results on the VOT2019 dataset.

Model	Year	VOT2019			FPS
		EAO	A	R	
Ours (CPFN)	2020	**0.280**	**0.583**	0.522	130
SPM [36]	2019	0.275	0.577	0.507	120
SiamDW [42]	2019	0.242	0.538	0.632	70
TADT [25]	2019	0.207	0.516	0.677	33
SiamRPN [18]	2018	0.260	0.573	0.547	160
SA_SIAM_R [12]	2018	0.253	0.559	0.492	50
RankingT [17]	-	0.270	0.525	**0.360**	-
SiamMsST [17]	-	0.252	0.575	0.552	-
gasiamrpn [17]	-	0.247	0.548	0.522	-
SSRCCOT [17]	-	0.234	0.495	0.507	-

Table 2 shows the comparison results of different tracking models in VOT2019. The proposed CPFN ranks first in VOT2019, and EAO of CPFN is 0.5% hinger than SPM, and CPFN has higher speed. Compared with SiamRPN, CPFN's EAO has 2.0% improvement. CPFN's EAO is significantly higher than some trackers provided by the VOT2019 challenge report [17] such as SA_SIAM_R, SiamMsST, gasiamrpn, SSRCCOT, TADT. The difference between VOT2018 and VOT2016 is that some simple videos are replaced with more difficult video sequences. Although the EAO of SPM is higher on VOT2016, CPFN shows better robustness. And in VOT2018, EAO of CPFN is better than SPM. In VOT2019, some more challenging video sequences are added into VOT2018. Therefore, CPFN and SPM have a significant performance decrease in VOT2019 compared with VOT2018, while EAO of CPFN is still better than SPM in VOT2019.

B. RESULTS ON OTB100

OTB100 dataset is one of the most commonly used benchmark, which includes 100 different video sequences and 11 different tracking challenges. The evaluation of OTB100 follows two metrics, i.e, precision plot and success plot. The precision plot reports the percentage of the center location error that is less than certain thresholds. The success plot reports the percentage of frames is that the overlap between the predicted and field bounding boxes is higher than the given ratio. In this experiment, CPFN model is compared with several representative trackers, including TADT [25], GradNet [24], SiamRPN [18], CIResNet22-FC [42], CF2 [29], DeepSRDCF [8], CNN-SVM [14], HDT [31], SRDCFdecon [5]. As shown in Figure 7, the proposed CPFN ranks 1st both in precision plot and success polt. In terms of precision plot and success plot, CPFN is 1.5% and 1.3% higher than SiamRPN, 2.5% and 1.9% higher than TADT and GradNet. In the success plot and precision plot, the CPFN curve is always on the top, indicating that it can better deal with target drift.

C. RESULTS ON LaSOT

LaSOT is a very large public training and testing dataset, including 1400 video sequences and 70 categories, the testset selects 280 video sequences from LaSOT with different challenges for tracking. The average number of video frames is 2500 frames larger than other testset, bringing greater challenges to tracking. The LaSOT toolbox provides the tracking results of a series of trackers on the LaSOT benchmark, including mainstream trackers such as SiamFC [2], VITAL [33], MDNet [30], ECO [6], StructSiam [41] etc. In addition, we also compared our model with SiamRPN and SiamDW. Figure 8 shows the success plot and normalization precision plot of all the trackers tested on the LaSOT testset, indicating that the proposed CPFN achieves better performance and ranks first. In terms of normalization precision plot and success plot, CPFN is 2.2% and 1.6% higher than SiamRPN, 8.8% and 6.8% higher than MDNet,

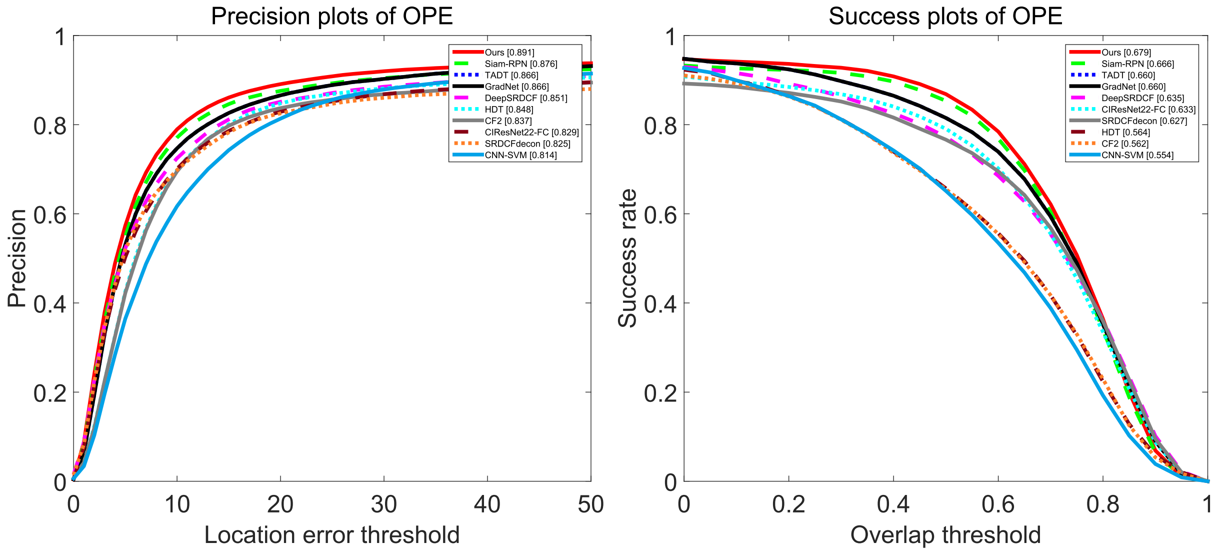

FIGURE 7. Results on the OTB100 dataset, left is precision plot and rigth is success plot.

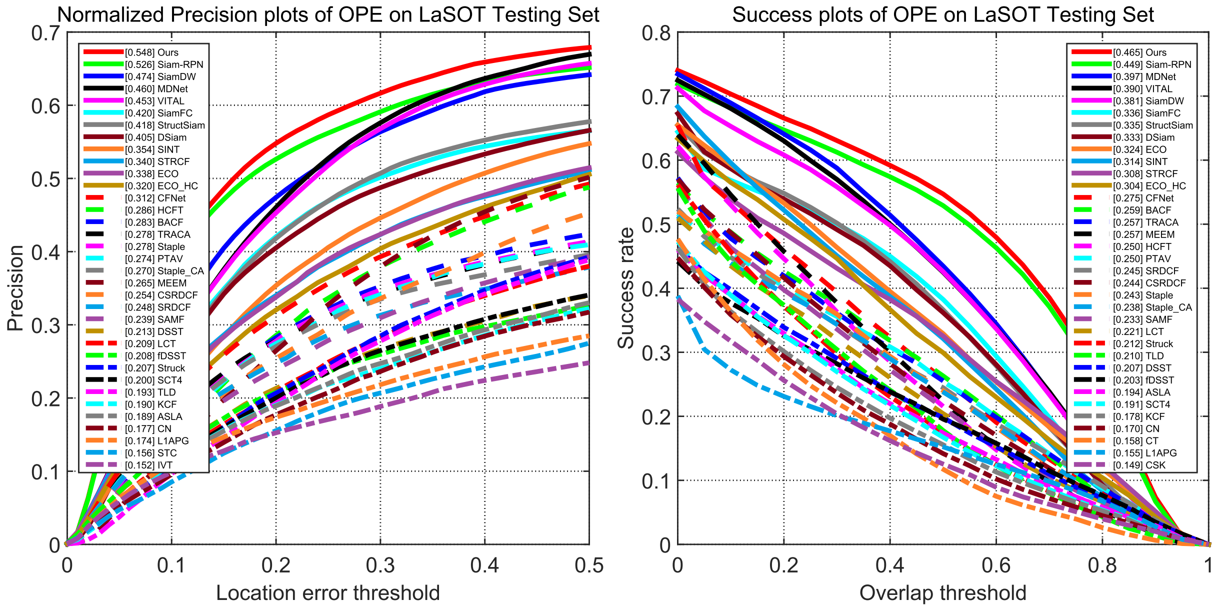

FIGURE 8. Results on the LaSOT Test dataset, left is normalization precision plot and rigth is success plot.

9.5% and 7.5% higher than VITAL. In the success plot and normalization precision plot, the CPFN curve is always on the top, indicating that it can better deal with target drift.

D. RESULTS ON UAV123

UAV123 dataset is captured from low-altitude unmanned aerial vehicles, which contains a total of 123 video sequences and 110K frames. The objects in the dataset mainly suffer from fast motion, large illumination, large scale variation, occlusions and variation, which is challenging for tracking. In this experiment, CPFN model compared with several representive trackers, including SiamRPN [18], ECO [6], ECO-HC [6], SiamFC [2], CFNet [34], SAMF [26], MEEM [40], DSST [7]. Table 3 shows the performance comparison of each model, where AUC is area under curve and Prec is precision score. On AUC and Prec, the proposed CPFN has an improvement of 0.9% and 1.9% compared to SiamRPN, an improvemet of 6.2% and 4.7% compared to the ECO, and an improvement of 8.1% and 6.3% compared to the ECO-HC. On AUC, the proposed CPFN has an improvement of 5.9% compared to MDNet.

TABLE 3. Results on the UAV123 dataset.

Model	UAV123	
	AUC	Prec
Ours(CPFN)	**0.587**	**0.788**
SiamRPN	0.578	0.769
MDNet	0.528	-
ECO	0.525	0.741
ECO-HC	0.506	0.725
SiamFC	0.498	0.726
SRDCF	0.464	0.676
CFNet	0.436	0.651
SAMF	0.396	0.592
MEEM	0.392	0.627
DSST	0.356	0.586

TABLE 4. Results on the GOT-10k testset.

Model	AO	$SR_{0.5}$	$SR_{0.75}$
Ours(CPFN)	**0.467**	**0.555**	**0.264**
SiamRPN	0.463	0.544	0.257
THOR	0.447	0.538	0.204
SRCRPN	0.451	0.528	0.212
DaSiamRPN	0.444	0.536	0.220
SARRPN	0.426	0.506	0.192
DCANet	0.403	0.466	0.150
SiamFC	0.348	0.353	0.098
C-COT	0.325	0.328	0.107
ECO	0.316	0.309	0.111
CF2	0.315	0.297	0.088

E. RESULTS ON GOT-10k

GOT-10K is a large high-diversity benchmark for generic object tracking in the wild. GOT-10k testset contains 180 video of real-world moving objects. Results of GOT-10k testset need to be uploaded to the official website for analysis. The provided evaluation indicators include average overlap (*AO*) and success rate (*SR*). The *AO* represents the average overlaps between ground-truth boxes and estimated bounding boxes. The $SR_{0.5}$ is the rate of successfully tracked frames that overlap more than 0.5, while $SR_{0.75}$ is rate of successfully tracked frames that overlap more than 0.75. We evaluate CPFN on GOT-10k testset and compare it with tracker with SiamRPN, DaSiamRPN [43], SiamFC, CF2, C-COT, ECO and other baselines or state-of-the art approaches. All the results are provided by the official website of GOT-10K. As shown in Table 4, our tracker ranks 1st in terms of all the indicators. Compared with SiamRPN, CPFN improves the scores by 0.4%, 1.1% and 0.7% for for relatively for *AO*, $SR_{0.5}$ and $SR_{0.75}$. Compared with DaSiamRPN, CPFN improves scores by 2.3%, 1.9% and 4.4% for relatively for *AO*, $SR_{0.5}$ and $SR_{0.75}$.

F. ABLATION STUDY

In this section, we analyze the effects of positive feedback feature channel (PFFC), negative feedback of interference feature channel (IFC), and update strategy (US) on VOT2016 and VOT2018 benchmarks. Table 5 presents the expected average overlap (EAO), accuracy (A), and robustness (R) of VOT2016 and VOT 2018 for each variation. Compared with the baseline tracker on VOT2016 and VOT2018, the additional PFFC method obtains gains (+0.5% and +2.0%), the additional PFFC and IFC method obtains significant gains (+2.7% and 3.8%). When integrating three method, the improvement becomes larger (+4.8% and +5.3% for VOT2016 and VOT2018), which demonstrates the effectiveness of our each method.

TABLE 5. Ablation studies on the VOT2016 and VOT2018 dataset.

PFFC	IFC	US	VOT2016			VOT2018		
			EAO	A	R	EAO	A	R
			0.380	0.614	0.261	0.308	0.578	0.351
√			0.385	0.620	0.247	0.328	0.586	0.342
√	√		0.407	**0.622**	0.219	0.346	**0.589**	0.304
√	√	√	**0.428**	0.620	**0.191**	**0.361**	0.586	**0.281**

TABLE 6. Ablation study of update scheme on the VOT2016 and VOT2018 dataset.

update scheme		US	simple templet update
VOT2016	EAO	0.428	0.386
	A	0.62	0.586
	R	0.191	0.247
VOT2018	EAO	0.361	0.328
	A	0.586	0.549
	R	0.281	0.318

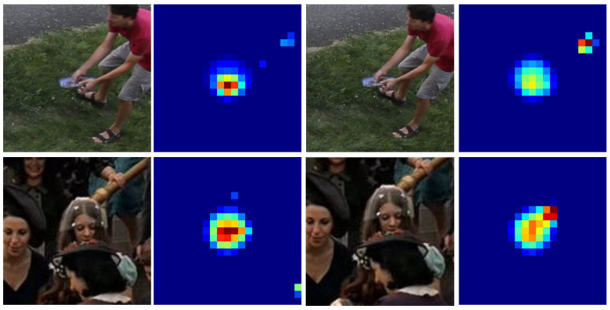

FIGURE 9. The first and third columns are the search areas of CPFN and simple template update schemes in the same video frame. The second column is the classification result of CPFN, and the fourth column is the classification result of simple template update strategy. The channel which locates the maximum value of classification score is the heat map visualized.

In order to verify the adverse effects of simple template update, we replace the update scheme in CPFN, that is, when the classification score is higher than a certain threshold, the following updates are made to the template: $template = template_{old} * \alpha + (1 - \alpha) * template_{new}, (cls > \theta)$. As shown in Table 6, after using a simple template update strategy, the EAO have significantly decreased. In addition, there is a significant decrease in the A metric, indicating that the overlapping effect of the regression box and the ground truth is bad, which shows that the template information is contaminated and the target location cannot be located well. As shown in Figure 9, it can be seen that a simple template update scheme will cause inaccurate positioning.

VI. CONCLUSION

In order to solve the problem of target drift caused by the interference of surrounding objects and the change of target state in the SiamRPN, this paper proposes a combined feature channel scheme and a novel update strategy. By comparing the feature map of each feature channel with a Gaussian kernel, the positive feedback feature channel and the interference feature channel can be selected quickly and effectively. Moreover, a negative feedback excitation for the interference channels and deep features is used more effectively. Gaussian score map is generated by integrating two types of channels. Combining the Gaussian score map with the classification branch of RPN, the network achieves the purpose of enhancing the target response and suppressing the response of similar objects, and further distinguishing the target and the background. Extensive experiments on datasets VOT2016, VOT2018, VOT2019, OTB100, UAV123, LaSOT and GOT-10k show that the CPFN model can further distinguish the background and the target, and outperforms the state-of-the-art methods based on small backbone network in terms of accuracy and tracking speed.

REFERENCES

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

[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] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2544–2550.

[4] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4670–4679.

[5] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1430–1438.

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

[7] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Discriminative scale space tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1561–1575, Aug. 2017.

[8] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 58–66.

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

[10] H. Fan and H. Ling, "Siamese cascaded region proposal networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7952–7961.

[11] H. Guo, Q. Shi, B. Du, L. Zhang, D. Wang, and H. Ding, "Scene-driven multitask parallel attention network for building extraction in high-resolution remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, early access, Aug. 19, 2020, doi: 10.1109/TGRS.2020.3014312.

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

[13] 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.

[14] 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.*, 2015, pp. 597–606.

[15] H. Kashiani, A. Abbas Hamidi Imani, S. Baradaran Shokouhi, and A. Ayatollahi, "Online visual tracking with one-shot context-aware domain adaptation," 2020, *arXiv:2008.09891*. [Online]. Available: http://arxiv.org/abs/2008.09891

[16] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1135–1143.

[17] M. Kristan *et al.*, "The seventh visual object tracking VOT2019 challenge results," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 2206–2241.

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

[19] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4904–4913.

[20] J. Li, E. Chen, Z. Ding, L. Zhu, K. Lu, and H. T. Shen, "Maximum density divergence for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 28, 2020, doi: 10.1109/TPAMI.2020.2991050.

[21] J. Li, M. Jing, K. Lu, L. Zhu, and H. T. Shen, "Locality preserving joint transfer for domain adaptation," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 6103–6115, Dec. 2019.

[22] K. Li, F.-Z. He, and H.-P. Yu, "Robust visual tracking based on convolutional features with illumination and occlusion handing," *J. Comput. Sci. Technol.*, vol. 33, no. 1, pp. 223–236, Jan. 2018.

[23] K. Li, F. He, H. Yu, and X. Chen, "A parallel and robust object tracking approach synthesizing adaptive Bayesian learning and improved incremental subspace learning," *Frontiers Comput. Sci.*, vol. 13, no. 5, pp. 1116–1135, Oct. 2019.

[24] P. Li, B. Chen, W. Ouyang, D. Wang, X. Yang, and H. Lu, "GradNet: Gradient-guided network for visual object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6162–6171.

[25] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1369–1378.

[26] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 254–265.

[27] Z. Liang and J. Shen, "Local semantic siamese networks for fast tracking," *IEEE Trans. Image Process.*, vol. 29, pp. 3351–3364, 2020.

[28] S. Liu, Q. Shi, and L. Zhang, "Few-shot hyperspectral image classification with unknown classes using multitask deep learning," *IEEE Trans. Geosci. Remote Sens.*, early access, Sep. 4, 2020, doi: 10.1109/TGRS.2020.3018879.

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

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

[31] 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. (CVPR)*, Jun. 2016, pp. 4303–4311.

[32] Q. Shi, M. Liu, X. Liu, P. Liu, P. Zhang, J. Yang, and X. Li, "Domain adaption for fine-grained urban village extraction from satellite images," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 8, pp. 1430–1434, Aug. 2020.

[33] Y. Song, C. Ma, X. Wu, L. Gong, L. Bao, W. Zuo, C. Shen, R. W. H. Lau, and M.-H. Yang, "VITAL: VIsual tracking via adversarial learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8990–8999.

[34] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2805–2813.

[35] C. Wang, L. Song, G. Wang, Q. Zhang, and X. Wang, "Multi-scale multi-patch person re-identification with exclusivity regularized softmax," *Neurocomputing*, vol. 382, pp. 64–70, Mar. 2020.

[36] G. Wang, C. Luo, Z. Xiong, and W. Zeng, "SPM-tracker: Series-parallel matching for real-time visual object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3643–3652.

[37] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3119–3127.

[38] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung, "Transferring rich feature hierarchies for robust visual tracking," 2015, *arXiv:1501.04587*. [Online]. Available: http://arxiv.org/abs/1501.04587

[39] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, "Joint group feature selection and discriminative filter learning for robust visual object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7950–7960.

[40] J. Zhang, S. Ma, and S. Sclaroff, "Meem: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 188–203.

[41] 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.

[42] Z. Zhang and H. Peng, "Deeper and wider siamese networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4591–4600.

[43] 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)*, 2018, pp. 101–117.

YUE CHEN received the B.S. degree in information and computing science from the Hunan University of Science and Technology, in 2018. He is currently pursuing the M.S. degree with Zhejiang Normal University. His current research interests include target tracking and deep learning.

XIAOWEI HE (Member, IEEE) is currently a Professor with the College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua, China. His research interests include image and video processing, machine learning, and blockchain.

ZHONGLONG ZHENG (Member, IEEE) received the B.Sc. degree from the China University of Petroleum, China, in 1999, and the Ph.D. degree from Shanghai Jiao Tong University, China, in 2005. He is currently a Full Professor with the College of Mathematics and Computer Science, Zhejiang Normal University, China. His research interests include machine learning, computer vision, and blockchain.

PENGCHENG BIAN received the B.Eng. degree from the Department of Computer Science, Hefei University, China, in 2017. He is currently pursuing the M.S. degree with the Department of Mathematics and Computer Science, Zhejiang Normal University, Jinhua, China. His research interests include deep learning and computer vision.

YI LI received the B.S. degree from the School of Information Science and Technology, Taishan University, in 2018. He is currently pursuing the M.S. degree with Zhejiang Normal University. His current research interests include Object Detection and Deep Learning.