IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 24, 2020, accepted March 18, 2020, date of publication March 20, 2020, date of current version March 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982261

Foreground Information Guidance for Siamese Visual Tracking

DAQUN LI[1,2] **AND YI YU**[1]

[1]Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China

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

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

ABSTRACT Existing Siamese network based trackers are easily disturbed by large deformation, occlusion and distractor objects in the background. By comparing these trackers, we observe that the monotonous positive pairs usually have limited challenging factors (Occlusion, Deformation, etc.), which may make the learned features less robust. In addition, the foreground information of the substantial training data is utilized directly without deeper exploration. Thus, the trackers cannot effectively discriminate the foreground from the semantic backgrounds. In this paper, we focus on modifying the Siamese tracker by enriching the positive pairs and taking further advantage of the foreground information. During the offline training phase, a simple sampling strategy is adopted to enrich the challenging factors in positive pairs, which can effectively enhance the robustness of the tracker. At the same time, we highlight the foreground information by padding the background, and the information is utilized to generate a novel padding loss, which guides the tracker to pay less attention to the distractors in the background. Moreover, an improved feature information fusion is adopted to update the template, so that the tracker can adapt to the drastic appearance changes. Comprehensive experiments on the OTB and the VOT benchmarks demonstrate that our proposed tracker can achieve outstanding performance in both accuracy and robustness.

INDEX TERMS Visual tracking, Siamese network, foreground information, feature information fusion.

I. INTRODUCTION

Visual tracking is one of the most important directions in the field of computer vision. In the most general setting, given an arbitrary target specified by a bounding box, the goal of the visual tracking is to locate the target in the subsequent frames. Although visual tracking has been greatly improved in recent research [1]–[7] and widely used in many applications [8]–[10], it has been regarded as a challenging task due to numerous complex scenes such as occlusion, deformation and background clutters, to name a few.

Recently, Siamese network based trackers [11]–[17] have drawn great attention in the tracking field owing to their balanced speed and accuracy. By defining the visual tracking as a matching problem, Siamese trackers aim to learn a general similarity function offline from substantial training videos. Among these trackers, the SiamFC tracker [11] first utilizes the fully-convolutional structure to achieve the end-to-end training, which allows the tracker can make full use of the substantial offline training data. The GOTURN tracker [13] integrates the regression method into the network, and CFNet [14] introduces the correlation filter for low level convolutional neural networks (CNNs) features to improve the tracking speed. By combining the region proposal network (RPN) with the Siamese network, the SiamRPN tracker [15] achieves a better performance than the above trackers. On the basis of SiamRPN, the DaSiamRPN tracker [16] generates more semantic pairs in the offline training phase and adopts the distractor-aware module to improve the discrimination power of the tracker. The SiamRPN++ [17] successfully trains a ResNet-driven Siamese tracker, which not only further improves the accuracy but also breaks the limitation of the shallow network structures.

Although the aforementioned trackers have obtained outstanding tracking performance, the Siamese network structures still suffer from some limitations. Firstly, during the offline training phase, the challenging factors in the positive

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

pairs are limited. Most Siamese trackers are trained on the ImageNet VID [18], which consists of about 4,000 videos and contains 30 categories. The limited categories of the training datasets are not sufficient to obtain a high-quality and robust tracker. Even though the DaSiamRPN [16] tracker expands the positive pairs by introducing other large-scale detection datasets, due to the long-tail distribution of some challenging factors (Occlusion, Deformation, etc.) [19], these factors may still not be included in those expanded positive pairs. Secondly, Siamese trackers cannot keep their high performance when the backgrounds are cluttered. Foreground information used in most Siamese trackers can effectively discriminate the target from the non-semantic background. However, the semantic backgrounds which are usually considered as distractors are the key to influence the tracking performance. When the backgrounds are cluttered, the bounding box will drift to the distractors, so that the tracker cannot track accurately. Though some recent works [16], [20], [21] aim to address the issue, the robustness is yet to reach a high level. Thirdly, most Siamese trackers cannot update the templates. The constant templates used in Siamese trackers make the methods lose the ability to adapt to the drastic appearance changes. Despite the high processing speed of these trackers, there is still a gap compared with the state-of-the-art tracking approaches.

In this work, we propose a foreground information guidance for Siamese visual tracking (FIGSiam), which overcomes the above limitations and promotes the tracking performance. We focus on modifying the Siamese tracker by enriching the positive pairs and taking further advantage of the foreground information. During the offline training phase, a simple sampling strategy is adopted to enrich the challenging factors in positive pairs, which can effectively enhance the robustness of the tracker. Moreover, we highlight the foreground information by padding the background, and the information is utilized to generate a novel padding loss, which guides the tracker to pay less attention to the distractors in the background. Furthermore, we adopt an improved adaptive feature information fusion to update the template, so that the tracker can adapt to the drastic appearance changes.

In summary, the main contributions of our work are listed below:

- We utilize a simple sampling strategy to enrich the challenging factors in positive pairs, which can effectively enhance the robustness of the tracker.
- We highlight the foreground information by padding the background, and the information is utilized to generate a novel padding loss, which guides the tracker to pay less attention to the distractors in the background.
- We adopt an improved adaptive feature information fusion to update the template, so that the tracker can adapt to the drastic appearance changes.

The rest of the work is organized as follows: We introduce the related works in Section II. In Section III, we present our proposed method. In Section IV, the experiments will be discussed. In Section V, we reach the final conclusions of the paper.

II. RELATED WORK

A. SIAMESE NETWORKS BASED TRACKING

Visual tracking has made astonishing progress in recent years, with the development of various methods. Recently, Siamese network based trackers have drawn great attention in the tracking field owing to their balanced speed and accuracy. Bertinetto *et al.* [11] first propose the fully-convolutional structure to estimate the regional feature similarity between the template and the search region. Tao *et al.* [12] use the Siamese network to offline train a matching function from substantial video sequences. In order to improve the tracking accuracy, the tracker adopts the region of interest pooling (ROI pooling) to deal with the stochastic size of the input, and utilizes the optical flow algorithm to filter the candidates. Held *et al.* [13] propose the GOTURN tracker by using the regression method and training a motion prediction model with the Siamese network. Thus, the tracker can run efficiently at 100 fps. Valmadre *et al.* [14] successfully adopt the Siamese network to learn the representation of the feature for correlation filter, which makes the tracker shallower but more efficient.

Although the above classical Siamese trackers have obtained outstanding tracking performance, the Siamese network structures still suffer from some limitations. In order to overcome the limitation of scale variation, the SiamRPN tracker [15] introduces the RPN into the Siamese network. Thus, the traditional multi-scale test can be removed. For further improving the tracking performance, the SiamRPN++ tracker [17] successfully trains a ResNet-driven Siamese tracker, which not only further improves the accuracy but also breaks the limitation of the shallow network structures. In order to overcome the limitation of background clutter, the DaSiamRPN tracker [16] combines a distractor-aware module with the Siamese network, and expands the positive pairs by introducing other large-scale detection datasets. The TDAT tracker [20] develops a regression loss and a ranking loss to learn target-aware features. The DSiam tracker [21] adopts the feature transformation to suppress the background and deal with the appearance variation. In order to overcome the limitation of template updating, Zhang *et al.* [22] utilize the CNNs called UpdateNet to update the templates. The optimal template can be obtained by fusing the accumulated templates and the template of current frame.

Despite the recent success of the aforementioned trackers, the limitations still need to be further considered. By comparing these trackers, we observe that the monotonous positive pairs usually have limited challenging factors (Occlusion, Deformation, etc.), which may make the learned features less robust. In addition, the foreground information of the substantial training data is utilized directly without deeper exploration. Thus, the trackers cannot effectively discriminate the foreground from the semantic backgrounds. In this

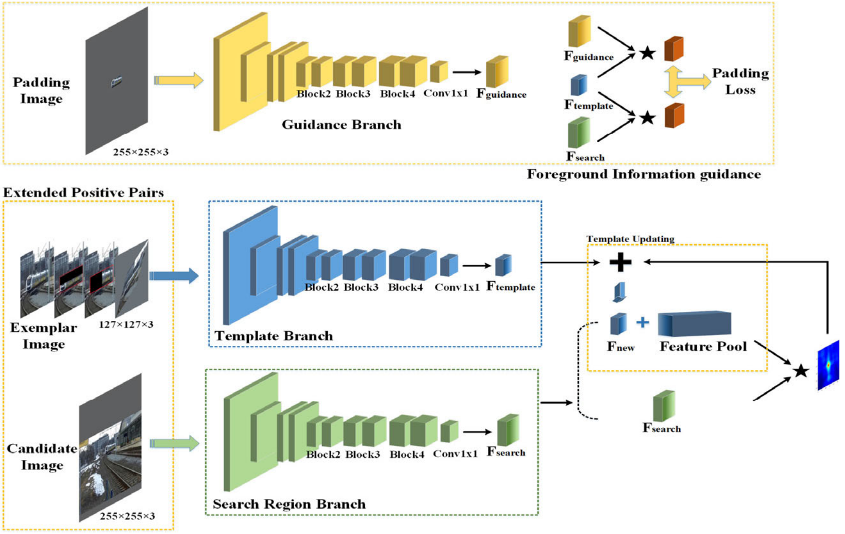


FIGURE 1. **The architecture of the proposed tracker FIGSiam. The Extended Positive Pairs module is adopted to enrich the challenging factors in the positive pairs. The Foreground Information Guidance module is utilized to generate a novel Padding Loss. The Template Updating module is used to update the output tensor of the Template Branch $F_{template}$. F_{search} means the output of the Search Region Branch, $F_{guidance}$ means the output of the Guidance Branch and F_{new} denotes the feature tensor of the tracking result extracted by the Template Branch. Feature Pool is used to store the credible features of the target.**

work, we enrich the positive pairs and take further advantage of the foreground information. The proposed tracker not only overcomes the limitation to a certain extent, but achieves outstanding performance in both accuracy and robustness.

B. TEMPLATE UPDATING

The target always suffers from drastic appearance changes when tracking is on-the-fly, while the fixed template cannot adapt to the changes since it only contains the previous foreground information. On the other hand, if the template is updated at high frequency, the distractor in the background will be introduced to the tracker, and the error will accumulate constantly. Thus, it is a significant research subject to adaptively update the template.

At present, most trackers usually adopt two ways to achieve the template updating: linear interpolation and multi-template updating [23]. Linear interpolation mainly interpolates the last tracking result into the current template linearly, such as BACF [24], CFNet [14], KCF [25], and MOSSE [26]. Although this simple mechanism can enhance the tracking performance to a certain degree, it is insufficient when the target suffers drastic appearance changes given by occlusion, fast motion, or deformation. To address this issue, the multi-template updating is utilized to maintain templates in previous frames. It adopts specific strategies to evaluate the reliability of the template and achieve adaptive updating. Yang and Chan [27] use the Long Short-Term Memory (LSTM) to evaluate the current template during online tracking, which is computed by storing previous templates in memory. Choi *et al.* [28] use the reinforcement learning to select the optimal template stored in the template memory. Yao *et al.* [29] efficiently use the stochastic gradient descent (SGD) offline to learn the updating coefficients for the correlation filter tracker.

Inspired by the above research works, we propose an improved feature information fusion to update the template. We adopt an effective yet simple strategy to obtain the important feature information not only based on the tracking result in each new frame, but also based on the accumulated templates in previous frames. By utilizing the information, we can update the template and make the tracker more adaptive to the drastic appearance changes.

III. METHOD

A. OVERVIEW

The main purpose of this paper is to improve the robustness of the tracker by enriching the positive pairs and take further advantage of the foreground information. As shown in Fig.1, our FIGSiam tracker mainly contains three modules, *i.e.* extended positive pairs, foreground information guidance and template updating. Then, we will briefly introduce these modules respectively.

In the extended positive pairs module, a simple sampling strategy is adopted to enrich the challenging factors in the positive pairs. Firstly, during the offline training phase, we utilize handcrafted occlusion masks which contain 11 different directions to extend the positive pairs. Secondly, the random affine transformation is added into the training. We adopt the rotation and the shear mapping to simulate the deformation of the target. Thus, the datasets can not only be enriched, but the robustness of the tracker in the attribute of occlusion and deformation can also be improved.

In the foreground information guidance module, we take further advantage of the foreground information. The structure of the guidance branch used in this module is the same as that of the template branch and the search region branch. The input is the padding image with the size 255 × 255 × 3. For improving the tracking performance, we use the improved ResNet-18 in our previous work [30] as the embedding function to extract the features. In the guidance branch, we highlight the foreground information by padding the background, and the information is utilized to generate a novel padding loss, which guides the tracker to pay less attention to the distractors in the background. Moreover, the guidance branch is enabled in training phase and disabled in tracking phase.

In the template updating module, we introduce an improved feature information fusion for the proposed tracker. We regard the tracking result of each new frame as the undecided template in tracking phase. The feature pool is utilized to store the credible feature tensors of the results in previous frames, and the undecided template is extracted by the template branch to obtain the new feature tensor. Furthermore, the global average pooling (GAP) is used to reduce the dimensions of the feature tensors, and the correlation between them can also be obtained. By utilizing the above important information, we can update the template and make the tracker more adaptive to the drastic appearance changes.

B. EXTENDED POSITIVE PAIRS

Training data with high quality is crucial for the success of end-to-end learning in Siamese visual tracking. At present, most Siamese trackers are trained on the large-scale detection datasets, such as ImageNet VID [18], COCO [31], or Youtube-BB [32]. However, the limited categories of the positive pairs are not sufficient to obtain a high-quality and robust tracker. Even though some trackers expand the positive pairs by combining some large-scale detection datasets, due to the long-tail distribution of some challenging factors (Occlusion, Deformation, etc.), these factors may still not be included in those expanded positive pairs. Thus, we adopt a simple sampling strategy to enrich the challenging factors in the positive pairs.

Firstly, we utilize the handcrafted occlusion masks to extend the positive pairs. During the offline training phase, we regard each exemplar image in positive pairs as the sample need to be extended. More specifically, given the target in exemplar image, we obtain the ground truth with size W × H. Then, we can generate masks with fixed size W/2 × H/2 in pre-designed directions, and drop out the values of each mask in the corresponding spatial location. The 11 directions are shown in Fig.2.

Secondly, we use the rotation and the shear mapping to simulate the deformation of the target. Given the exemplar image, we rotate the image in the range of $\theta = \pm 30°$. At the same time, we also perform the shear mapping in both X direction and Y direction. The range of each direction can be denoted as $\phi = \pm 25°$ and $\psi = \pm 25°$ respectively. All the transformations are combined randomly, and we will select 6 samples at random for training. Some representative samples can be seen in Fig.3.

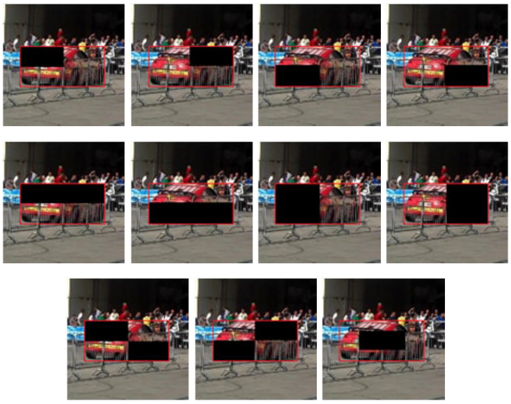


FIGURE 2. 11 directions of the handcrafted occlusion masks.

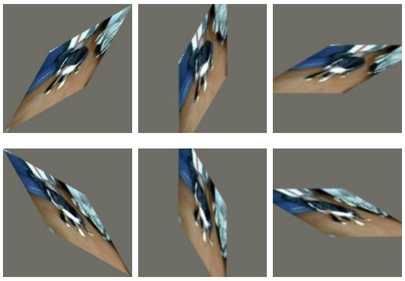


FIGURE 3. Representative samples of the random affine transformation.

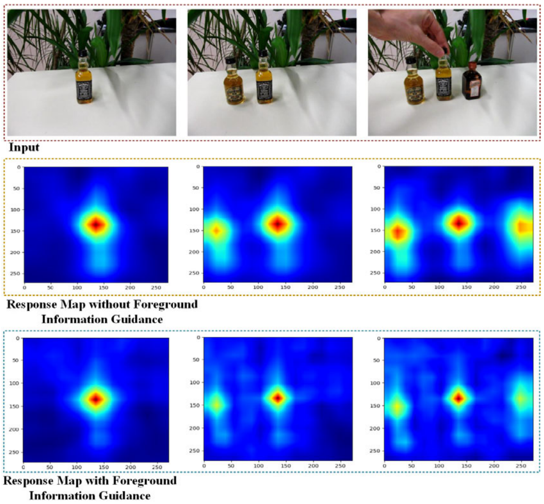


FIGURE 4. Visualization of response maps w or w/o the foreground information guidance.

C. FOREGROUND INFORMATION GUIDANCE

The extended positive pairs in the last subsection can improve the robustness and the discrimination power of the tracker in the offline training stage. However, it is still hard to discriminate the foreground from the semantic backgrounds like Fig.4. To address this issue, we take further advantage of the foreground information and generate a novel padding loss, which guides the tracker to pay less attention to the distractor in the background.

In SiamFC [11], the AlexNet [33] is used as the embedding function φ to extract the features of both template branch and the search region branch. Meanwhile, the tracker learns a similarity metric *R(z,x)* to compare the exemplar image z and the candidate image x:

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

where $*$ denotes the cross correlation between the feature maps, and $b\mathbf{1}$ denotes the equal bias in every location. Furthermore, the tracker defines the loss function $loss_{training}$ as follows:

$$loss_{training}(k, s) = \frac{1}{|M|} \sum_{p \in M} l(k\,[p]\,, s\,[p]) \tag{2}$$

Here, M denotes the score map after cross correlation, $k\,[p] \in \{+1, -1\}$ and $s\,[p]$ indicate the label and the real-valued score for each position $p \in M$ in the score map, l is the logistic loss:

$$l\,(k, s) = log\,(1 + exp\,(-ks)) \tag{3}$$

Inspired by SiamFC, we use the improved ResNet-18 in our previous work [30] as the embedding function to extract the features. However, we cut off most of the convolution kernels with 1×1 size, and only maintain only one at the end of the network to reduce the dimension. Moreover, in order to lighten the heavy computational burden, the channel of the final output tensor is adjusted to 256, and the size is cropped to retain the central region (8×8 for template branch, 24×24 for search region branch). For breaking the restriction of the strict translation invariance, we also adopt the spatial aware sampling strategy which is first proposed in [17].

To take further advantage of the foreground information, we add a guidance branch into the framework and combine a novel padding loss with the logistic loss used in SiamFC. As can be seen from Fig.1, the input of the guidance branch is a padding image with size $255 \times 255 \times 3$. We highlight the foreground information by padding the background with the mean value of the whole image:

$$Im\,age = \begin{cases} Im\,age(a, b) & (a, b) \in fg \\ Mean(Image) & (a, b) \in bg \end{cases} \tag{4}$$

Here, *a* and *b denote the* a^{th} row and b^{th} column of the image, fg means the foreground, bg means the background. Given the feature tensors extracted by the improved ResNet-18, we calculate the Depthwise Cross Correlation (DW-XCorr) [17] between the output tensors of template branch and search region branch, as well as the output tensors of template branch and guidance branch. The result can be denoted as S and S'. The padding loss can be computed as:

$$loss_{padding} = \sum_{c}^{C} \sum_{a}^{H} \sum_{b}^{W} \left\| \frac{S_{c,a,b}}{max(\|S\|_2)} - \frac{S'_{c,a,b}}{max(\|S'\|_2)} \right\|_2 \tag{5}$$

where C, H and W represent the channel, the height and the width of the tensor respectively.

By using this padding loss, the tracker can be guided to filter the features of background and improve the discrimination power of the tracker. The final loss can be formulated as:

$$loss_{final} = (1 - \lambda) loss_{training} + \lambda loss_{padding} \tag{6}$$

where λ means the weight of the padding loss.

D. TEMPLATE UPDATING

The constant templates used in most Siamese trackers make the methods lose the ability to adapt to the drastic appearance changes. In order to address this issue, we add an improved feature information fusion into the tracker.

In the initialization stage, a feature pool is utilized to store the credible feature tensors $F_{pool} = \left\{F_1, F_2, F_3, \ldots, F_j | j \leq N\right\}$ which are extracted from the tracking results, and N denotes the upper limit of the pool. Given the tracking result of a new frame, we first put it into the template branch and regard the extracted tensor as the new feature tensor F_{new} with size $8 \times 8 \times 256$. Moreover, for lightening the computational burden, we use the global average pooling (GAP) to reduce the dimension of F_{new} and features in F_{pool}. Thus, the size can be compressed to $1 \times 1 \times 256$ (${F_{new}}'$ and ${F_j}'$). In the next step, we use the L2 normalization to normalize the features, and calculate the matching score between the ${F_{new}}'$ and the ${F_j}'$:

$$S^{j}_{matching} = \left| (F'_{new})^{T} F'_{j} \right| \tag{7}$$

Then, we can adaptively fuse the features as follows:

$$F_{final} = \varepsilon F_{new} + (1 - \varepsilon) \sum_{j=1} F_j \delta_j \tag{8}$$

Here, $\delta_j = \frac{S^{j}_{matching}}{\sum\limits_{j=1} S^{j}_{matching}}$ means the adaptive weight of each feature tensor in the feature pool, ε denotes the weight of F_{new}.

In our module, when the number of the credible feature tensors is less than N, we will directly add F_{new} into the feature pool if $Mean(S^{j}_{matching})$ is higher than the threshold Th_{update}. However, when the number is more than N, we will use F_{new} to replace the feature tensor with the lowest matching score in the feature pool.

IV. EXPERIMENTS

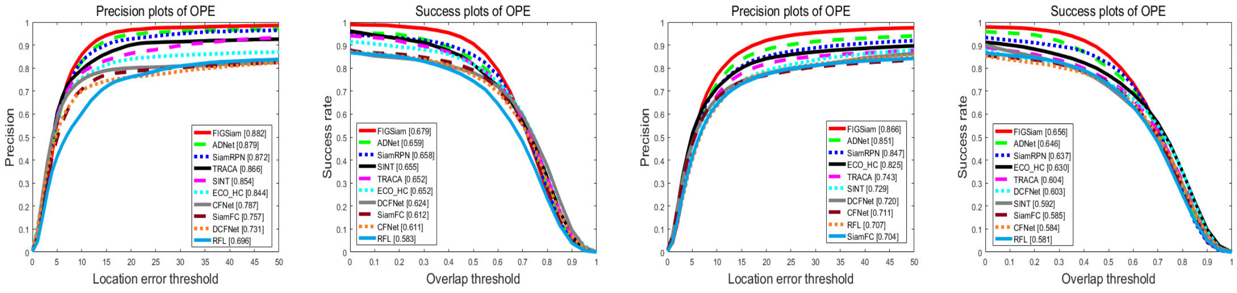
A. EXPERIMENTAL DETAILS

1) ENVIRONMENT

Our experiments are implemented using PyTorch-0.4.1 on PC with Intel i7-9800X CPU (3.80GHz), 64GB RAM and NVDIA TITAN RTX GPU. The average testing speed on short-term benchmarks can reach 32 fps.

2) TRAINING

The improved ResNet-18 network is trained on ImageNet VID [18] and COCO [31] datasets. Moreover, we augment



(a) OTB2013 (b) OTB100

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

the positive pairs as described in Section "*METHOD*". The momentum of 0.9 and the weight decay of 0.0005 are used. The template branch, the search region branch and the guidance branch are trained for 50 epochs, and the learning rate is decayed in log space from 0.01 to 0.0001. In training stage, the inputs of three branches are images with the size $127 \times 127 \times 3$, $255 \times 255 \times 3$, $255 \times 255 \times 3$ respectively. In testing stage, the guidance branch will be cut.

3) OTHER SETTINGS

The weight of the padding loss λ is set to $\lambda = 0.5$. For the improved information fusion, we set the upper limit $N = 20$, the weight $\varepsilon = 0.6$ and the threshold $Th_{update} = 0.65$. All the above parameters which represent the best performance are selected by extensive experiments on short-term benchmarks.

B. OTB DATASET

1) OVERALL PERFORMANCE COMPARISON

For testing the performance of our proposed FIGSiam tracker, we adopt the standardized OTB [34], [35] datasets to achieve the state-of-the-art comparison. We choose Siamese network based trackers (SiamFC [11], SINT [12], CFNet [14], SiamRPN [15], DCFNet [36]), recurrent neural network (RNN) based tracker RFL [37], correlation filter based trackers (ECO_HC [38], TRACA [39]) and deep reinforcement learning based tracker ADNet [40] to implement the comparison. Moreover, the success plots and the precision plots are used to present the results of the evaluation which are shown in Fig.5.

As can be seen from Fig.5 (a), FIGSiam is able to produce leading results in both precision and the area under the curve (AUC) score. Compared with the SiamRPN tracker, FIGSiam improves 1% in precision and 2.1% in overlap on OTB2013 dataset. Meanwhile, as shown in Fig.5 (b), the precision of FIGSiam on OTB100 dataset is 0.866 and the AUC score is 0.656, which improves 1.9% compared with SiamRPN respectively.

Compared with other Siamese trackers, the superiority of the FIGSiam tracker is more obvious. FIGSiam adopts the handcrafted occlusion masks and the random affine transformation to enrich the challenging factors in positive pairs. The extended positive pairs can effectively enhance the robustness of the tracker. At the same time, the foreground information is highlighted by padding the background, and the information is utilized to generate a novel padding loss, which guides the tracker to pay less attention to the distractor in the background. Moreover, the constant templates used in other Siamese trackers make the methods lose the ability to adapt to the drastic appearance changes. Unlike these trackers, FIGSiam introduces an improved feature information fusion to the framework, so that the tracker can overcome the restriction and perform well when the target suffers drastic appearance changes given by occlusion, fast motion, or deformation.

Compared with the correlation filter based trackers (ECO_HC and TRACA), FIGSiam still achieves an outstanding performance. By using this padding loss, the tracker can be guided to filter the features of background and improve the discrimination power of the tracker. Furthermore, the adaptive template updating strategy is also the key to compete with these trackers in complex scenes

2) ATTRIBUTE COMPARISON

For further analyzing the performances of the trackers, we compare these trackers by using 11 annotated attributes in OTB100 dataset, and the success plots are shown in Fig.6.

As can be seen from Fig.6, FIGSiam can achieve leading results in most challenging scenes. However, compared with the SiamRPN and ADNet, it attains much inferior tracking performance in the attributes of low resolution and scale variation. The ultimate reason lies in that FIGSiam utilizes the strategy used in SiamFC to deal with the scale variation. It only adopts the limited scale factors to find out the optimum size of the tracking result. In addition, the target which only contains no more than 400 pixels may lose the information when extracted by the deep network.

3) FAILURE ANALYSIS

Although our proposed FIGSiam can achieve leading results in most challenging scenes, it cannot perform well in all of the OTB sequences. Some failure cases on Matrix and Soccer sequences are shown in Fig.7.

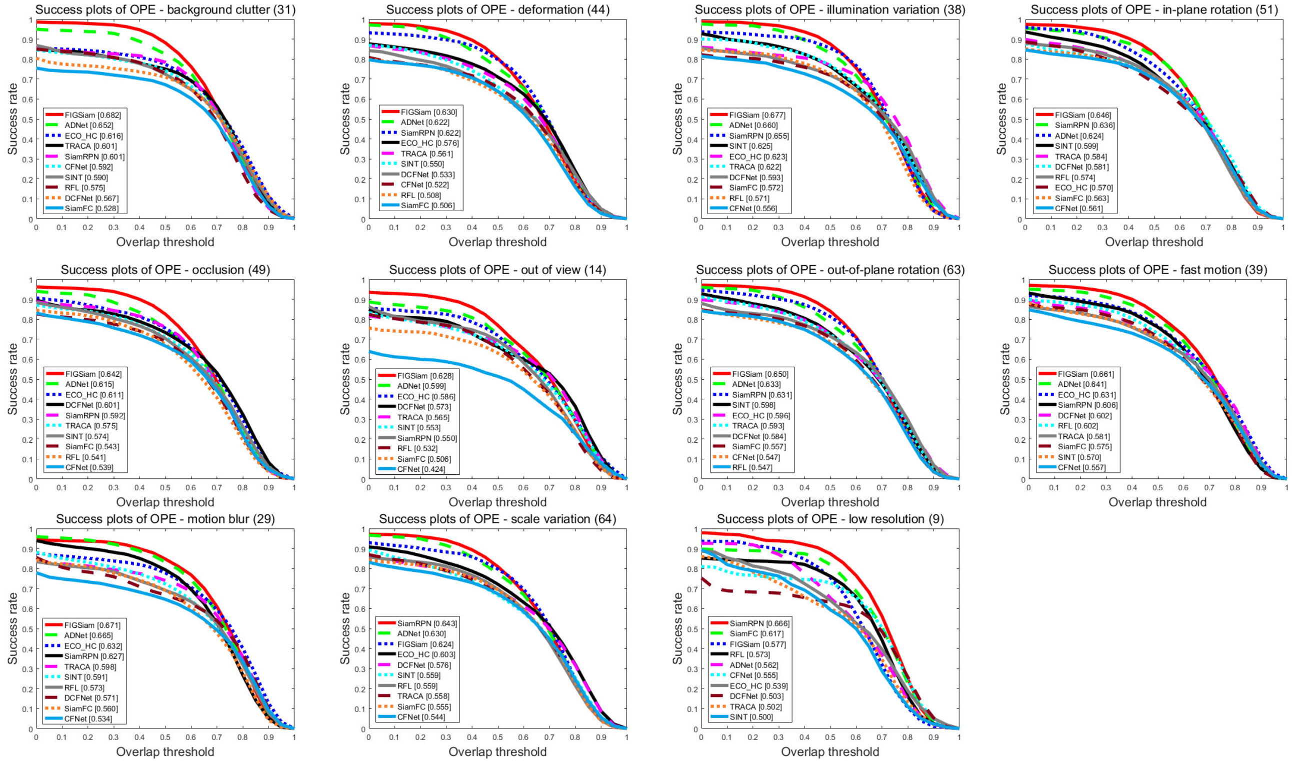


FIGURE 6. **Attribute-based success plots on the OTB100 dataset. The later digits of the title denote the number of the sequences with that attribute.**

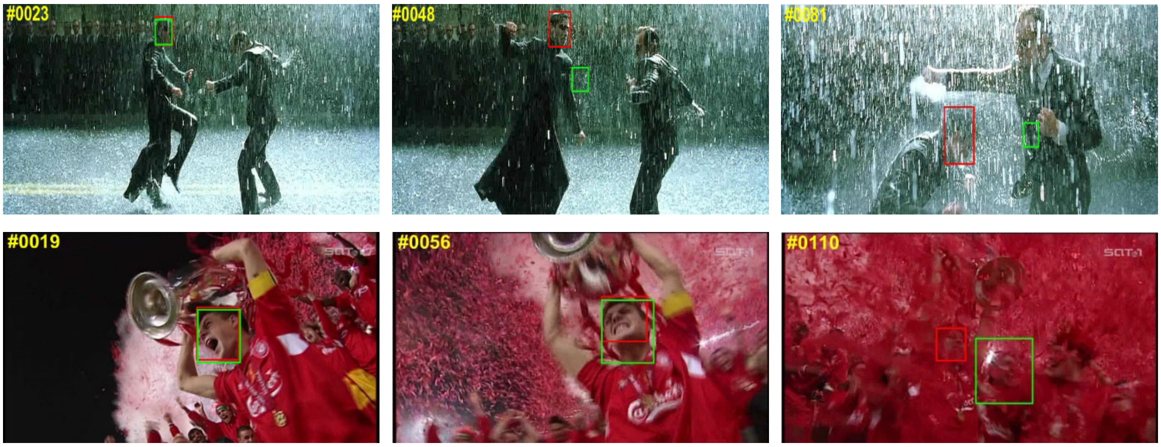


FIGURE 7. **Failure cases on Matrix and Soccer sequences. The red boxes are the ground truth and the green ones are results of FIGSiam.**

In Matrix sequence, our FIGSiam tracker can perform well in the beginning, such as the 23-*th* frame. However, when the target suffers more drastic appearance changes given by fast motion and the deformation, the tracker will lose the target and drift to the background. Though the handcrafted template updating strategy can enhance the tracker to deal with these challenges, it is not powerful enough to face complex scenes.

In Soccer sequence, our FIGSiam tracker cannot adapt to the scale variation and the out-of-plane rotation since the 56-*th* frame. Moreover, when the background becomes more cluttered and the target suffers severe occlusion in the 110-*th* frame, the tracker fails to locate the target. Thus, how to improve the template updating strategy and optimize the network to face more serious challenges still need to be further researched.

C. VOT2016

The VOT2016 [41] dataset consists of 60 sequences. The performance of the tracker is evaluated by using accuracy (average overlap while tracking successfully) and robustness

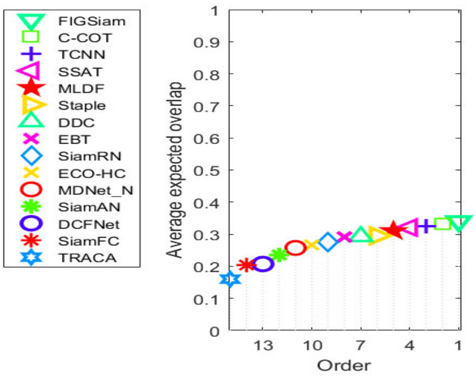


FIGURE 8. Expected average overlap performance on VOT2016.

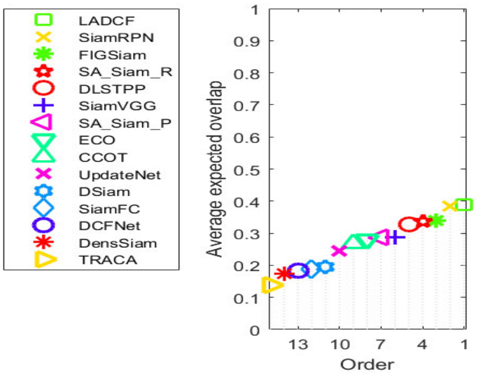


FIGURE 9. Expected average overlap performance on VOT2018.

TABLE 1. Detail information about the trackers.

Tracker	EAO	Accuracy	Failures
TCNN [43]	0.3249	0.55	17.94
ECO-HC [38]	0.2665	0.51	21.40
SSAT [41]	0.3207	0.57	19.27
MLDF [41]	0.3106	0.49	15.04
DDC [41]	0.2929	0.53	20.98
SiamRN [41]	0.2766	0.55	24.00
MDNet N [44]	0.2572	0.54	21.08
SiamAN [41]	0.2352	0.53	29.80
Staple [41]	0.2952	0.54	23.89
C-COT [45]	0.3310	0.53	16.58
TRACA [39]	0.1599	0.46	37.95
DCFNet [36]	0.2071	0.49	24.94
SiamFC [11]	0.2039	0.51	24.67
EBT [46]	**0**.2913	**0**.45	15.19
FIGSiam	**0.3402**	**0.58**	20.14

TABLE 2. Detail information about the trackers.

Tracker	EAO	Accuracy	Failures
DLSTPP [42]	0.3273	**0.542**	**13.695**
ECO [38]	0.2809	0.476	17.663
SiamFC [11]	0.1875	0.498	34.026
SiamVGG [42]	0.2872	0.527	20.453
SA_Siam_P [42]	0.2863	0.521	19.467
SA_Siam_R [47]	0.3374	**0.557**	15.555
TRACA [39]	0.1385	0.431	53.678
LADCF [48]	**0.3889**	0.502	**8.774**
C-COT [45]	0.2674	0.485	20.414
DSiam [21]	0.1952	0.506	40.087
DCFNet [36]	0.1827	0.465	35.202
UpdateNet [22]	0.2442	0.510	26.872
DensSiam [49]	0.1732	0.456	44.448
SiamRPN [15]	**0.3837**	**0.580**	17.661
FIGSiam	**0.3389**	0.501	**14.530**

(failure times). Furthermore, the overall performance of the tracker can also be evaluated by utilizing EAO which takes both accuracy and robustness into account. In our work, we compare our FIGSiam tracker with 14 excellent trackers. Fig.8 shows the EAO ranking and TABEL I lists the details about the Accuracy, the Robustness and the EAO (red, blue and green denote 1st, 2nd and 3rd respectively).

As can be seen from TABEL I, Although FIGSiam cannot perform well in Failure, it is able to rank 1st in both accuracy and EAO. Thus, we believe that the improvements used in our tracker can achieve an outstanding performance.

D. VOT2018

We evaluate our FIGSiam tracker on the VOT2018 [42] dataset in comparison with 14 state-of-the-art methods. The VOT2018 dataset is developed by replacing some least challenging videos in VOT2016 with some more difficult ones. It still contains 60 sequences and evaluates the trackers by using Accuracy, Robustness and EAO. The EAO ranking and the results are shown in Fig.9 and TABLE 2 respectively.

Among these trackers, the LADCF tracker achieves the best Failures and EAO. The SiamRPN tracker ranks 1st in accuracy and 2nd in EAO. Though our FIGSiam tracker can perform better than most Siamese trackers, it attains much inferior accuracy and robustness compared to the outstanding trackers. We believe the ultimate reason lies in that the FIGSiam uses the handcrafted template updating strategy, and the network needs to be further improved to face more serious challenges, such as scale variation and low resolution.

E. ABLATION STUDY

In order to verify the contributions of each module in our proposed tracker, we evaluate three variations of our method. The evaluative results include AUC score in OTB100 [35] and EAO in VOT2018 [42].

TABLE 3. Ablation analysis on OTB100 and VOT2018.

Component	SiamRPN	FIGSiam		
Extended positive pairs?		✓	✓	✓
Foreground information guidance?			✓	✓
Template updating?				✓
AUC in OTB100	0.637	0.641	0.652	0.656
EAO in VOT2018	0.3889	0.2976	0.3361	0.3389

As shown in TABLE 3, we use the SiamRPN as the baseline method. In OTB100, the AUC score increases to 0.641 from 0.637 when the extended positive pairs module is added in training. Similarly, when the foreground information guidance module is adopted in training, the performance increases by near 1.5%. When the template updating module is utilized in inference, the performance can increase by near 2%. Though SiamRPN has a better performance than FIGSiam in VOT2018, we can reduce the gap by adopting our proposed modules. The EAO criterion increases to 0.3361 from 0.2976 when the foreground information guidance module is added. Moreover, when the template updating module is adopted, the FIGSiam tracker can increase by 4%, which represents the best tracking performance in VOT2018.

V. CONCLUSION

In this paper, we focus on modifying the Siamese tracker by enriching the positive pairs and taking further advantage of the foreground information. A simple sampling strategy is adopted to enrich the challenging factors in positive pairs. Meanwhile, we highlight the foreground information by padding the background, and the information is utilized to generate a novel padding loss. By using this padding loss, the tracker can be guided to filter the features of background and improve the discrimination power of the tracker. In addition, an improved feature information fusion is adopted to update the template, so that the tracker can adapt to the drastic appearance changes. Comprehensive experiments on the OTB and the VOT benchmarks demonstrate that our proposed tracker can achieve excellent performance in both accuracy and robustness. However, the proposed tracker still attains much inferior accuracy and robustness compared to some outstanding trackers. In the next step, we will improve the template updating strategy and optimize the network to face more serious challenges. Moreover, we will also pay more attention to the background appearance information rather than ignore it.

ACKNOWLEDGMENT

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

REFERENCES

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

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

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

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

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

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

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

[8] L. Liu, J. Xing, H. Ai, and X. Ruan, "Hand posture recognition using finger geometric feature," in *Proc. Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 565–568.

[9] J. Xing, H. Ai, and S. Lao, "Multiple human tracking based on multi-view upper-body detection and discriminative learning," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 1698–1701.

[10] G. Zhang and P. A. Vela, "Good features to track for visual SLAM," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1373–1382.

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

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

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

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

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

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

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

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

[19] X. Wang, C. Li, B. Luo, and J. Tang, "SINT++: Robust visual tracking via adversarial positive instance generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4864–4873.

[20] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," Apr. 2019, *arXiv:1904.01772*. [Online]. Available: http://arxiv.org/abs/1904.01772

[21] 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. (ICCV)*, Oct. 2017, pp. 1781–1789.

[22] L. Zhang, A. Gonzalez-Garcia, J. van de Weijer, M. Danelljan, and F. S. Khan, "Learning the model update for siamese trackers," Aug. 2019, *arXiv:1908.00855*. [Online]. Available: http://arxiv.org/abs/1908.00855

[23] Y. Zha, M. Wu, Z. Qiu, S. Dong, F. Yang, and P. Zhang, "Distractor-aware visual tracking by online Siamese network," *IEEE Access*, vol. 7, pp. 89777–89788, 2019.

[24] 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. 1144–1152.

[25] 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.

[26] 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. (CVPR)*, Jun. 2010, pp. 2544–2550.

[27] T. Yang and A. B. Chan, "Learning dynamic memory networks for object tracking," Mar. 2018, *arXiv:1803.07268*. [Online]. Available: http://arxiv.org/abs/1803.07268

[28] J. Choi, J. Kwon, and K. Mu Lee, "Real-time visual tracking by deep reinforced decision making," Feb. 2017, *arXiv:1702.06291*. [Online]. Available: http://arxiv.org/abs/1702.06291

[29] Y. Yao, X. Wu, L. Zhang, S. Shan, and W. Zuo, "Joint representation and truncated inference learning for correlation filter based tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 560–575.

[30] D. Li, X. Wang, and Y. Yu, "Siamese visual tracking with deep features and robust feature fusion," *IEEE Access*, vol. 8, pp. 3863–3874, 2020.

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

[32] E. Real, J. Shlens, S. Mazzocchi, X. Pan, and V. Vanhoucke, "YouTube-BoundingBoxes: A large high-precision human-annotated data set for object detection in video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7464–7473.

[33] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2012, vol. 25, no. 2, pp. 1097–1105.

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

[35] 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.

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

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

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

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

[40] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1349–1358.

[41] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. Cehovin, T. Vojir, G. Häger, A. Lukežič, G. Fernández, A. Gupta, A. Petrosino, A. Memarmoghadam, A. Garcia-Martin, and A. S. Montero, "The visual object tracking VOT2016 challenge results," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 777–823.

[42] M. Kristan *et al.*, "The sixth visual object tracking VOT2018 challenge results," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 3–53.

[43] H. Nam, M. Baek, and B. Han, "Modeling and propagating CNNs in a tree structure for visual tracking," Aug. 2016, *arXiv:1608.07242*. [Online]. Available: http://arxiv.org/abs/1608.07242

[44] 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.

[45] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 621–629.

[46] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 943–951.

[47] A. He, C. Luo, X. Tian, and W. Zeng, "Towards a better match in Siamese network based visual object tracker," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 132–147.

[48] T. Xu, Z.-H. Feng, X.-J. Wu, and J. Kittler, "Learning adaptive discriminative correlation filters via temporal consistency preserving spatial feature selection for robust visual tracking," Jul. 2018, *arXiv:1807.11348*. [Online]. Available: http://arxiv.org/abs/1807.11348

[49] M. H. Abdelpakey, M. S. Shehata, and M. M. Mohamed, "DensSiam: End-to-end densely-Siamese network with self-attention model for object tracking," Sep. 2018, *arXiv:1809.02714*. [Online]. Available: http://arxiv.org/abs/1809.02714

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

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