IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 4 November 2021, accepted 25 November 2021, date of publication 8 December 2021, date of current version 26 August 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3134066

Siamese Visual Tracking With Residual Fusion Learning

XINGLONG SUN [1,2], **GUANGLIANG HAN** [1], **AND LIHONG GUO** [1]

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

Corresponding author: Guangliang Han (hangl@ciomp.ac.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61401425 and Grant 61602432.

ABSTRACT Multi-stage feature fusion is pretty effective for deep Siamese trackers to promote tracking performance. Unfortunately, conventional fusion approaches, such as weighted average, are so simple that they are inappropriate to combine the features with diverse characteristics. In addition, the fusion module is generally optimized along with Siamese network module, which may result in the performance degradation of the whole tracker. In this paper, we propose a novel feature fusion network for Siamese tracker by exploiting the expression capacity of residual fusion learning (SiamRFL). Specifically, the network employs the deep-layer features as direct input to semantically recognize the object from background, and refines the object state with local detail patterns by exploring the shallow-layer features through residual channel. The classification and the regression features can be fused respectively by deploying multiple fusion units. To avoid the degradation problem, we also present an ensemble training framework for our tracker, in which different loss functions are introduced to individually optimize the Siamese and the fusion modules. Compared to the baseline SiamRPN++ tracker, the proposed tracker achieves favorable gains by 0.696→0.709, 0.285→0.308, 0.603→0.624, 0.496→0.520 and 0.517→0.559 on OTB100, VOT2019, UAV123, LaSOT and GOT10k datasets, outperforming other approaches by an obvious margin.

INDEX TERMS Visual tracking, Siamese network, feature fusion, residual learning, ensemble training.

I. INTRODUCTION

Visual tracking is one of the most fundamental research directions in computer vision, which has a capacity to infer the state of an arbitrary object in a sequence, only with its initial state in the first frame as reference. The technique is required by various visual issues, such as visual surveillance [1], robotics [2], human computer interaction [3] and augmented reality [4]. Despite great progress has been realized, most of trackers still struggle with several challenging factors, such as background clutters, occlusion, illumination variation, etc.

With the development of Convolutional Neural Networks, a few more efficient tracking paradigms are gradually presented to address the above difficult factors, such as Siamese network. The network aims to match the features of template and search region patches to predict the object state, which has a significant advantage in speed and precision. Following the seminal works of SiamFC [5] and SINT [6], massive efforts are performed to furtherly promote tracking performance. Some [7], [8] expected to improve the quality of feature representation by introducing effective backbone networks, like ResNet [9] and GoogleNet [10], etc. While others devoted themselves to completing more reliable decisions by designing powerful matching modules, i.e., Region Proposal Networks (RPN) [11], [12] and Anchor-free networks [13], [14]. In addition, a number of training approaches [15], [16] and online update strategies [17], [18] were explored to achieve better tracking results.

In one neural network, the modules in different depths vary in abstract levels and receptive fields, so they are able to learn features with diverse attributes. The features from shallow layers consist of abundant local detail patterns which are valuable for perceiving the location variations of the object, while deep-layer features with high-level semantic information are important to discriminate the object from background. In this circumstance, most of the previous Siamese trackers [7], [13], [19] try to fuse multi-layer features to benefit from their complementary attributes. However, existed fusion approach, i.e., weighted average, is very simple

The associate editor coordinating the review of this manuscript and approving it for publication was Mehul S. Raval.

and still suffers from several drawbacks. Firstly, the method is so simple that it cannot aggregate features in an adaptive way, even though the aggregated weights are trainable. Since it ignores the attribute difference between multiple-layer features and treats them equally, these features maybe disturb each other during fusion that makes trackers fail to adapt to the drastic appearance variations of object. Moreover, conventional works usually train the whole network entirely, in which both Siamese network module and fusion module are optimized by only one loss function. This manner is insufficient to ensure the training quality of every module, and degrades the performance of Siamese trackers.

In this paper, we propose a novel feature fusion framework for Siamese trackers based on the attributes of different-stage features, which is comprised of multiple residual units. When tracking an object, an appropriate tracking strategy is that the tracker first explores abstract semantic patterns to discriminate the tracked object from a global view, and then utilizes spatial detailed patterns to refine the state of object. Inspired by this idea, the features from deep layers are adopted as the direct component of our residual unit to coarsely identify the object from background, while the shallow-layer features are inputted into residual channel to eliminate the prediction deviations of direct channel. There exist two sub-networks in the proposed fusion module to combine the classification and the regression features respectively, and each sub-network is constructed by cascading the fusion units. Through introducing the fusion architecture, a Siamese tracker is able to predict the object state in a coarse-to-fine manner.

Furthermore, an ensemble training approach is presented for our tracker to avoid the performance degradation in the testing phase. Concretely, several basic losses are adopted to optimize the Siamese network module including backbone and decision networks, and a fusion loss is utilized to only train the fusion module. By decomposing the optimizations of diverse blocks, our presented tracker would be trained with high-quality. Figure 1 shows several representative visual response maps, illustrating that all decision modules become more efficient under our optimizing scheme, and the proposed fusion strategy produces more robust and reliable tracking responses.

The major contributions presented in the work mainly consist of the following points.

1. We propose a novel feature fusion scheme by exploiting residual learning, which has an ability to take full advantage of the attribute information of multi-layer features, and generate more reliable tracking results.
2. An ensemble training approach is designed to optimize our Siamese tracker. By using multiple loss functions to separately train different network modules, it is very effective to promote the training quality of the proposed tracker.
3. Extensive experiments on some challenging benchmark datasets manifest that the proposed tracker is superior to some state-of-the-art trackers with very promising performance.

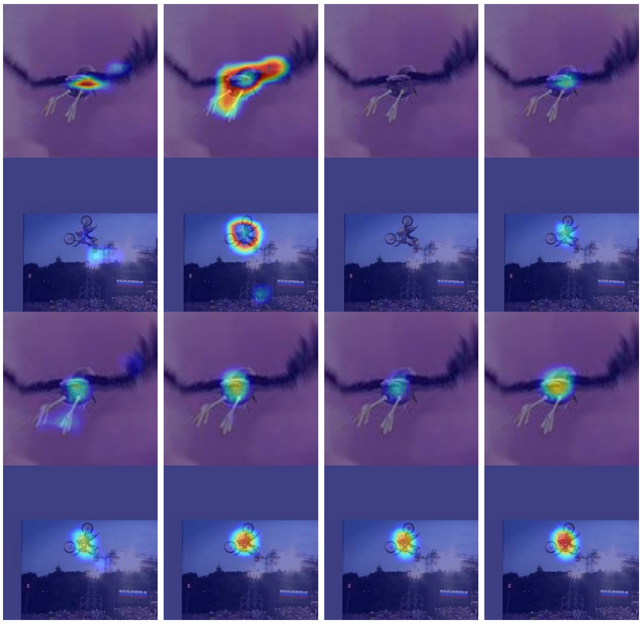

FIGURE 1. Classification resulting maps of SiamRPN++ [7] and our SiamRFL on some typical videos. The first two rows show the results of SiamRPN++, while the rest of rows illustrate our responses. The columns from 1st to 4th express the resulting maps output by three RPN modules (*conv-3*, *conv-4* and *conv-5*) and fusion module, respectively.

The rest of the paper is organized as follows. We first review the related works in Section II, and then describe the Siamese tracker with our presented residual fusion network as well as its training approach in Section III. The experiments and results on several latest datasets are analyzed in Section IV, in which our tracker is compared with most of state-of-the-art methods. At last, the paper will be concluded in Section V.

II. RELATED WORKS

In this section, we just briefly review the recent researches related to our work, including Siamese trackers, feature fusion approaches and loss functions. More elaborate introduction about visual tracking can be found in some review literatures [20], [21].

A. SIAMESE TRACKERS

Siamese networks serve as a popular tracking paradigm which have received intensive attentions in the last few years. Inspired by the pioneering work of SiamFC [5] that presented a cross-correlation layer to compare the features of template and search patches, abundant strategies were exploited to lift the potential of the networks. Among these, a very representative direction is to study how to predict the object state. Concretely, SiamRPN [11] combined Siamese network with Region Proposal Network [22] to parallelly perform object-background classification and bounding box regression, realizing high-speed and impressive tracking. Following the instance, several more complicated and successful structures were developed, such as SPM [23] and C-RPN [12]. Furtherly, anchor-free networks were also explored to avoid complex hyperparameters in RPN modules. SiamBAN [13] presented a box adaptive network without anchors, which can detect the bounding box of an object in a per-pixel manner. SiamFC++ [24] described a set of guidelines for the object state prediction, while Ocean [14] designed an object-aware anchor-free network for tracking. Moreover, other decision blocks such as segmentation network [25] and corner detection network [26] were proved to be powerful, too. Another important evolution for Siamese networks is to introduce deeper backbone network for more abstract feature representation. SiamRPN++ [7] used spatial aware sampling to overcome the negative influence of padding operation, and employed ResNet-50 [9] as backbone. SiamDW [8] straightly proposed a novel residual unit without padding. In addition to design network model, both adversarial learning [16] and distractor-aware sampling [15] were utilized to improve training quality, while some online update methods [17], [18], [27] were adopted to help trackers to achieve satisfactory performance.

B. FEATURE FUSION APPROACHES

Feature aggregation is a valuable way for lifting the tracking performance of neural networks, which has been widely applied in previous works. A popular solution is to transmit multi-layer convolutional features into Discriminative Correlation Filters (DCF) [28]–[30], which were able to combine these features to form a kernel to recognize the object. In addition, FCNT [31] presented a switch mechanism to alternately select the features from diverse stages for tracking. Nevertheless, these methods are all artificially designed, which could not benefit from large-scale training datasets, as well as satisfy challenging tracking requirements. In Siamese networks, it is more meaningful to aggregate multi-stage features when utilizing deeper backbone networks such as ResNet, since the abstract levels and receptive fields varies a lot [7]. As a result, SiamRPN++ [7], SiamBAN [13] and some of the rest Siamese trackers [19], [32] attempted to accumulate the tracking responses computed on diverse-layer features using fixed weight ratios. However, the linear average strategy is so simple that trackers were incapacity of taking full advantage of the features with diverse attributes, even though the weights are trainable. The drawback would limit the role of feature fusion to some extent. In contrast to these traditional approaches, this paper proposes a more powerful fusion scheme, which can combine the low-level detail and the high-level semantic information in a more adaptive way by exploiting residual learning.

C. TRAINING LOSS FUNCTIONS

Loss function plays a vital role to guide the optimization of neural networks, and a variety of losses have been proposed to train Siamese networks. In several initial studies, i.e., SiamFC [5] and CFNet [17], a simple classification loss was presented to generate the similarity confidence map. Then, Dong *et al.* [33] described a triplet loss to find the relative relationship among exemplars, positive instances and negative instances. For some state-of-the-art Siamese networks, such as SiamRPN [11] and SiamRPN++ [7], both classification and regression losses were required to discriminate the object and predict its location state. As a result, they usually accumulated a classification loss and a regression loss as the final training loss. Although these losses are mature and effective, they are not suitable for optimizing our proposed tracker. The core reason is that all these adopted only one loss to train all network modules, which cannot endow every network module with different capability. PG-Net [32] put forward to a multi-stage loss function, where multiple sub-losses were introduced to train the corresponding decision modules, while a fusion loss was adopted for the whole network. The loss is more specific, but it still can't separate the training procedure of Siamese network module and fusion module completely. In this case, we need to design a novel optimization scheme to train the proposed tracker more efficiently.

III. SIAMRFL TRACKER

In this section, we describe the proposed SiamRFL tracker in detail. After giving the overview of the overall architecture, we introduce the baseline SiamRPN++ tracker [7] and present our residual fusion network. Next, we analyze the fatal drawbacks of conventional training way, and illustrate our ensemble training method for model optimization. Last of all, the implementation details about offline training and online testing are explained.

A. OVERVIEW

The architecture of the proposed tracker is depicted in Figure 2. Concretely, the tracker first extracts the template and the search region features with a weight-shared backbone network, and then matches their features in different stages using three RPN blocks. Subsequently, multi-layer output response features are aggregated by the residual fusion network, which consists of multiple residual fusion units (RF Unit) to fuse the classification and the regression features, respectively. The fusing results would be adopted to predict the final state of object. In offline training phase, the Siamese network module, i.e., the backbone as well as Region Proposal Networks and the fusion module are optimized with diverse loss functions, which is very productive to improve the performance of our tracker.

B. SIAMRPN++ TRACKER

Siamese networks generally infer the state of object through comparing the candidate samples in search region x with the initial template z, which can be formulated as

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

in which, φ represents the weight-shared backbone network for feature extraction, while G indicates the similarity

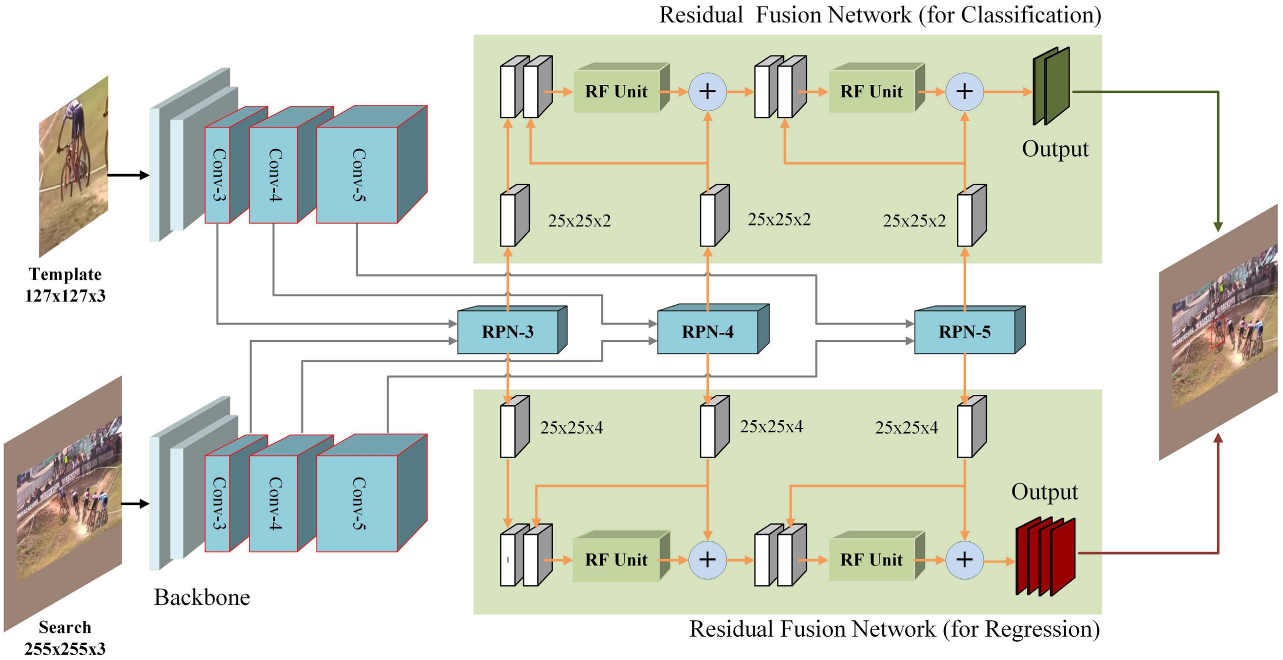

FIGURE 2. Illustration of the proposed framework, consisting of the weight-shared backbone, Region Proposal Networks [7] and Residual Fusion Network. The presented fusion network is comprised by two subnetworks, which can combine the classification and the regression outputs of multiple RPN modules, respectively.

matching module which is used to find the most similar candidate sample with template. b is a bias factor and f denotes the matching results of all candidate samples.

Considering previous works, SiamRPN++ [7] is an important development in field of Siamese visual tracking, which exploits deeper backbone module and aggregates features from multiple stages to find the tracked object. Due to more powerful feature expression, this work could produce very promising tracking results. To describe our feature fusion network and validate its effectiveness, we take the tracker as the baseline, and their main modules, i.e., backbone network and Region Proposal Network, are introduced as follows.

1) BACKBONE

SiamRPN++ has declared that Siamese networks can benefit from more abstracting feature representation, and thus employs ResNet-50 [9] as the backbone. Besides, it adjusts the backbone with several extra trails to make it more appropriate for tracking. Specifically, the sampling strides in the fourth and the fifth residual blocks, i.e., *conv-4* and *conv-5* blocks, are first reduced to 1 pixel to improve the dimensions, while dilated convolution is introduced into these blocks to maintain the receptive fields. To boost tracking ability using features with different attributes, this tracker takes advantage of the last three residual blocks to output features, in which an additional 1×1 convolutional layer is appended to each of block end to align the channels to 256. For template samples, only the features in central 7×7 regions are used to express the objects.

2) REGION PROPOSAL NETWORK

Region Proposal Network is a typical anchor-based decision block. It is proposed for object detection [22], but has gradually become popular in visual tracking domain due to the advantage of prediction precision. There are two different task branches in the block, i.e., a classification branch for identifying the object from background as well as a regression branch for finding the bounding box of object. After adjusting input features, a depth-wise cross-correlation layer is first used in each branch to match a pair of input features. Then, a decision head is constructed to finish object classification or bounding-box regression. In SiamRPN++, three RPN blocks are employed corresponded to the output layers of backbone, whose function can be formulated as

$$\begin{aligned} C_i &= H_i^{cls}\left(a_i^{cls}\left(\varphi_i(z)\right) * \beta_i^{cls}\left(\varphi_i(x)\right)\right) \\ L_i &= H_i^{loc}\left(a_i^{loc}\left(\varphi_i(z)\right) * \beta_i^{loc}\left(\varphi_i(x)\right)\right) \end{aligned} \tag{2}$$

where, $\varphi_i(z)$ and $\varphi_i(x)$ are the template and the search region features, respectively. a or β denotes a 1×1 convolutional layer to adjust the features. $i \in [3, 4, 5]$ depicts diverse output stages. $*$ denotes the cross-correlation operation, while H represents a classification or regression head. C_i and L_i are the classification and the regression results of different layers, respectively.

C. RESIDUAL FUSION NETWORK

Since the above RPN blocks can finish object state prediction using the features with different characteristics, more

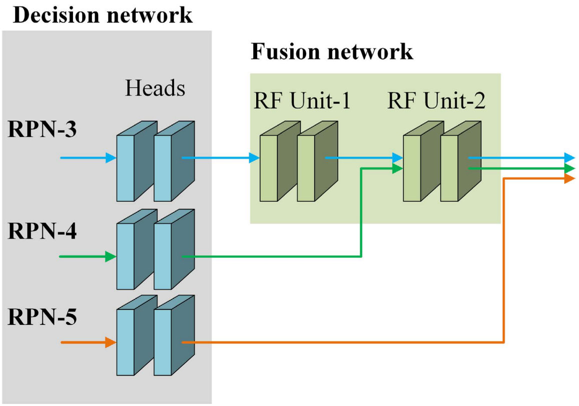

FIGURE 3. Instance of the depth displacement between multiple decision branches. The decision depths of diverse stages are inconsistent once the fusion modules are mistaken as the decision layers, degrading the tracking performance severely.

precise and reliable tracking results will be produced if we combine the response outputs of these blocks in a proper way. The feature expression gradually becomes more abstract with the increasement of the network depths. As a result, the deep-layer features that encode more high-level semantic patterns are suitable for discriminating the object from background globally, while the features provided by shallow layers should be used to refine the tracking results of deep-layer blocks with massive local detail patterns. We observe that the residual learning framework presented in ResNet [9] is very suitable for combining these features, which constructs two independent branches to learning feature representation. Inspired by the issue, we propose a residual fusion network to utilize multi-layer features, named as RFNet.

The network is composed of some residual fusion units, each of which consists of two cascading 1 × 1 convolutional layers. Specifically, the first one compresses the channels of features in half, and the second just adjusts the features without reducing the quantity of channels. A Rectified Linear Unit (RELU) layer is inserted between two convolutional layers to enhance nonlinearity. The features from two diverse stages are required simultaneously by the unit to learn how to aggregate them. In consequence, these features are first concatenated and then inputted into the unit for forward propagation. We accumulate the results with the original deep-layer features to remove the tracking errors of deep-layer modules based on residual learning, which can be formulated as

$$y_r = y_j + R\left(y_i, y_j\right) \tag{3}$$

where, y_i and y_j indicate the features from the shallow and the deep layers, respectively. R depicts the residual fusion unit, while y_r is the fusion result.

In our framework, there are two subnetworks to aggregate the classification and the regression features, respectively. Every subnetwork is comprised of two residual units, which is able to combine the features from three stages. The features from the first two stages are transmitted into the first unit, whose outputs are adopted as the shallow-layer inputs of the last unit. In one subnetwork, the last unit produces the final results of feature fusion:

$$\begin{aligned} C_f &= C_{i+2} + R_s^{cls}\left(C_{i+1} + R_d^{cls}\left(C_i, C_{i+1}\right), C_{i+2}\right) \\ L_f &= L_{i+2} + R_s^{loc}\left(L_{i+1} + R_d^{loc}\left(L_i, L_{i+1}\right), L_{i+2}\right) \end{aligned} \tag{4}$$

in which, R_d and R_s represent the first and the second residual units in a subnetwork, while $i = 3$ denotes the first output stage. C_f and L_f denote the fusion results of classification and regression features, respectively.

In reality, multi-layer feature fusion is a kind of ensemble learning technique, for which one of the most important issues is to design an ensemble module to combine several weak sub-learners into a stronger learner. The technique has been widely discussed and proved to be effective in some previous trackers [34], [35]. For a Siamese network, every decision block can be regarded as a sub-learner, while the fusion approach plays the role of ensemble module. In this view, it is easy to observe that previous fusion methods [19], [32] are too simple to adaptively integrate sub-learners and maximize the advantage of ensemble learning. In contrast, the proposed fusion network is presented based on analyzing the characteristics of each sub-learner, and has an ability to benefit from the training on large-scale image datasets. Therefore, it can accomplish more efficient feature aggregation.

D. ENSEMBLE TRAINING WITH MULTIPLE LOSSES

At present, Siamese networks are generally optimized under a standard training framework, in which only one loss function is used to train the whole network model. However, the tracking performance of our proposed tracker will degrade severely if we follow the traditional training route. The core reason is that one loss function is insufficient to guide all network modules to master the corresponding capabilities. For instance, the decision and the fusion modules are directly cascaded in our Siamese tracker. If there is only one loss for training them, the optimizer may regard them as one functional block, and deliver them with the uniform capability. A possible extreme situation is that the fusion module is mistaken as a part of decision layers, which learns how to predict the object state rather than how to fuse the multi-layer features. Moreover, the depths of diverse decision blocks are unbalanced in this condition, as displayed in Figure 3, which may result in the further reduction of tracking quality.

Analyzing this problem with ensemble learning, we discover that all sub-learners, i.e., RPN blocks, and the ensemble module, i.e., fusion module are synchronously optimized using only one loss function in conventional training paradigm. This manner is inappropriate since it cannot ensure the basic performance of sub-learners and the validity of ensemble learning. To yield the problem, we present an ensemble training framework for our Siamese tracker, as shown in Figure 4. In the framework, every RPN block and its corresponding feature extraction layers are individually optimized by one basic loss function, and a fusion loss function is adopted to only optimize the proposed residual fusion module.

1) BASIC LOSS

The role of basic loss functions is to guide the sub-learners to learn how to track an object, so there are multiple basic losses corresponding to diverse sub-learners. In practice, we introduce the training loss presented in SiamRPN++ [7] as the basic loss function, which consists of a classification loss for identifying the object and a regression loss for estimating the bounding box of object. One RPN block and its feature extraction layers are optimized with the loss

$$\mathcal{L}_i = \mathcal{L}_{cls}\left(C_i, \ell_{cls}\right) + \lambda \cdot \mathcal{L}_{reg}\left(L_i, \ell_{reg}\right) \tag{5}$$

in which, C_i and L_i denote the classification and the regression results in diverse stages of $i \in [3, 4, 5]$, respectively. $\mathcal{L}_{cls}$ is the Cross Entropy Loss for classification, and $\mathcal{L}_{reg}$ is the standard smooth $L1$ Loss for regression. ℓ_{cls} represents the binary label of classification, while ℓ_{reg} depicts the ground-truth bounding box of object. λ denotes a weight factor for balancing two kinds of losses. Then, the basic losses of all stages are aggregated

$$\mathcal{L}_m = \sum_{i=3}^{5} \mathcal{L}_i \tag{6}$$

where $\mathcal{L}_m$ denotes the aggregated result of multiple basic losses. We could complete the optimization of backbone network and all Region Proposal Networks through backward propagating the gradient of the loss.

2) FUSION LOSS

In addition to train sub-learners with the basic losses, an extra loss function is required to guide the fusion module to combine the decision results of sub-learners. Keeping consistent with the training process of sub-learners, we optimize the residual fusion network with the same loss

$$\mathcal{L}_f = \mathcal{L}_{cls}\left(C_f, \ell_{cls}\right) + \lambda \cdot \mathcal{L}_{reg}\left(L_f, \ell_{reg}\right) \tag{7}$$

where C_f and L_f represent the classification and the regression fusing results output from fusion network, respectively.

During offline training, all network modules are optimized jointly. Specifically, two different optimizers are constructed to train Siamese network module and fusion module, respectively. In every batch, we extract several sample pairs of templates and search regions, and then forward propagate them to compute the aggregated basic loss and the fusion loss. Next, we backward propagate the gradients of basic loss and use the first optimizer to train backbone and RPN blocks. The gradients of the fusion loss are backward propagated by the other optimizer to train the residual fusion network. By combining two kinds of losses, the whole network is trained in an end-to-end manner.

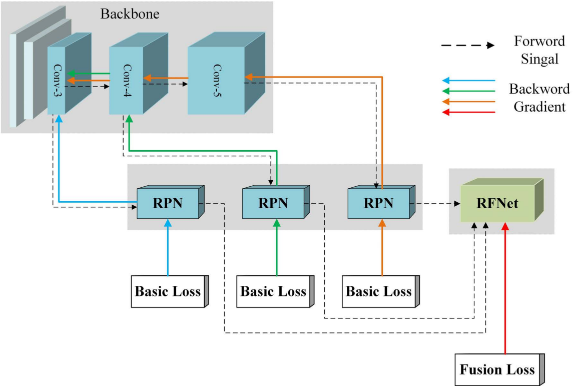

FIGURE 4. Framework of the proposed ensemble training method. Multiple basic losses are adopted to train the RPN blocks and the corresponding backbone layers, while the residual fusion network is optimized using an independent fusion loss.

E. IMPLEMENTATION DETAILS

1) TRAINING

The proposed Siamese network is optimized on the training datasets of ImageNet VID [36], YouTube-BoundingBoxes [37], COCO [38], ImageNet DET [36], LaSOT [39] and GOT10k [40]. We extract a pair of template and search region samples from different frames of a video sequence or a still image with diverse data augmentations, where the sizes of object template and search region patches are set to 127 and 255, respectively. The anchor boxes in RPN blocks are deployed according to the way described in [11]. An anchor would be labelled as positive sample if its IOU ratio with ground-truth is larger than 0.6, while it would be viewed as negative sample if the IOU ratio is lower than 0.3. In one training image pair, we only extract 16 positive and 32 negative samples for network optimization.

After initializing the backbone module with the parameters pretrained on ImageNet dataset [36], we optimize our network using Stochastic Gradient Descent (SGD) method with a weight decay of 0.0005 and a momentum of 0.9. The network is trained 20 epochs with a minibatch of 32, and one million sample pairs are utilized in each epoch. We use a warm-up learning rate for network optimization. Concretely, the learning rate increases from 0.001 to 0.005 in the first 5 epochs, and decays from 0.005 to 0.00005 in the last 15 epochs. Moreover, the first two residual blocks of backbone network are frozen throughout the training, and only the rest of residual blocks are optimized in the last 10 epochs. The learning rate of backbone is smaller 10 times than other network modules. The hyperparameters λ of losses in Eq.5 and Eq.7 are set to 1.2. Our work is performed using PyTorch on a computer with two NVIDIA Titan Xp GPUs. Because we need two optimizers to train the network iteratively, more time is required to make the model converge. The whole training phase takes about 6 days.

2) INFERENCE

Following some previous works [7], we extract the template features using backbone network only in the initial frame, and don't perform update during the tracking process for stability. In each subsequent frame, we extract the search region sample based on the object state in the previous frame, and compare its features with template features. After aggregating the response maps of multiple RPN blocks with the proposed residual fusion network, cosine window penalty and scale change penalty are adopted to re-rank the classification scores of all anchors [11]. The anchor with the highest classification score is selected to regress the bounding box of object. The target size is changed by linear interpolation to maintain the shape changing smoothly. The hyperparameters in the above penalty and linear interpolation operations are automatically computed using the tracking toolkit [13]. The classification and regression results are displayed in Figure 5, where we find that the proposed tracker can provide very accurate and robust tracking results through adaptively fusing multi-layer features.

IV. EXPERIMENTS AND DISCUSSION

To evaluate the performance of the proposed Siamese tracker, we conduct extensively experiments on several public popular benchmark datasets, including OTB-2015 [41], VOT-2019 [42], UAV123 [43], LaSOT [39] and GOT-10k [40]. Our tracker is first compared with some state-of-the-art trackers to highlight its superiority, where the comparison results with other Siamese trackers manifest the advantage of our fusion scheme. Besides, we perform the ablation experiments on LaSOT dataset to show the role of each contribution in our method. In all experiments, the evaluation protocols presented by the above benchmarks are followed rigorously. In the experiments, our tracker runs at a speed of over 40 Frames-Per-Second (FPS), which is very close with the speed of SiamRPN++.

A. COMPARISON WITH THE STATE-OF-THE-ART TRACKERS

1) OTB-100

Online Tracking Benchmark is classic benchmark for visual tracking, and the latest version, i.e., OTB-100 [41] consists of 100 fully-annotated video sequences. These sequences cover 11 kinds of challenging attributes, like background clutter, motion blur, occlusion, etc. Both center location error and overlap ratio are used to evaluate the performance of trackers in the standard protocol. Concretely, center location error indicates the relative distance between the predicted location and ground-truth center, and Precision metric could be furtherly computed by counting the percentage of frames where center location errors are within a given threshold. Overlap ratio measures the Intersection over Union (IoU) ratios of the predicted and ground-truth bounding boxes, where Success metric is used to represent the percentage of images where overlap ratios are larger than a given threshold. We conduct the evaluation in the One-Pass Evaluation (OPE) formulation.

We compare our tracker with twelve state-of-the-art trackers: TransT [44], SiamBAN [13], SiamR-CNN [45], SiamCAR [19], SiamRPN++ [7], SiamRPN [11], UDT [46], DIMP [47], ATOM [48], ECO [30], CREST [49] and MDNet [50]. To be specific, the first six trackers belong to Siamese tracking frameworks, while others are discriminated trackers. The overall comparison results of success and precision plots are displayed in Figure 6. It is worth noticing that the proposed tracker achieves the best performance on both Success and Precision metrics. Compared to the baseline SiamRPN++ tracker, our SiamRFL framework gains a 1.3% improvement on Success with an AUC score of 0.709. For the second-ranked SiamR-CNN tracker in terms of Success score, our method outperforms it by 2.8% on Precision. Among these comparison algorithms, SiamBAN and SiamCAR also take the ResNet-50 as backbone and output convolutional features from the last three residual blocks. It can be seen that our SiamRFL is superior to them because of the feature fusion ability.

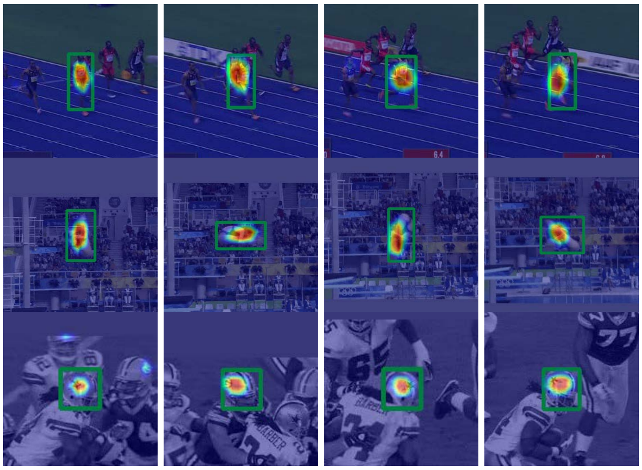

FIGURE 5. Tracking results of our proposed SiamRFL tracker on several typical sequences. Each object is annotated by the response heatmap produced by classification branch and the bounding box output from regression branch.

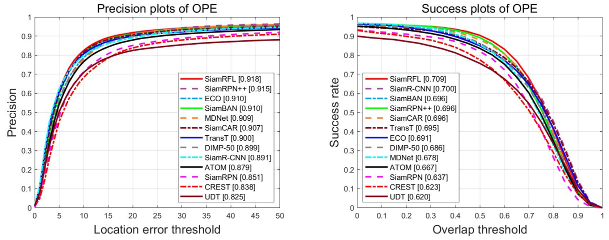

FIGURE 6. Precision and success plots of OPE for all trackers on OTB-100. These trackers are ranked according to the performance score. The performance score of precession plot is at error threshold of 20 pixels, while the performance score of success plot is the value of area under curve (AUC).

To analyze the performance of all trackers more carefully, we also give success and precision plots in multiple challenging attributes, as displayed in Figure 7 and Figure 8. The results manifest that our tracker realizes very satisfactory performance in these attributes. Especially in the attributes of Illumination Variation (IV), Deformation (DEF) and Out-of-Plane Rotation (OPR), the proposed method ranks first on both Success and Precision. For the Success

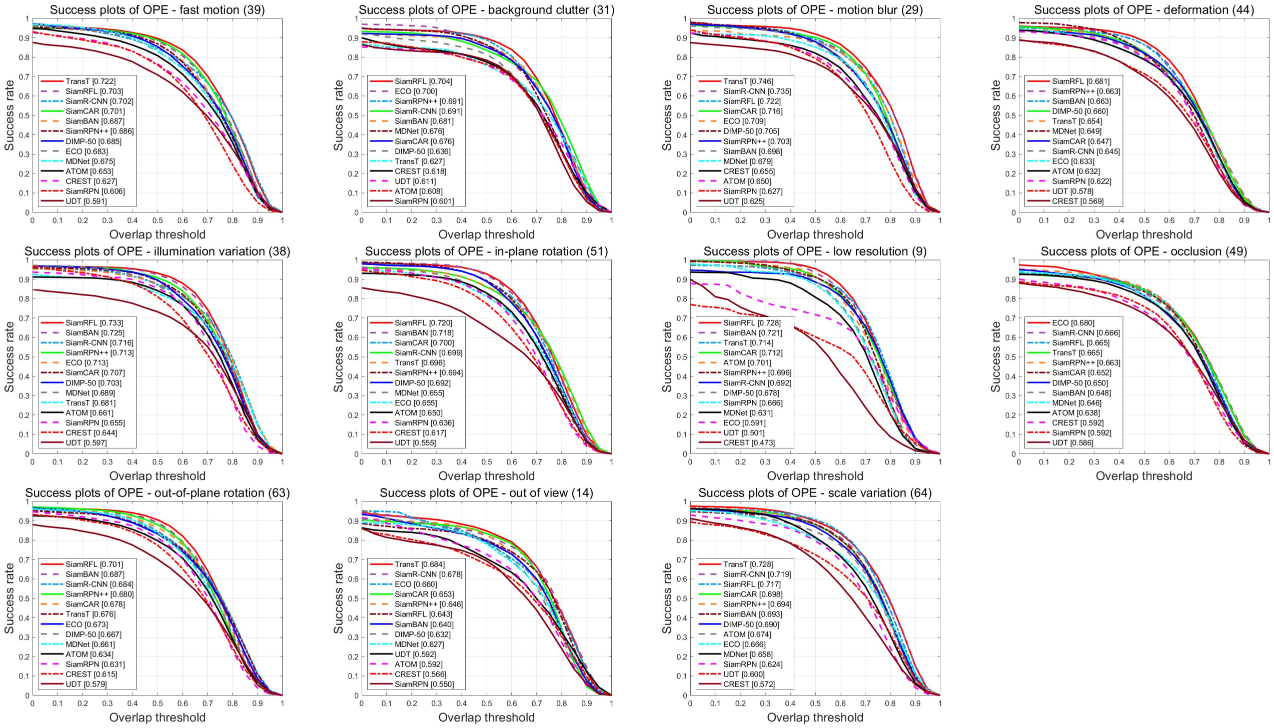

FIGURE 7. **Success plots of OPE for different attributes on OTB-100. The number in the parenthesis denotes the number of sequences within the attribute. These trackers are ranked according to the performance scores of success.**

score, our approach exceeds the second-ranked by 1.8% in DEF attribute and 1.4% in OPR attribute. Compared with the SiamRPN++, SiamRFL obtains more than 1.0% gains in several diverse attributes, including Fast Motion (FM), Motion Blur (MB), Low Resolution (LR), Scale Variation (SV) and so on. These results demonstrate that SiamRFL tracker has an ability to adapt to all kinds of complex appearance variations. This is because the proposed fusion network can aggregate the multi-layer features with diverse attributes more effectively, which helps the tracker to complete robust object classification and accurate object location.

2) VOT-2019

Visual Object Tracking challenge 2019 Dataset (VOT-2019) [42] is a popular benchmark to test online model-free single object trackers, containing 60 video sequences. We conduct the comparison experiments on the dataset with 10 top participants including MemDTC [51], SA-Siam [52], Siam-CRF [42], SPM [23], SiamRPN++ [7], SiamMask [25], ARTCS [42], SiamDW [8], ATOM [48] and CLNet [53]. Following the official evaluation protocol, the trackers would be reset with ground-truths when tracking failures occur. The tracking performance is measured by three metrics: Accuracy (average overlap on successful tracking periods), Robust (failure times) and EAO (Expected Average Overlap). The tracking results are reported in Figure 9 and Table 1, which testify that the proposed tracker performs better than most of compared trackers in term of EAO. Among these, only CLNet slightly outperforms our SiamRFL, which explored a compact latent network to capture the sequence-specific features for fast adjustment. Compared with the SiamRPN++, our tracker yields a relative gain of 2.3%. In addition, our tracker achieves the top-ranked performance on Accuracy.

TABLE 1. **Comparison results on the VOT-2019 Dataset. The best three results are highlighted in red, blue and green fonts.**

Trackers	Robustness↓	Accuracy↑	EAO↑
MemDTC [51]	0.587	0.485	0.228
SA-Siam [52]	0.507	0.562	0.252
Siam-CRF [42]	0.346	0.549	0.262
SPM [23]	0.507	0.577	0.275
SiamRPN++ [7]	0.482	0.599	0.285
SiamMask [25]	0.461	0.594	0.287
ARTCS [42]	0.482	0.602	0.287
SiamDW [8]	0.467	0.600	0.299
ATOM [48]	0.411	0.602	0.301
CLNet [53]	0.461	0.606	0.313
SiamRFL	0.467	0.618	0.308

3) UAV123

UAV123 [43] dataset consists of 123 aerial videos captured from the low-attitude UAV platform, whose average length is about 915 frames. It is pretty challenging to track the object in the dataset due to frequent distractors, such as fast motion, scale change, illumination variation, occlusion, etc. We compare our SiamRFL tracker with several recently proposed methods and present the success and precision

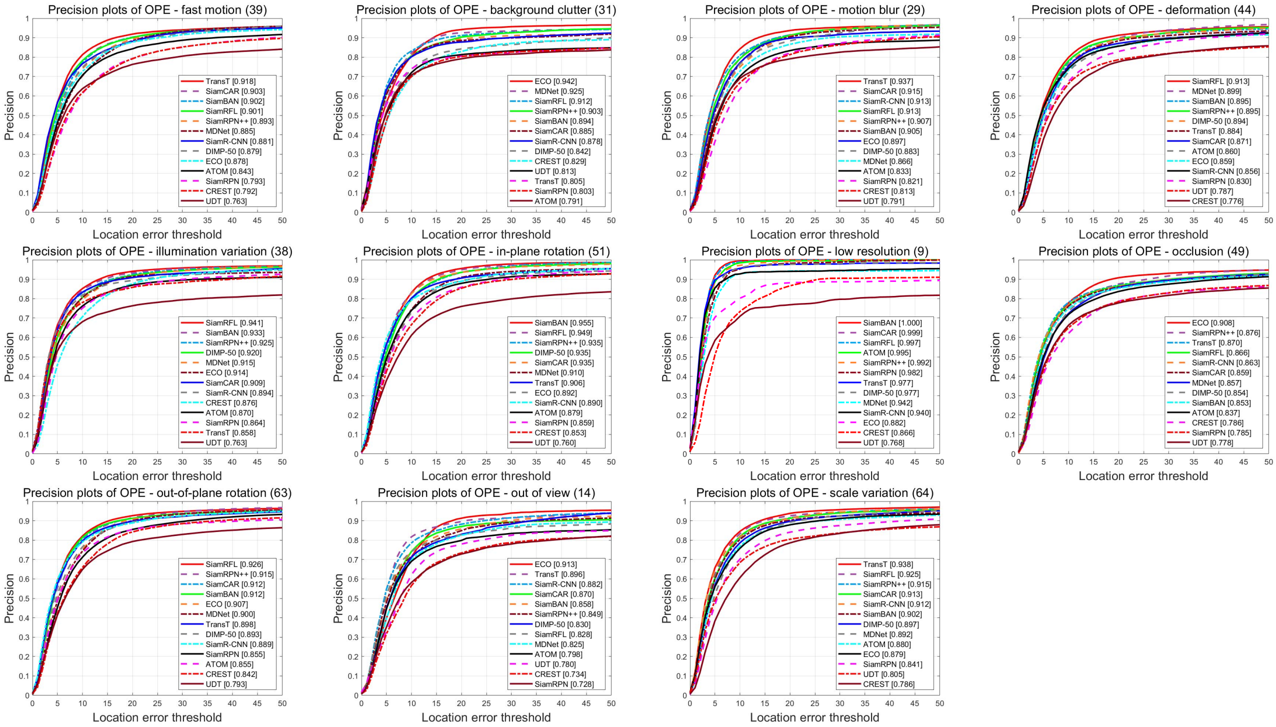

FIGURE 8. Precision plots of OPE for different attributes on OTB-100. The number in the parenthesis denotes the number of the sequences within the attribute. These trackers are ranked according to the performance scores of precision.

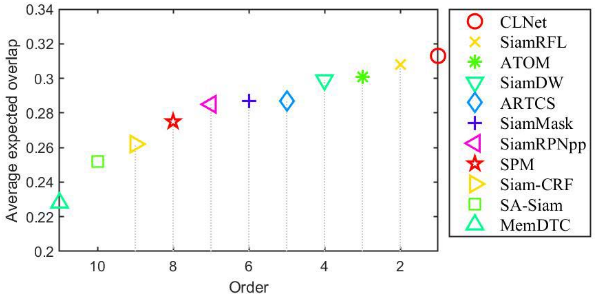

FIGURE 9. Expected averaged overlap (EAO) scores of all methods on VOT-2019.

plots in Figure 10. The proposed tracker exhibits satisfactory results and surpasses most of recent remarkable approaches on both metrics. The only exception in the comparison results is the SiamBAN tracker [13], which is top-performing among all trackers by exploring anchor-free network for object state prediction.

4) LASOT

LaSOT [39] is a recent public large-scale tracking benchmark dataset containing 1400 fully-annotated video sequences, where 280 sequences belonging to 70 diverse classes are selected for testing. The dataset is more challenging than typical short-term tracking datasets [41], [42] due to much longer sequences whose average length is about 2500 frames. We validate our proposed tracker following the standard One-Pass Evaluation (OPE). The success and normalized precision plots are illustrated in Figure 11, in which the state-of-the-art SiamBAN [13], SiamRPN++ [7], ATOM [48], SiamMask [25], SiamDW [8], VITAL [56], C-RPN [12], MDNet [50], DSiam [18] and ECO [30] trackers are adopted for comparison. Our SiamRFL tracker outperforms all aforementioned trackers by a significant margin. In comparison with the baseline SiamRPN++, our tracker produces substantial gains of 2.5% on Success and 3.8% on Normalized Precision. These results demonstrate that the proposed fusion network is more effective than the fusion strategy in SiamRPN++, i.e., weighted average. In addition, our method performs better than SiamBAN tracker, which achieves the leading performance among all comparison methods.

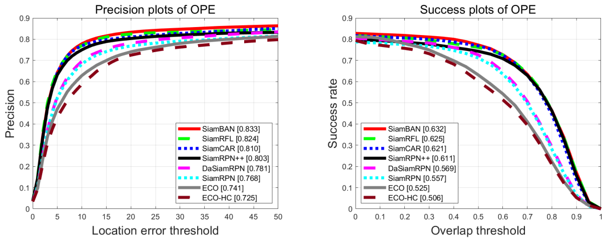

FIGURE 10. Precision and success plots of OPE for all trackers on UAV123. These trackers are ranked according to the performance scores.

5) GOK-10K

The dataset [40] is a recent high-diversity benchmark for generic object tracking including 10k video sequences for

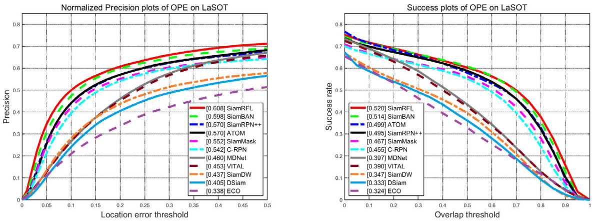

FIGURE 11. Normalized precision and success plots of OPE for all trackers on LaSOT. These trackers are ranked according to the performance scores.

TABLE 2. Comparison with state-of-the-art trackers on GOT-10k. The best three results are highlighted in red, blue and green fonts.

Trackers	AO↑	$SR_{0.5}$↑	$SR_{0.75}$↑
BACF [54]	0.260	0.262	0.101
CFNet [17]	0.293	0.265	0.087
MDNet [50]	0.299	0.303	0.099
ECO [30]	0.316	0.309	0.111
C-COT [55]	0.325	0.328	0.107
SiamFC [5]	0.374	0.404	0.144
SiamRPN [11]	0.483	0.581	0.270
SPM [23]	0.513	0.593	0.359
SiamRPN++ [7]	0.517	0.616	0.325
SiamMask [25]	0.514	0.587	0.366
ATOM [48]	0.556	0.635	0.402
SiamRFL	0.559	0.651	0.420

training and 180 sequences for testing. These testing videos cover 84 types of objects in the wild with diverse motions. Notably, there is no overlap in object classes between training and testing sets to evaluate the generation of trackers, and all comparison methods should be trained only on the dataset for equity. Following the testing protocol, we assess the tracking performance using average overlap (AO) and success rates (SR) corresponding to two overlap thresholds of 0.5 and 0.75. The results are reported in Table 2. The presented tracker achieves the top performance on all evaluation metrics, which surpasses the second-ranked ATOM [48] by 1.6% in terms of $SR_{0.5}$ and 1.8% in terms of $SR_{0.75}$. Compared with SiamRPN++, our method improves the performance by 4.5% on AO and 9.5% on $SR_{0.75}$.

B. COMPARISON WITH SIAMESE TRACKERS

To highlight the potential of the proposed fusion network, we compare our SiamRFL with several typical Siamese trackers on OTB-100 dataset. Among these comparison methods, SiamFC [5], SA-Siam [52], StructSiam [57], SiamRPN [11], DaSiamRPN [15], C-RPN [12] and SPM [23] adopt the features output from the last convolutional layer of AlexNet [58], while the rest of SiamRPN++ [7], PG-Net [32], SiamBAN [13] and SiamCAR [19] employ the ResNet-50 [9] as backbone and combine the features from multi-layers for tracking. According to Table 3, our tracker achieves the leading performance on both Success and Precision scores. We can discover that fusing multi-layer features of one deeper backbone is very effective to lift the tracking performance of Siamese trackers, but existed ways [7], [13], [19], [32] have no capacity to maximize the role of feature aggregation. In contrast, the proposed fusion scheme is more adaptive and powerful, whose outperformance and effectiveness have been verified by the comparison results.

TABLE 3. Success and precision scores of some typical Siamese trackers on OTB-100 dataset. The best three results are highlighted in red, blue and green fonts.

Trackers	Success↑	Precision↑
SiamFC [5]	0.582	0.771
Sa-Siam [52]	0.657	0.865
StructSiam [57]	0.621	0.851
SiamRPN [11]	0.637	0.851
DaSiamRPN [15]	0.658	0.880
C-RPN [12]	0.663	–
SPM [23]	0.687	0.899
SiamRPN++ [7]	0.696	0.915
PG-Net [32]	0.691	0.892
SiamBAN [13]	0.696	0.910
SiamCAR [19]	0.696	0.907
SiamRFL	0.709	0.918

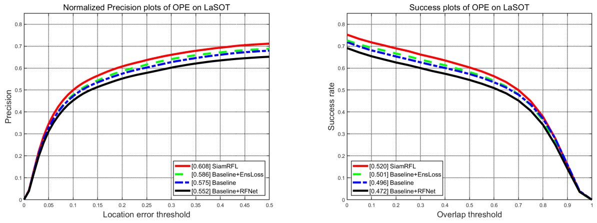

FIGURE 12. The ablation study of the proposed tracker on LaSOT datasets.

C. ABLATION STUDIES

We compare four variants of the proposed tracker on LaSOT dataset [39] to manifest the impact of our contributions, which consist of *Baseline*, *Baseline+EnsTrain*, *Baseline+RFNet* and *SiamRFL*. Concretely, "*Baseline*" represents the original SiamRPN++ tracker [7] under standard optimizing paradigm, while "*Baseline+EnsTrain*" denotes that the tracker is trained using our present ensemble training method. For "*Baseline+RFNet*", we replace traditional fusion strategy in SiamRPN++ with our residual fusion network, but still train the network using a standard optimizer. "*SiamRFL*" indicates our final tracker, in which both residual fusion network and ensemble training framework are employed.

The success and precision plots of ablation study on LaSOT are shown in Figure 12. Compared with "*Baseline*", Our ensemble training framework (*EnsTrain*) lifts the tracking performance by 0.5% on Success and 1.1% on Normalized Precision, which proves that the framework is also useful for some simple fusion mechanisms, like weighted average. However, "*Baseline+RFNet*" performs with 2.4% drops on Success and 2.3% drops on Normalized Precision. It is to say that the performance of tracker will degrade severely if we adopt the proposed residual fusion network but

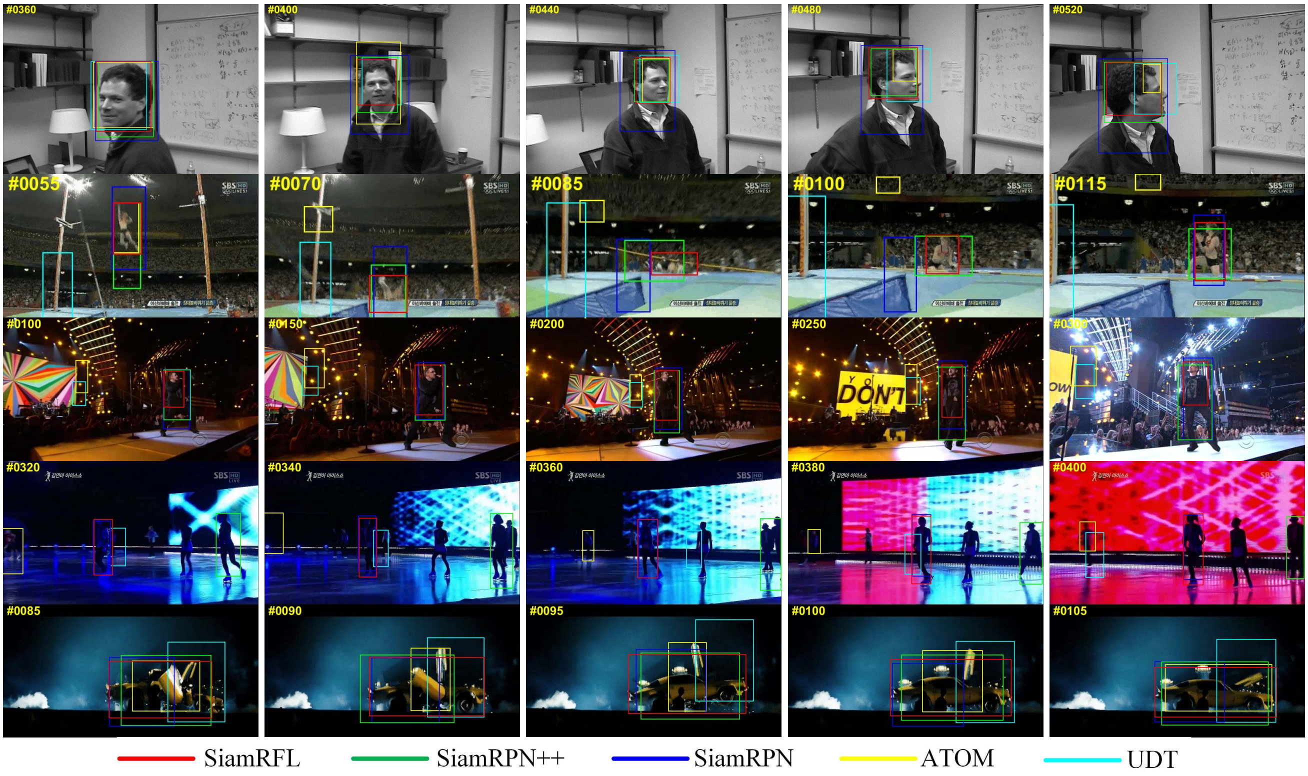

FIGURE 13. Qualitative comparison of our tracker with four state-of-the-art approaches on some challenging sequences of OTB-100 dataset (Fleetface, Jump, Singer-2, Skating1, Trans).

TABLE 4. Overall comparison on LaSOT datasets for multiple-layer feature fusion using diverse fusion methods. L3, L4 and L5 represent *conv-3*, *conv-4* and *conv-5*, respectively. The best three results are highlighted in red, blue and green fonts.

L3	L4	L5	Fusion	Success↑	Norm Precision↑
✓			-	0.464	0.557
	✓		-	0.475	0.562
		✓	-	0.452	0.537
✓	✓		WA	0.475	0.565
	✓	✓	WA	0.482	0.566
✓		✓	WA	0.466	0.548
✓	✓	✓	WA	0.496	0.575
✓	✓		RFNet	0.503	0.595
	✓	✓	RFNet	0.511	0.592
✓		✓	RFNet	0.488	0.564
✓	✓	✓	RFNet	0.520	0.608

do not adjust the training way. The final SiamRFL tracker surpasses all other variants, which obtains 2.4% Success increments and 3.3% Precision increments compared with the baseline. The phenomenon declares that our fusion network is very powerful for visual tracking once we introduce appropriate training method, i.e., the presented ensemble training.

To furtherly highlight the advantages of our fusion mechanism, we present the tracking results for aggregating the features from diverse layers and compare it with weighted average (WA), as shown in Table 4. When aggregating two stages, "*WA*" yields slight improvements on combining *conv-4* and *conv-5*, but no improvement is gained on the other two combinations. It means that weighted average has no ability to fully reflect the effect of feature fusion. In contrast, our residual network improves the tracking performance more significantly. Taking *conv-3* and *conv-4* as instance, our fusion method exceeds the conventional weighted average by 2.6% on AUC score and 2.9% on Normalized Precision score. It is even better than the model that combines *conv-3*, *conv-4* and *conv-5* via weighted average. In addition, the best results can be achieved by exploiting our fusion network to combine all three stages.

D. QUALITATIVE RESULTS

The qualitative tracking results of some recent trackers on a subset of OTB-100 [41] sequences are exhibited in Figure 13. These results demonstrate that our SiamRFL tracker is able to achieve very satisfactory visual performance and performs better than other popular comparison methods. The main reason is that the presented fusion mechanism can fuse low-ranked detail features and high-level semantic features in an adaptive and efficient way, which prompts our tracker to be more robust and accurate when facing all kind of interferences.

In the video sequence of Fleetface, our approach can address the great challenge of in-plane and out-of-plane rotations well, and track the object closely. In the sequences of Jump and Trans, there are severe scale and deformation variations for the objects. The presented tracker successfully adapts to these variations as well as precisely infers the bounding boxes, when other trackers suffer from significant scale and shape drifts. In video Singer2, our SiamRFL tracker accurately distinguishes the object from background, which proves that our tracker is strong to tackle the background clutters. In the sequence of Skating1, our method can identify the object more robustly although it is frequently occluded by other similar objects, which is since our approach can effectively perceive the detailed and semantic differences between two objects with the proposed fusion framework.

V. CONCLUSION

In this paper, we proposed a novel residual fusion network for Siamese tracker, which can aggregate multi-stage features in a powerful way. Specifically, the network utilizes deep-layer features as direct input to identify the object from background in a semantic view, and refines the object state by exploiting the local detail patterns in shallow-layer features through residual channel. When incorporating the network into Siamese tracker, an ensemble training approach was presented to address the degradation problem, which optimizes Siamese network and fusion network separately by arranging multiple loss functions. The experimental results on five popular benchmark datasets demonstrated the effectiveness of our residual fusion network, as well as the proposed tracker performs favorably against the state-of-the-art trackers.

Although has achieved promising performance, there are obviously some drawbacks in our proposed Siamese tracker. For instance, we do not design an effective online updater for our Siamese model. Since Siamese networks are generally optimized in an offline manner, an appropriate online updating mechanism is very important to help the trackers to dynamically learn the object appearances, which can lift the performance of trackers in complex scenes. In the future, we would pay more attention to solve the online updating problem.

ACKNOWLEDGMENT

Xinglong Sun would like to thank Guangliang Han and Lihong Guo for the support. The authors would like to thank the anonymous reviewers for their constructive comments.

REFERENCES

[1] Y. Zeng, X. Fu, L. Gao, J. Zhu, H. Li, and Y. Li, "Robust multivehicle tracking with Wasserstein association metric in surveillance videos," *IEEE Access*, vol. 8, pp. 47863–47876, 2020.

[2] Z. Chen, S. Li, N. Zhang, Y. Hao, and X. Zhang, "Eye-to-hand robotic visual tracking based on template matching on FPGAs," *IEEE Access*, vol. 7, pp. 88870–88880, 2019.

[3] C.-Y. Tsai and S.-H. Tsai, "Simultaneous 3D object recognition and pose estimation based on RGB-D images," *IEEE Access*, vol. 6, pp. 28859–28869, 2018.

[4] 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.

[5] 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. (ECCV)*, Oct. 2016, pp. 850–865.

[6] 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.

[7] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4277–4286.

[8] 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. 4586–4595.

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

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

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

[12] 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. 7944–7953.

[13] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6667–6676.

[14] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 771–787.

[15] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware Siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 101–117.

[16] 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.

[17] 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.

[18] 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.

[19] D. Guo, J. Wang, Y. Cui, Z. Wang, and S. Chen, "SiamCAR: Siamese fully convolutional classification and regression for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6268–6276.

[20] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, "Deep learning for visual tracking: A comprehensive survey," *IEEE Trans. Intell. Transp. Syst.*, early access, Jan. 28. 2021, doi: 10.1109/TITS.2020.3046478.

[21] M. Ondrasovic and P. Tarabek, "Siamese visual object tracking: A survey," *IEEE Access*, vol. 9, pp. 110149–110172, 2021.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2015, pp. 91–99.

[23] 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. 3638–3647.

[24] Y. Xu, Z. Wang, Z. Li, Y. Yuan, and G. Yu, "SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proc. AAAI. Conf. Artif. Intell. (AAAI)*, Feb. 2019, pp. 1–9.

[25] Q. Wang, L. Zhang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1328–1338.

[26] F. Du, P. Liu, W. Zhao, and X. Tang, "Correlation-guided attention for corner detection based visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6835–6844.

[27] L. Zhang, A. Gonzalez-Garcia, J. V. D. Weijer, M. Danelljan, and F. S. Khan, "Learning the model update for Siamese trackers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4009–4018.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] B. Liao, C. Wang, Y. Wang, Y. Wang, and J. Yin, "PG-Net: Pixel to global matching network for visual tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 429–444.

[33] X. Dong and J. Shen, "Triplet loss in Siamese network for object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 459–474.

[34] L. Wang, W. Ouyang, X. Wang, and H. Lu, "STCT: Sequentially training convolutional networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1373–1381.

[35] Z. Hu, H. Chen, and G. Li, "Deep ensemble object tracking based on temporal and spatial networks," *IEEE Access*, vol. 8, pp. 7490–7505, 2020.

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

[37] 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.

[38] 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.

[39] H. Fan, H. Ling, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, and C. Liao, "LaSOT: A high-quality benchmark for large-scale single object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5369–5378.

[40] L. Huang, X. Zhao, and K. Huang, "GOT-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 5, pp. 1562–1577, May 2021.

[41] 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.

[42] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, and R. Pflugfelder, "The seventh visual object tracking VOT2019 challenge results," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2019, pp. 2206–2241.

[43] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 445–461.

[44] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8126–8135.

[45] P. Voigtlaender, J. Luiten, P. H. S. Torr, and B. Leibe, "Siam R-CNN: Visual tracking by re-detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6577–6587.

[46] N. Wang, Y. Song, C. Ma, W. Zhou, W. Liu, and H. Li, "Unsupervised deep tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1308–1317.

[47] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6181–6190.

[48] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4655–4664.

[49] Y. Song, C. Ma, L. Gong, J. Zhang, R. W. H. Lau, and M.-H. Yang, "CREST: Convolutional residual learning for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2574–2583.

[50] 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.

[51] T. Yang and A. B. Chan, "Learning dynamic memory networks for object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 152–167.

[52] 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. (CVPR)*, Jun. 2018, pp. 4834–4843.

[53] X. Dong, J. Shen, L. Shao, and F. Porikli, "CLNet: A compact latent network for fast adjusting Siamese trackers," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 378–395.

[54] 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.

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

[56] 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. (CVPR)*, Jun. 2018, pp. 8990–8999.

[57] 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)*, Sep. 2018, pp. 351–366.

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

XINGLONG SUN received the M.S. degree from the Beijing Institute of Technology, in 2018. He is currently pursuing the Ph.D. degree with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science. His current research interests include deep learning, object tracking, and image registration.

GUANGLIANG HAN received the M.S. and Ph.D. degrees from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science, in 2000 and 2003, respectively. He is currently a Research Fellow with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science. His current research interests include computer vision, image processing, and object tracking.

LIHONG GUO received the M.S. and Ph.D. degrees from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science, in 1999 and 2003, respectively. She is currently a Research Fellow with the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Science. Her current research interests include computer vision and photoelectric system design.