IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 6, 2021, accepted August 28, 2021, date of publication September 9, 2021, date of current version September 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3111532

End-to-End Correlation Tracking With Enhanced Multi-Level Feature Fusion

GUANGEN LIU AND GUIZHONG LIU, (Member, IEEE)
School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China

Corresponding author: Guizhong Liu (liugz@xjtu.edu.cn)

This work was supported in part by the Education Ministry Joint Foundation of China under Grant 614A0223.

ABSTRACT Discriminative correlation filters (DCF) have drawn increasing interest in visual tracking. In particular, a few recent works treat DCF as a special layer and add it into a Siamese network for visual tracking. However, most of them adopt shallow networks to learn target representations, which lack robust semantic information of deeper layers and make these works fail to handle significant appearance changes. In this paper, we design a novel Siamese network to fuse high-level semantic features and low-level spatial detail features for correlation tracking. Specifically, to introduce more semantic information into low-level features, we specially design a residual semantic embedding module to adaptively involve more semantic information from high-level features to guide the feature fusion. Furthermore, we adopt an effective and efficient channel attention mechanism to filter out noise information and make the network focus more on valuable features that are beneficial for visual tracking. The overall architecture is trained end-to-end offline to adaptively learn target representations, which are not only enabled to encode high-level semantic features and low-level spatial detail features, but also closely related to correlation filters. Experimental results on widely used OTB2013, OTB2015, VOT2016, TC-128, and UAV123 benchmarks show that our proposed tracker performs favorably against several state-of-the-art trackers.

INDEX TERMS Visual tracking, correlation filters, deep features, multi-level feature fusion.

I. INTRODUCTION

Visual object tracking is a fundamental research topic in computer vision and plays an important role for its various applications, e.g., vehicle navigation, robotics, surveillance, and so on. And visual object tracking aims at estimating states of a target in a video sequence given its appearance template in the first frame. Despite many tracking algorithms proposed in the past few decades, it's still a challenging problem to develop a robust tracker due to the factors such as illumination variations, pose changes, scale changes, cluttered background, severe occlusions, fast motion and out-of-plane rotations.

Recently, the trackers based on discriminative correlation filters (DCF) method [1], [2] have received significant attention due to their state-of the-art performance and high tracking speed. The DCF is an algorithm that learns to discriminate between images and their circulant translations by solving a ridge regression problem extremely efficiently in Fourier domain. The conventional DCF based trackers exploit hand-crafted features (e.g., Histogram of Oriented Gradient (HOG) [3], color attributes [4]) for correlation tracking. Inspired by the great success of deep convolutional neural networks (CNN) in image recognition [5], [6], semantic segmentation [22] and object detection [7], [8], some DCF based trackers [9]–[11] replace hand-crafted features with CNN features and achieve significant improvement in performance. However, these trackers treat the extraction of deep convolutional features and DCF as two separated parts. In deep learning algorithms, the optimal performance tends to be achieved by adopting end-to-end deep network architectures for jointly learning all parameters of multiple parts.

Meanwhile, Siamese network based trackers have also gained attention recently. The pioneering work, fully-convolutional Siamese network based tracker (SiamFC) [20], formulates tracking as a problem of similarity computation between the target template and the search region. To get more precise bounding boxes, the Siamese region proposal network (SiamRPN) [28] improves the SiamFC by adding a region proposal network (RPN). These two works adopt shallow networks that restrict their performance. Li *et al.* [26] design a deeper Siamese network to perform layer-wise

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

aggregation by constructing multiple RPN modules on it and fusing the outputs of these modules as a final prediction. The later work [64] adopts an anchor-free strategy to predict object bounding boxes, and fuses low-level and high-level features by concatenating multi-layer deep features along channel dimension for tracking. Although these trackers achieve state-of-the-art performance, the lack of online learning makes them hard to adapt to appearance variations of target. Therefore, they are easily disturbed by background clutters.

Recently, several works [12], [13], [24] have begun to investigate the integration of CNN features and DCF. They regard DCF as a special differentiable layer and integrate this layer into a lightweight Siamese network. Moreover, this layer can be updated online efficiently to adapt to appearance variations of target [13], [24]. Although these network models can capture convolutional features suited to correlation tracking by offline training, their algorithms are hard to handle significant appearance changes and complex scenes during tracking due to the lack of robust semantic information from high-level features in deep convolutional networks.

In this paper, we use a deep network for feature extraction and design a sub-network to enhance the fusion of low-level and high-level deep features for correlation tracking. Concretely, we treat DCF as an independent layer and integrate it with our novel Siamese network to construct an end-to-end deep architecture for visual tracking, as shown in Fig. 1. The Siamese network is used to generate features suitable for tracking task and divided into two parts: multi-level feature extraction sub-network (MFEN) and multi-level feature fusion sub-network (MFFN), as Fig. 2 shows. MFEN uses VGG-16 [5] as a backbone network and is employed to extract features on multiple levels, each of which characterizes inputs from different aspects. In the MFFN, in order to suppress noise information and enhance useful features, we adopt an efficient channel attention (ECA) module on high-level features to select the channels which play an important role for visual tracking. Moreover, we propose a residual semantic embedding (RSE) module to adaptively introduce semantic information into low-level features, which contributes to reducing the gap in semantic levels and spatial resolution, and enhancing the fusion of low-level and high-level features. Then we adopt a top-down architecture with lateral connections to fuse the multi-level convolutional features into a group of single resolution feature maps, which simultaneously contain high-level semantic information and spatial details and are sent into the correlation filter layer for visual tracking.

In short, our contributions are summarized as follows:

1. We construct a novel Siamese network and combine it with correlation filter (CF) layer to end-to-end learn the fusion of high-level semantic features and low-level detail features for visual tracking.

2. We adopt an effective and efficient channel attention mechanism to make the tracking network focus on important features and suppress unwanted noise information, which can enhance the feature fusion and representation ability of the target object.

3. We design a residual semantic embedding module and integrate it into the Siamese network, which can adaptively involve semantic information from high-level features to guide the fusion of high-level semantic features and low-level spatial details.

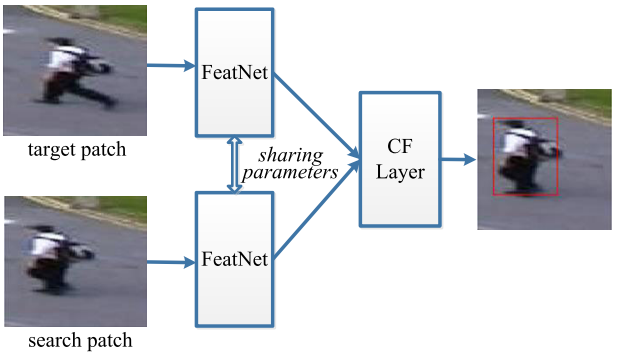


FIGURE 1. The overall network architecture of the proposed tracker. The target patch and search patch have the same size (128 × 128). Discriminative Correlation filters can be formulated as a layer in the network, which is called the CF layer in this paper.

II. RELATED WORK

Visual tracking is a significant subject in computer vision and has been studied extensively in a series of approaches in recent years. Since our main contribution is an end-to-end multi-level deep feature fusion framework for correlation tracking, we give a brief review on four directions closely related to this work: DCF based tracking, CNN based trackers, multi-level feature fusion and attention mechanisms.

A. DCF BASED TRACKING

In recent years, discriminative correlation filters (DCF) based tracking methods have drawn increasing attention because of their high computational efficiency. Minimum Output Sum of Squared Error (MOSSE) tracker [14] is the first work that applies DCF to visual tracking, which uses gray features to characterize object appearance and achieves outstanding performance at a high speed. Henriques *et al.* [1] replace the gray features with multi-channel features (e.g. HOG [3]) and use circulation matrices to interpret correlation filters. Color Names (CN) [4] takes color information into account for correlation tracking. Sui *et al.* [16] experimentally demonstrate that the performance of the DCF tracker can be improved by using more robust loss functions. The Discriminative Scale Space Tracker (DSST) [2] solves the problem of scale estimation by learning adaptive multi-scale filters. Danneljan *et al.* [17] alleviate the inherent boundary effects in DCF tracking by penalizing the filter coefficients.

To further improve the tracking performance, some DCF trackers characterize the object appearance with deep convolution features that have superior representation power. Hierarchical Convolutional Features (HCF) [9] and Hedged Deep Tracking (HDT) [11] combine pre-trained multi-layer

deep convolutional features for DCF tracking, which online learn DCF trackers on different CNN layers respectively and fuse the responses of these trackers for object tracking. In Deep Spatially Regularized DCF (DeepSRDCF) [15], the experimental results show that using convolutional features from the first layer can achieve superior tracking performance compared to the deeper layers in a spatially regularized DCF framework. Continuous Convolution Operator Tracker (CCOT) [10] constructs a novel DCF formulation for employing multi-resolution features. The above mentioned methods usually extract deep convolutional features with pre-trained networks, which are trained for different tasks. Therefore, they treat the feature extraction and correlation filter tracking as two independent components. So the achieved tracking performance may be suboptimal. While CFNet [12] and DCFNet [13] design correlation filters as a differentiable layer and integrate it into a lightweight Siamese network, thus achieving an end-to-end representation learning. Flow correlation Tracking (FlowTrack) [24] further takes the flow information in consecutive frames into account. Although these works are able to learn target representations suitable for correlation tracking, they adopt shallow networks and cannot take full advantage of features from deep networks, such as VGG-16 or deeper. Our work constructs an end-to-end deep network to fuse multi-level convolutional features and integrates the fused features with the CF layer for visual tracking.

B. CNN BASED TRACKERS

Apart from the DCF trackers using CNN features, other CNN based trackers have also made significant progress. Multi-Domain Network (MDNet) [18] offline trains a discriminative CNN with multiple branches structure and online fine-tunes the pre-trained network for tracking. Song *et al.* [23] formulate ridge regression as a convolution layer and employ a gradient descent technique to solve regression weight coefficients. Recently, Siamese network-based trackers [19], [20], [25], [26], [28]–[30], [63]–[66] have attracted much attention due to their accuracy and speed. The pioneering works, such as Siamese Instance search Tracker (SINT) [19] and SiamFC [20], use Siamese networks to offline learn a similarity metric between the target template and candidate image patches. SiamRPN [28] introduces a RPN into the Siamese network to get more accurate object bounding boxes. The follow-up works [26], [25] further improve the performance of the SiamRPN by designing deeper and wider networks. After that, the recent works [63]–[65] also use deeper networks for feature extraction and adopt an anchor-free strategy to predict bounding boxes directly without using predefined anchor boxes. To better distinguish the target from background clutters, Cheng *et al.* [66] introduce a novel Relation Detector into the Siamese network. Gao *et al.* [30] construct a Graph Convolutional Tracking (GCT) framework to exploit both spatial-temporal and context information of the target. These methods use time-consuming online fine-tuning paradigms to update network models, or lack online learning of target models. Different from them, our tracker can effectively and efficiently update the correlation filter layer online to adapt to the appearance variations of the target.

C. MULTI-LEVEL FEATURE FUSION

The feature representations of target play an important role in visual tracking. In general, different features describe the target from different views and properly fusing multiple features may achieve good tracking performance. The trackers [9], [19], [21], [24], [31]–[34], [54] integrate multi-level deep features for visual tracking. HCF [9] and Multi-Cue Correlation filter based Tracker (MCCT) [31] construct correlation filters on multi-level deep features respectively and combine correlation response maps in an empirical manner. The work [32] also uses multi-level deep features and exploits end-to-end offline training to learn the weight coefficients used to combine response maps. The above three methods calculate the response maps separately on multi-level deep features and then merge the response maps to achieve the final tracking result. While the works [19], [21], [33] directly combine multi-level deep features by channel-wise concatenation or addition, and then the fused deep features are sent into correlation filters or prediction layer to detect targets. Cascaded Region Proposal Networks (C-RPN) [34] adopts a cascade way to combine multiple predictions and exploits element-wise addition to fuse high-level and low-level features. Zhu *et al.* [24] combine flow information between consecutive frames and deep features to improve tracking performance. Zhong *et al.* [67] adopt a probabilistic method to fuse multiple imperfect oracles for visual tracking. To get a robust motion model, Zhong *et al.* [68] design a hierarchical tracker by reinforcement learning based searching and coarse-to-fine verifying. In this paper, we did not use simple feature addition or concatenation to fuse features. By introducing the attention mechanism and semantic embedding module, we construct an end-to-end deep network framework for the harmonious fusion of low-level spatial details and high-level semantic features.

D. ATTENTION MECHANISMS

Attention mechanisms have been used in many computer vision tasks [35]–[37], which could reduce the irrelevant features and focus on the important ones. In visual tracking algorithms, Wang *et al.* [38] improve the target representations by learning dual attention and channel attention for target branch in the Siamese architecture. Zhu *et al.* [24] develop a spatial-temporal attention mechanism to adaptively aggregate historical frame features. Yu *et al.* [39] use spatial attention to learn context information, and channel attention to emphasize interdependent channel-wise features. Zhang *et al.* [32] incorporate a residual channel attention mechanism into the backbone network. In this paper, we adopt a lightweight channel attention module to suppress the noisy features and select more discriminative ones.

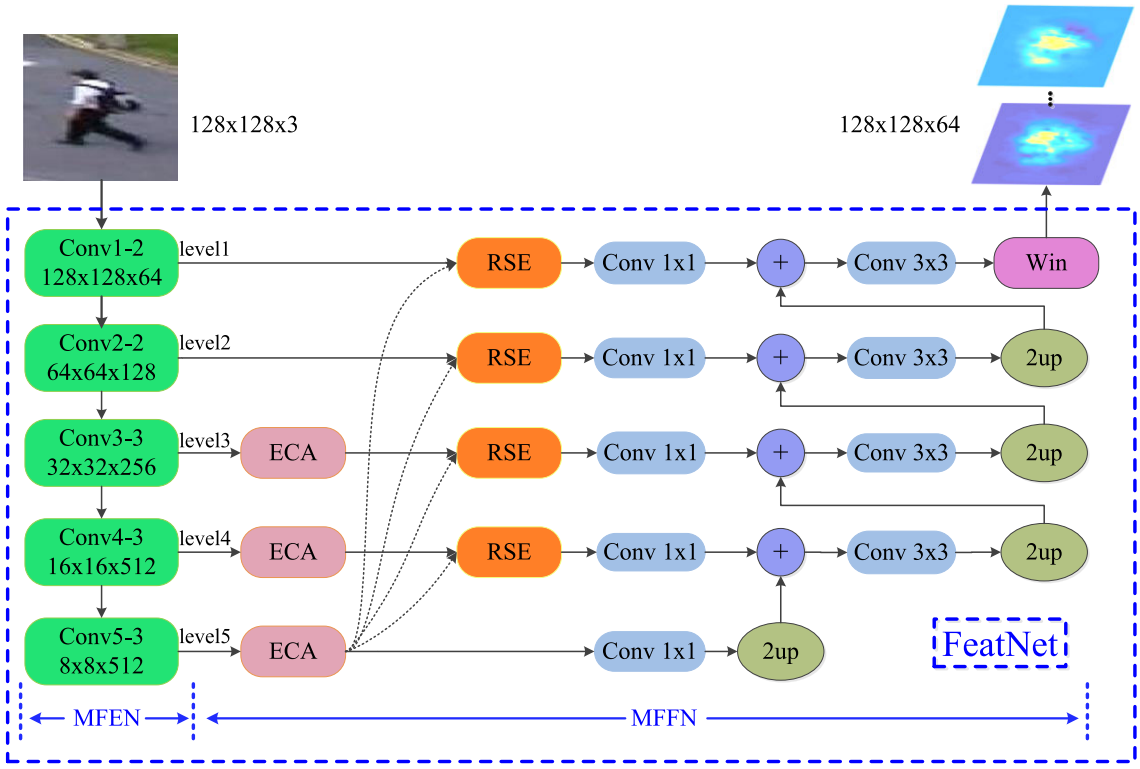


FIGURE 2. Detailed structure of FeatNet module used for generating fused features of the target patch and search patch. FeatNet consists of multi-level feature extraction sub-network (MFEN) and multi-level feature fusion sub-network (MFFN). ECA means Efficient Channel Attention (ECA) module. RSE indicates Residual Semantic Embedding (RSE) module. "+" and "2up" respectively represent element-wise addition operation and 2 times bilinear upsampling operation. "Win" indicates cosine window operation. "Conv NxN" means a convolutional layer composed of 64 convolution kernels of size NxN, which is followed by a BN layer and a relu layer.

III. OUR PROPOSED METHOD

In this section, we first introduce DCF framework and correlation filter layer. Then the overall correlation network architecture is described in detail. Finally, online tracking is described consisting of model updating and scale estimation.

A. DCF AND CORRELATION FILTER LAYER

Here, we briefly introduce the DCF tracking method. More details can be found in paper [2]. Let $x \in \mathbb{R}^{M\times N\times C}$ be a C-channel feature map of spatial size $M \times N$ extracted from a target patch. The regression target $y \in \mathbb{R}^{M\times N}$ is designed as a Gaussian function associated with the feature map x, whose peak value is located at the center. The goal of DCF training is to find an optimal filter that minimizes the following loss function:

$$J = \left\| \sum_{l=1}^{C} f^l \star x^l - y \right\|^2 + \lambda \sum_{l=1}^{C} \left\| f^l \right\|^2 \tag{1}$$

In the above formula, the superscript $l \in \{1, \ldots, C\}$ denotes channel number, $\star$ represents circular correlation, λ is a regularization parameter. The closed-form solution of (1) is:

$$f^l = F^{-1}\left(\frac{\hat{y}^* \odot x^l}{\sum_{r=1}^{C} (\hat{x}^r)^* \odot \hat{x}^r + \lambda} \right) \tag{2}$$

Here, the hat represents the discrete Fourier transform (DFT) of a function, F^{-1} denotes the inverse DFT, $*$ indicates the complex conjugate of corresponding variables, $\odot$ denotes point-wise multiplication.

In the tracking, the feature map z is extracted from a search region having the same size with the target patch in the new frame. Its corresponding confidence score map g is then calculated as

$$g = F^{-1}\left(\sum_{l=1}^{C} \hat{f}^{l*} \odot \hat{z}^l \right) \tag{3}$$

The new target state can be estimated by searching the maximum value of the score map g.

In order to construct DCF as a single layer in the deep neural network as shown in Fig. 1, we derive the forward propagation and backward propagation through this layer, which is called correlation filter (CF) layer. The forward propagation of the CF layer can be easily implemented by using formula (3). For backward propagation, we calculate the derivatives of loss function L with respect to the inputs x and z respectively. The loss function L for network training is formulated as:

$$\begin{aligned} L(\theta) &= \|g(\theta) - \tilde{g}\|^2 + \mu \|\theta\|^2 \\ s.t.\ g(\theta) &= F^{-1}\left(\sum_{l=1}^{C} \hat{f}^{l*} \odot \hat{z}^l(\theta) \right) \\ \hat{f}^l &= \frac{\hat{y}^* \odot \hat{x}^l(\theta)}{\sum_{r=1}^{C} \left(\hat{x}^r(\theta)\right)^* \odot \hat{x}^r(\theta) + \lambda} \end{aligned} \tag{4}$$

where $\tilde{g}$ denotes the desired response that is a Gaussian distribution centered at the real target location. θ refers to the parameters of the feature network (FeatNet module in the Fig. 1). Here we give the derivatives directly, which are also derived in [13].

$$\begin{cases} \dfrac{\partial L}{\partial x^l} = F^{-1}\left(\dfrac{\partial L}{\partial (\hat{x}^l)^*} + \left(\dfrac{\partial L}{\partial \hat{x}^l}\right)^*\right) \\ \dfrac{\partial L}{\partial (\hat{x}^l_{mn})^*} = \dfrac{\partial L}{\partial \hat{g}^*_{mn}} \dfrac{(\hat{z}^l_{mn})^* \hat{y}^*_{mn} - \hat{g}^*_{mn}(\hat{x}^l_{mn})^*}{\sum_{r=1}^{C} \hat{x}^r_{mn}(\hat{x}^r_{mn})^* + \lambda} \\ \dfrac{\partial L}{\partial \hat{x}^l_{mn}} = \dfrac{\partial L}{\partial \hat{g}^*_{mn}} \dfrac{-\hat{g}^*_{mn}\hat{x}^l_{mn}}{\sum_{r=1}^{C} \hat{x}^r_{mn}(\hat{x}^r_{mn})^* + \lambda} \end{cases} \tag{5}$$

$$\begin{cases} \dfrac{\partial L}{\partial z^l} = F^{-1}\left(\dfrac{\partial L}{\partial (\hat{z}^l)^*}\right) \\ \dfrac{\partial L}{\partial (\hat{z}^l_{mn})^*} = \dfrac{\partial L}{\partial \hat{g}^*_{mn}} \hat{f}^l_{mn} \end{cases} \tag{6}$$

where $\frac{\partial L}{\partial \hat{x}^l_{mn}}$ denotes the partial derivative with respect to each element of $\hat{x}^l$ (indexed by m and n).

B. CORRELATION NETWORK ARCHITECTURE

As shown in Fig. 1, the overall training framework of our correlation network consists of FeatNet (feature extraction network) and CF tracking layer. The overall training architecture adopts symmetric Siamese network with parameter sharing. As can be seen from Fig. 2, the FeatNet can be divided into multi-level feature extraction sub-network (MFEN) and multi-level feature fusion sub-network (MFFN). In MFEN, we take conv1-2, conv2-2, conv3-3, conv4-3 and conv5-3 layers of VGG-16 to extract multi-level features. Note that the VGG-16 network can be easily replaced with other different deep convolutional networks such as VGG-19 [5] or ResNet [27]. In MFFN, we take conv3-3, conv4-3 and conv5-3 in VGG-16 as high-level features and impose channel-wise attention on them to suppress noisy features. In general, high-level features have obvious semantic distinctions among different channels. Meanwhile, we adopt residual semantic embedding (RSE) module to adaptively introduce semantic information into lower-level features for more effectively feature fusion among different feature levels. Then we adopt a convolution kernel with size 1×1 on each level of features to reduce their channel dimensions to 64. And we exploit a top-down feature fusion way that hierarchically propagates high-level semantic information into the lower layers. Finally, the fused feature maps from two branches of symmetric Siamese network are fed into subsequent correlation filter layer for training. All the modules are differentiable and trained end-to-end. In the next section, we will describe channel attention mechanism, RSE module and multi-level feature fusion strategy.

1) CHANNEL ATTENTION MECHANISM

Most feature fusion methods simply apply concatenation or element-wise sum operation to incorporate high-level semantic cues and low-level spatial detail features. However, this can degrade the quality of predictions because background clutters and noise information can also be passed through. So it is important to filter these features and focus more on valuable features. Some channel attention methods [35], [36], [40], [41] are proposed to recalibrate the importance of features. Especially, the experiments in recent works [40], [41] show dimensionality reduction in squeeze-and-excitation (SE) network [35] prevents the SE block from achieving the full potential of channel attention mechanism. The reason may be that dimensionality reduction can destroy the direct correspondence between the channel and its weight. And experiments in work [41] indicate that batch normalization (BN) can further improve the performance of the attention module. Inspired by these works, we design an efficient and effective channel attention method as shown in Figure 3.

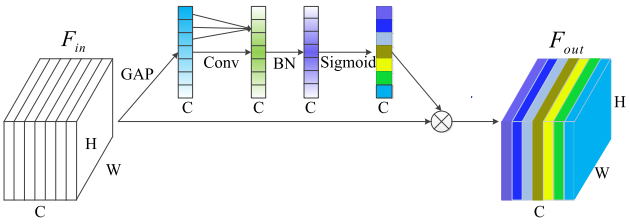


FIGURE 3. Diagram of channel attention module. GAP indicates global mean pooling layer. The input features are passed through it to obtain a channel-wise vector. Conv means 1D convolution and is used to capture local channel-wise dependencies. BN and sigmoid denote batch normalization layer and sigmoid operation respectively.

We unfold input features $F_{in} \in \mathbb{R}^{W\times H\times C}$ as $F_{in} = [f^1, f^2, \cdots, f^C]$, where $f^i \in \mathbb{R}^{W\times H}$ is the $i-th$ slice of F_{in} and C is the total channel number. First, we apply global average pooling (GAP) to each f^i to obtain a channel-wise feature vector $v \in \mathbb{R}^C$. Then we adopt a one-dimensional convolution layer with kernel size of k to capture local channel-wise dependencies. After that, through using BN and sigmoid operation, we take normalization measures for the encoded channel feature vectors and map them to [0, 1].

$$CA = \sigma\left(BN\left(Conv1D\left(GAP\left(F_{in}\right)\right)\right)\right) \tag{7}$$

where $Conv1D$ and σ represent $1D$ convolution and sigmoid operation respectively. The final output F_{out} of the module is obtained by weighting the input features with CA.

$$F_{out} = CA \cdot F_{in} \tag{8}$$

It is worth noting that the channel attention module only involves k parameters, which is much smaller than the number of parameters $(2C^2/r)$ in the SE block [35]. As shown in Fig. 2, we apply this efficient channel attention (ECA) module to conv3-3, conv4-3 and conv5-3 of VGG-16. These high-level features contain abstract image semantics and have semantic distinctions among different channels. And it is important to emphasize the feature channels that characterize objects and filter out the ones that characterize background by using the channel attention mechanism. We don't use channel-wise attention for conv1-2 and conv2-2, because there are almost no semantic distinctions among different channels of low-level features.

2) RESIDUAL SEMANTIC EMBEDDING

Low-level and high-level features are complementary by nature, where low-level features are rich in spatial details but lack semantic information and vice versa. However, low-level features are noisy and hard to provide sufficient high resolution semantic guidance. And high-level features with little spatial information cannot take full advantage of low-level features. Therefore feature fusion could be enhanced by introducing more semantic concepts into low-level features to alleviate the gap between low-level and high-level features [42]. On the other hand, introducing too much semantic information may suppress some necessary details in the lower layers, which are important for visual tracking task.

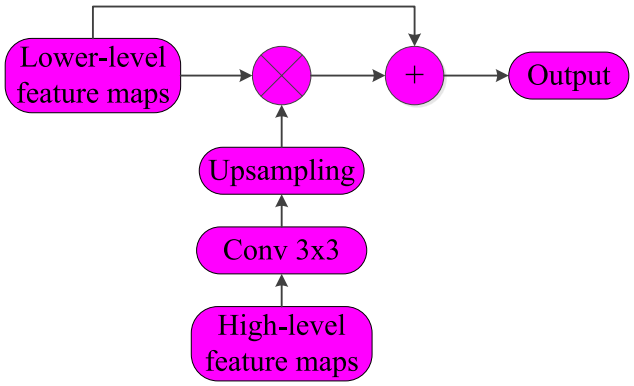


FIGURE 4. Diagram of residual semantic embedding (RSE) module. The "x" sign and "+" sign indicate element-wise multiplication and addition, respectively. "Upsampling" denotes a bilinear up-sampling layer.

Taking the above considerations together, we construct the way of introducing semantic information into low-level features as a residual form, as shown in Fig. 4. We call it residual semantic embedding (RSE) module. The inputs of RSE module are high-level and lower-level features maps. The RSE module can be expressed as

$$F^{out} = F^{L} \odot Up\left(Conv_{3\times3}\left(F^{H}\right)\right) + F^{L} \tag{9}$$

where F^{L} and F^{H} represent lower-level feature maps and high-level feature maps respectively. Up refers to the bilinear upsampling operation. F^{out} means the output of the RSE module. By introducing residual connection, this module can adaptively involve semantic information from high-level features to guide the feature fusion. When lower-level feature maps are sufficient enough to obtain perfect prediction, then the residual is simply driven towards zero. As shown in Fig. 2, we take the refined feature maps from conv5-3 by ECA module as high-level feature maps in RSE module. And we apply the RSE module for features of lower levels (see Fig. 2).

3) MULTI-LEVEL FEATURE FUSION

The feature maps from shallower layers encode low-level details and spatial information, which can be exploited to achieve better localization. However, such features are sensitive to scene changes and make tracking algorithms less robust. Meanwhile, the features of deeper layers encode context and semantic information, which are robust to target appearance changes. However, these features lack spatial details, resulting in poor localization. Hence, it is important to fuse features from different layers in order to achieve better tracking performance.

As shown in Fig. 2, we adopt a top-down fusion way that hierarchically propagates high-level semantic information into the lower layers. We take the feature integration of level5 and level4 as an example to describe the feature fusion process. After being recalibrated by the ECA module, the convolutional features from conv5-3 layer are followed by a 1×1 convolutional layer to reduce the channel dimension to 64. Then a deconvolution or upsampling layer is applied to the feature maps consisting of 64 channels to double their spatial resolution. Meanwhile, the feature maps from conv4-3 layer are passed through the ECA module, RSE module and a 1×1 convolutional layer used for reducing channel dimensionality in turn. Subsequently, the features of level5 and level4 are fused by summing the up-sampled features from the level5 branch and the channel dimension reduction features from the level4 branch. To mitigate the gridding effect caused by upsampling operation, a 3×3 convolutional layer is attached on the merged features to generate the final fused feature maps. So far, we have fused the features from level5 and level4. Finally, features of all five levels are hierarchically integrated by repeating the above feature fusion process. In this way, the final integrated features will simultaneously encode semantic information and spatial details.

C. ONLINE TRACKING

Once we have completed the training of the tracking network, online tracking can be implemented by evaluating the network in forward-mode. When a new frame comes, the search patch is cropped at the estimated object location of the previous frame. The search patch has the same size with the target patch and is sent into the tracking network to generate the correlation response map. The object position in the new frame is predicted by searching for the maximum response value.

Besides, in order to adapt our algorithm to the appearance changes of the target and preserve the robustness of target model, we only update the correlation filters with weighted historical target templates as in [13], [24]. We consider historical templates $\{x_t : t = 1, \ldots, p\}$ of the target from the first frame till the current frame p. Thus the optimization problem in equation (1) can be reformulated as follows.

$$J = \sum_{t=1}^{p} \beta_t \left(\left\| \sum_{l=1}^{C} f_p^l \star x_t^l - y \right\|^2 + \lambda \sum_{l=1}^{C} \left\| f_p^l \right\|^2 \right) \tag{10}$$

where the parameter β_t is the weight of template x_t and t represents the frame index. The closed-form solution in

equation (2) can be extended to time series as in [4], [13].

$$f_p^l = F^{-1}\left(\frac{\sum_{t=1}^{p}\beta_t \hat{y}^* \odot x_t^l}{\sum_{t=1}^{p}\beta_t\left(\sum_{r=1}^{C}(\hat{x}_t^r)^* \odot \hat{x}_t^r + \lambda\right)}\right) \quad (11)$$

In practice, the weight β_t is set by using a learning rate parameter, denoted by γ. We set$\beta_1 = (1-\gamma)^{p-1}$, $\beta_j = \gamma(1-\gamma)^{p-j}, j = 2, \ldots, p$, $\gamma \in (0, 1)$. This equation puts more weight on recent frames and makes the effect of previous frames decay exponentially over time.

Moreover, scale estimation is also crucial to object tracking. We extract search patches centered at the previously predicted object position in different scales in each frame. Similar to [24], we use patch pyramid with the scale factors $\left\{a^s | s = \left\lfloor -\frac{S-1}{2} \right\rfloor, \cdots, \left\lfloor \frac{S-1}{2} \right\rfloor\right\}$. Where S denotes the number of scale layers and a is used to restrict the sampling granularity in the scale space. These patches are then resized into the same size with the target patch. Next these patches are fed into the tracking network to generate the response maps. The target size $(\mathrm{w}_p, \mathrm{h}_p)$ at $p-th$ frame is computed as:

$$(\mathrm{w}_p, \mathrm{h}_p) = \alpha(\mathrm{w}_p^{s\circ}, \mathrm{h}_p^{s\circ}) + (1-\alpha)(\mathrm{w}_{p-1}, \mathrm{h}_{p-1}) \quad (12)$$

where $(\mathrm{w}_p^{s\circ}, \mathrm{h}_p^{s\circ})$ is denoted as the size of the scaled search patch with maximum response value at $p-th$ frame, α is a weighted coefficient.

IV. EXPERIMENTS AND ANALYSIS

A. IMPLEMENTATION DETAILS AND TRACKING DATASETS

1) IMPLEMENTATION DETAILS

We use the ILSVRC2015 VID dataset [6] for training and validation, which contains almost 4500 videos. We crop a training sample pair from every two frames (the interval between them is less than 10) in a video. Both elements in this pair are target patches with 2 times the size of the target bounding box. In total we have cropped more than 240000 sample pairs for training. All the cropped patches are resized to 128×128. The multi-level feature extraction network (backbone network) is pre-trained on ImageNet [6]. The network parameters are optimized by minimizing the square loss between predicted response maps and desired Gaussian label using stochastic gradient descent (SGD) with a learning rate of 0.0001. We use a momentum of 0.9 and set the weight decay parameter and mini-batch size to 0.0005 and 32, respectively. Training is performed for 40 epochs. The regularization parameter λ in equation (1) is set to 1e-4.

In online tracking, the learning rate parameter of filter γ is set to 0.005. The standard deviation for the desired correlation output is set to 1/10 of the target size. We set the weighted parameter α in equation (12) to 0.0075. The scale step a and number S are set to 1.0275 and 3, respectively. The proposed tracking method is implemented using Pytorch on a PC with an Intel 2.4GHz CPU, 32 GB RAM, GeForce GTX 1080Ti GPU. Average speed of the tracker is 20 FPS.

2) TRACKING DATASETS

We evaluate our method on five benchmark datasets, including OTB2013 [43], OTB2015 [44], VOT2016 [45], UAV123 [58] and Temple-color-128 (TC-128) [47] benchmarks. In the following, we briefly introduce these datasets and their corresponding performance measures.

OTB2013, OTB2015, UAV123, TC128. OTB2013 [43] and OTB2015 [44] are widely used tracking benchmark datasets that are composed of 51 and 100 sequences respectively. And UAV123 [58] and TC-128 [47] consist of 123 and 128 sequences respectively. These four benchmarks all adopt center location error (CLE) and overlap ratio (OR) as basic metrics to measure the tracking performance on a single frame. Based on CLE and OR respectively, the precision and success plots are introduced in these benchmarks to evaluate the overall tracking performance on all sequences. Concretely, the precision plot computes the percentage of frames with a CLE lower than a given threshold, which is usually set to 20 pixels to rank tracking methods. The success plot measures the percentage of the successful frames whose OR is larger than a given threshold. The area under the curve (AUC) of the success plot is usually used as the primary metric to rank trackers.

VOT2016. The VOT2016 [45] benchmark dataset consists of 60 challenging sequences. The VOT benchmark will re-initialize the tracker 5 frames after detecting a failure, which is defined as the case when the overlap between the predicted bounding box and ground truth becomes zero. With this re-initialization method, the metrics of accuracy, robustness and expected average overlap (EAO) are used to evaluate the trackers. Specifically, accuracy measures the average overlap between the predicted results and ground truths during successful tracking periods. Robustness measures the failure times. Based on accuracy and robustness, EAO is computed to measure the overall tracking performance.

B. ABLATION STUDY

1) MULTI-LEVEL FEATURE FUSION

We perform a detailed ablation study to validate the effectiveness of our proposed method. In this paper, the proposed tracker is named as Enhanced Feature Fusion Correlation Tracking (EFFCT). The network architecture of EFFCT is shown in Figure 1 and Figure 2. In this experiment, we train and evaluate the EFFCT and its nine variants. The nine variants are obtained by redesigning the structure of FeatNet module (Figure 2). Thus their configurations are as follows.

(i) FFCT_W_ECA. This variant is obtained by removing RSE sub-modules (see Figure 2) from the FeatNet module.

(ii) FFCT_W_RSE. This variant is obtained by removing ECA sub-modules (see Figure 2) from the FeatNet module.

(iii) FFCT-L1-5. This variant is obtained by removing both ECA and RSE sub-modules from the FeatNet module. So FFCT-L1-5 fuses the features extracted from five layers (conv1-2, conv2-2, conv3-3, conv4-3 and conv5-3) of VGG-16 model for correlation tracking.

TABLE 1. Performance comparison among the variants of proposed tracker on OTB2013, OTB2015, TC-128 and UAV123 DATASETS. The variants fuse different level features extracted from VGG-16 model. The precision (Denoted as "PREC" in this table) and AUC scores are reported. The best three results are shown in Red, Green and Blue FONTS, respectively.

Variants	OTB2013		OTB2015		TC128		UAV123	
	Prec	AUC	Prec	AUC	Prec	AUC	Prec	AUC
FFCT-L4-5	0.823	0.588	0.774	0.564	0.636	0.461	0.672	0.468
FFCT-L1-2	0.828	0.631	0.795	0.613	0.695	0.532	0.710	0.509
FFCT-L1-4	0.876	0.673	0.838	0.646	0.717	0.537	0.738	0.525
FFCT-L3-5	0.852	0.653	0.823	0.630	0.706	0.517	0.718	0.506
FFCT-L2-5	0.875	0.664	0.842	0.642	0.720	0.535	0.740	0.514
FFCT-L1-5	0.889	0.676	0.850	0.648	0.737	0.546	0.745	0.523

TABLE 2. Performance comparison of the proposed tracker and its variants on OTB2013, OTB2015, TC-128 and UAV123 DATASETS. The precision (Denoted as "PREC" in this table) and AUC scores are reported. The best two results are shown in Red and Blue fonts, respectively.

Variants	ECA	RSE	OTB2013		OTB2015		TC128		UAV123	
			Prec	AUC	Prec	AUC	Prec	AUC	Prec	AUC
FFCT-L1-5			0.889	0.676	0.850	0.648	0.737	0.546	0.745	0.523
FFCT-Concat			0.865	0.656	0.824	0.632	0.726	0.540	0.741	0.529
FFCT_W_ECA	√		0.904	0.686	0.860	0.657	0.755	0.563	0.751	0.536
FFCT_W_RSE		√	0.910	0.682	0.862	0.653	0.747	0.557	0.755	0.530
EFFCT	√	√	0.918	0.691	0.874	0.665	0.766	0.573	0.763	0.541

(iv) FFCT-L2-5. Similar to FFCT-L1-5, this variant fuses the features extracted from conv2-2, conv3-3, conv4-3 and conv5-3 for correlation tracking.

(v) FFCT-L3-5. This variant fuses the features extracted from conv3-3, conv4-3 and conv5-3 for correlation tracking.

(vi) FFCT-L1-4. This variant fuses the features extracted from conv1-2, conv2-2, conv3-3 and conv4-3 for correlation tracking.

(vii) FFCT-L1-2. This variant fuses the features extracted from conv1-2 and conv2-2 for correlation tracking.

(viii) FFCT-L4-5. This variant fuses the features extracted from conv4-3 and conv5-3 for correlation tracking.

(ix) FFCT-Concat. Based on FFCT-L1-5, this variant is obtained by replacing the feature addition ("+" in Figure 2) with feature concatenation (concatenating the features along channel dimension).

The proposed EFFCT and its nine variants are evaluated on OTB2013, OTB2015, TC128 and UAV123 datasets. The quantitative results of the ablation study are reported in Table 1 and Table 2. According to Table 1, the FFCT-L1-5, FFCT-L1-4 and FFCT-L2-5 obtain better performance than FFCT-L4-5 and FFCT-L1-2 on the four datasets. The reason may be that the FFCT-L1-5, FFCT-L1-4 and FFCT-L2-5 employ both high-level semantic features and low-level detail features to characterize the target, while the FFCT-L4-5 and FFCT-L1-2 only use either high-level semantic features or low-level detail features. Specially, the FFCT-L1-5 achieves the best performance on 3 out of 4 datasets, and significantly outperforms the FFCT4-5 and FFCT-L1-2 on all datasets, which demonstrates the effectiveness of FFCT-L1-5 for fusing high-level semantic features and low-level spatial detail features.

As shown in Table 1, we compare the performance of the variants that fuse the features extracted from different layers of the VGG-16 model. We observe that the FFCT-L1-5 performs the best on OTB2013, OTB2015 and TC128 datasets. Moreover, the FFCT-L1-5 achieves better or competitive results on the UAV123 dataset. Overall, the FFCT-L1-5 performs the best while the FFCT-L1-4 and FFCT-L2-5 obtain sub-optimal performance. In addition, the performance of the other three variants (FFCT-L4-5, FFCT-L1-2 and FFCT-L3-5) is inferior to that of the variants mentioned above. Therefore, our proposed tracker learns to fuse the deep features of five levels for correlation tracking.

The Table 2 shows performance comparison of different components of our proposed tracker. To explore the effects of feature addition and concatenation on the fusion method, we compare the performance of FFCT-L1-5 and FFCT-Concat, and observe that FFCT-L1-5 outperforms FFCT-Concat on 3 out of 4 datasets. Therefore, we adopt feature addition to fuse the deep features of each two levels. Additionally, compared with the FFCT-L1-5, both FFCT_W_ECA and FFCT_W_RSE obtain better performance on the four datasets, which indicates that both ECA and RSE modules are beneficial for improving the performance of FFCT-L1-5. Based on FFCT-L1-5 and equipped with ECA and RSE modules, the proposed method EFFCT performs the best on the four datasets.

2) PARAMETERS ANALYSIS IN ECA MODULE

To analyze the influence of the kernel size k in our channel attention module, we implement and evaluate four different parameter settings on OTB2015 benchmark. As shown in Fig. 5, we empirically find that the proposed tracker EFFCT achieves the best performance when the kernel size of 1D convolution layer in ECA module is set to 3. Our ECA module becomes Accuracy Booster block (AB) proposed in [41] when the kernel size is set to 1. The tracking performance drops when the kernel parameter k is 7. Thus, we set the convolution kernel size in ECA module to 3 which is then used in all experiments.

C. RESULTS ON OTB

1) QUANTITATIVE EVALUATIONS

The proposed tracker EFFCT is compared with 26 existing state-of-the-art trackers, including DCF-based trackers with handcrafted features such as SRDCF [17], background-aware correlation filters (BACF) [55], MCCT using Hand-crafted features (MCCT-H) [31], efficient convolution operators using HOG and CN (ECO-HC) [21], SRDCF with decontamination (SRDCFdecon) [50]; DCF-based trackers with deep features such as HCF [9], HDT [11], DeepSRDCF [15], CCOT [10], ECO [21], multi-task correlation particle filter based tracker (MCPF) [48], MCCT [31]; the end-to-end DCF-based trackers such as DCFNet [13], CFNet [12], target-aware deep tracking (TADT) [51], unsupervised deep tracking (UDT) [49], context-aware based tracker (TRACA) [53], FlowTrack [24]; and some other deep trackers such as convolutional residual learning based tracker (CREST) [23], distractor-aware SiamRPN (DaSiamRPN) [52], gradient-guided network (GradNet) [29], GCT [30], SINT_flow [19], cropping-inside residual networks with 22 weighted convolution layers (CIResNet22) [25], C-RPN [34], Siamese box adaptive network (SiamBAN) [65]. All of these trackers are evaluated on OTB datasets [43], [44]. The sequences in the OTB datasets comprise of a wide variety of tracking challenges, such as illumination variations, scale variations, deformation, occlusion, fast motion, rotation, and background clutters. As the raw tracking results of some state-of-the-art trackers are not available, and in order to make representation concise, we only show the comparison results of the proposed tracker and 12 of the state-of-the-art trackers in Fig. 6, and the complete comparison results can be found in Table 3.

As can be seen from Fig. 6 (a), the proposed tracker EFFCT achieves the best performance in both precision and success plots. In particular, in terms of success plot, the EFFCT has obtained 69.1% AUC score which is 1.1% better than the second best performer TADT (68.0% AUC score) on OTB2013 dataset. As shown in Fig. 6 (b), the proposed tracker obtains competitive performance on OTB2015 dataset compared with other state-of-the-art trackers.

Table 3 shows the precision, AUC scores and running speed of four categories of trackers on OTB2013 and OTB2015 datasets. In Table 3, from top to bottom, the trackers are broadly categorized into four classes: DCF-based trackers with handcrafted features, DCF-based trackers with deep features, end-to-end DCF-based trackers and other deep trackers. In the group of end-to-end DCF-based trackers, the proposed tracker EFFCT achieves the best performance in AUC score on both OTB2013 and OTB2015 datasets. And the proposed EFFCT achieves similar performance with FlowTrack in precision, which exploits the flow information among frames for correlation tracking. Compared with the trackers using shallow networks including DCFNet, CFNet and UDT, the EFFCT achieves a significant performance improvement, which demonstrates the effectiveness of the proposed multi-level feature fusion method. Compared with the trackers in the fourth category in Table 3, the proposed method obtains the second best performance in terms of AUC scores on both the OTB2013 and OTB2015 datasets. This can be largely attributed to the fused deep features that exploit the spatial details and semantic information of the target, which makes our tracker robust to appearance variations of the target, and discriminative to background clutters. Overall, the trackers (MCCT, SiamBAN and ECO) obtain the best performance among all the compared trackers. However, the MCCT and ECO suffer from time-consuming algorithm computations and online model training. In summary, the proposed tracker achieves competitive performance against the state-of-the-art trackers in Table 3 and a close to real-time tracking speed (20 FPS).

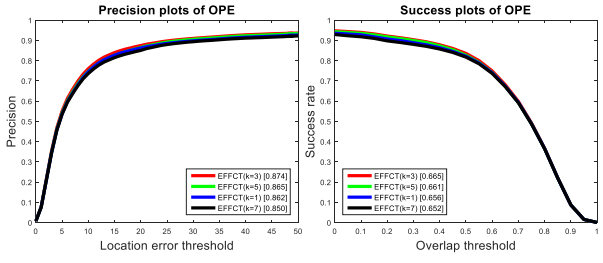
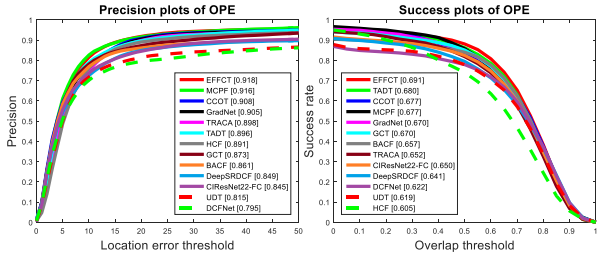


FIGURE 5. Precision and success plots for the analysis of convolution kernel size k in ECA module on OTB2015 benchmark. The legend of precision plot contains threshold scores at 20 pixels, while the legend of success rate contains area-under-the-curve score for each tracker.

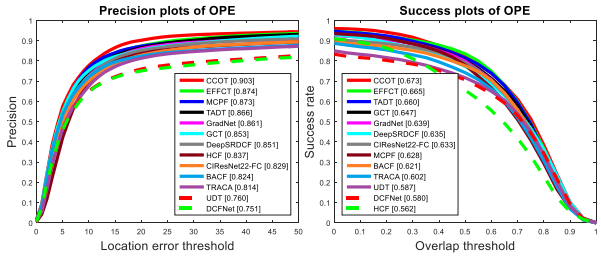
2) ATTRIBUTE-BASED EVALUATIONS

We also perform the attribute-based performance evaluation on the OTB2015 dataset containing 11 different tracking challenges: Illumination Variation (IV), Scale Variation (SV), DEFormation (DEF), Occlusion (Occ), Fast Motion (FM), In-Plane Rotation (IPR), Out-of-Plane Rotation (OPR), Background Clutter (BC), Motion Blur (MB), Out-of-View (OV) and Low Resolution (LR). In this attribute-based evaluation, the proposed tracker EFFCT is compared with 12 state-of-the-art trackers including CCOT [10], DeepSRDCF [15], BACF [55], HCF [9], MCPF [48], DCFNet [13], TADT [51], UDT [49], TRACA [53], GradNet [29], GCT [30] and CIResNet22 [25]. Table 4 shows the attribute-based performance comparison of the proposed tracker and the state-of-the-art trackers in term of precision and success rate on the OTB2015 dataset. The results demonstrate that our proposed tracker performs well in the attributes of BC, DEF, IPR, OPR, OV, SV and IV.

Although the proposed tracker can achieve excellent performance in most tracking challenges, it cannot perform well in the attributes of LR, Occ and FM. In contrast, the tracker CCOT achieves higher scores in the attributes of Occ, FM and MB, which can be attributed to its online updating scheme, powerful DCF model and large search region. While the proposed tracker EFFCT simply uses the tracking result of each frame to update target model, and its model will be contaminated when the long-term occlusion occurs, which may further lead to the subsequent tracking failure. In addition,



(a) Results on the OTB2013 dataset



(b) Results on the OTB2015 dataset

FIGURE 6. Precision and success plots on OTB2013 and OTB2015 datasets. The legend of precision plot contains threshold scores at 20 pixels, while the legend of success rate contains area-under-the-curve score for each tracker.

TABLE 3. Performance comparison with state-of-the-art trackers on OTB2013 and OTB2015 DATASETS. The performance is reported in terms of precision (PREC) at a threshold of 20 pixels and AUC for success rate. The best three results are shown in Red, Green, and Blue fonts, respectively. The notation $*$ denotes the running speed is reported by the authors as the source code is not available.

Trackers	OTB2013		OTB2015		FPS
	Prec	AUC	Prec	AUC	
SRDCF [17]	0.838	0.628	0.788	0.598	3.1
BACF [55]	0.861	0.657	0.824	0.621	15
MCCT-H [31]	0.856	0.651	0.841	0.633	23.5
ECO-HC [21]	0.874	0.652	0.856	0.643	13
SRDCFdecon [50]	0.870	0.653	0.825	0.627	1.3
HCF [9]	0.891	0.605	0.837	0.562	5.8
HDT [11]	0.889	0.603	0.848	0.564	5.3
DeepSRDCF [15]	0.849	0.641	0.851	0.635	<1
CCOT [10]	0.908	0.677	0.903	0.673	<1
ECO [21]	0.930	0.702	0.910	0.694	3.1
MCPF [48]	0.916	0.677	0.873	0.628	1.8
MCCT [31]	0.928	0.714	0.915	0.695	1.2
DCFNet [13]	0.795	0.622	0.751	0.580	45
CFNet [12]	0.807	0.611	0.748	0.568	40
TADT [51]	0.896	0.680	0.866	0.660	33
UDT [49]	0.815	0.619	0.760	0.587	44
TRACA [53]	0.898	0.652	0.814	0.602	65
FlowTrack [24]	0.921	0.689	0.881	0.655	12*
EFFCT	**0.918**	**0.691**	**0.874**	**0.665**	**20**
CREST [23]	0.908	0.673	0.838	0.623	2.4
DaSiamRPN [52]	0.890	0.655	0.880	0.658	95
GradNet [29]	0.905	0.670	0.861	0.639	80
GCT [30]	0.873	0.670	0.853	0.647	46
SINT_flow [19]	0.882	0.655	0.789	0.592	<1
CIResNet22 [25]	0.845	0.650	0.829	0.633	65
C-RPN [34]	0.884	0.675	0.853	0.663	36*
SiamBAN [65]	0.920	0.704	0.910	0.696	40

due to the inherent boundary effects of standard DCF and the lack of robust motion model, the EFFCT cannot cope with FM and MB as well as CCOT. In terms of LR challenge, the tracker GradNet performs the best.

3) QUALITATIVE EVALUATIONS

To qualitatively evaluate the performance of the proposed tracker EFFCT, we present the tracking results of the EFFCT and 9 existing trackers on key frames of 12 challenging sequences selected from OTB100 dataset, as shown in Fig. 7. The 9 existing trackers used for comparison include CCOT, TADT, GCT, BACF, GradNet, MCPF, CIResNet22, DCFNet and TRACA. In the following, we analyze the tracking challenges and their associated sequences in detail.

DEF: The sequences Diving and Singer2 contain DEF challenge. Compared with other trackers, the proposed tracker EFFCT is able to predict relatively better target positions in the two sequences, which can be attributed to the enhanced multi-level feature fusion method. In Diving, DCFNet tracker also performs well. In case of Singer2, in addition to the tracker EFFCT, TRACA and BACF can also successfully track the target object.

BC: The sequences Bolt2, Matrix and Ironman contain the BC challenge. In case of Bolt2, EFFCT, GCT and BACF trackers can successfully locate the target in the whole tracking process. While the tracker CCOT loses the target in the $24-th$ frame because it fails to distinguish the target from surrounding distractors. The trackers, EFFCT and CCOT, perform well for both Ironman and Matrix sequences. The results can illustrate that our proposed tracker can cope with the distractors in the complex scenes and is robust to background clutters.

IPR and OPR: The sequence Board contains the OPR tracking challenge. The trackers, CIResNet22, BACF and EFFCT, can tackle with this challenge successfully while other trackers slightly drift or lost the target. The main challenge of the MotorRolling sequence is IPR. As shown in Fig. 7, the trackers, EFFCT, GradNet and MCPF, are able to consistently locate target's positions in the whole tracking process.

IV: In the sequence Skating1, the main challenge in the last part of this sequence is IV. The tracker MCPF performs the best. The proposed tracker EFFCT can locate the position of the target well, but it fails to accurately estimate the scale of the target. The other trackers lost the target.

SV: The CarScale sequence main contains the SV and Occ challenges. As shown in Fig. 7, almost all the trackers can cope with the Occ challenge well. In terms of the SV challenge, EFFCT, MCPF and TADT perform better than the other trackers. However, none of the trackers perfectly solve the problem of SV in this sequence. The proposed tracker EFFCT adopts a fixed aspect ratio and only utilizes three scale factors to overcome this problem, which makes our tracker hard to cope with the large SV.

Occ: The main challenge of the Human4 sequence is the occlusion. The target undergoes partial and nearly full occlusion, and only EFFCT, CCOT and BACF trackers perform well and achieve stable tracking results. The multi-level fused features with rich semantic information can make our tracker robust to the occlusion challenge.

TABLE 4. Attribute-Based performance comparison of the proposed and existing state-of-the-art trackers in terms of success rate on OTB2013 and OTB2015 datasets. the AUC for success rate is reported. To save space, we abbreviate the trackers DEEPSRDCF and CIRESNET22 to DSRDCF and CIRESNET. The best three results are shown in Red, Green and Blue fonts, respectively.

Trackers	IV	SV	DEF	Occ	FM	IPR	OPR	BC	MB	OV	LR
EFFCT	64.4|66.7	67.3|63.9	70.4|62.4	66.4|63.0	62.9|62.6	67.0|63.8	68.0|64.7	67.1|66.5	62.9|64.9	70.5|61.9	53.8|59.3
CCOT	64.5|68.2	66.5|65.8	65.5|61.4	70.2|67.4	65.9|67.3	63.3|62.7	66.5|65.2	61.2|65.2	65.9|71.6	73.5|64.8	58.1|61.9
TADT	64.1|67.6	68.1|65.6	64.5|60.4	67.8|64.1	63.4|65.3	64.0|62.1	66.5|64.6	63.2|62.2	63.4|68.1	68.0|62.5	57.9|64.6
MCPF	62.1|62.8	67.3|60.4	65.1|57.0	67.1|62.0	62.3|58.3	63.6|62.0	66.2|61.9	64.6|60.1	62.3|59.7	66.0|55.3	59.4|59.8
GradNet	62.7|64.3	65.7|61.8	63.4|57.2	65.4|61.6	60.5|62.3	64.8|62.8	64.8|62.8	62.3|61.1	60.5|66.0	65.1|58.3	63.1|65.6
GCT	64.1|67.4	66.7|63.0	68.6|61.8	63.9|60.2	61.2|62.6	65.3|62.5	66.7|62.9	62.6|62.7	61.2|65.3	59.0|53.6	61.3|61.7
CIResNet	60.5|62.7	65.1|62.2	62.2|56.2	63.4|60.6	59.1|63.7	60.2|61.3	62.2|61.7	59.0|58.2	59.1|66.8	61.5|59.5	58.7|64.8
TRACA	62.3|62.2	61.3|55.8	68.8|56.1	64.4|57.0	57.8|57.2	61.0|58.0	64.0|59.3	61.8|59.3	57.8|60.0	63.0|54.7	39.2|49.5
HCF	56.0|54.0	53.1|48.8	62.6|53.0	60.6|52.5	57.8|55.2	58.2|55.9	58.7|53.4	62.3|58.5	57.8|57.3	57.5|47.4	55.7|42.4
BACF	61.9|64.3	61.5|57.9	64.4|58.3	64.2|57.6	61.0|60.2	63.5|58.4	64.3|58.4	62.9|62.5	61.0|59.7	63.3|55.2	43.6|51.2
DSRDCF	58.6|62.1	62.8|60.9	61.7|56.6	62.8|60.1	60.8|62.5	59.6|58.9	63.0|60.7	59.1|62.7	60.8|65.6	61.9|55.3	35.2|47.4
UDT	55.7|55.0	59.2|55.3	60.2|51.2	61.6|54.6	57.0|58.5	58.6|57.1	60.5|56.6	59.0|57.1	57.0|58.2	64.5|51.1	49.7|51.0
DCFNet	59.6|58.1	61.9|57.0	60.6|49.7	64.5|57.3	53.4|54.4	57.2|55.7	61.2|57.5	57.9|56.9	53.4|56.4	69.0|55.7	49.6|55.1

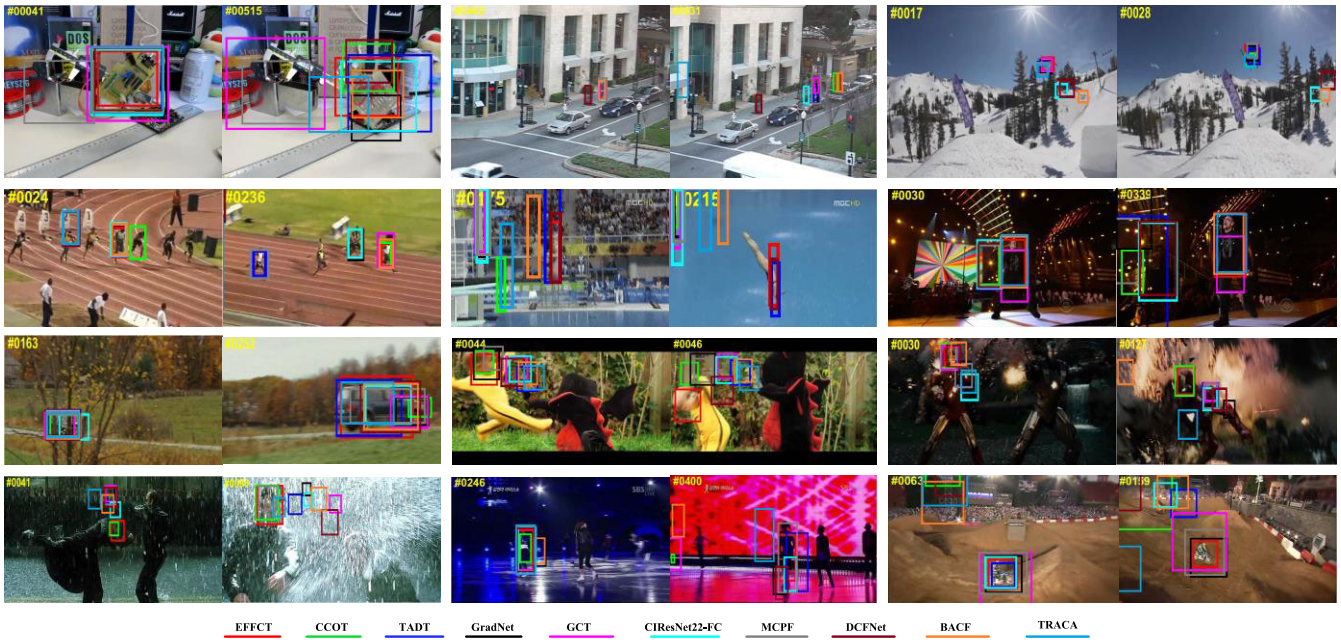


FIGURE 7. Qualitative comparisons of the proposed EFFCT tracker with current state-of-the-art trackers including CCOT [10], TADT [43], GradNet [29], GCT [30], CIResNet22 [25], MCPF [48], DCFNet [13], BACF [22] and TRACA [53] on 12 challenging sequences selected from OTB2015 dataset. The 12 sequences (from top to bottom, from left to right) are Board, Human4, Skiing, Bolt2, Diving, Singer2, CarScale, DragonBaby, Ironman, Matrix, Skating1 and MotorRolling.

FM and MB: As shown in Fig. 7, one of the main challenges of the Matrix sequence is the FM. EFFCT and CCOT are able to successfully track the target while the other compared trackers drift. The main challenges of the DragonBaby sequence are FM, MB and OPR around 44 − *th* frame. As can be seen from Figure 7, only the tracker EFFCT is able to predict relatively better target positions in this sequence. This may be because the EFFCT can tackle with these three challenges simultaneously.

LR and OV: The Skiing Sequence contains the LR challenge in which EFFCT, CCOT, TADT, GradNet, GCT, and MCPF trackers perform well. The Board and Ironman sequences also contain the OV challenge. The proposed tracker EFFCT is able to predict relatively better target positions in these two sequences, which demonstrates that our proposed tracker are robust to OV challenge.

D. RESULTS ON TC128

The temple-color-128 (TC-128) dataset [47] consists of 128 video sequences with 11 various challenging factors, which focuses more on color information. We also adopt success and precision plots to evaluate different trackers. We compare our tracker EFFCT with 11 state-of-the-art trackers, including ECO [21], adaptive spatially-regularized correlation filters (ASRCF) [57], MCPF [48], TADT [51], parallel tracking and verifying (PTAV) [56], DeepSRDCF [15], UDTplus [49], SiameseFC [20], HCF [9], DCFNet [13] and CFNet [12]. Figure 8 shows the performance comparison of the proposed

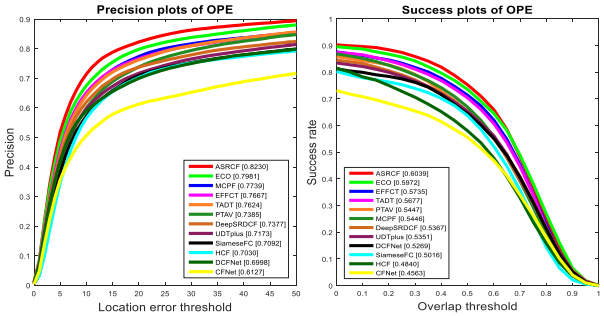


FIGURE 8. Precision and success plots on TC128 dataset. The legend of precision plot contains threshold scores at 20 pixels, while the legend of success rate contains area-under-the-curve score for each tracker.

trackers EFFCT with the state-of-the-art trackers on TC-128 dataset.

As shown in Fig. 8, the DCF-based trackers (ASRCF and ECO) with deep features perform the best, which can be attributed to their powerful correlation filters models. In particular, the proposed tracker EFFCT achieves the third best performance in success plot. Although the TC128 dataset is more challenging compared to the OTB2013 and OTB2015 datasets, the proposed EFFCT achieves competitive performance against the state-of-the-art trackers. These results demonstrate the effectiveness of the proposed multi-level feature fusion method. This experiment also illustrates that the EFFCT can tackle with various challenging factors well.

E. RESULTS ON UAV123

We also use UAV123 dataset [57] to evaluate the proposed method. The UAV123 dataset consists of 123 video sequences, which are captured from low-altitude UAVs and inherently different from videos in popular tracking datasets like OTB2015, TC128, and VOT2016. In this experiment, we compare our tracker EFFCT with 8 state-of-the-art trackers, including ECO [21], CCOT [10], real-time MDNet (RTMDNet) [61], GCT [30], ECO-HC [21], SRDCF [17], MEEM [59], and MUSTER [60]. Fig. 9 shows the comparative performance in terms of precision and success plots of the proposed tracker with other state-of-the-art trackers on UAV123 dataset. In terms of success plot, the proposed tracker EFFCT obtains the best performance. In terms of precision plot, the EFFCT achieves competitive performance against the RTMDNet that performs the best. Compared with the ECO method, which achieves impressive performance on other datasets like OTB2015 and TC128, the proposed tracker obtains better performance on the UAV123 dataset. Therefore this experiment further demonstrates that the proposed tracker is robust for various tracking challenges.

F. RESULTS ON VOT2016

We also validate the proposed tracker on the VOT2016 dataset [45], which consists of 60 video sequences with various challenges. Table 5 shows the experimental results of the proposed tracker and compared state-of-the-art trackers on

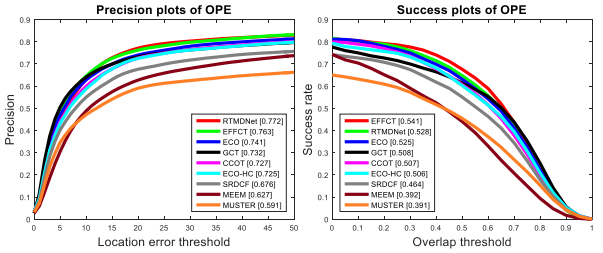


FIGURE 9. Precision and success plots on UAV123 dataset. The legend of precision plot contains threshold scores at 20 pixels, while the legend of success rate contains area-under-the-curve score for each tracker.

TABLE 5. Experimental results on the VOT2016 dataset.

Tracker	EAO ↑	Accuracy ↑	Failure ↓
HCF [9]	0.2203	0.4354	23.8569
TRACA [53]	0.1599	0.4600	37.9500
CCOT [10]	0.3310	0.5351	15.5817
DeepSRDCF [15]	0.2763	0.5249	20.3462
deepMKCF [45]	0.2323	0.5430	26.0329
MCCT-H [31]	0.3049	0.5714	21.7616
TADT [51]	0.3006	0.5443	19.9735
MDNet_N [18]	0.2572	0.5433	21.0817
CIResNet22[25]	0.3033	0.5388	19.3109
SA-Siam [46]	0.2911	0.5442	19.5602
RFD_CF2 [45]	0.2415	0.4531	22.9993
ECO [21]	0.3742	0.5407	11.6734
DCFNet [13]	0.2071	0.4900	24.9400
UDTplus [49]	0.3015	0.5158	20.4831
MemTrack [62]	0.2729	0.5457	24.3618
EFFCT	**0.3111**	**0.5485**	**20.3397**

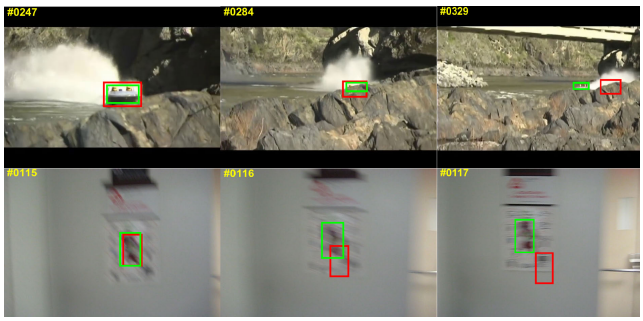


FIGURE 10. Failure cases (sequence Boat_ce2 from TC128 dataset, sequence BlurOwl from OTB100 dataset), where the red bounding boxes show our tracking results and the green ones are ground truths, respectively.

the VOT2016 dataset. The best three results are shown in red, green and blue fonts, respectively. The proposed tracker performs favorably against the state-of-the-art trackers on this dataset. Concretely, the EFFCT achieves the third best EAO score with the second best accuracy and a favorable robustness score. ECO equipped with robust DCF model achieves the best EAO score and robustness score. Overall, the proposed tracker performs well in terms of accuracy and robustness. This experiment demonstrates the effectiveness of our enhanced multi-level feature fusion method, which helps to distinguish between the target object and the background.

G. FAILURE CASES

In Fig. 10, we show two different failure cases of the proposed tracker. In the Boat_ce2 sequence, our tracker fails to track the

boat when it undergoes long term and full occlusion. In this circumstance, integrating a target re-detection module into our method may be able to improve the tracking performance. In the second row of Fig. 10, the proposed method drifts from the target when the target encounters fast motions (the motion displacement of the target between consecutive frames shown in Fig. 10 is larger than 45 pixels in the horizontal direction). This tracking failure can be alleviated by designing a robust target motion model.

V. CONCLUSION

In this paper, we propose an end-to-end multi-level feature fusion framework for correlation tracking. Specifically, in order to suppress the transmission of noise information and focus on important features, we develop an efficient channel attention mechanism to recalibrate the weight of high-level features that have semantic distinctions among different channels. Meanwhile, we also adopt a residual semantic embedding module that can adaptively involve semantic information from high-level features to guide the feature fusion. Extensive experimental results on five public datasets demonstrate that our algorithm performs favorably against the state-of-the-art trackers in both robustness and accuracy. In future, we will adopt more robust model update method to improve the robustness of our algorithm. Besides, the scheme of constructing CF layer on multiple hierarchical fused features respectively and then fusing their correlation responses will also be considered.

REFERENCES

[1] 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.

[2] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11, doi: 10.5244/C.28.65.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 886–893.

[4] M. Danelljan, F. S. Khan, M. Felsberg, and J. Van De Weijer, "Adaptive color attributes for real-time visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1090–1097.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[6] 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, pp. 211–252, Dec. 2015.

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

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2016, pp. 21–37.

[9] 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)*, Santiago, Chile, Dec. 2015, pp. 3074–3082.

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

[11] 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.

[12] 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.

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

[14] 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.*, San Francisco, CA, USA, Jun. 2010, pp. 2544–2550.

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

[16] Y. Sui, Z. Zhang, G. Wang, Y. Tang, and L. Zhang, "Real-time visual tracking: Promoting the robustness of correlation filter learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 662–678.

[17] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 4310–4318.

[18] 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.

[19] 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.

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

[21] 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.

[22] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Nov. 2014.

[23] 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)*, Venice, Italy, Oct. 2017, pp. 2574–2583.

[24] Z. Zhu, W. Wu, W. Zou, and J. Yan, "End-to-end flow correlation tracking with spatial-temporal attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 548–557.

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

[26] 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 Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4291.

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

[28] 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.

[29] P. Li, B. Chen, W. Ouyang, D. Wang, X. Yang, and H. Lu, "GradNet: Gradient-guided network for visual object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 6161–6170.

[30] J. Gao, T. Zhang, and C. Xu, "Graph convolutional tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4649–4659.

[31] N. Wang, W. Zhou, Q. Tian, R. Hong, M. Wang, and H. Li, "Multi-cue correlation filters for robust visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4844–4853.

[32] C. Zhang, H. Wang, J. Wen, and L. Peng, "Deeper Siamese network with stronger feature representation for visual tracking," *IEEE Access*, vol. 8, pp. 119094–119104, 2020.

[33] 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.

[34] 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)*, Long Beach, CA, USA, Jun. 2019, pp. 7944–7953.

[35] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.
[36] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.
[37] Z. Zhong, Z. Q. Lin, R. Bidart, X. Hu, I. B. Daya, Z. Li, W.-S. Zheng, J. Li, and A. Wong, "Squeeze-and-attention networks for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 13062–13071.
[38] Q. Wang, Z. Teng, J. Xing, J. Gao, W. Hu, and S. Maybank, "Learning attentions: Residual attentional Siamese network for high performance online visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4854–4863.
[39] Y. Yu, Y. Xiong, W. Huang, and M. R. Scott, "Deformable Siamese attention networks for visual object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 6727–6736.
[40] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient channel attention for deep convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11534–11542.
[41] P. Singh, P. Mazumder, and V. P. Namboodiri, "Accuracy booster: Performance boosting using feature map re-calibration," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Snowmass Village, CO, USA, Mar. 2020, pp. 873–882.
[42] Z. Zhang, X. Zhang, C. Peng, X. Xue, and J. Sun, "ExFuse: Enhancing feature fusion for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 273–288.
[43] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 2411–2418.
[44] 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.
[45] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pugfelder, L. Ä. Zajc, G. F. Dominguez, A. Gupta, A. Petrosino, A. Memarmoghadam, A. Garcia-Martin, A. Montero, A. Vedaldi, A. Robinson, A. Ma, A. Varfolomieiev, and Z. Chi, "The visual object tracking VOT2016 challenge results," in *Proc. Eur. Conf. Comput. Vis.*, vol. 9914, Oct. 2016, pp. 777–823.
[46] 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.*, Salt Lake City, UT, USA, Jun. 2018, pp. 4834–4843.
[47] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: Algorithms and benchmark," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015.
[48] T. Zhang, C. Xu, and M.-H. Yang, "Learning multi-task correlation particle filters for visual tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 365–378, Feb. 2019.
[49] 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)*, Long Beach, CA, USA, Jun. 2019, pp. 1308–1317.
[50] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 1430–1438.
[51] X. Li, C. Ma, B. Wu, Z. He, and M.-H. Yang, "Target-aware deep tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 1369–1378.
[52] Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, and W. Hu, "Distractor-aware Siamese networks for visual object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 103–119.
[53] 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.*, Salt Lake City, UT, USA, Jun. 2018, pp. 479–488.
[54] G. Liu and G. Liu, "Integrating multi-level convolutional features for correlation filter tracking," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 3029–3033.
[55] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1144–1152.
[56] H. Fan and H. Ling, "Parallel tracking and verifying: A framework for real-time and high accuracy visual tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 5487–5495.
[57] K. Dai, D. Wang, H. Lu, C. Sun, and J. Li, "Visual tracking via adaptive spatially-regularized correlation filters," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4665–4674.
[58] M. Mueller, N. Smith, and B. Ghanem, "A benchmark and simulator for UAV tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 445–461.
[59] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 188–203.
[60] Z. Hong, Z. Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, "MUlti-store tracker (MUSTer): A cognitive psychology inspired approach to object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 749–758.
[61] I. Jung, J. Son, M. Baek, and B. Han, "Real-time MDNet," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 83–98.
[62] T. Yang and A. B. Chan, "Visual tracking via dynamic memory networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 360–374, Jan. 2021.
[63] Z. Zhang, H. Peng, J. Fu, B. Li, and W. Hu, "Ocean: Object-aware anchor-free tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 771–787.
[64] 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.
[65] Z. Chen, B. Zhong, G. Li, S. Zhang, and R. Ji, "Siamese box adaptive network for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 6667–6676.
[66] S. Cheng, B. Zhong, G. Li, X. Liu, Z. Tang, X. Li, and J. Wang, "Learning to filter: Siamese relation network for robust tracking," 2021, *arXiv:2104.00829*. [Online]. Available: http://arxiv.org/abs/2104.00829
[67] B. Zhong, H. Yao, S. Chen, R. Ji, T.-J. Chin, and H. Wang, "Visual tracking via weakly supervised learning from multiple imperfect oracles," *Pattern Recognit.*, vol. 47, no. 3, pp. 1395–1410, Mar. 2014.
[68] B. Zhong, B. Bai, J. Li, Y. Zhang, and Y. Fu, "Hierarchical tracking by reinforcement learning-based searching and coarse-to-fine verifying," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2331–2341, May 2019.

GUANGEN LIU received the B.S. degree from Taiyuan University of Science and Technology, in 2012. He is currently pursuing the Ph.D. degree with Xi'an Jiaotong University, Xi'an, China. His research interests include visual object tracking and deep learning.

GUIZHONG LIU (Member, IEEE) received the B.S. and M.S. degrees in computational mathematics from Xi'an Jiaotong University, Xi'an, China, in 1982 and 1985, respectively, and the Ph.D. degree in mathematics and computing science from Eindhoven University of Technology, Eindhoven, The Netherlands, in 1989. He is currently a Full Professor with the School of Electronics and Information Engineering, Xi'an Jiaotong University. His current research interests include non-stationary signal analysis and processing, image processing, computer vision, and multimedia compression, transmission, and retrieval.