IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 4, 2019, accepted October 29, 2019, date of publication November 5, 2019, date of current version November 15, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2951600

Weighted Multiple Instance-Based Deep Correlation Filter for Video Tracking Processing

XU CHENG[1], YONGXIANG GU[1], BEIJING CHEN[1], YIFENG ZHANG[2], AND JINGANG SHI[3]

[1]School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[2]School of Information Science and Engineering, Southeast University, Nanjing 210096, China
[3]Center for Machine Vision and Signal Analysis, University of Oulu, FI-90014 Oulu, Finland

Corresponding author: Xu Cheng (chengxu20052005@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61802058, Grant 61911530397, and Grant 61972206, in part by the Equipment Advance Research Foundation Project of China under Grant 61403120106, in part by the Open Project Program of the State Key Laboratory of CAD&CG under Grant A1919, Zhejiang University, in part by the Startup Foundation for Introducing Talent of the Nanjing University of Information Science and Technology under Grant 2018r057, and in part by the Project funded by the China Postdoctoral Science Foundation under Grant 2019M651650.

ABSTRACT With the development of internet technology, the video data has been widely used in multimedia devices, such as video surveillance, webcast, and so on. Lots of visual processing algorithms are developed to handle the corresponding visual task, but the challenging problems still exist. In this paper, we propose a weighted multiple instances based deep correlation filter for visual tracking processing, which utilizes the importance of instances for training of deep learning model and correlation filter. First, the initial object appearance is modeled based on the confidence of the object and background at the first frame. During the tracking, the superpixel is used to capture the object appearance variations. Most importantly, our tracker can enhance the discriminative ability of the object using deep residual network and improve the tracking efficiency with correlation filter. Second, we introduce the sample importance into residual deep learning model to improve the training performance. We define the importance of each instance by computing the sore of all the pixels within the corresponding instance. Third, we update the parameters of deep learning network and correlation filter in a fixed interval frames to reduce the object drift. Extensive experiments on the OTB2015 benchmark and VOT2018 dataset demonstrate that the proposed object tracking algorithm outperforms the state-of-the-art tracking algorithms.

INDEX TERMS Visual tracking, deep learning, correlation filter, multiple instance learning.

I. INTRODUCTION

For video security, the sensitive video content needs to be protected before transmission. Data encryption is an efficient way to achieve this purpose [1], [2]. Compared with the text and binary data, the video data has large volume, and requires real-time processing. Since the traditional encryption algorithms don't consider the video characteristics, efficient video encryption algorithms should be designed for video data security.

Object tracking is an important issue in computer vision and is applicable in object recognition, action analysis, anomaly detection, intelligent transportation, pattern classification and human-computer interaction. In the past decade, various attempts, such as IVT [3], VTD [4], Fragment [5], TLD [6], MIL [7], SCM [8], are made to address some challenging issues. Existing tracking algorithms can be classified into either correlation filter or deep learning approaches. Despite of decades of extensive research, visual object tracking technique in the context of complex background and illumination appearance changes remains an open problem.

In recent years, correlation filter (CF) tracking has received considerable attention due to fast speed and higher accuracy. CF trackers learn a filter by the object state of the first frame, and then the learned filter tracks the object in successive frames. CF trackers lie in the approximate dense sampling achieved by circulantly shifting the object patch [9]. The remarkable runtime performance is achieved by solving the underlying ridge regression problem in Fourier domain [10]. Since the inception of CF trackers with single-channel features [10], [11], they have been developed

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

with multi-channel features [12], scale adaptation [13] and kernels [14]. In addition, many CF trackers improve the original work by adding context [15], [16], learning continuous filters and spatially regularizing the learned filters [17].

Beside the CF trackers, the tracking algorithms based on deep learning have also been developed at present. First, generic features of the object are learned on ImageNet object detection dataset, then fine-tuning the parameters domain-specific layers to be object-specific in an online way. MDNet [18] shows the best tracking performance on the VOT2015 challenge. Another approach consists of training a fully convolutional network and uses a feature map selection method to choose discriminative features between shallow and deep layers [19]. However, their computational complexity prohibits these deep learning trackers from being deployed in real application. To reduce the computational load, Siamese network based trackers [20]–[23], [40]–[42] and Generative Adversarial Networks (GAN) methods [43]–[45] are proposed to predict motion between consecutive frames. During the tracking, only a forward-pass is executed due to their simple network architecture and lack of offline fine-tuning mechanism. Their speed is up to 100fps on a GPU while results achieve competitive accuracy.

However, in many works, the CF and deep learning are simply combined and separately trained. Deep learning based trackers are often offline trained in an image classification dataset, and then online fine-tuning the parameters of deep learning network during the tracking. Thus, these trackers are less discriminative in various objects tracking domain. CF might result in tracking failure when the object suffers from heavy appearance variations. The complementary of deep learning and CF requires to be further researched.

The major contributions of this paper are of threefold:

(1) We propose a weighted multiple instances based deep correlation filter for visual tracking. We exploit superpixel as mid-level cue to extract the object feature for the appearance representation. The object's drifting can be reduced based on the learned object appearance model.

(2) Our object tracking method takes the sample importance into account for deep residual network and correlation filter procedure. The learned appearance model is used to evaluate the importance of instances. Then we utilize these weighted instances to update the parameters of network and filters.

(3) Our tracking method achieves a comparable performance to state-of-the-art on OTB2015 benchmark. The experimental results demonstrate that the proposed visual object tracking algorithm performs favorably against other conventional deep learning trackers.

This paper is structured as follows. Section II reviews related work on object tracking. In Section III, we briefly introduce the correlation filter tracker. In Section VI, we introduce our tracking scheme and its advantages over state-of-the-art algorithms in details. Section V gives the experimental results. Section VI concludes the paper.

II. RELATED WORKS

There are extensive surveys of visual object tracking in the literature [24]. In this section, we mainly focus on visual object tracking methods that are based on deep learning and correlation filters.

A. CORRELATION FILTERS BASED TRACKERS

Correlation filters for visual tracking have achieved the superior performance due to the computational efficiency in the Fourier domain. Correlation filters based tracking methods regress all the circular-shifted versions of the input features to a Gaussian function. The MOSSE tracker [10] encodes object appearance changes by optimizing the output sum of squared error. Later, several works have been developed to improve tracking performance, such as kernelized correlation filters [14], context learning [25], scale estimation [12], subspace learning [26], re-detection [27], spatial regularization [28], short-term and long-term memory [29]. Danelljan *et al.* [30] propose an effective correlation filter visual tracker that can cope with the scale changes of the object. Choi *et al.* [31] propose a tracker with an attention mechanism using previous object appearance and dynamics.

B. DEEP LEARNING BASED TRACKERS

Deep learning technique has brought remarkable performance improvements in many computer vision areas, such as object detection, tracking, body analysis, classification and semantic segmentation. They can build accurate visual object tracking algorithms without online adaptation due to powerful deep learning features.

Recurrent networks have been applied to visual tracking task [32]–[35] by considering temporal information. Wang *et al.* [19] simultaneously utilize shallow and deep convolutional features to consider contextual information of the object. Nam and Han [18] propose a training method by adding a classification layer to a convolutional network structure. Lei *et al.* [36] employ a similarity function trained by Siamese network to predict the position of the object. Tao *et al.* [37] develop a novel deep learning network which can be trained by a reinforcement learning scheme with weakly labeled benchmark. However, deep learning based trackers require frequent fine-tuning of the networks to capture the object appearance variations, which is slow and prohibits real-time tracking application. In [47], VITAL tracker is used to address the augmentation of positive samples and the issue of class imbalance via adversarial learning. Li *et al.* [48] propose the Siamese region proposal network (Siamese-RPN) which is end-to-end trained off-line with large-scale image pairs. In [49], the spatial distribution of feature is considered in structural support vector machine for visual tracking. Song *et al.* [50] apply residual learning to take appearance changes into account to reduce model degradation during the tracking. Cheng *et al.* [51] propose an Auto-Encoder pair model for visual tracking.

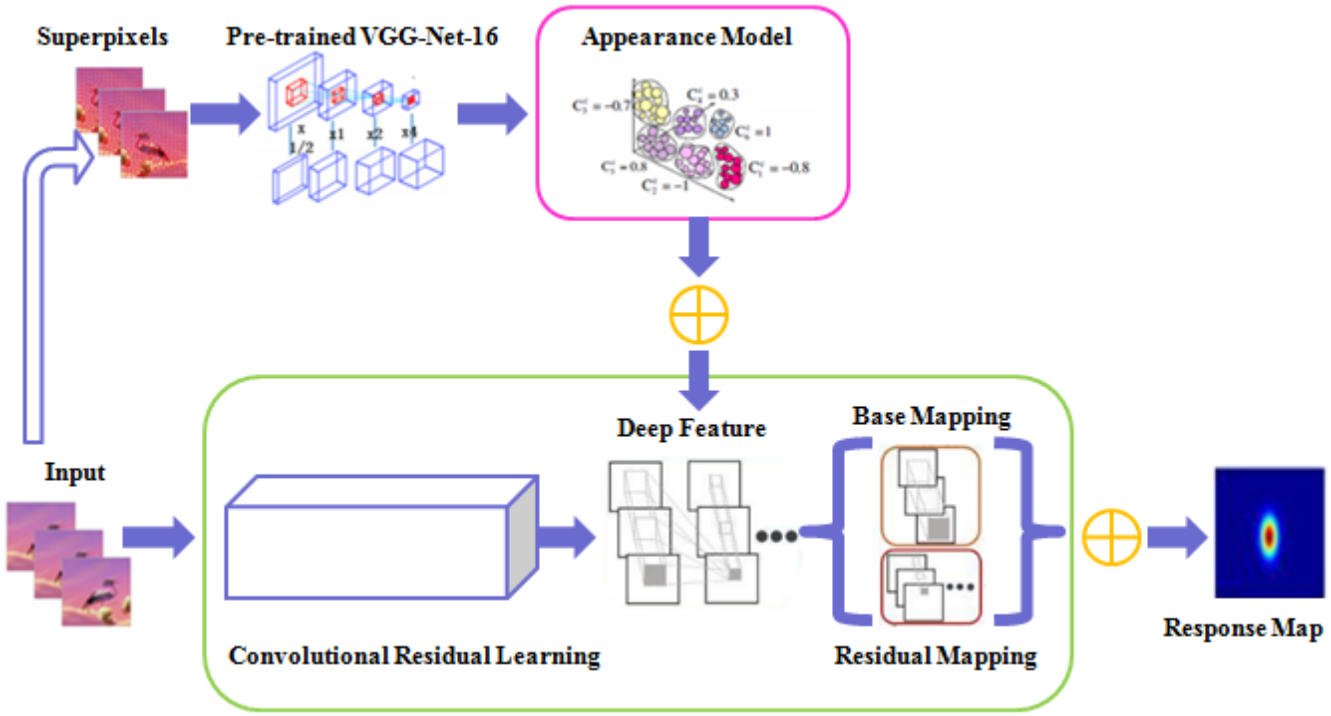


FIGURE 1. The workflow of the proposed object tracking algorithm.

C. COMBINING TWO FRAMEWORKS FOR TRACKING

Recently, deep learning technique has begun to play a key role in correlation filters [23]. MDNet, with deep feature extraction and a deep discriminative classifier, achieved significantly tracking performance. Yun *et al.* [38] improve the regularized correlation filter by using deep convolutional features. Zhang *et al.* [39] estimate the object state by combining different response maps which are obtained from convolutional features. However, there are too many channels in raw deep convolutional features to be handled in real-time, even though each correlation filter works fast. The deep feature redundancy is not fully suppressed.

III. CORRELATION FILTER FOR VISUAL TRACKING

DCF tracker learns a discriminative learner and predicts the object position by searching the maximum score in the response map.

$$\mathbf{w} = \arg\min_{\mathbf{w}} = \|\boldsymbol{y} - \mathbf{w} * \boldsymbol{x}\|^2 + \lambda \|\mathbf{w}\|^2 \tag{1}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ denote the input sample and its Gaussian function label, respectively. λ is the regularization parameter. $\mathbf{w}$ is a correlation filter learned from the minimum optimization problem of Eq.(1). The convolution operation between input $\boldsymbol{x}$ and correlation filter $\mathbf{w}$ can be formulated into a dot product in the Fourier domain.

The kernelized correlation filter is a well-known object tracking algorithm. Before detailing the proposed tracking algorithm, we briefly introduce the functionality of conventional correlation filters using a single-channel feature map. Correlation filters based trackers can be quickly trained and lead to a high tracking accuracy under low computational load based on the property of the circulant matrix in the Fourier domain.

Given the vectorized training feature map $\mathbf{z} \in \mathrm{R}^{wh \times 1}$ and vectorized object response map $\mathbf{y}$ obtained from a 2-D Gaussian with size $w \times h$ and variance σ_y^2, the vectorized correlation filter $\mathbf{w}$ can be estimated by:

$$\mathbf{w} = F^{-1}\left(\frac{\hat{z} \odot \hat{y}}{\hat{z} \odot \hat{z}^* + \lambda}\right) \tag{2}$$

where λ is a predefined regularization factor; F^{-1} represents an inverse Fourier transform function; $\hat{y}$ and $\hat{z}$ denote the Fourier-transformed vector of $\boldsymbol{y}$ and $\boldsymbol{z}$, respectively; $\odot$ denotes an element-wise multiplication; $\hat{z}^*$ is the conjugated vector of $\hat{z}$.

For a test feature map $\hat{z'}^* \in \mathrm{R}^{wh \times 1}$, the response map $\mathbf{r}$ can be obtained by:

$$\mathbf{r} = F^{-1}\left(\hat{\boldsymbol{w}} \odot \hat{\mathbf{z}}'^*\right) \tag{3}$$

Finally, the object position with the maximum peak position is obtained from a 2-D response map $\mathbf{R}^{w \times h}$ which is rebuilt from $\mathbf{r}$.

IV. THE PROPOSED TRACKING ALGORITHM

A. FORMULATION

The main flow of the proposed object tracking algorithm is shown in Fig.1. First, we utilize superpixel as mid-level

features for modeling the initial object appearance. The confidence of the object and background are computed by the sum of pixels responses within the corresponding instance. When a new frame of a video arrives, object candidates around the position of last frame are randomly drawn. Then the feature of each state is extracted using superpixel and deep residual network, respectively. Second, we consider the contribution for each instance and the contribution for each instance is obtained based on the initial object template. The importance of instances is used for training of deep learning model and correlation filter. During the tracking, the superpixel is used to capture the tracked object appearance changes. The state with the maximum confidence is regarded as tracking result. Third, it is necessary to update the appearance model in a fixed length frame interval to reduce the object drift. Finally, extensive experiments demonstrate that the proposed tracking algorithm outperforms the state-of-the-art visual object tracking algorithms.

B. MID-LEVEL BASED STRUCTURAL APPEARANCE REPRESENTATION

In this paper, we use superpixels as the mid-level cue to mine the structural information of the object. To model the object appearance, we first track the object in the first ten frames of a video with simple tracker [3], and then segment the surrounding region of the tracking result of the last frame into N superpixels. We find that the size of the candidate region does not have a direct impact on the number of superpixels. But computational complexity will grow as the candidate region of object increases. To reduce the computational time, superpixel segmentation is applied to the surrounding region of the object for effective object tracking. Each superpixel s_i $(i = 1, \ldots, N)$ is represented by a feature vector $\mathbf{f}_i$ which is a Haar-like feature. The locations of some small rectangles are randomly generated in each superpixel, and these rectangles consist of a set of feature templates which are used to widely capture specific object appearance variations. The number of rectangles in each superpixel ranges from 2 to 4. We randomly generate the weights and the heights. We define that the pixels in the same rectangle have the same weight, and these initial weight is randomly generated from the range (0, 1]. Each Haar-like feature is computed by the sum of weighted pixels. Then, we exploit the mean shift clustering method that can automatically cluster r classes based on the size of mid-level feature vectors of the object region. Superpixel members of the i-th cluster cover the regions of object to indicate how probable its superpixel members belong to the foreground or background. Therefore, a confidence measure for the i-th cluster is defined as:

$$Con_i^0 = \frac{S^+ - S^-}{S^+ + S^-}, \quad i = 1, \ldots, r \tag{4}$$

where S^+ represents overlapping between the size of cluster area and the object area, and S^- is the size of the cluster region outside the object region. Con_i^0 denotes an initial appearance model which is used as a prior knowledge of the object from the first ten frames.

During the tracking, the object state and its surrounding region based on object position of last frame are segmented into M mid-level features. We evaluate the confidence score of the k-th mid-level feature region as follows.

$$Con_k = \exp(\eta \times ||f_k - f_{c,i}||_2) \times Con_i^0, \quad k = 1, \ldots, M \tag{5}$$

where η denotes a normalization term (2 in this paper); f_k and Con_k represent the mid-level feature vector of the k-th area and the corresponding confidence score, respectively; $f_{c,i}$ denotes the feature center of the i-th superpixel area that f_k belongs to. We will utilize the confidence score of the superpixel to determine the quality of tracking results.

C. WEIGHTED MIL CORRELATION FILTER TRACKING

The deep feature vector of the object is extracted by the VGG-Net. It takes an object region X as the input. Then each pixel on the search region of the input is assigned a score based on the mid-level feature response. In this paper, we integrate all the sample contributions into the training process using weighted sum of instance probability. The contribution of each candidate instance can be obtained by accumulating the scores of all the pixels of superpixel area within the corresponding search region of the object.

$$w_l = \sum_{(i,j)\in S_l} v_l(i,j) \tag{6}$$

where $v_l(i,j)$ is the response score at location (i,j) within the l-th candidate instance S_l.

Then, we train the correlation filter with these importances of samples in the first frame and consider it as the object template.

During the training, n positive samples and m negative samples are obtained at the first frame. Positive samples are drawn around the object state ($\mathbf{I_1}$) of first frame as positive bag X^+, which satisfies $||\mathbf{I}_{pos}\text{-}\ \mathbf{I}_1|| < \alpha(\alpha = 5)$. Negative bag in an annular region specified by $\alpha < ||I_{neg} - I_1|| < \beta$, where α and $\beta = 2\alpha$ are inner and outer radii, respectively.

The positive samples probability is defined as follows.

$$p(y = 1|X^+) = \sum_{j=0}^{n-1} w_j p(y = 1|x_j) \tag{7}$$

where w_j denotes the importance corresponding to the j-th candidate in X^+; the candidates with the higher scores contribute more to the positive bag probability than those with the lower response scores. y is a binary label; $p(y = 1|x_j)$ is the posterior probability of candidate x_j to be positive. Sample x_j can be represented by concatenating mid-level feature and deep feature $\mathbf{f}(x_j)$. The posterior probability of x_j to be positive is obtained as follows.

$$p(y = 1|x_j) = \sigma\left(\ln\left(\frac{p(\mathbf{f}(x_j)|y = 1)p(y = 1)}{p(\mathbf{f}(x_j)|y = 0)p(y = 0)}\right)\right) \tag{8}$$

where σ is a sigmoid function.

All of the candidate samples contribute equally to the negative bag, which are far away from the tracked object region. The posterior probability of negative bag X^- can be written as:

$$p(y=0|\mathrm{X}^-)=\sum_{j=n}^{n+m-1}(1-p(y=1|x_j)) \tag{9}$$

Similar to MIL [11], correlation filter $H_K()$ is considered as a weak classifier and defined as

$$H_K(x_j)=\ln\left(\frac{p(\mathbf{f}(x_j)|y=1)p(y=1)}{p(\mathbf{f}(x_j)|y=0)p(y=0)}\right) \tag{10}$$

We assure that the features in $\mathbf{f}(x_j)=[f_1(x_j),\ldots,f_K(x_j)]^{\mathrm{T}}$ are independently distributed. Further, Eq.(10) can be represented as

$$H_K(x_j)=\sum_{k=1}^{K}\ln\left(\frac{p(f_k(x_j)|y=1)}{p(f_k(x_j)|y=0)}\right)=\sum_{k=1}^{K}h_k(x_j) \tag{11}$$

The conditional distributions are considered as a Gaussian function, which can be modeled as $p(f_k(x)|y=1)\sim N(u_1,\sigma_1)$ and $p(f_k(x)|y=0)\sim N(u_0,\sigma_0)$.

In addition, the parameters (u_1,σ_1) of model are updated by using the following Eq.(12) and Eq.(13).

$$u_1=\gamma u_1+(1-\gamma)\bar{u} \tag{12}$$

$$\sigma_1=\gamma\sigma_1+(1-\gamma)\sqrt{\frac{1}{n}\sum_{j|y_i=1}(f_k(x_{ij})-u_1)^2} \tag{13}$$

where n denotes the number of positive candidate instances; γ is a learning rate parameter, which is set to 0.8 in this work.

Finally, the objective function not only minimizes the regression loss, but also imposes a constraint for an effective deep network learning by maximizing the positive and negative samples in log-likelihood function $L(H)$.

$$L(H)=\sum_{s=0}^{1}\left(y_s\log(\sum_{j=0}^{n-1}w_jp(y=1|x_j))+(1-y_s)\log(\sum_{j=n}^{n+m-1}(1-p(y=1|x_j)))\right) \tag{14}$$

The Eq.(14) is solved using Stochastic Gradient Descent (SGD) method, which has been widely utilized in deep neural network training process. The correlation filter is used discriminate the object location.

D. UPDATE SCHEME

We define the length of the retained sequence as L which is set to 20. The object appearance template is updated in every U frames. IV-E and Fig. 2 show that the updating interval U is set to 10 to reach a compromise between computational complexity and the tracking accuracy. Then, we put the object tracked result of every frame into the end of L sequence. At the same time, the k-th information in L $(k<L, k=4)$ is deleted.

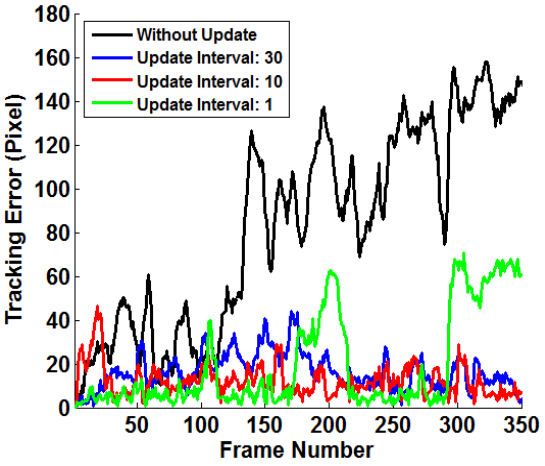


FIGURE 2. The influence of tracking errors and update frame interval.

TABLE 1. Average per-frame runs time on OTB2015 benchmark.

Update Interval U	Average Run Time/Frame
$U=0$	0.19s
$U=30$	0.26s
$U=10$	0.28s
$U=1$	0.46s

Frequently object template updating mechanism may easily introduce the noisy into the updating process. Therefore, the first frame information of the object needs to be retained to reduce the object drifting problem.

E. TRACKING

The object state from the first frame is initialized manually with a bounding box. A cropped rectangle region is utilized as training data to initialize the object template and fine-tuning parameters of network by using the object information of the first frame. During the tracking, a search region of the object at t-th frame is cropped based on the last location. The deep feature is extracted by VGG-Net and passed through the correlation filter framework to obtain the response map. The object position is determined in the t-th frame based on the location of the maximum response value. In the training stage, the object tracking results are regarded as the training samples which are utilized for updating the deep networks parameters with the loss function using SGD.

V. EXPERIMENTS

A. EXPERIMENTAL SETTING

The proposed object tracking algorithm is carried out on MATLAB platform with Intel Core 2 Duo 2.93GHz CPU and 2.96GB RAM. To facilitate a fair performance comparison for the proposed object tracking algorithm, we evaluate our tracking algorithm on a large benchmark OTB100 dataset that contains 100 challenging videos. OTB100 is manually

tagged with 11 attributes, which represents the challenging scenario in visual tracking. Most of the computation complexity of the proposed tracker is spent on mid-level feature generation. We employ the SLIC (Simple Linear Iterative Clustering) algorithm [13] to segment the search region into superpixel.

Our method is compared against 29 state-of-the-art object tracking algorithms. To facilitate a fair comparison, these object tracking algorithms are broadly categorized into three classes: (1) correlation filter trackers with hand-crafted features, including DSST [12], SAMF [13], KCF [14] and SRDCF [28]; (2) correlation filter trackers with deep learning, including CFNet [23], HCF [40] and DeepSRDCF [46]; (3) some representative trackers, including LCT [27], HCF [40], VITAL [47], CREST [50], AEPCF [51] and MCPF [54]. All source codes are provided by the authors' websites for fair comparison. In this paper, three metrics are used to evaluate all the tracking methods. First, distance precision error gives the percentage of frames whose estimated object position is within the predefined threshold of the ground truth. Second, center location error indicates the average Euclidean distance between the estimated center position and the ground truth. Third, overlap success rate is defined as the percentage of frames where the bounding box of the tracked object overlap surpasses a given threshold.

The CNN network is trained online using SGD. The tracked object appearance template is initialized in the first frame. During the tracking, the object appearance template is online updated by a standard SGD scheme. The regularization parameter λ is set to 0.005. All the parameters are fixed throughout all video sequences.

B. OVERALL PERFORMANCE

1) OTB2015 BENCHMARK [53]

The videos in OTB2015 benchmark are annotated with 11 attributes to describe the different challenges in the tracking, e.g., Illumination variation, occlusion, background clutter, motion blur, out-of-plane rotation, low resolution, deformation, scale variation, out-of-view, fast motion and in-plane rotation. These attributes are useful for analyzing the performance of tracker.

The quantitative comparisons, distance precision at 20 pixels, overlap success rate at 0.6 and tracking speed, are reported in Table 2. From Table 2, we can see that the proposed tracker performs favorably against state-of-the-art trackers in three metrics. Among the trackers, the MEEM tracker achieves the best performance with an average DP of 74.4% and OS of 64.9%. However, the proposed tracking algorithm can work well with DP of 85.4% and OS of 76.9%. The KCF, CSK and STC trackers achieve higher frame rate than our tracker that performs well at 28 frames per second.

The one-pass evaluation (OPE) protocol [10] is used to report the success and precision plots based on bounding box overlap metrics and the position error with respect to the ground truth. For success plots, the area under the curve (AUC) is computed. For precision plots, the distance precision at a threshold of 20 pixels (DP) is given. In addition, each method is able to process the frames per second (FPS) is discussed.

TABLE 2. Comparisons with state-of-the-art tracker on the OTB100 benchmark. Our tracker performs favorably against state-of-the-art trackers in distance precision (DP) at a threshold of 20 pixels, overlap success (OS) rate at an overlap threshold 0.6 and center location error (CLE). The first and second highest values are highlighted by red bold and blue bold fonts.

Trackers	DP(%)	OS(%)	CLE(pixel)	Speed(FPS)
LCT	76.7	61.8	25.8	27.4
HCF	81.2	71.4	20.9	4.5
KCF	73.2	59.3	35.5	**39.1**
MIL	47.5	37.3	62.3	28.1
Struck	65.6	55.9	50.6	10.0
CT	40.6	34.1	**78.9**	38.8
ASLA	53.2	51.1	73.1	7.5
SCM	64.9	61.6	54.1	0.4
MEEM	74.4	64.9	41.6	19.4
TGPR	70.5	62.8	51.3	0.7
TLD	60.8	52.1	48.1	21.7
CFNet	74.8	56.8	58.9	15.2
VITAL	83.6	62.2	36.4	28.9
CREST	82.4	59.3	27.3	34.6
Ours	**85.6**	**64.2**	**76.8**	**44.2**

Fig. 3 shows the precision and success plots over all the 100 videos, reporting AUC scores in the legend. MCPF and SRDCF, both based on correlation filters, achieve AUC scores of 61.7% and 62.1% respectively. Our tracker significantly outperforms VITAL with a relative gain of 2%. In addition, it is noticeable that the proposed tracking algorithm respectively runs 1.3 times and 1.6 times faster than the CREST and LCT. Overall, the experimental results on OTB100 demonstrate that the proposed tracking algorithm achieves competitive performance against the most relevant trackers in this benchmark.

2) VOT2018 BENCHMARK [52]

VOT2018 dataset is the one of the most recent public datasets for evaluating the performance of single object trackers in challenging scenarios. We evaluate our tracker on this dataset in comparison with 18 state-of-the-art methods. The VOT2018 benchmark consists of 60 videos with different challenging factors. Different from the evaluation criteria of OTB2015 benchmark, we exploit four primary measures to analyze the compared trackers performance: robustness (R), accuracy (A), Expected Average Overlap (EAO) and speed (in EFO units). The robustness measures how many times the tracker loses the object during the tracking. A tracking fails when the overlap ratio drops to zero. When such failure occurs, the tracker is reinitialized to continue tracking. The accuracy is the average overlap rate between the predicted object position and ground truth bounding boxes during the tracking process. The detailed experiments are reported in Table 5.

From Table 5, we observe that the weighted multiple instances based deep correlation filter tracker achieves the

TABLE 3. Attribute distributions for OTB100 benchmark.

OTB100	IV	OPR	SV	OCC	DEF	MB	FM	IPR	OV	BC	LR
IV	38	24	24	20	15	12	12	17	5	17	2
OPR	24	63	45	38	29	16	24	42	11	19	7
SV	24	45	64	33	29	21	28	35	11	17	9
OCC	20	38	33	49	25	14	19	25	12	14	5
DEF	15	29	29	25	44	10	15	17	5	12	3
MB	12	16	21	14	10	29	24	16	8	8	1
FM	12	24	28	19	15	24	39	22	11	10	2
IPR	17	42	35	25	17	16	22	51	8	14	6
OV	5	11	11	12	5	8	11	8	14	6	2
BC	17	19	17	14	12	8	10	14	6	31	1
LR	2	7	9	5	3	1	2	6	2	1	9

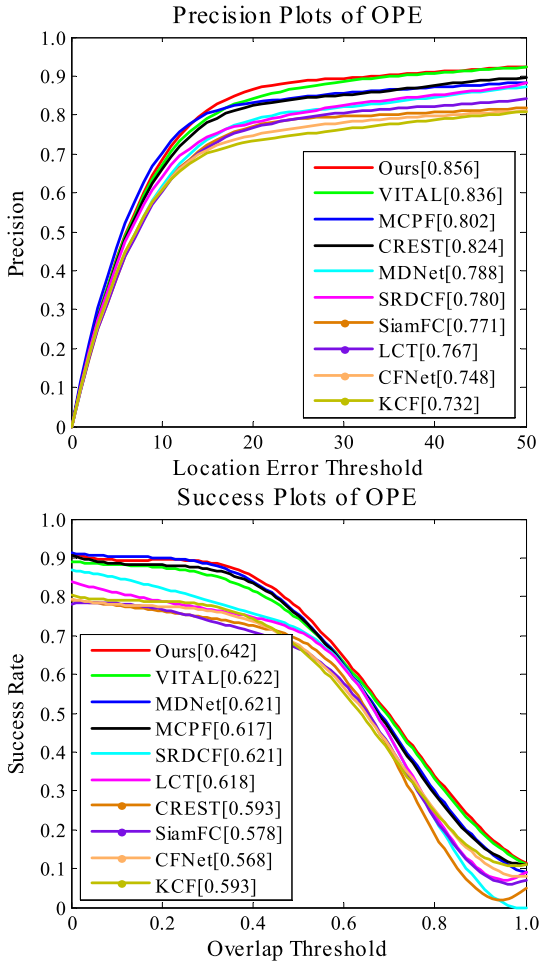


FIGURE 3. Distance precision and overlap success plots in the OTB100 benchmark sequences using one-pass evaluation (OPE). The legend contains the area-under-the-curve score for each tracker.

top-ranked performance on ECO and A. Our tracker provides an EAO score of 0.331 and maintains a competitive accuracy. In the comparison, KCF achieves the best speed. Among the top three deep feature based trackers, SiamFC obtains the best speed with an EFO of 0.235. Compared with VITAL, the proposed tracker obtains a performance gain of 0.6%. For the speed, the Frames-Per-Second (FPS) on VOT2018 is reported from the last row in Table 5. The reported speed is evaluated on a machine with an NVIDIA Titan Xp GPU, other results are provided by the VOT2018 official results.

C. QUALITATIVE EVALUATION

In the section, we give more detailed analysis of the strength and weakness of the trackers. Video sequences in OTB100 benchmark are categorized into 11 attributes, including illumination variation (IV), scale variation (SV), background clutter (BC), fast motion (FM), deformation (DEF), occlusion (OCC), out-of-plane rotation (OPR), in-plane rotation (IPR), motion blur (MB), out-of-view (OV) and low resolution (LR). Each attribute contains a specific challenging factor. Due to space limitation, five main challenging attributes in the paper are selected for the detailed analysis.

We compare our tracker with other five state-of-the-art trackers on ten challenging sequences. KCF tracking algorithm is based on a correlation filter framework learned from conventional HOG features. It performs well in handing significant background clutter and occlusion due to the robust HOG features representation. However, it drifts when the object undergoes heavy occlusion and doesn't redetect the object in the case of tracking failure (Tiger2 and Jumping). KCF tracker cannot cope with the background clutter. The struck tracker doesn't work well in rotation, background clutter and heavy occlusion. This is because that they are less discriminative in handling appearance change with one single classifier. TLD tracker is able to capture the lost object in the case of tracking failure. However, it ignores the temporal motion cues and does not capture the change of the object appearance (Tiger2, SUVs, and Singer2). TLD tracker updates its detector frame-by-frame introducing the noise into the object template. The proposed tracking algorithm

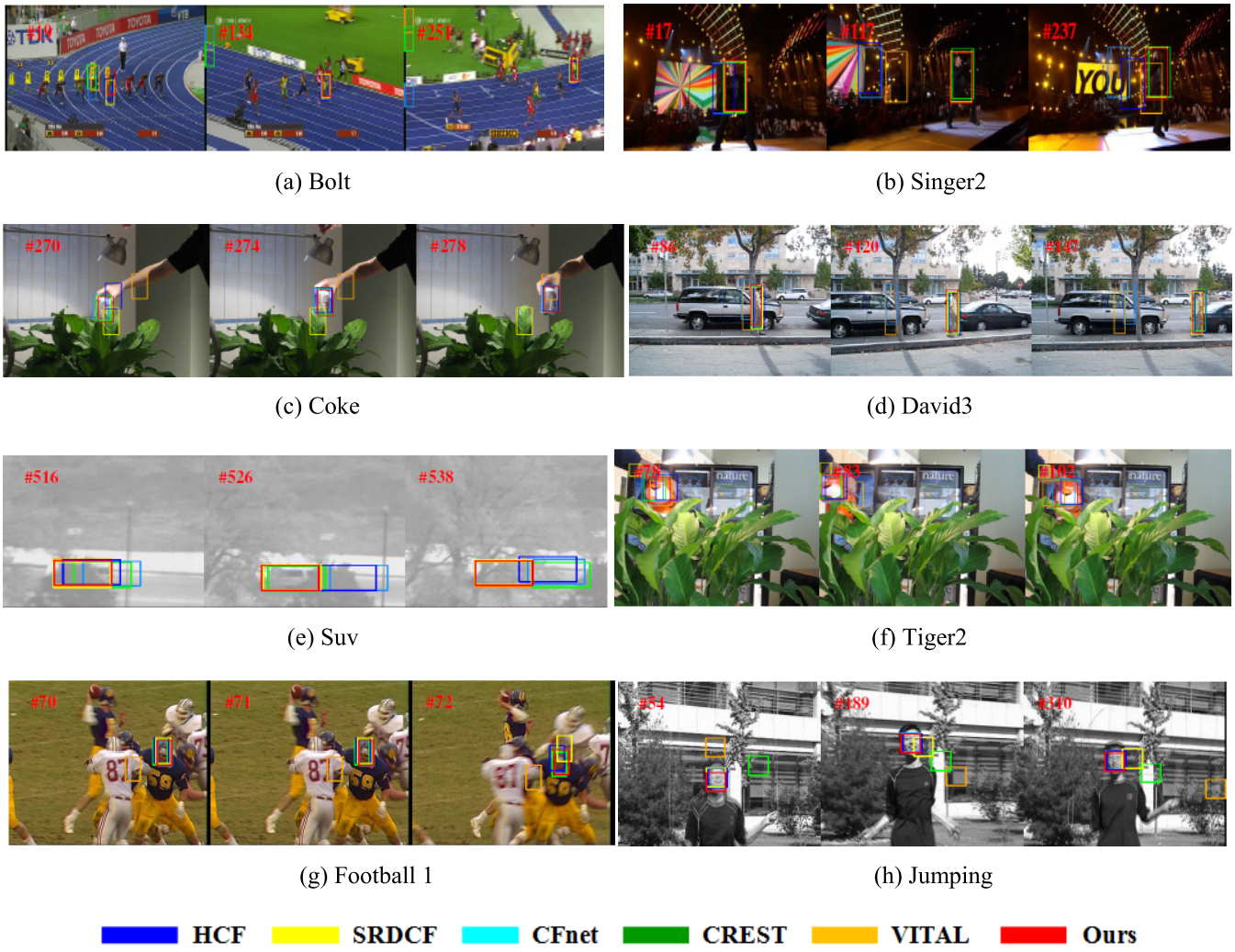


FIGURE 4. Screenshots of tracking results on 8 challenging benchmark sequences. For the sake of clarity, we only show the results of six trackers.

works well in estimating both the object positions and scales on the challenging sequences.

During the tracking, the tracked object often suffers from the occlusion and motion blur. In this case, it is difficult to distinguish the object from the background. However, occlusion is one of the major factors resulting in tracking failure. In Bolt sequence, the tracked object causes the in-plane rotation and partial occlusion. The qualitative evaluation results in several representative frames are reported in Fig. 4(a). The proposed tracking algorithm respectively outperforms the second best performance in this case in terms of success rate and precision. Unfortunately, MIL cannot recover the lost object due to drastic object appearance variations. TLD can't adapt the variations of the object size, leading to tracking failures when the object scale changes significantly. While other trackers CFNet, SRDCF, DSST, SAMP and DeepSRDCF drift to the background. Our tracker can achieve a satisfied result.

Fig. 4(b) and Fig. 4(g) show the evaluation results in several representative frames where no-rigid deformation occurred. Singer1 sequence contains scale variations, camera

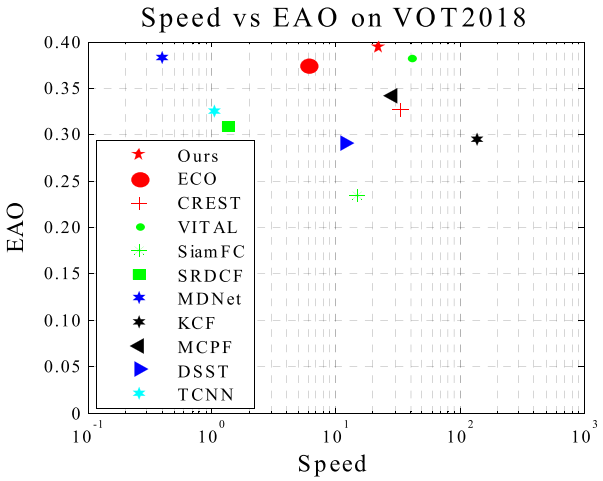


FIGURE 5. A comparison of EAO and the speed of state-of-the-art trackers on VOT2018 benchmark. We visualize the EAO with respect to the frames-per-second (FPS). The FPS axis is in the log scale.

motion as well as illumination variations, which result in most of the object tracking algorithms drift. In shaking sequence, DSST and SRDCF can track the object well

TABLE 4. Success rate scores with different attributes on OTB2015 benchmark. The bold fonts of results denote the best performance.

Tracker	OCC	LSV	FM	LI	TC	SO	DEF
Ours	**0.74**	**0.65**	**0.65**	**0.66**	**0.64**	**0.61**	**0.58**
VITAL	0.53	0.61	0.55	0.48	0.55	0.47	0.40
CREST	0.62	0.58	0.51	0.55	0.44	0.46	0.49
MDNet	0.62	0.58	0.57	0.51	0.61	0.50	0.47
SiamFC	0.54	0.49	0.52	0.46	0.60	0.44	0.41
SIT	0.39	0.43	0.35	0.35	0.41	0.39	0.34
MUSter	0.46	0.53	0.45	0.47	0.50	0.46	0.46
DSST	0.51	0.60	0.50	0.53	0.57	0.54	0.53
CFNet	0.50	0.47	0.41	0.44	0.48	0.53	0.55
KCF	0.34	0.40	0.36	0.36	0.34	0.29	0.37
HCF	0.54	0.47	0.51	0.50	0.43	0.40	0.38

TABLE 5. State-of-the-art in terms of expected average overlap (EAO), robustness, accuracy and speed on VOT2018 dataset. We compare with the state-of-the-art trackers, and only the top 10 are shown in the legend for clarity.

	DSST	MDNet	KCF	SRDCF	TCNN	MCPF	SiamFC	ECO	VITAL	CREST	Our
EAO	0.291	0.383	0.295	0.308	0.325	0.323	0.235	0.374	0.325	0.326	0.331
Robustness	0.90	1.12	1.35	0.74	0.96	0.85	0.24	0.72	0.276	0.337	0.56
Accuracy	0.44	0.55	0.54	0.42	0.54	0.56	0.53	0.54	0.566	0.569	0.571
FPS	12	0.4	138	1.37	1.05	29	15	6	40	33	12.06

except for some minor errors in some frames; while other conventional tracking algorithms lose the object. This is because that the light and the pose of the object are drastically varied due to the head shaking. Deer sequence show that the object suffers from appearance variations drastically. The proposed tracking algorithm succeeds in tracking the object and obtains reliable results in such challenging scenarios.

In addition, other attributes such as LR and OV, usually bears the ambiguous object appearance changes, leading to inferior tracking performance and visual object tracking failures.

In Table 4, our tracking method substantially outperforms all baselines in all attributes, demonstrating the effectiveness of our method in Table 4. The results demonstrate the importance of weighted multiple instances learning in visual tracking, especially in BC and OCC. In such scenarios, the proposed tracking algorithm can provide more reliable information.

VI. CONCLUSION

In this paper, a weighted multiple instances based deep correlation filter is developed. In the first frame, the initial object appearance is modeled with superpixel feature. During the tracking, residual network is used to extract the feature of the object, which improves tracking efficiency and captures the object appearance variations. Then, the sample importance is introduced into residual deep learning to train the parameters of network and correlation filter. Furthermore, we update the parameters of deep learning network and correlation filter in a fixed interval frames to alleviate drift to some extent. The experiments demonstrate that the proposed tracking algorithm significantly improves tracking performance over the state-of-the-art methods on OTB2015 benchmark and VOT2018 dataset.

REFERENCES

[1] Z. Pan, C.-N. Yang, V. S. Sheng, N. Xiong, and W. Meng, "Machine learning for wireless multimedia data security," *Secur. Commun. Netw.*, vol. 2019, Apr. 2019, Art. no. 7682306.
[2] J. Lei, J. Duan, W. Feng, N. Ling, and C. Hou, "Fast mode decision based on grayscale similarity and inter-view correlation for depth map coding in 3D-HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 3, pp. 706–718, Mar. 2018.
[3] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 125–141, 2008.
[4] J. Kwon and K. M. Lee, "Visual tracking decomposition," in *Proc. CVPR*, Jun. 2010, pp. 1269–1276.
[5] A. Adam, E. Rivlin, and I. Shimshoni, "Robust fragments-based tracking using the integral histogram," in *Proc. CVPR*, Jun. 2006, pp. 798–805.
[6] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, Jul. 2012.
[7] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.
[8] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2356–2368, May 2014.
[9] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2012, pp. 702–715.
[10] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2544–2550.
[11] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: Complementary learners for real-time tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1401–1409.
[12] M. Danelljan, G. Häger, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proc. Brit. Mach. Vis. Conf.*, Sep. 2014, pp. 1–5.
[13] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Computer Vision—ECCV Workshops*, L. Agapito, M. Bronstein, and C. Rother, Eds. Cham, Switzerland: Springer, 2015, pp. 254–265.

[14] 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.

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

[16] M. Mueller, N. Smith, B. Ghanem, "Context-aware correlation filter tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1396–1404.

[17] 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)*, Oct. 2016, pp. 472–488.

[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] 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.

[20] Z. Pan, H. Qin, X. Yi, Y. Zheng, and A. Khan, "Low complexity versatile video coding for traffic surveillance system," *Int. J. Sensor Netw.*, vol. 30, no. 2, pp. 116–125, 2019.

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

[22] 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. 1763–1771.

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

[24] P. Li, D. Wang, H. Lu, and L. Wang, "Deep visual tracking: Review and experimental comparison," *Pattern Recognit.*, vol. 76, pp. 323–338, Apr. 2018.

[25] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, "Fast visual tracking via dense spatio-temporal context learning," in *Proc. ECCV*, 2014, pp. 127–141.

[26] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proc. CVPR*, Jun. 2015, pp. 4902–4912.

[27] C. Ma, X. Yang, C. Zhang, and M.-H. Yang, "Long-term correlation tracking," in *Proc. CVPR*, Jun. 2015, pp. 5388–5396.

[28] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proc. ICCV*, Dec. 2015, pp. 4310–4318.

[29] 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. CVPR*, Jun. 2015, pp. 749–758.

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

[31] J. Choi, H. J. Chang, J. Jeong, Y. Demiris, and J. Y. Choi, "Visual tracking using attention-modulated disintegration and integration," in *Proc. CVPR*, Jun. 2016, pp. 4321–4330.

[32] S. E. Kahou, V. Michalski, C. Pal, P. Vincent, and R. Memisevic, "RATM: Recurrent attentive tracking model," in *Proc. CVPR Workshop*, Jul. 2017, pp. 1613–1622.

[33] Q. Gan, Q. Guo, Z. Zhang, and K. Cho, "First step toward model-free, anonymous object tracking with recurrent neural networks," 2015, *arXiv:1511.06425*. [Online]. Available: https://arxiv.org/abs/1511.06425

[34] D. Gordon, A. Farhadi, and D. Fox, "Re3 : Real-time recurrent regression networks for visual tracking of generic objects," 2017, *arXiv:1705.06368*. [Online]. Available: https://arxiv.org/abs/1705.06368

[35] T. Yang and A. B. Chan, "Recurrent filter learning for visual tracking," in *Proc. ICCV*, Oct. 2017, pp. 2010–2019.

[36] J. Lei, B. Peng, C. Zhang, X. Mei, X. Cao, X. Fan, and X. Li, "Shape-preserving object depth control for stereoscopic images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 12, pp. 3333–3344, Dec. 2018.

[37] R. Tao, E. Gavves, and A. W. M. Smeulders, "Siamese instance search for tracking," in *Proc. CVPR*, Jun. 2016, pp. 1420–1429.

[38] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, "Action-decision networks for visual tracking with deep reinforcement learning," in *Proc. CVPR*, Jul. 2017, pp. 2711–2720.

[39] J. Zhang, S. Ma, and S. Sclaroff, "MEEM: Robust tracking via multiple experts using entropy minimization," in *Proc. ECCV*, 2014, pp. 188–203.

[40] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proc. ICCV*, Dec. 2015, pp. 3074–3082.

[41] Z. Gao, S. X. Xia, Y. K. Zhang, R. Yao, J. Zhao, Q. Niu, and H. Jiang, "Real-time visual tracking with compact shape and color feature," *Comput., Mater. Continua*, vol. 55, no. 3, pp. 509–521, 2018.

[42] R. Amin, I. Inayat, and A. Li, "A bio-inspired global finite time tracking control of four rotor test bench system," *Comput., Mater. Continua*, vol. 57, no. 3, pp. 365–388, 2018.

[43] Y. Tu, Y. Lin, J. Wang, and J.-U. Kim, "Semi-supervised learning with generative adversarial networks on digital signal modulation classification," *Comput. Mater. Continua*, vol. 55, no. 2, pp. 243–254, 2018.

[44] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (GANs): A survey," *IEEE Access*, vol. 7, pp. 36322–36333, 2019.

[45] W. Fang, F. Zhang, V. S. Sheng, and Y. Ding, "A Method for improving CNN-based image recognition using DCGAN," *Comput., Mater. Continua*, vol. 57, no. 1, pp. 167–178, 2018.

[46] M. Danelljan, G. Häger, F. S. Khan, and M. Felsberg, "Convolutional features for correlation filter based visual tracking," in *Proc. ICCVW*, Dec. 2015, pp. 58–66.

[47] 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. CVPR*, Jun. 2018, pp. 8990–8999.

[48] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. CVPR*, Jun. 2018, pp. 8971–8980.

[49] Y. Zheng, L. Sun, S. Wang, J. Zhang, and J. Ning, "Spatially regularized structural support vector machine for robust visual tracking," *IEEE Trans Neural Netw. Learn. Syst.*, vol. 30, no. 10, pp. 3024–3034, Oct. 2019.

[50] 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. ICCV*, Oct. 2017, pp. 2555–2564.

[51] X. Cheng, Y. Zhang, Y. Zheng, and L. Zhou, "Visual tracking via auto-encoder pair correlation filter," *IEEE Trans. Ind. Electron.*, to be published.

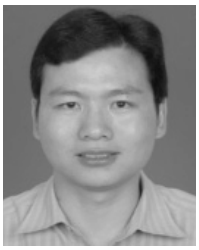
[52] M. Kristan *et al.*, "The sixth visual object tracking VOT2018 challenge results," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 1–52.

[53] 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.

[54] 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. 2018.

XU CHENG was born in 1983 in Taiyuan, China. He received the B.E. and M.E. degrees in information engineering from the Taiyuan University of Technology, in 2007 and 2010, respectively, and the Ph.D. degree in information and communication engineering from Southeast University, Nanjing, in 2015. He is currently an Associate Professor with the Nanjing University of Information Science and Technology. His research interests include computer vision, pattern recognition, and image processing.

YONGXIANG GU is currently pursuing the master's degree with the Department of Software Engineering, Nanjing University of Information Science and Technology, China. His research interests include deep learning and object tracking.



BEIJING CHEN received the Ph.D. degree in computer science from Southeast University, Nanjing, China, in 2011. He is currently an Associate Professor with the School of Computer and Software, Nanjing University of Information Science and Technology, China. His research interests include color image processing, image forensics, image watermarking, and pattern recognition.

YIFENG ZHANG was born in Wuhu, China. He received the B.E. degree in electrical engineering from Southeast University, Nanjing, China, in 1984, the M.E. degree in computer engineering from the Harbin Institute of Technology, Harbin, China, in 1989, and the Ph.D. degree in electrical engineering from Southeast University, Nanjing, in 1999. From 1999 to 2001, he was a Postdoctoral Fellow of the Department of Radio Engineering, Southeast University. In 2009, he joined Southeast University, where he is currently an Associate Professor with the School of Information Science and Engineering. He has published an academic book in Chinese. His research interests include visual tracking, watermarking and information hiding, chaotic neural information processing, and machine learning. He has received the Best Paper Award from the IEEE Asia Pacific Conference on Circuits and Systems.

JINGANG SHI received the B.S. and Ph.D. degrees from the Department of Information Engineering, School of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China. He is currently a Postdoctoral Researcher with the Center for Machine Vision and Signal Analysis, University of Oulu, Finland. His current research topics include image super-resolution, face analysis, and biomedical signal processing.