IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 22, 2020, accepted March 9, 2020, date of publication March 17, 2020, date of current version March 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2981525

ADT: Object Tracking Algorithm Based on Adaptive Detection

YUE MING, (Member, IEEE), AND YASHU ZHANG
Beijing Key Laboratory of Work Safety Intelligent Monitoring, School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Yue Ming (myname35875235@126.com)

This work was supported by the Beijing Natural Science Foundation (Grant no. L182033), and fund for the Fundamental Research Funds for the Central Universities (Grant no. 2019PTB-001).

ABSTRACT Object tracking is one of the most fundamental and important fields in computer vision with a wide range of applications. Although great progress has been made in object tracking combined with detection, there is still enormous challenges in real-time applications and for the computer cannot effectively capture the temporal correlations of targets and background clutter. In order to improve the performance of tracking algorithms under complex unconstrained conditions, we propose a novel tracking framework based on adaptive detection, called adaptive detection tracking (ADT). First, we exploit the temporal correlation of the recurrent neural network to predict the target's motion direction and efficiently update the region of interest (RoI) in the narrow range of the next frame. Then, the algorithm utilizes the correlation filter to initialize the defined region of interest based on the threshold. If the Interaction of Union (IoU) of the predicted bounding box and the groundtruth bounding box is greater than the set threshold, the predicted bounding box will be directly output as the tracking results, whereas the detection is adaptively carried out in the determined RoI. Finally, the predicted bounding box refines the direction model as the input of the next frame to complete the whole tracking flow. Our proposed adaptive detection tracking mechanism can efficiently realize non-frame-by-frame adaptive detection with excellent tracking accuracy and is more robust in the unconstrained scenes, especially for occlusion. Comprehensive experiments demonstrate that our approach consistently achieves state-of-the-art results and runs in real-time on six large tracking benchmarks, including OTB100, VOT2016, VOT2017, TC128, UAV123 and LaSOT datasets.

INDEX TERMS Recurrent neural network, adaptive detection, object tracking, correlation filtering, model compression.

I. INTRODUCTION

Generally, object tracking utilizes the bounding box of the target to predict the target's position and the whole trajectory in the subsequent frames, which has been widely used in various aspects of human daily life and military security [1], [2]. After sustainable development from the classic Kalman filter [3], particle filter [4], meanshift algorithm [5] to correlation filter [6] and deep-learning based algorithms [7], more and more excellent trackers have been proposed and achieved more robust performance in the large tracking benchmarks. However, most of the current datasets generally focus on the specific challenges of the visual tracking. In complex unconstrained conditions, there are still numerous interference factors including camera dithering, scale variance, target occlusion and so on [8]. At the same time, most of the existing trackers have limited tracking speeds, and jamming phenomenon of the video is obvious during tracking. Therefore, the temporal information between adjacent frames should be considered to better establish the location relationship of the target to reduce the dependency on object detection and the computational cost. Our previous work [9] has already developed a direction model, which can effectively narrow down the search area of the target in the next frame. However, our previous tracker [9] based on frame-by-frame object detection significantly slowed down the tracking speed. How to realize the non-frame-by-frame adaptive detection in real-time application and avoid boundary effect and motion shift

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

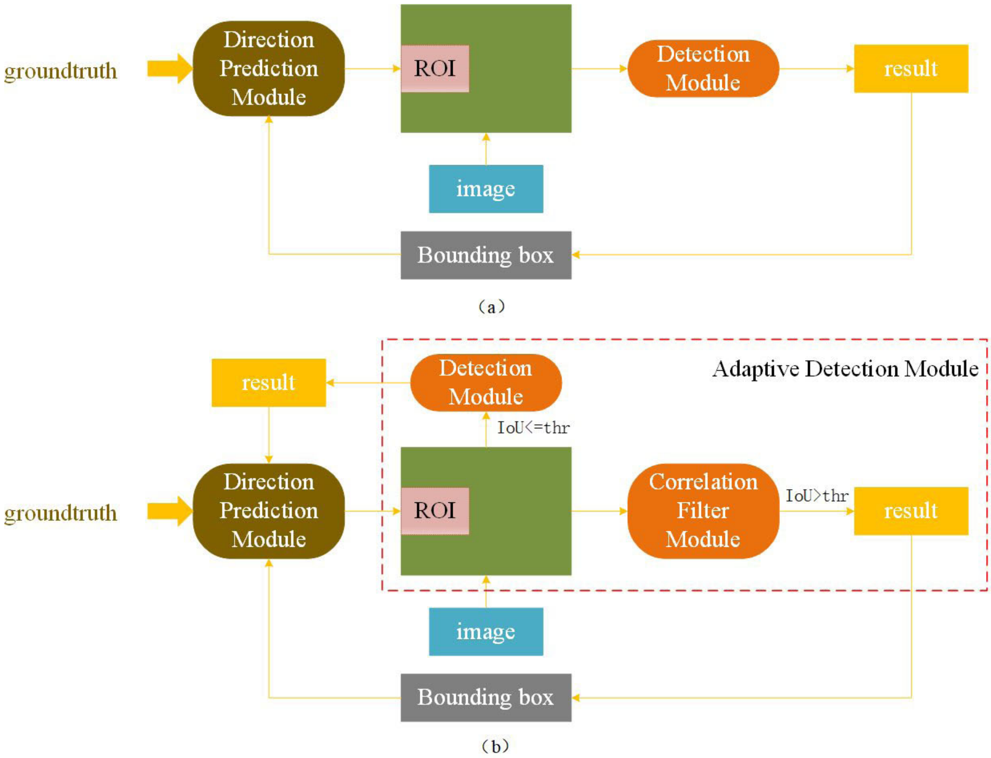


FIGURE 1. The frameworks of our previous work (a) and our improved framework (b). The two pipelines: Direction Prediction Module and Adaptive Detection Module.

is still a problem. Meanwhile, the tracking performance will be challenged by partial or fully occlusion with degressive performance.

Here, a novel object tracking algorithm is proposed by adaptive detection, called adaptive detection tracker (ADT). First, the temporal correlation based on recurrent neural network is exploited to predict the motion direction of the next frame and the region of interest (RoI) is determined along this direction. Then, the adaptive tracker developed from correlation filter and the direction prediction model is proposed, which is initialized by the predicted RoI. When the IoU of the predicted bounding box and the groundtruth is greater than the set threshold, the predicted bounding box will be directly output as the tracking results. Finally, the results of adaptive detection model are transmitted to the direction prediction model, which updates as the input of the next frame, for refining the whole tracking. As shown in Figure 1, our proposed algorithm effectively combines the adaptive detection module, which can adaptively realize object detection based on the variances of object and scene, compared with our previous frame-by-frame detection tracking mechanism. The improved algorithm can effectively take a trade-off between the speed and accuracy. The main contributions in this paper are summarized as follows:

1) Adaptive detection: we propose a novel adaptive detection mechanism to realize non-frame-by-frame detection, which can further update and localize an adaptive bounding box in real-time tracking as the object changes shape and size in complex unconstrained conditions.
2) Occluded robustness: our proposed tracker combined with direction prediction model and correlation filter, which can make full use of temporal reliability and spatial effectiveness to highlight the importance of the motion state overtime, especially for heavy occlusion.
3) Superior performance: extensive experiments conducted on six large tracking benchmarks, namely OTB100 [10], TC128 [11], UAV123 [12], VOT2016 [13], VOT2017 [14] and LaSOT [15] datasets, demonstrate that our proposed algorithm obtains better performance in accuracy and efficiency compared with the state-of-the-art algorithms.

The rest of this paper is organized as follows. We survey related works on detection and tracking in Section 2. Our adaptive detection tracking framework is described in Section 3. The following Section 4 contains the experimental evaluations and results. Finally, we conclude the paper in Section 5.

II. RELATED WORK

Since the key point of this paper is tracking based on adaptive detection, we provide a brief review on two aspects, which can be roughly categorized as object detection and object tracking.

A. OBJECT DETECTION

With the emergence of convolutional neural network (CNN), deep learning methods have been proposed to object detection [16] on many challenging datasets, which can be divided into two categories. One focuses on improving accuracy, called two-stage detection algorithm, while the other focuses on speed, called one-stage detection algorithm.

Two-stage detection first extracts several candidate regions from an input image, then obtains convolution feature using CNN [17], and finally classifies and regresses each candidate region. One representative method, called R-CNN [18], extracted about 2k candidate regions by the traditional method and obtained convolution features on each candidate region by CNN, which is 19% mAP higher than the best traditional detector. Due to disadvantages of complex region selection and feature repeated extraction, SPP-Net [19] proposed a pyramid pooling layer to extract features on the whole image and was 170 times faster than RNN. Fast R-CNN [20] changed the feature pyramid pooling layer into RoI pooling layer and put the classification and regression into an end-to-end network for training. Faster R-CNN [21] proposed Region Proposal Network (RPN) to extract the candidate region of an image. RPN and detection network shared convolution layer, and the 73.2 mAP detection accuracy and 5 fps detection speed were achieved. Object detection for 3D images is also proposed to many areas such as industrial processing. For example, an effective microscopic detection method based on the form-invariant [22] was proposed to detect the brain section of three-dimensional imaging, which can effectively realize the automated silicon-substrate ultramicrotome (ASUM). Although Faster R-CNN had made great improvement in the detection speed, the accuracy is improved at the expense of the algorithm's speed. Therefore, one-stage algorithms have emerged.

One-stage detection algorithm utilizes an end-to-end network to classify and locate targets directly without region proposal. Thus, it has more advantages in the computational speed. YOLO [23] and SSD [24] were the two most representative methods. YOLO transformed the classification problem into a regression problem, and only used a CNN network to directly predict the category and location information of different targets. However, the number of positive and negative samples in training was extremely unbalanced, which resulted in the low accuracy of the training model. Two improved algorithms for YOLO were proposed successively, namely YOLO v2 [25] and YOLO v3 [26]. The accuracy of YOLO v2 is improved by 15.2% compared with YOLO, and the detection effect of YOLO v3 on small targets was much better than YOLO. On the other hand, SSD combined the regression of YOLO and anchor mechanism of Fast R-CNN to obtain target location and category information. The difference is that YOLO only used the top-level feature map for classification and regression, while SSD used the feature pyramid structure for detection. Thus, SSD can improve the detection performance of small targets due to multi-scale object detection. Some improved methods based on SSD were also proposed, such as RON [27] and DSOD [28]. Although the one-stage detection algorithm can achieve real-time detection speed, the accuracy still needs to be improved due to the extremely unbalanced number of positive and negative samples. In addition, sufficient shallow feature extraction drops the detection effect for small targets.

B. OBJECT TRACKING

We discuss the trackers, which can be roughly categorized as: correlation filter trackers and deep learning trackers. In addition, we have also summarized the similar tracking methods combined with detection, e.g. region proposal methods.

1) CORRELATION FILTER TRACKERS

KCF algorithm [6] is one of the most classical trackers based on correlation filter. KCF took full advantage of the property that cyclic matrix can be diagonalized by Fourier matrix, and converted the matrix operation into Hadamad product to reduce the computational complexity. SRDCF [29] improved the boundary effect caused by the periodicity assumption of cyclic matrix. Deep SRDCF [30] replaced hand-craft features with CNN features to improve the tracking performance. C-COT [31] utilized deeper multi-layer convolution features and a continuous spatial interpolation conversion to solve the problem of different resolution. ECO algorithm [32], developed from C-COT, changed the model update strategy to sparse updating for reducing the model size and sample set size. The above tracking methods based on correlation filter can be divided into two categories: ones based on hand-craft features, such as KCF, SRDCF, CSF [33] and Staple [34]; ones based on CNN features, such as Deep-SRDCF, C-COT and ECO, etc. The former ones are fast but have yet to be improved in accuracy, while the latter ones are excellent in performance but the speed is low.

In order to overcome the challenges above, recently, more improved correlation filter algorithms have been proposed for object tracking. Background aware correlation filter (BACF) for robust tracking [35] was presented to improve the location of targets with higher precision in complex scenarios. A novel approach to repress the aberrances happening during the detection was presented, called aberrance repressed correlation filter (ARCF) [36], which outperformed other 20 state-of-the-art trackers based on discriminative correlation filter (DCF) on different UAV datasets. However, the above methods only demonstrate the top performance on some specific datasets, lack of the versatility testing on the different datasets. Although the adaptive spatially regularized correlation filters (ASRCF) [37] was favorably against many current algorithms, two CF models within them estimated the location and scale separately, thus making the tracker slower than the other CF methods.

2) DEEP LEARNING TRACKERS

In 2013, Deep Learning Tracking (DLT) [38] was first proposed by Wang *et al.* The general target feature was obtained by pre-training and tracking model with a large-scale dataset. The model had stronger classification performance by fine-tuning the pre-train model with limited training data. Representative algorithms of object tracking based on deep learning include CNN-SVM [39], FCNT [40], MDNet [41], TCNN [42], SiamFC [43], [44] and so on. Accurate Tracking by Overlap Maximization (ATOM) [45] consisted of dedicated target estimation and classification to guarantee high discriminative power in presence of distractors. Bhat *et al.* [46] proposed an end-to-end architecture, called Discriminative Model Prediction (DiMP) for tracking to fully exploit both target and background appearance information. However, most tracking algorithms based on CNN does not have satisfactory support for temporal correlation. The location information of previous frames holds the key to predicting the target location in the next frame and improving the tracking performance.

Therefore, many scholars introduce the recurrent neural network into object tracking, such as RTT [47], ROLO [48], SANet [49] and so on. Huang *et al.* [50] proposed a Bidirectional Tracking for tracking based on recursive orthogonal least squares to update model strategy for model drift problem. However, the original RNN [51] has encountered vanished gradient problem when the image sequence is too long, which may hinder information learning spanning over a long sequence. Yan *et al.* [52] developed an algorithm for robust long-term tracking, called ''Skimming-Perusal'' Tracking (SPLT), which can effectively choose the most possible regions from a large number of sliding windows. Simultaneously, the tracking performance is largely dependent on the object detection, so the bottleneck of detection limits the improvement of object tracking.

3) TRACKING COMBINED DETECTION

Currently, many advanced object tracking methods related to region proposal and detection have been proposed to improve the robustness of object tracking, including KCF, Siamese network and so on. A fast and robust approach [53] was first presented by integrating an adaptive object detection within a kernelized correlation filter (KCF). Li *et al.* [54] presented a novel gradient-guided network (GradNet) to capture the temporal variations of targets and background and updated the template in the Siamese network. To achieve accelerated tracking, Siamese region proposal network (Siamese-RPN) [55] was also proposed for feature extraction and region proposal prediction, which had reach at 160 fps with superior performance on the VOT datasets. Increasing deeper and wider Siamese network (SiamDW) [56] was also proposed for real-time visual tracking, which can effectively control receptive field size and network stride. In order to solve similar distractors and large variation, a multi-stage tracking framework [57], namely Siamese Cascade RPN (C-RPN), was proposed to make location more accurate. Li *et al.* [58] further developed a simple yet effective spatial aware sampling strategy for the Siamese tracking with very deep networks (SiamRPN++), which can successfully train a ResNet-driven tracker with significant performance improvement. For real-time scale and angle estimation, a one-shot Siamese network [59] utilized a single search network to estimate the target bounding box. Xu *et al.* [60] utilized the fine-tuning Alexnet to train the Siamese network of fused response map and weighted the fusion of score map for feature extraction and performance improvement. Although a large number of works focus on improving the accuracy and robustness of object tracking with detection and have achieved great progress, how to take a trade-off between the speed and accuracy of temporal related update on the different datasets under complex unconstrained conditions has not yet been fully resolved and need long-term studies, especially for heavy occlusion.

III. ADAPTIVE DETECTION TRACKING

In order to reduce the dependency of target detection and overcome occlusion, we propose a novel object tracking framework, called adaptive detection tracking (ADT), as shown in Figure 2. First, given an initial frame of target video, the target's motion direction of the next frame can be obtained by the direction prediction module [9]. Then, the region of interest (RoI) is determined along this direction, that is, the rough positioning of the target for adaptive detection. Next, RoI serves as the input of the adaptive detection module, and the output includes the precise location and category information of the predicted target. In the adaptive detection module, the correlation filter is initialized with the RoI instead of directly detecting the RoI. If the IoU of the predicted bounding box of the trained correlation filter and the groundtruth is greater than the threshold, the tracking is treated as to be successful. The results of correlation filter prediction are transmitted back to the direction prediction module to complete the whole tracking. If the IoU is less than the threshold, the detection will be carried out adaptively and the prediction results obtained by the detection module will be sent back to the direction prediction module to complete the tracking. Here, we will introduce the details of our proposed framework, including the adaptive detection module, correlation filtering module and detection module.

A. ADAPTIVE DETECTION MECHANISM

The trackers based on correlation filter can transform the complex matrix into Hadamad product of vector with real-time tracking. Simultaneously, when the target is heavily occluded, the online update mechanism of correlation filter is used to extract a frame in the historical template to determine the occluded target position of the next frame. Therefore, we combine the correlation filter tracker with the direction prediction module and propose a novel adaptive detection tracking mechanism as shown in Figure 3, which enables the object detection in a non-frame-by-frame manner. It not only improves the tracking accuracy under the heavy occlusion, but also further improves the efficiency of the tracking algorithm. The adaptive detection module consists of RoI determination module proposed by our previous research [9], correlation filter module and detection module. The correlation filter module and detection module in the adaptive detection tracker will be discussed in the following subsections.

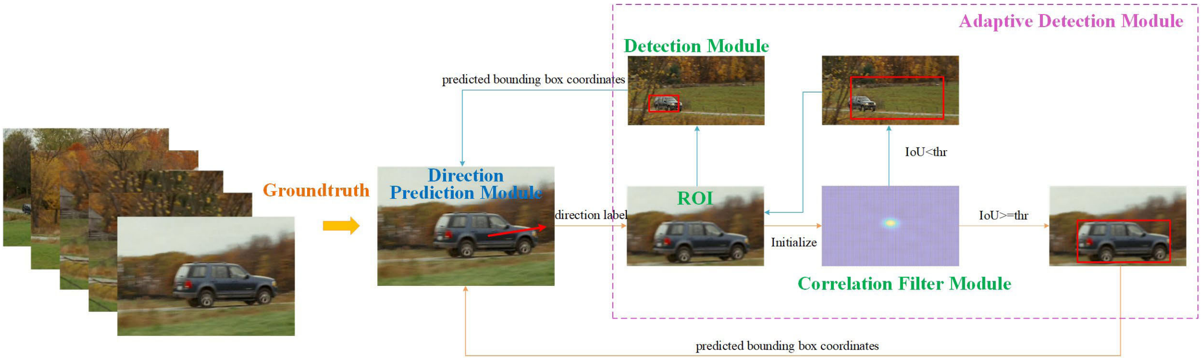

Adaptive Detection Module
Detection Module
predicted bounding box coordinates
Groundtruth
Direction Prediction Module
direction label
ROI
IoU<thr
Initialize
IoU>=thr
Correlation Filter Module
predicted bounding box coordinates

FIGURE 2. The framework of our proposed tracker. Our tracker consists of direction prediction module and adaptive detection module.

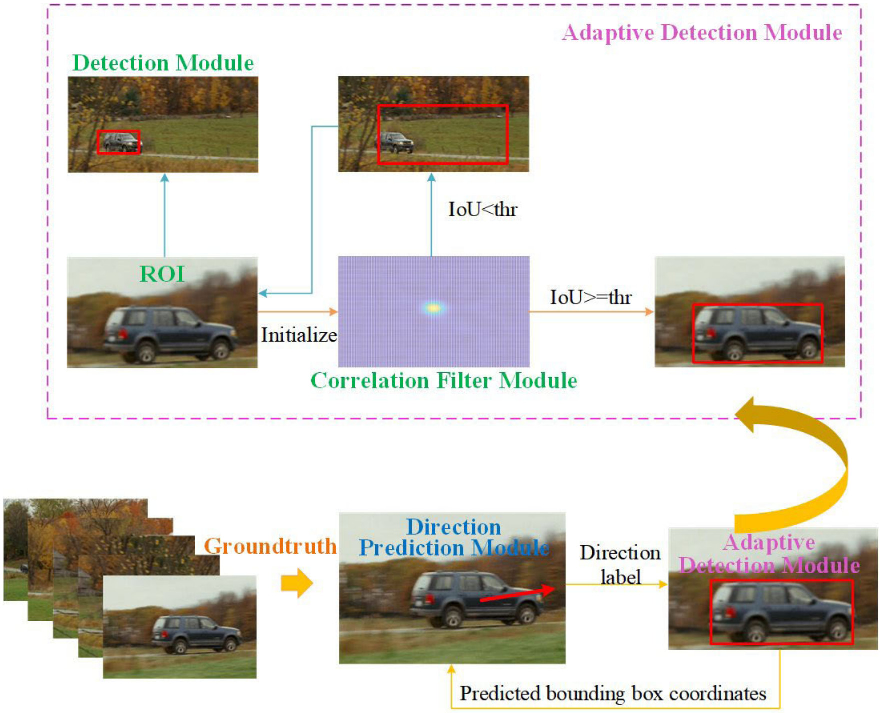

Adaptive Detection Module
Detection Module
IoU<thr
ROI
IoU>=thr
Initialize
Correlation Filter Module
Groundtruth
Direction Prediction Module
Direction label
Adaptive Detection Module
Predicted bounding box coordinates

FIGURE 3. The adaptive detection module includes the region of interest determination module, correlation filter module and detection module.

B. CORRELATION FILTER MODULE

The original trackers initialize the correlation filter with the patch of the target position in the first frame for model training. Then, the predicted position for each subsequent frames is updated to the position of the response peak, and a new correlation filter is trained at the new position. However, the significant deformation or occlusion caused by the target's fast motion will result in drift and boundary effect as shown in Figure 4. We propose an improved correlation filter strategy by adding a template correlation. That is, the target position coordinates predicted by the previous frames are used to predict the motion direction of the target in the next frame, and then RoI determined by the direction is used to train the correlation filter of the frame. Therefore, the motion state information of the previous frame will help correct the tracking drift and improve the track performance as illustrated in Figure 5.

On the other hand, the predicted RoI is treated as the target region, and the cyclic shift operation of correlation filter is applied to the target region to obtain different training samples. If the occluded part of the target does not exceed 20% of the whole target size, and the currect frame is used to detect the next frame, and the model parameters are updated through the online update mechanism. The online update model can be written as,

$$\alpha = (1-\beta)\cdot\alpha_{pre} + \beta\cdot\alpha_x, \tag{1}$$

where α is a vector of coefficients, β is a constant, α_{pre} is trained in the previous frames respectively. Then, the current frame is returned to the direction prediction model to

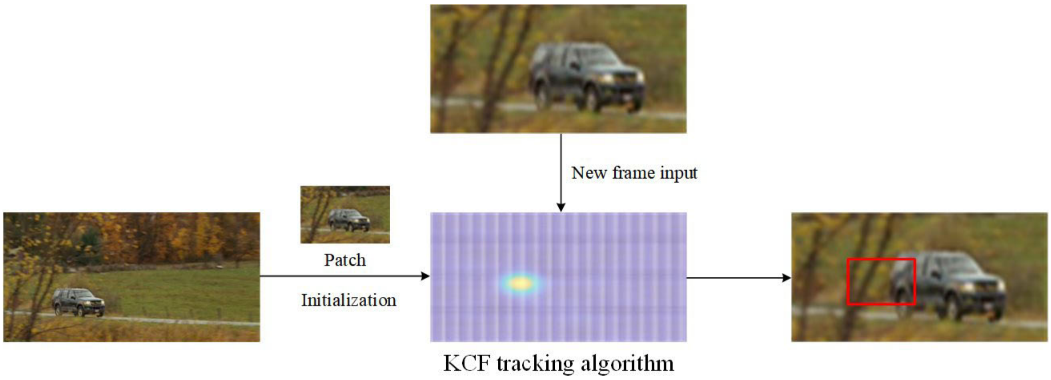


FIGURE 4. The original KCF algorithm initialized by the patch of the video first frame, which fails to track.

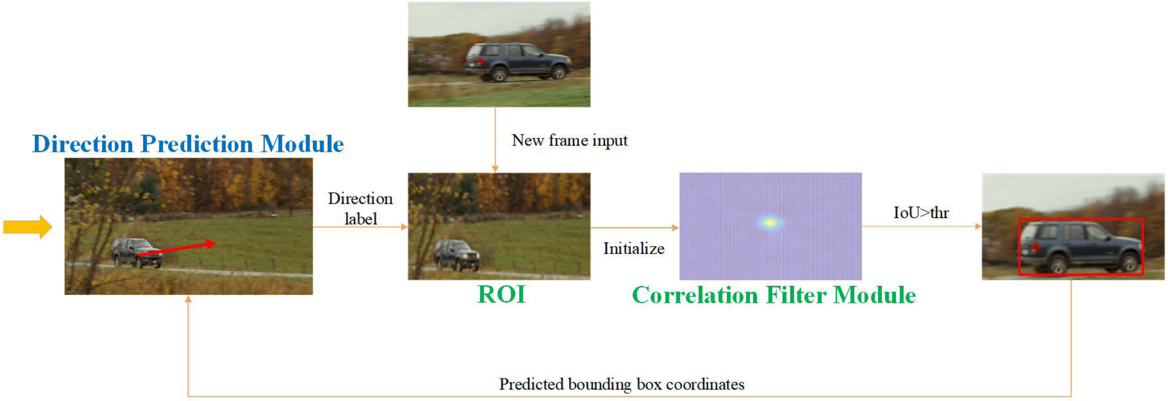

Direction Prediction Module
New frame input
Direction label
IoU>thr
Initialize
ROI
Correlation Filter Module
Predicted bounding box coordinates

FIGURE 5. The KCF algorithm with template correction is successfully tracked. The region of interest determined by the DPM is used to train the correlation filter.

complete the whole tracking. If the target is heavily occluded, we abandon the current frame and sample the video frames in the historical template. The sampled frame is used to update the model parameters to predict the target position of the next frame. Then, the sampled frames are returned to the direction prediction model to complete the whole tracking. The flow of correlation filter module is shown in Figure 6. Due to the online update and template correlation mechanisms, our proposed correlation filter module demonstrates more robustness for the scene when the target deforms or drifts.

C. DETECTION MODULE

Based on our previous research [9], we find that SSD does not only have high accuracy, but also achieves real-time detection speed on single object detection. However, the backbone of the original SSD detection network is VGGNet [61], which has a large number of parameters with information redundancy. In order to further enhance the detection efficiency, we introduce the lightweight network ShuffleNet [62], instead of VGGNet, which has the smallest classification error among the different model compression algorithms as shown in Table 1. First, the input feature gragh is grouped convoluted, and then channel shuffle is used to communicate information between channels. Next, 3*3 depthwise convolution is utilized for reducing parameters. When the stride is equal to 2, the number of channels increases while the size of the feature map decreases, resulting in the mismatch of the input and output dimensions. ShuffleNet takes the mean pooling for the original input, so as to obtain the feature map with the same size as the output. And then, we concat the

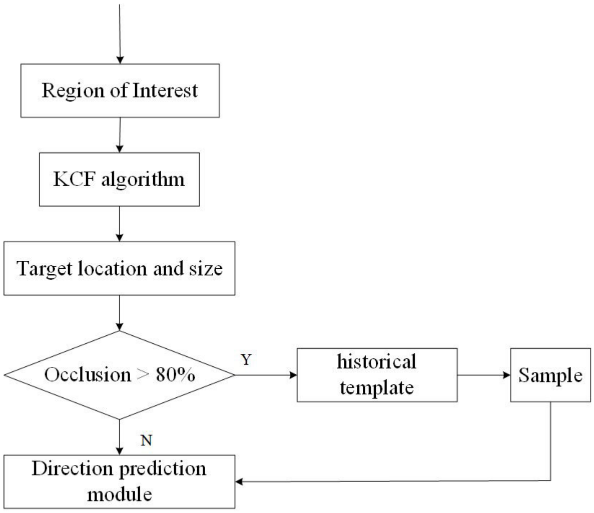


FIGURE 6. The running process of the correlation filter module based on our deigned tracker.

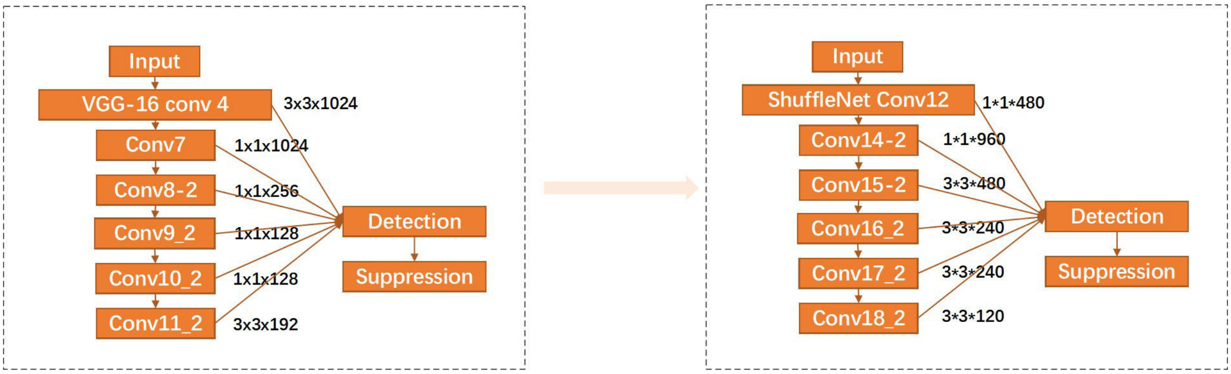


FIGURE 7. The ShuffleNet changes of feature pyramid structure based on VGGNet and SSD.

TABLE 1. Classification errors of different model compression algorithms with the same complexity, where the complexity is 140 MFLOPS. ShuffleNet has the lowest classification error under the same complexity.

algorithm	Classification error
VGG-like [62]	50.7
ResNet [64]	37.3
Xception-like [65]	33.6
ResNeXt [66]	33.3
MobileNet [67]	36.3
ShuffleNet [63]	32.4

feature map with the output to reduce the computational complexity and parameter size. In addition, our SSD-ShuffleNet is also a full convolution network and four convolution layers are added behind the ShuffleNet. Classification and regression are conducted by combining shallow and deep features. The original $conv4_3$, $conv7(fc7)$, $conv6_2$, $conv7_2$, $conv8_2$ and $conv9_2$ convolution layers are replaced by the six feature layers of $conv12$, $conv14_2$, $conv15_2$, $conv16_2$, $conv17_2$ and $conv18_2$, as shown in Figure 7. As a result, our proposed detection module can be effectively applied to various hardware platforms due to the lower demand for the storage space and computational resources.

IV. EXPERIMENTS

To validate the effectiveness and efficiency of our proposed algorithm for object tracking, we carry out extensive experiments with comparisons to state-of-the-arts methods on six different large-scale datasets, that is OTB100 [10], VOT2016 [13], VOT2017 [14], UAV123 [12], TC128 [11] and LaSOT [15] datasets, covering challenging scenarios, including scale variations, occlusions, background cluster and fast motion as shown in following subsections. In our experiment design, the training and verification experiments of the network is conducted under Linux Ubuntu (16.04) system. The configuration of the machine is shown in Table 2. First, the direction prediction model is implemented in Python using Tensorflow developed from our previous research [9]. Then, adaptive detection model is implemented in C++ using Caffe with eight cores of 3.4GHz Inter Core i7-3770 and NVIDIA TITAN X GPU. The dimensions of hidden layers of LSTMs are 128. The learning rates of LSTMs are initialized to be 0.001 and decay exponentially with the rate of 0.9. The parameters of detection model are set to the same as the literature [24]. Finally, we provide the visualized qualitative analysis of our approach with comparison to the existing tracking methods.

TABLE 2. Machine configuration for network training and validation experiments.

Hardware	Number	Model
GPU	3	NVIDIA TITAN-X
CPU	1	Inter®Xeon®E5-2630 V3@2.4GHz

A. FRAMEWORK IMPLEMENTATIONS

First, OTB100 [10] is used to test the performance of the object tracking network model based on motion direct prediction [9] with model compression. The performance is compared with the state-of-the-art trackers. The precision plots and success plots are shown in Figure 8 and Figure 9. To evaluate whether the model compression method is beneficial for reducing model parameters and speeding up the detection speed, the comparative experimental analysis of running time is carried out as shown in Table 3. The experimental results demonstrate that the tracking algorithm based on the motion prediction model with ShuffleNet achieves accuracy of 0.834 and success rate of 0.629. At the same time, the speed of our improved algorithm is 7 fps faster than our previous one [9]. It is proved that our method with model compression helps reduce model parameters and improve the detection and tracking speed.

B. ADAPTIVE THRESHOLD SELECTION

As discussed above, it is necessary to determine the IoU threshold of a predicted bounding box and the groundtruth bounding box. Therefore, we carry out the experimental analysis on the accuracy of trackers with different thresholds. The value range of α is 0.1-1.0 and the interval is 0.1. As shown in Table 4, $\alpha = 0.0$ means that the correlation filter is initialized completed by the determined region of interest and the response peak position is the location of the predicted target,

TABLE 3. Comparison of running times among different trackers. Runtime is the processing time of forward propagation of an image. The unit is fps.

Tracker	Ours-A	Ours-P	Struck	ROLO	KCF	SiamFC	TCNN	MDNet
Runtime	48	41	21.4	35	172	58	1.5	1
Tracker	ARCF	ASRCF	ATOM	DiMP	SiamRPN	SiamRPN++	SPLT	GradNet
Runtime	15.3	28	30	40	32	35	25.7	80

TABLE 4. Accuracy comparison of object tracking algorithm based on adaptive detection for different α values. The value of α is 0.0, which means that the correlation filter is initialized completely by the determined region of interest, and the response peak position obtained is the location of the predicted target. And the value of α is 1.0, which means that detects directly in the region of interest.

α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Precision	0.729	0.766	0.802	0.829	0.843	0.858	0.861	0.869	0.857	0.841	0.834

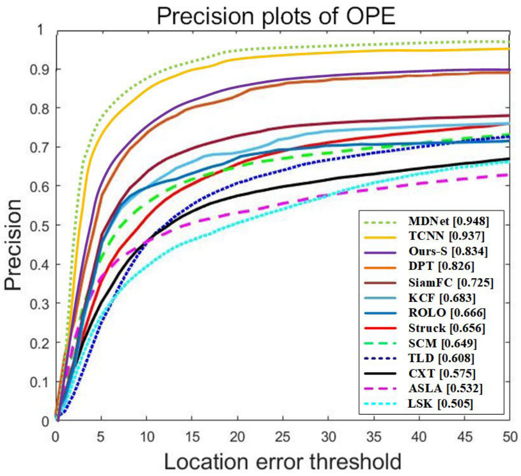


FIGURE 8. Precision plots for all 50 sequences. The proposed tracker (purple) outperform state-of-the-art systems, such as TLD and Struck. The numbers in the legend indicate the representative precision at 20 pixels for precision plots.

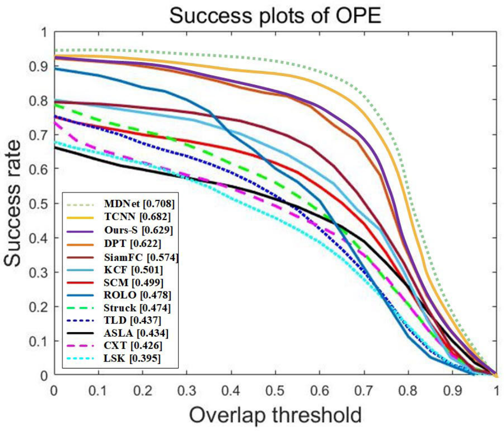


FIGURE 9. Success plots for all 50 sequences. The proposed tracker (purple) outperforms state-of-the-art systems, such as TLD and Struck. The numbers in the legend indicate the area-under-curve scores for success plots.

rather than detecting in the region of interest. $\alpha = 1.0$ means that the predicted bounding box is exactly coincident with the groundtruth bounding box, that is, the tracking algorithm detects directly in the region of interest. When α is less than 0.7, the accuracy increases as α increases. When α is greater than 0.7, the accuracy decreases as α increases. Therefore, $\alpha = 0.7$ is selected as the threshold of IoU for adaptive detection with the highest accuracy.

C. EVALUATION ON THE DIFFERENT DATABASES

In this subsection, we compare the performance of our proposed method with state-of-the-arts trackers. We evaluate our algorithm based on the selected videos on recent benchmarks, including OTB100 [10], TC128 [11], VOT2016 [13], VOT2017 [14], UAV123 [12] and LaSOT [15], which contains various complex environmental scenarios.

1) OTB100 DATASET

The OTB100 dataset [10] is one of the most popular benchmarks, which consists of 100 challenging video clips annotated with 11 different attributes, which contains partial or complete occlusions, background illumination variations, fast motion target and so on. Figure 10 illustrates the tracking results from five test videos based on our adaptive detection tracking. The precision plots and success plots shown in Figure 11 and Figure 12 presents the superiority of our proposed algorithm qualitatively over other trackers, including TLD [61], ROLO [48], TCNN [42], MDNet [41] and other traditional tracking algorithms. In order to evaluate the computational efficiency of our proposed algorithm, comparative experimental analysis of frame rate is carried out, and the experimental results are shown in Table 3. As illustrated in Figure 11 and Figure 12, our proposed tracking algorithm achieves precision of 0.869 and success of 0.645, which is higher than our previous research [9] based on direction prediction module and object detection. At the same time, the frame rate of our improved method is 48 fps faster than our previous one. Although the precision of our proposed adaptive detection tracking is still lower than some state-of-the-art algorithms, such as TCNN and MDNet, the tracking speed of these methods are only 15 fps and 1 fps as shown in Table 3. Therefore, it is proved that the correlation filter is beneficial for improving the tracking performance, and reducing the dependency on object detection.

In addition, we also tested the performance of KCF tracking algorithm with temporal prediction based on direction prediction model (DPM). The KCF tracking algorithm without DPM is initialized with the first frame of the video

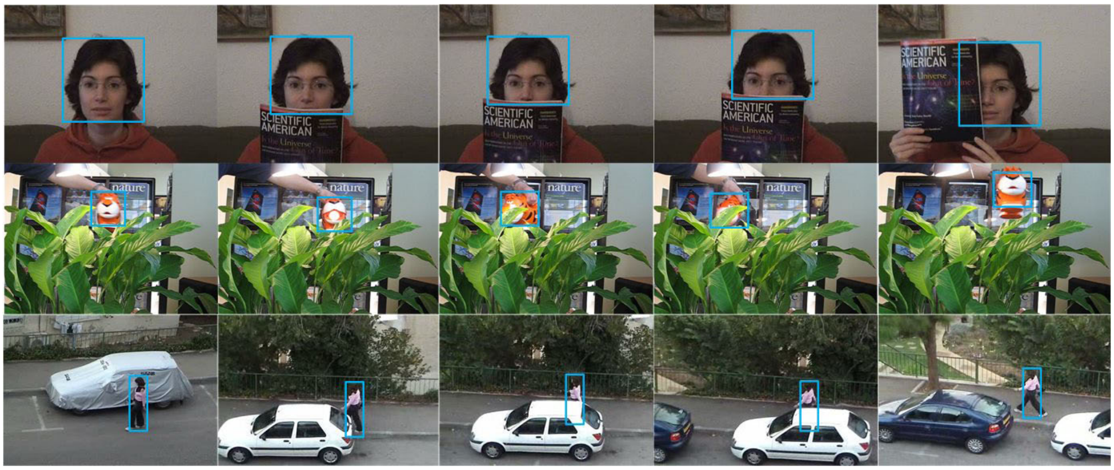


FIGURE 10. The visualization results of the samples, face (top), tiger (middle) and person (bottom), respectively.

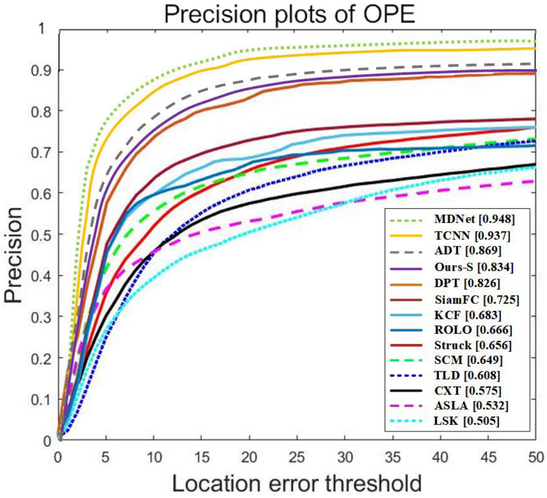


FIGURE 11. Precision plots for all 50 sequences. The proposed tracker(gray) outperform state-of-the-art systems, such as TLD and Struck. But not as good as TCNN or MDNet. The numbers in the legend indicate the representative precision at 20 pixels for precision plots.

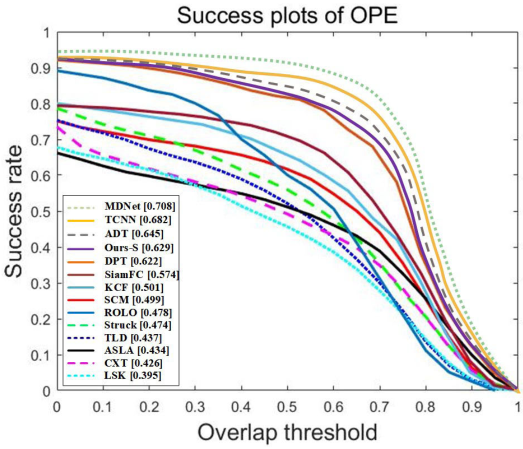


FIGURE 12. Success plots for all 50 sequences. The proposed tracker (gray) outperforms state-of-the-art systems, such as TLD and Struck. But not as good as TCNN or MDNet. The numbers in the legend indicate the area-under-curve scores for success plots.

sequence, and the position is updated to the response peak for each subsequent frame. The KCF algorithm with DPM initializes the correlation filter with the determined region of interest, and then updates the position of each subsequent frame to the region of interest determined by DPM in the next frame. The experimental results are shown in Figure 13 and Figure 14. The KCF algorithm initialized by DPM achieves the precision of 0.729 and the success of 0.576. Both are higher than the originial KCF tracking algorithm, which proves the importance and effectiveness of using adaptive detection for temporal prediction.

2) VOT DATASETS

The VOT2016 and VOT2017 datasets contain 60 challenging sequences respectively with the different attributes. The evaluation criteria in these benchmarks are the accuracy (A), robustness (R) and expected average overlap (EAO). First, we evaluate the performance of our proposed tracker with other state-of-the-arts trackers based on the randomly selected videos, including SRDCF [29], Staple [34], Siam-FC [43], ECO [32], SiamRPN [55], SiamDW [56], ASRCF [37], TCNN [42], C-COT [31], on the VOT2016 database. Table 5 demonstrates that our tracker achieves the second performance among the comprehensive evaluation of the three criteria. Although ASRCF obtains the better results than ours, its speed is slower than our ADT tracker as shown in Table 3. Then, we further test the tracking results between ours and other top trackers based on the randomly selected videos, such as SRDCF [29], Staple [34], Siam-FC [43], SiamRPN [55], SiamDW [56], GradNet [54], C-COT [31], ECO [32], ASRCF [37], on the VOT2017 database. Although the accuracy of our method is worse than GradNet and SiamDW, our robustness and EAO are better than them. Compared with the computational cost demonstrated in Table 3, our method can achieve 48 fps for real-time applications compared with ECO

TABLE 5. Performance evaluation on the VOT2016 [13] dataset. In this table, we compare our method with state-of-the-arts tracker. The results are presented in terms of expected average overlap (EAO), accuracy rank (A) and robustness (R). The best three results are shown in red, blue and green colors, respectively.

VOT2016	SRDCF	Staple	Siam-FC	ECO-HC	SiamRPN	SiamDW	ASRCF	TCNN	C-COT	Ours
A	0.54	0.54	0.53	0.54	0.56	0.54	0.563	0.554	0.539	0.572
R	0.42	0.38	0.46	0.3	0.26	0.38	0.187	0.268	0.238	0.25
EAO	0.25	0.3	0.24	0.32	0.34	0.3	0.391	0.325	0.331	0.342

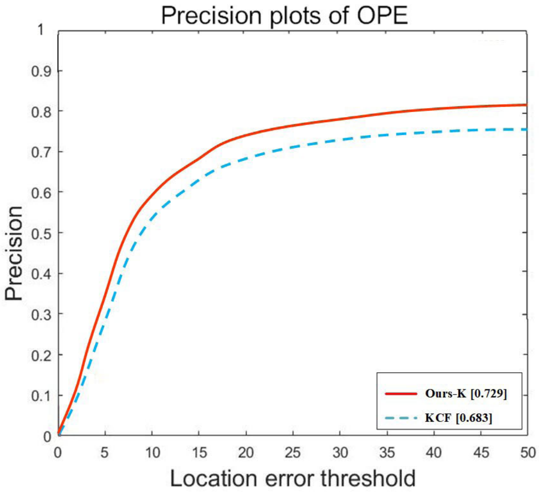


FIGURE 13. Precision plots of KCF algorithm with and without direction prediction model. The numbers in the legend indicate the representative precision at 20 pixels for precision plots.

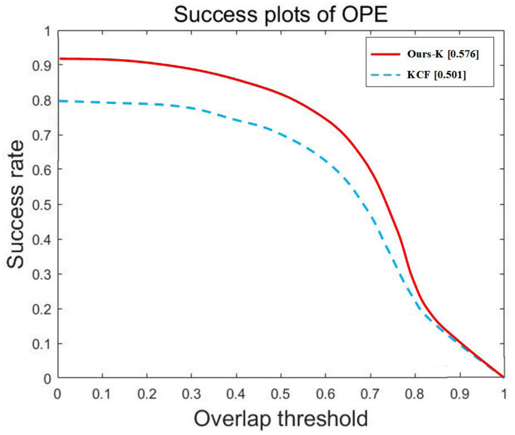


FIGURE 14. Success plots of KCF algorithm with and without direction prediction model. The numbers in the legend indicate the area-under-curve scores for success plots.

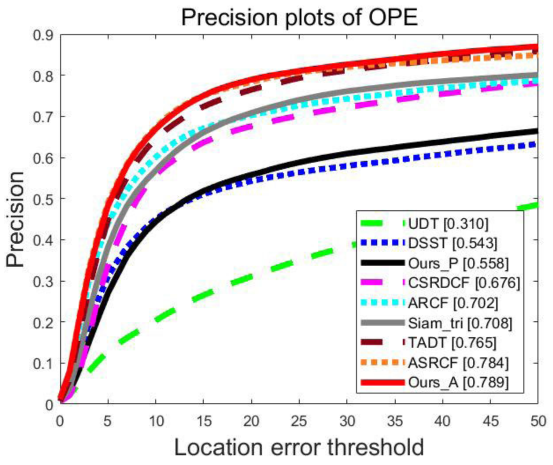


FIGURE 15. Precision plots of our proposed tracker (Ours-A) with other trackers on the TC128 dataset. The numbers in the legend indicate the representative precision at 20 pixels for precision plots.

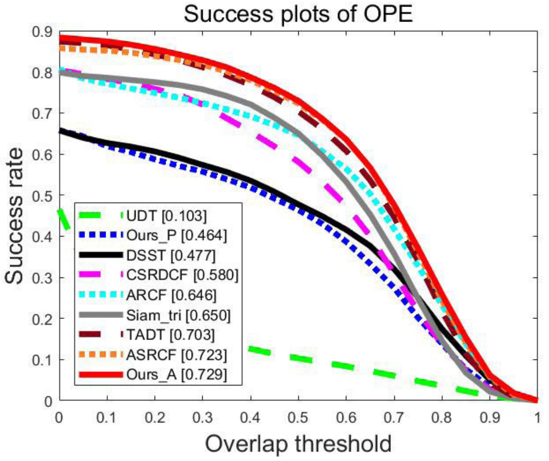


FIGURE 16. Success plots of our proposed tracker (Ours-A) with other trackers on the TC128 dataset. The numbers in the legend indicate the area-under-curve scores for success plots.

and SiamDW. Thus, our method can effectively balance the simple implementation and high accuracy with computational efficiency.

3) TC128 DATASET

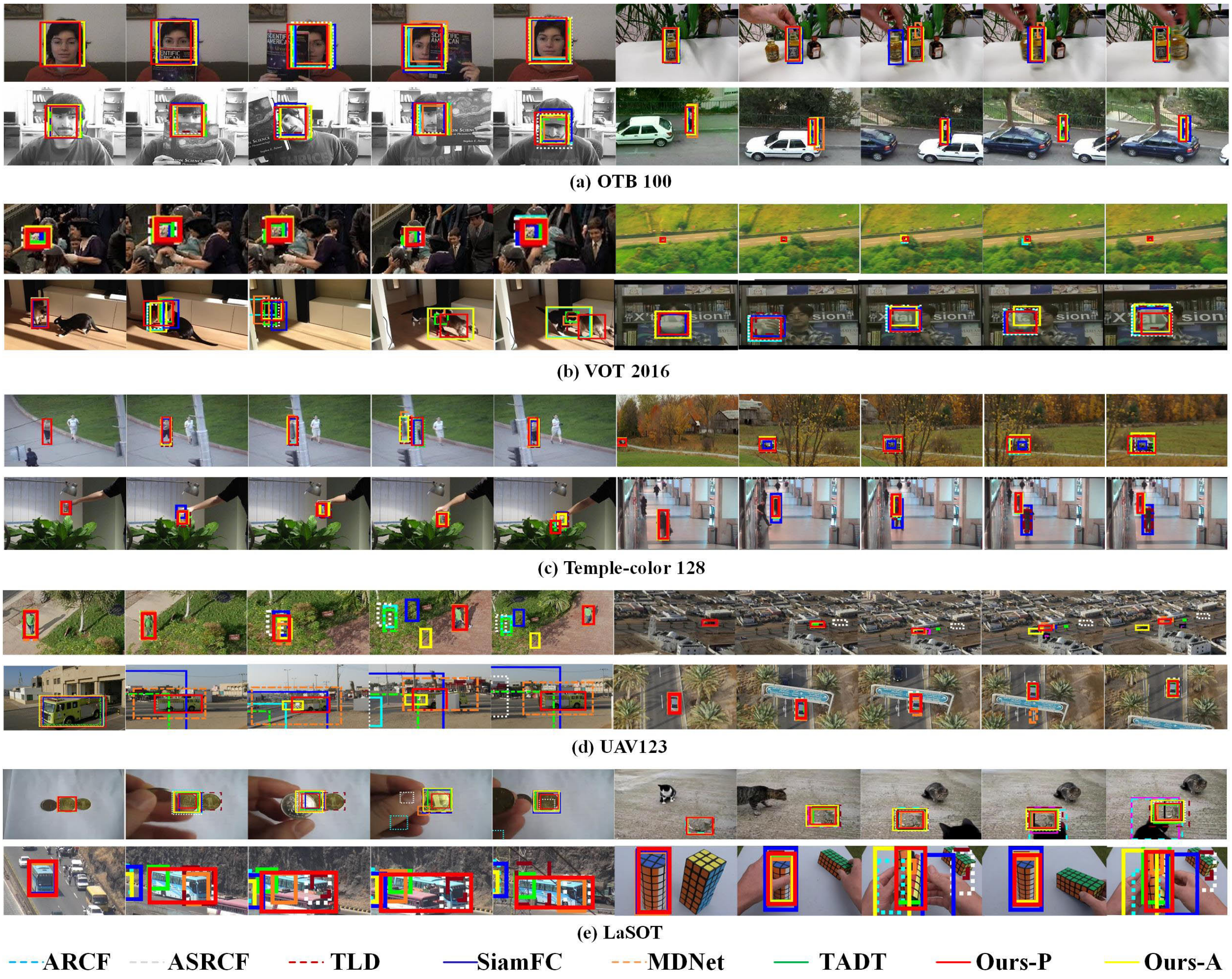
TC128 database consists of 128 annotated sequences with more color information, which covers 11 various challenging factors, such as scale variation, low resolution, fast motion, in/out plane rotation, out of view, background clutter, illumination variation, motion blur, occlusion and deformation. We also utilize both success and precision plots to evaluate the tracking performance among the different trackers, including ASRCF [37], ARCF [36], UDT [67], DSST [68], CSRDCF [69], SiamTri [70], TADT [71], our previous tracker and our ADT tracker. As illustrated in Figure 15 and Figure 16, our proposed tracker obtains the best results among all of the compared trackers with various challenges. In addition, compared with our previous tracker, our adaptive detection tracker also has a great progress on both accuracy and computational efficiency.

4) UAV123 DATASET

UAV123 dataset includes 123 low altitude aerial videos for unmanned aerial vehicles with different variations of illumination, shape, scale and rotations. We choose 100 challenging

TABLE 6. Performance evaluation on the VOT2017 [14] dataset. In this table, we compare our method with state-of-the-arts tracker. The results are presented in terms of expected average overlap (EAO), accuracy rank (A) and robustness (R). The best three results are shown in red, blue and green colors, respectively.

VOT2017	SRDCF	Staple	Siam-FC	SiamRPN	SiamDW	GradNet	C-COT	ECO	ASRCF	Ours
A	0.49	0.52	0.49	0.49	0.50	0.507	0.494	0.483	0.494	0.495
R	0.97	0.69	0.44	0.46	0.49	0.375	0.318	0.276	0.328	0.315
EAO	0.12	0.17	0.24	0.24	0.23	0.247	0.267	0.28	0.234	0.241


(a) OTB 100

(b) VOT 2016

(c) Temple-color 128

(d) UAV123

(e) LaSOT

FIGURE 17. Tracking qualitative results of our method and the other six current trackers on five different datasets. (a) OTB100 dataset (b) VOT2016 dataset (c) TC128 dataset (d) UAV123 dataset (e) LaSOT dataset.

videos to test the performance of our tracker on UAV-like objects. We introduce the same evaluation protocol and annotations the same as the literature [46]. Table 7 demonstrates the AUC score of the compared trackers, including ECO [32], DaSiamRPN [72], ATOM [45], CCOT [31], MDNet [41], SiamRPN++ [58], UPDT [73], DiMP [46] and ours. Our tracker achieves an AUC score of 62.5%, which is lower than ATOM and DiMP. But our tracker realizes a faster frame rate of 18 fps and 8 fps respectively, than these two methods. Thus, our ADT tracker can obtain the comparative result for small UAV-like objects.

5) LaSOT DATASET

A larger and more challenging dataset, that is LaSOT dataset [15], is introduced to further evaluate the performance of our ADT tracker, which is composed of 1400 large-scale, high-quality dense annotated videos and 280 testing videos. The average frame length is more than 2500 frames, which is suitable for evaluating the long-term sequence tracking. We adopt normalized precision and success as the evaluation criteria. Table 8 shows the compared result between our proposed tracker and other 10 state-of-the-art trackers based on the randomly selected videos, including STRCF [74],

TABLE 7. AUC score comparison on UAV123 dataset.

UAV123	ECO	DaSiamRPN	ATOM	CCOT	MDNet	SiamRPN++	UPDT	DiMP	Ours
AUC score (%)	50.6	58.6	64.4	51.3	52.8	61.3	54.5	64.3	62.5

TABLE 8. Performance comparison on LaSOT dataset.

LaSOT	STRCF	SINT	ECO	DSiam	StructSiam	SiamFC	VITAL	MDNet	DaSiamRPN	ATOM	Ours
Norm. Prec (%)	34.0	35.4	33.8	40.5	41.8	42.0	45.3	46.0	49.6	57.6	55.3
Success (%)	30.8	31.4	32.4	33.3	33.5	33.6	39.0	39.7	41.5	51.5	48.7

SINT [43], ECO [32], DSiam [75], StructSiam [76], SiamFC [43], VITAL [77], MDNet [41], DaSiamRPN [72] and ATOM [45], which demonstrate that our tracker outperforms most of the state-of-the-art trackers under two protocols. Although ATOM obtains better results than ours, its speed is slower than ours for real-time requirement. These results demonstrate the powerful model adaptation capability for long-term sequences.

6) QUALITATIVE EVALUATION

We also present the qualitative comparison of our ADT tracker against the 6 competing trackers on five different datasets. In Figure 17(a), we can observe that tracker is extremely adaptive to the change in partial occlusion as well as in-plane rotation and similar appearance. For VOT2016 dataset, it is experimentally found that all the compared trackers achieve acceptable outputs for small objects with fast motion. But tracking drift will occur in the face of fuzzy appearance and targets interaction based on SiamFC, TADT and TLD methods. Only our ARCF and our method can obtain the satisfied tracking results. In Figure 17(c), we evaluate the tracking performance on the TC128 dataset. Almost all the other trackers fail in completely occluded, whereas our method can effectively continue to track the target location in subsequent frames. The tracking objects of UAV123 dataset are mostly small targets with in/out-plane rotation and severe occlusion. It is clear that from the Figure 17(d) that our method has achieved good results in these cases, while the other methods are faced with serious tracking drift. Similarly, in Figure 17(e), we evaluate all the trackers under scale variations, partial occlusion and out-plane rotation. ARCF, ASRCF and SiamFC are incapable to keep track of the object under such constraints. However, our ADT tracker is not affected by such challenges and keep good track of the objects. As a result, our method has good tracking effect for different datasets and different challenges under complex unconstrained conditions. The method has certain data universality.

V. CONCLUSION

In this paper, we propose a novel adaptive algorithm called adaptive detection tracking, which combined correlation filter and our proposed direction precision module. First, the proposed framework initializes correlation filter with the region of interest by setting an adaptive threshold. When the IoU is greater than the threshold, the position of each subsequent frame will be updated to the region of interest determined by the direction prediction model. When the IoU is less than the threshold, the detection is carried out in the region of interest. As a result, non-frame-by-frame detection can be realized, which not only further reduces the dependency on object detection, but also improves the speed of object tracking for real-time applications. Then, our proposed algorithm utilizes the online update mechanism based on KCF method. When the object is heavily occluded, the current frame can be effectively discarded, and the effective frame in the historical template can be used to predict the position and return to the direction prediction model, which makes the proposed tracking method more robust to the heavy occluded scene. Extensive experiments on six challenging datasets demonstrate our tracker performs favorably against state-of-the-art trackers, which can effectively reduce computation redundancy and improve tracking accuracy.

REFERENCES

[1] K.-H. Lee and J.-N. Hwang, “On-road pedestrian tracking across multiple driving recorders,” *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1429–1438, Sep. 2015.

[2] S. Tang, M. Andriluka, B. Andres, and B. Schiele, “Multiple people tracking by lifted multicut and person re-identification,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3539–3548.

[3] R. Singer, “Estimating optimal tracking filter performance for manned maneuvering targets,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-6, no. 4, pp. 473–483, Jul. 1970.

[4] M. Isard and A. Blake, “Condensation–conditional density propagation for visual tracking,” *Int. J. Comput. Vis.*, vol. 29, no. 1, pp. 5–28, 1998.

[5] Y. Cheng, “Mean shift, mode seeking, and clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, 1995.

[6] 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.

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

[8] X. Wang, Z. Hou, W. Yu, Z. Jin, Y. Zha, and X. Qin, “Online scale adaptive visual tracking based on multilayer convolutional features,” *IEEE Trans. Cybern.*, vol. 49, no. 1, pp. 146–158, Jan. 2019.

[9] Y. Zhang, Y. Ming, and R. Zhang, “Object detection and tracking based on recurrent neural networks,” in *Proc. 14th IEEE Int. Conf. Signal Process. (ICSP)*, Aug. 2018, pp. 338–343.

[10] 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.

[11] 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.

[12] M. Mueller, N. Smith, and B. Ghanem, “A benchmark and simulator for UAV tracking,” in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 445–461.

[13] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, P. R. Pflugfelder, L. Cehovin, T. Vojir, G. Hager, A. Lukezic, and G. Fernandez, "The visual object tracking VOT2015 challenge results," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2016, pp. 1–23.

[14] M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, L. C. Zajc, T. Vojir, G. Hager, A. Lukezic, A. Eldesokey, and G. Fernandez, "The visual object tracking VOT2017 challenge results," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 1949–1972.

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

[16] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. H. S. Torr, "Deeply supervised salient object detection with short connections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 815–828, Apr. 2019.

[17] Krizhenvshky, A and Sutskever, Ilya and Hinton, G, "Imagenet classification with deep convolutional networks," in *Proc. Conf. Neural Inf. Process. Syst. (NIPS)*, 2010, pp. 1097–1105.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Jan. 2015.

[20] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

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

[22] L. Cheng and W. Liu, "An effective microscopic detection method for automated silicon-substrate ultra-microtome (ASUM)," *Neural Process. Lett.*, pp. 1–18, Nov. 2019.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[24] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.

[25] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7263–7271.

[26] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

[27] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, and Y. Chen, "RON: Reverse connection with objectness prior networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5936–5944.

[28] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, and X. Xue, "DSOD: Learning deeply supervised object detectors from scratch," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1919–1927.

[29] 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)*, Dec. 2015, pp. 4310–4318.

[30] 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.

[31] 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.*, pp. 472–488, 2016.

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

[33] J. A. 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.*, 2012, pp. 702–715.

[34] 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.

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

[36] Z. Huang, C. Fu, Y. Li, F. Lin, and P. Lu, "Learning aberrance repressed correlation filters for real-time UAV tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2019, pp. 2891–2899.

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

[38] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 809–817.

[39] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 597–606.

[40] 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.

[41] 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.

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

[43] 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.

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

[45] M. Danelljan, G. Bhat, F. S. Kham, and M. Felsberg, "ATOM: Accurate tracking by overlap maximization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 4660–4669.

[46] G. Bhat, M. Danelljan, L. Van Gool, and R. Timofte, "Learning discriminative model prediction for tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6182–6190.

[47] Z. Cui, S. Xiao, J. Feng, and S. Yan, "Recurrently target-attending tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1449–1458.

[48] G. Ning, Z. Zhang, C. Huang, X. Ren, H. Wang, C. Cai, and Z. He, "Spatially supervised recurrent convolutional neural networks for visual object tracking," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2017, pp. 1–4.

[49] H. Fan and H. Ling, "SANet: Structure-aware network for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 42–49.

[50] Z. Huang, Y. Yu, and M. Xu, "Bidirectional tracking scheme for visual object tracking based on recursive orthogonal least squares," *IEEE Access*, vol. 7, pp. 159199–159213, 2019.

[51] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, 1990.

[52] B. Yan, H. Zhao, D. Wang, H. Lu, and X. Yang, "'Skimming-perusal'tracking: A framework for real-time and robust long-term tracking," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 2385–2394.

[53] S. P. Bharati, S. Nandi, Y. Wu, Y. Sui, and G. Wang, "Fast and robust object tracking with adaptive detection," in *Proc. IEEE 28th Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2016, pp. 706–713.

[54] P. Li, B. Chen, W. Ouyang, D. Wang, X. Yang, and H. Lu, "GradNet: Gradient-guided network for visual object tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6162–6170.

[55] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with Siamese region proposal network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8979.

[56] Z. Zhang and H. Peng, "Deeper and wider Siamese networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4591–4599.

[57] H. Fan and H. Ling, "Siamese cascaded region proposal networks for real-time visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1–10.

[58] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4282–4290.

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

[60] L. Xu, L. Wang, Y. Zhang, and S. Cheng, "Visual tracking based on Siamese network of fused score map," *IEEE Access*, vol. 7, pp. 151389–151398, 2019.

[61] K. Simonyon and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. IEEE Int. Conf. Learn. Represent.*, Sep. 2015, pp. 1–14.

[62] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1–9.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*. [Online]. Available: http://arxiv.org/abs/1512.03385

[64] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1–8.

[65] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

[66] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861

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

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

[69] A. Lukezic, T. Vojír, L. C. Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4847–4856.

[70] X. Dong and J. Shen, "Triplet loss in Siamese network for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 1–16.

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

[72] 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.*, 2018, pp. 1–17.

[73] G. Bhat, J. Johnander, M. Danelljan, F. S. Khan, and M. Felsberg, "Unveiling the power of deep tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 1–22.

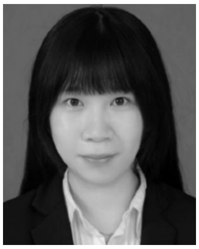
[74] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4904–4913.

[75] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan, and S. Wang, "Learning dynamic Siamese network for visual object tracking," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1781–1789.

[76] Y. Zhang, L. Wang, J. Qi, D. Wang, M. Feng, and H. Lu, "Structured Siamese network for real-time visual tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 1–16.

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

YUE MING received the B.S. degree in communication engineering, the M.Sc. degree in human–computer interaction engineering, and the Ph.D. degree in signal and information processing from Beijing Jiaotong University, China, in 2006, 2008, and 2013, respectively. She was a Visiting Scholar with Carnegie Mellon University, USA, from 2010 to 2011. Since 2013, she has been working as a Faculty Member with the Beijing University of Posts and Telecommunications. Her research interests include biometrics, computer vision, computer graphics, information retrieval, and pattern recognition.



YASHU ZHANG was born in Meihekou, Jilin, in 1994. She received the bachelor's degree in communication engineering from Jilin University, in 2016. She is currently pursuing the master's degree with the School of Electronic Engineering, Beijing University of Posts and Telecommunications. Her current research interests include object detection and object tracking in computer vision.