Multi-Feature Object Tracking with Fusion of LBP and Attention Based on Siamese Networks

Zhangfang Hu, Yuan Yuan, Yi Wang, Bokun Wang

Abstract—Siamese networks utilize deep learning models to achieve a balance between tracking speed and accuracy in visual object tracking. However, in low light and other challenging lighting conditions, the contours and textures of the tracked object may not be accurately represented in the feature maps. This can result in blurriness and interference from similar distractions, as non-edge features may be overlooked. To tackle these challenges, this paper proposes a multi-feature object tracking approach that combines Siamese networks with Local Binary Patterns (LBP) and attention mechanisms. After preprocessing the video frames for LBP feature extraction, the resulting local binary pattern maps are input into a SiamRPN network, with ResNeSt as its backbone. The incorporation of local binary patterns enhances feature representation by capturing texture information from the target's appearance. Additionally, a coordinated attention mechanism is applied after each feature layer of the network to further improve tracking accuracy. This mechanism dynamically adjusts the weights of different features to optimize performance across various scenarios. The target's position is then determined by merging the deep and shallow features retrieved by the network, which are subsequently fed into the RPN network for regression classification. Experimental results demonstrate the effectiveness of the proposed algorithm in tracking objects.

***Index Terms*—Attention Mechanism, Feature Fusion, LBP Features, Siamese Networks**

Manuscript received April 4, 2024; revised November 6, 2024.

This work was supported in part by the Youth Fund Program of the National Natural Science Foundation of China (Grant No. 61703067), the Chongqing Basic Science and Frontier Technology Research Program (Grant No. Cstc2017jcyjAX0212), and the Science and Technology Research Program of Chongqing Municipal Education Commission (KJ1704072).

Zhangfang Hu is a Professor at the Key Laboratory of Optical Information Sensing and Technology, School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065 China(e-mail: huzf@cqupt.edu.cn)

Yuan Yuan is a graduate student of the School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065 China (corresponding author phone: 152-0286-6616; e-mail: 2250599628@qq.com)

Yi Wang is a graduate student of the School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065 China (e-mail:2253356946@qq.com)

Bokun Wang is a graduate student of the School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065 China (e-mail: bkwang2018@163.com)

I. INTRODUCTION

OBGECT tracking is one of the main areas of study in computer vision, with a great deal of potential for advancement and practical utility.It is widely used in military equipment[1], smart healthcare[2], robotics[3], and autonomous driving[4], among other fields. Predicting a target's location and form in every frame of a movie is the goal of object tracking. It is crucial to identify the background and other distracting elements from the target in the video and annotate the target object appropriately, given the target's initial position and scale information in the first frame. Subsequently, the ongoing localization and size estimation of the target are performed in the following video frames.

Convolutional neural networks have become a widely used method due to advancements in deep learning, and object tracking using deep learning has become a popular research area. One prevalent approach is the Siamese network-based target tracking method, which trains trackers end-to-end using datasets and frames the target tracking issue as a similarity matching problem. Siamese networks leverage deep learning models to achieve a balance between tracking speed and accuracy in visual object tracking. However, variations in lighting can lead to overexposure of the target image, resulting in a loss of features and hindering real-time tracking, ultimately causing the target to be lost. A significant challenge in visual object tracking technology is its struggle to effectively extract information from the target when it closely resembles its surroundings. The target may undergo deformation, the network model may fail to capture its features adequately, and important feature points may be lost due to unclear local texture features in the image, which can lead to the extraction of highly similar background information.

This research suggests a shallow-deep fusion target tracker based on Siamese networks that integrate coordinated attention and LBP characteristics in order to overcome the aforementioned problems. It is mostly divided into two sections:

1) Conventional target tracking network models usually use just deep features for target feature extraction, or they use classic manual feature extraction methods, ignoring the impact of shallow and local texture features on target tracking. In order to address the issue of low accuracy and resilience of single-depth feature extraction in the environment, this paper

uses a fusion strategy combining local binary pattern features with deep and shallow features. As a traditional technique for describing textures, LBP features exhibit considerable resilience to variations in lighting. Following preprocessing, we first extract the target's LBP features and then combine them with shallow and deep features that we retrieved from video frames using Siamese networks. The poor pace of other deep learning networks caused by employing pre-trained networks for feature extraction can be addressed with Siamese network-based target trackers. Since Siamese networks provide robust tracking capabilities along with higher speed, this paper also explores this kind of target tracker.

2) We suggest adding a coordinated attention module after each feature layer of the backbone network to enhance the selection of relevant information for the task within the network and link feature information between convolutional layers with spatial information when extracting features from images. In order to effectively address the issue of losing positional information, the coordinated attention module assists deep networks in adaptively adjusting the focus on various feature layers. This increases the network's focus on semantic information associated with the target. Grayscale invariance of LBP features highlights the edge properties of video information, which makes them appropriate for target and background separation. Shallow features are useful for target localization because they provide appearance information that helps differentiate the target from comparable interfering items and from the backdrop. Thus, we first extract LBP features from video frames, and then we use Siamese networks to extract shallow and deep features. Finally, we use skip connections to merge the shallow and deep features.

Numerous testing outcomes suggest that our suggested method performs well in current benchmark tests. This paper's remaining sections are arranged as follows: Part III outlines our methodology, while Part II introduces relevant work. Part V concludes; Part IV covers the network's performance test findings and training procedure.

II. Related Work

Three phases can be identified in the evolution of object tracking technology. The initial phase, which started in 2000, focused mostly on object tracking using machine learning and classical techniques. These methods required little in the way of hardware resources, operated quickly, and had minimal computational complexity, but their accuracy and resilience were lacking. Correlation filter-based trackers emerged in the second stage, which spanned 2010 to 2016, most notably with the launch of the MOSSE[5] tracker. These trackers' great speed and precision allowed them to perform admirably on a variety of evaluation datasets. In the third stage, which began in 2016 and is ongoing, deep learning algorithms have become more popular. One notable example of this is the use of Siamese networks in object tracking technologies. These algorithms continuously enhance their robustness and accuracy in object tracking with the progressively rich dataset settings thanks to their strong end-to-end learning capabilities. As such, they have attracted a great deal of interest from scholars lately.

The template branch and the search branch are the two branches used for object tracking in the Siamese network, which uses weight sharing between two neural networks. Essentially, the search branch's frame features are subjected to convolution operations by the template branch, which functions as a kernel in the process. By means of end-to-end training, the tracking problem is converted into a similarity matching problem as the network immediately learns the representation of the target from raw data.Siamese Instance Search for Tracking (SINT), the first object tracking technique utilizing the Siamese network framework, was put forth by Tao Ran et al. [6]. SINT offers a unique method for doing object tracking by treating it as target matching. Later, fully convolutional networks (FCNs) were included into the Siamese network framework by L. Bertinetto et al. [7], leading to the introduction of Fully-Convolutional Siamese Networks for Object Tracking (SiamFC). While SiamFC, an early Siamese network-based tracking algorithm, satisfies real-time tracking criteria, accuracy and resilience are still lacking.In order to tackle this, Li et al. [8] integrated the Region Proposal Network (RPN) into the SiamFC framework, proposing SiamRPN, taking inspiration from the Faster R-CNN [9]technique used in object detection. SiamRPN is made up of two functional modules: the RPN module for proposal generation, which generates candidate target regions, and the Siamese module for feature extraction, which has the same structure as the SiamFC network. There are two branches in the RPN module: one for regression and one for classification. The regression branch pinpoints the exact location of the targets, whereas the classification branch distinguishes between targets and backgrounds. Figure 1 shows the SiamRPN network architecture.

Wang et al. [10] have developed SiamMask, a different approach that can simultaneously do object segmentation and tracking, as a means of enhancing SiamFC. Zhang et al. [11] created the SiamDW method, which optimizes the Siamese network's backbone network. By using Cropping-Inside Residual (CIR) units—which are enhanced based on residual modules—it lessens the detrimental effects of biases brought on by padding operations on object tracking. By improving SiamRPN's backbone network, Li et al.'s SiamRPN++ [12] produces a deep Siamese network model with several layers of feature fusion. During the feature learning process, our model overcomes the propensity of deep networks to give more weight to an image's central position.

Local Binary Pattern elements have been added to object tracking by certain academics in the past few years. LBP characteristics are renowned for their superior texture description capabilities and low sensitivity to variations in lighting. Researchers want to improve their ability to represent the target and capture object properties at a higher level by combining LBP features with deep learning techniques.

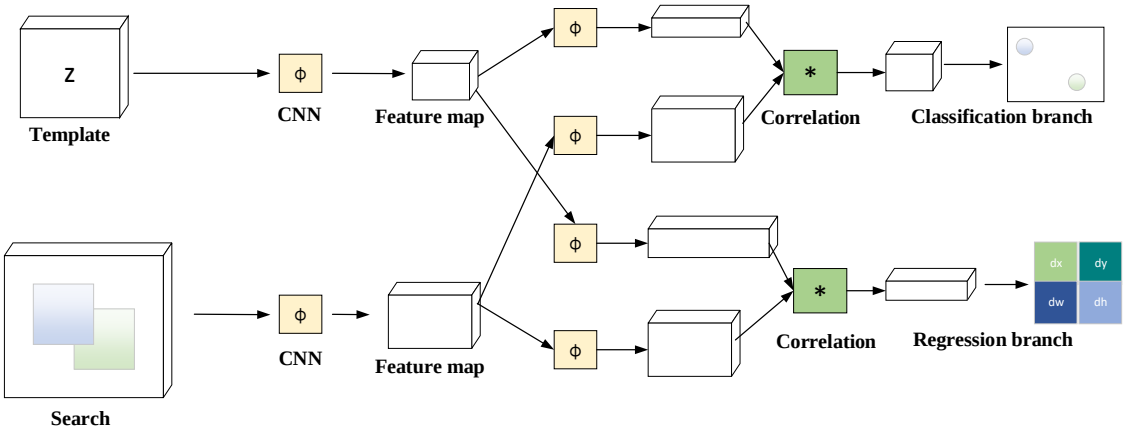


Fig. 1. SiamRPN network structure

Effectively integrating LBP features with deep learning techniques is still a challenge in the research that has already been done. Furthermore, it is still urgently needed to address the issue of varying the attention of features at different layers in deep networks. The goal of this research is to improve the problem of multi-level information mining in object tracking by presenting a coordinated attention mechanism and an information fusion approach based on LBP characteristics. Next, we will present an in-depth description of our methodology and use tests to show how effective it is in object tracking tasks.

III. Our Approach

A. LBP feature extraction

The process of extracting unique and reliable features from image sequences in order to characterize the appearance and motion information of targets is known as feature extraction, and the quality of this step has a direct impact on how well tracking algorithms perform. Conventional tracking techniques frequently track target objects using only one feature, which can lead to incomplete feature representation during tracking, which in turn causes poor tracking performance and results. Moreover, the majority of deep learning-based tracking techniques rely only on deep semantic features extracted by artificial neural networks, ignoring the enhancement effect of image local texture features and shallow features containing appearance information on tracking performance.. In this research, we first extract LBP features from video frames following preprocessing. The video frames with the LBP features that were extracted are then sent into a Siamese network for training. We use skip connections to combine shallow and deep features that we extract from the Siamese network, which improves the features' capacity to be represented.

Since its initial proposal by Ojala et al. [13] in 1994, the Local Binary Pattern has been widely used because of its exceptional capacity to characterize local texture elements in images. The LBP operator has great discriminability, high computational efficiency, and invariance to monotonic grayscale changes. This technique compares each pixel's grayscale value to that of its nearby neighboring pixels, converting the comparison findings into binary values to create local binary patterns. It is based on the grayscale disparities between local pixels in a picture. The primary disadvantage of the basic LBP operator, however, is that it can only accommodate a limited area and a fixed radius range, making it unable to accommodate textures with varying sizes and frequencies. Ojala et al. [14] extended the 3×3 neighborhood to any neighborhood and substituted a circular neighborhood for the square one in order to increase the LBP operator's ability to adapt to texture features of varied scales and achieve grayscale and rotation invariance. An arbitrary number of pixels can be included in a circular neighborhood with radius R via the enhanced LBP operator. This results in LBP operators, which may be described mathematically as follows: these operators have a radius of R and contain P sampling points within the circular region.

$$\mathrm{LBP}_{P,R}(x_c, y_c) = \sum_{m=1}^{P} s(I(m) - I(c)) * 2^m \tag{1}$$

Where $\mathrm{I(c)}$ indicates the grayscale value of the center pixel, $\mathrm{I(m)}$ indicates the grayscale value of the m th point on the circular boundary, and m indicates the m th sampling point out of a total of P sampling points within the circular region. The circular boundary has m points in total, and these points' coordinates are as follows:

$$\begin{cases} x_m = x_c + R * \cos(\dfrac{2\pi m}{p}) \\ y_m = y_c - R * \sin(\dfrac{2\pi m}{p}) \end{cases} \tag{2}$$

The formula for $\mathrm{s(x)}$ is still the same as it was in the initial LBP and is written like this:

$$s(x) = \begin{cases} 1, x \geq 0 \\ 0, otherwise \end{cases} \tag{3}$$

Figure 2(c) shows the LBP feature extraction for the video frame with R=1 and P=8. Since picture (a) is a color image, the first step after preprocessing the video sequence is to convert it to grayscale. Image (b) shows the sequence's resulting grayscale image. After that, LBP feature extraction is carried out, producing the local binary pattern that is seen in image (c).

(a) Original sequence diagram (b) Grayscale sequence diagram (c) Local binary plots

Fig. 2 LBP feature extraction

In comparison to the original video, the frames processed through Local Binary Pattern feature extraction exhibit enhanced texture information. These texture images are proficient in detecting edges and corners, thereby emphasizing their edge characteristics, even in the presence of noise and low resolution. Additionally, LBP features demonstrate robustness against noise and variations in lighting, which aids in alleviating feature distortion caused by environmental changes. The integration of LBP features with the feature learning capabilities of Siamese networks facilitates improved adaptability to target tracking tasks across various challenging conditions. The advantages of LBP features, including invariance to rotation and illumination, as well as a strong correlation with target variations, are fully utilized when training the deep learning network SiamRPN on a dataset comprising textured video frames. Consequently, the SiamRPN model is able to identify target feature information with greater accuracy when trained on textured video frames compared to conventional video frames.

B. twin network tracker with multi-feature fusion to coordinate attention mechanisms

Conventional trackers such as SiamFC and SiamRPN extract features using the conventional five-layer AlexNet backbone. We substitute ResNeSt for the conventional AlexNet backbone in order to deepen the network. We do not apply score-level fusion to the feature information once feature extraction is finished, with the exception of the last layer. It is evident by visualizing the graphics of each depth layer that the resolution of the feature maps reduces as network layers and depth rise. They are separated into three layers: shallow, medium, and deep. While the medium layer can roughly identify the target's appearance, the deep layer cannot recognize the target's appearance but contains semantic information that makes it more robust to deformations in target images and suitable for target classification. The features in the shallow layer have the highest resolution and contain detailed appearance information. In light of these findings, we suggest fusing shallow and deep features to compensate for each other's shortcomings and enhance target tracking performance.

This paper selects picture features through the feature fusion process by incorporating an attention mechanism [15] since not all features in the network layers are helpful. It has less of an impact on tracking performance when greater weights are given to feature information that is relevant to the tracking job and lower weights are given to irrelevant feature information. This helps with improved feature extraction by guaranteeing efficient feature extraction for the target object across all channels. Figure 3 shows the process of feature fusion.

The Convolutional Block Attention Module (CBAM) is the most widely used framework at the moment [16]. Although it takes into account both spatial position and channel information, the large-scale pooling it uses may cause positional information in image frames to be lost. As a result, we present Coordinate Attention, a rather more recent technique [17]. Figure 4 illustrates its particular structure. The operation of the coordinate attention block consists of two parts: coordinate information embedding and coordinate attention generation. The latter focuses on acquiring positional information and generating weight values, while the former encodes channel information in both horizontal and vertical dimensions.

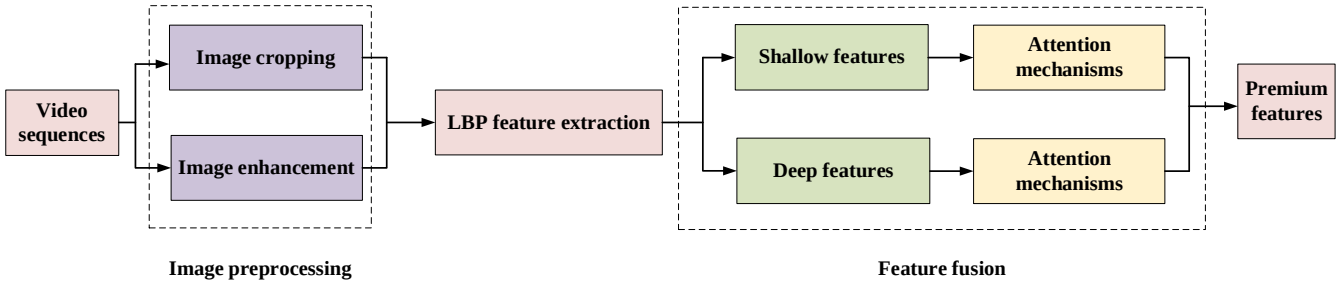


Fig. 3 Feature fusion process

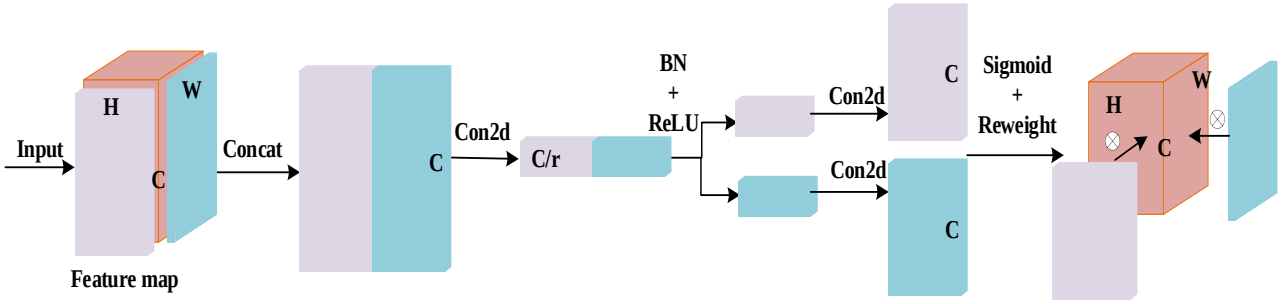


Fig. 4 Coordinating the attention calculation process

1） Coordinate information embedding

Based on the given input feature $X=[x_1,x_2,\ldots,x_c]\in\mathbb{R}^{C\times H\times W}$, for each channel, a pooling kernel of size (H,1) or (1,W) is used to encode in the horizontal and vertical directions, respectively. Therefore, the output of the c-th channel at height H can be represented as:

$$z_c^h(h)=\frac{1}{W}\sum_{0\le i\le w}x_c(h,i) \tag{4}$$

In a similar vein, channel C's output at width W is:

$$z_c^w(w)=\frac{1}{H}\sum_{0\le i\le \mathrm{H}}x_c(j,w) \tag{5}$$

2） Coordinate attention generation

The length of the two encoded features is then converted to $(H+W)$ by concatenating them along the spatial dimension. They are then processed with a common convolutional transformation function F_1, producing the following outcomes:

$$f=\delta(F_1([z^h,z^w])) \tag{6}$$

In this case, $f\in\mathbb{R}^{c/r\times(H+W)}$ is the intermediate feature mapping that encodes spatial information in both horizontal and vertical directions, $[z^h,z^w]$ stands for concatenation operation along the spatial dimension, and δ indicates a nonlinear activation function.After that, f is divided into two distinct tensors along the spatial dimension. To convert f^h and f^w into tensors with the same number of channels as the input *X*, two convolution transformations are applied to them, accordingly. Last but not least, a sigmoid activation function is used to produce:

$$\begin{cases}g^h=\delta(F_h(f^h))\\ g^w=\delta(F_w(f^w))\end{cases} \tag{7}$$

F_h and F_w are two convolutions, g^h and g^w are two-dimensional weights.

Ultimately, the input feature X is fused with g^h and g^w to produce the Coordinate Attention Module's output.

$$y_c(i,j)=x_c(i,j)\times g_c^h(i)\times g_c^w(j) \tag{8}$$

To enhance the attentional scope while reducing computational demands, the Coordinate Attention Module incorporates spatial location information into channel attention mechanisms. This module effectively mitigates the loss of positional information that arises from the two-dimensional global pooling utilized in the Convolutional Block Attention Module (CBAM). It achieves this by amalgamating both channel and spatial coordinate information into the generated attention map through the use of two concurrent one-dimensional feature encoders. It functions as a plug-and-play, adaptable, and lightweight model that successfully handles long-range dependence difficulties in addition to taking into account channel and spatial information.

On the basis of the aforementioned research, we suggest a shallow-deep multi-feature fusion network with LBP-based attention for SiamRPN. Figure 5 shows the network architecture. The search image is set to 255×255, and the template image is set to 127×127. Following preprocessing, LBP feature extraction is applied to the video frames. The SiamRPN network is then trained using the local binary pattern maps that are produced. Deep and shallow features are then retrieved and combined at the feature level. To complement deep and shallow information, the conv3 and conv5 layers from both branches are merged using skip connections after going through the coordinated attention method. We utilize Max pooling and 1×1 convolutional modules to standardize the dimensions of the fused feature maps, thereby preserving spatial information, as the size and channel dimensions of feature maps differ across various layers of the network. The RPN network receives the fused feature maps after that in order to process them further. To improve model representation and real-time performance, the weighted fused final outputs from the regression and classification branches are combined. To differentiate between the target and background, the classification branch is required to compute the intersection over union (IOU) between the predicted and actual bounding boxes of the target. Regression branch adjusts for anticipated scale changes during early tracking stages by precisely matching the predicted bounding box with the actual target state. Coordinated attention is added to the shallow-deep feature fusion network to filter out far-off disturbances with little effect. This results in more convergent peaks without subtle or dispersed disturbances in the score map, which increases tracker accuracy.

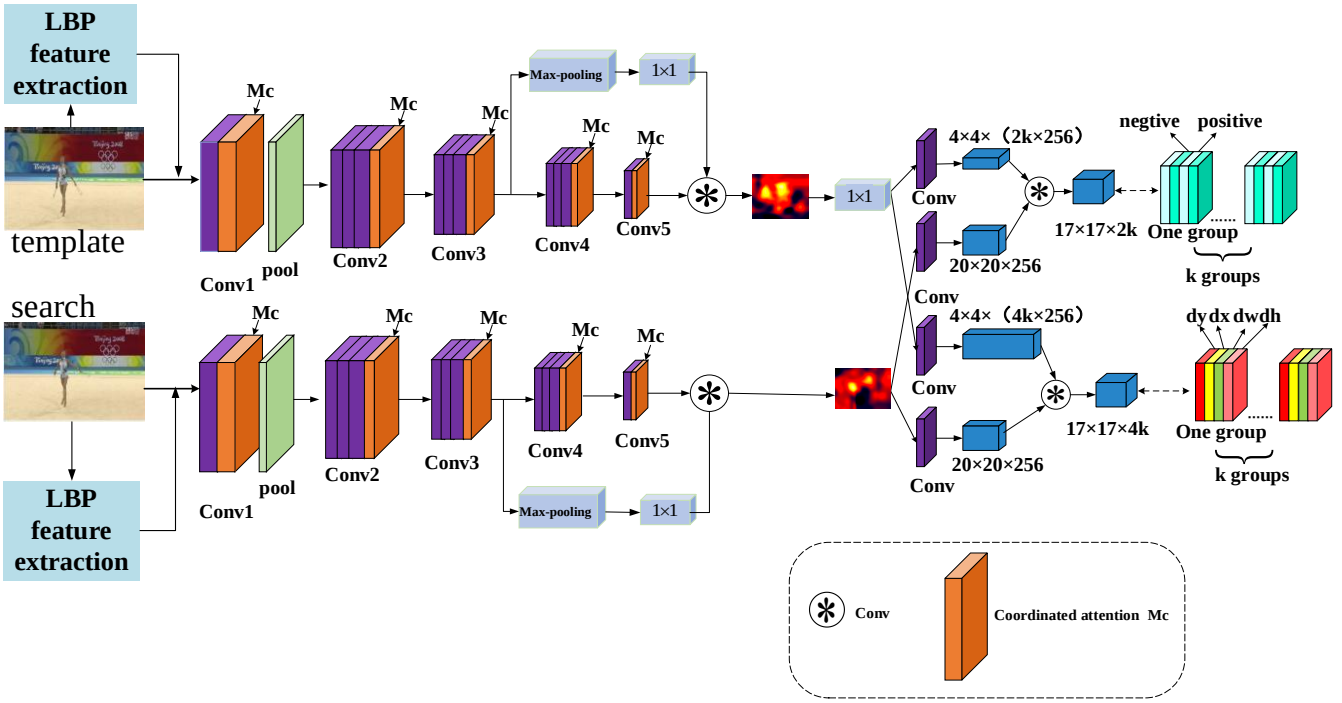


Fig.5 Network structure diagram of LBP and shallow deep multi-feature fusion attention

IV. EXPERIMENT

A. *Details of the experiment*

The experimental setup employed the Ubuntu 16.04 LTS operating system, equipped with an Intel Xeon E5 central processing unit (CPU) and an NVIDIA 2080 Ti GPU. The software utilized for the experiment was MATLAB version 2018b. The datasets incorporated in this research included GOT-10k[18],ILSVRC[19],OTB2015[20], and VOT2018[21]. The network hyperparameters were configured with a learning rate of 0.005, a batch size of 16, and a total of 80 epochs.

B. *Training details*

We trained the network more thoroughly on the ILSVRC and GOT-10k datasets because we used ResNeSt as the backbone network instead of the more conventional AlexNet, and ResNeSt had previously been populated with image labels on the ImageNet dataset. In order to create the final score map, we first extracted features from the template and search photos using CNN, then we performed pertinent procedures. The following sums up the complete procedure:

$$S(z,x) = f(\varphi(z), \varphi(x)) \tag{9}$$

In this case, $\varphi(z)$ stands for the features of the template picture, $\varphi(x)$ for the features of the search image, $f(\cdot)$ for related operations, $S(z,x)$ for the similarity between the template and search images, and the network's ultimate objective is to maximize $S(z,x)$. The logical loss function, which is defined as follows, is used to train the network.

$$L(y,v) = \frac{1}{D}\sum_{u\in D}\log(1+\exp(-y[u]v[u])) \tag{10}$$

In the score map, u stands for a score point, $v[u]$ for its similarity score, and $y[u]$ for its ground truth label. Stochastic gradient descent (SGD) is used to optimize the loss function and update and obtain the weight parameters of the network. $y[u]$ is defined based on the position of any point on the score map with respect to the target center.

$$y[u] = \begin{cases} +1 & k \parallel u-c \parallel \leq R \\ -1 & otherwise \end{cases} \tag{11}$$

Where c is the target image center point and k is the network step size.

To obtain patches measuring 127×127 pixels for the template image and 255×255 pixels for the search image, we perform cropping around the designated target position during the training process. In instances where the cropped area is insufficient in size, we utilize the average RGB values to populate the incomplete sections.

The RPN network is then trained using the outputs $\varphi(z)$ and $\varphi(x)$ from the two branches of the Siamese network. Using two distinct convolution procedures, $\varphi(z)$ is initially divided into two branches, $\varphi(z)_{\text{cls}}$ and $\varphi(z)_{\text{reg}}$, which correspond to 2k and 4k channels, respectively, in the correlation operation. Convolutions are also used to divide $\varphi(x)$ into two branches, $\varphi(x)_{cls}$ and $\varphi(z)_{\text{reg}}$. The number of channels in $\varphi(x)$ stays the same, in contrast to $\varphi(z)$. After that, a unique "convolution" operation is carried out with the feature maps of $\varphi(x)_{cls}$ and $\varphi(x)_{reg}$ as kernels, producing outputs with sizes of 17x17x2k and 17x17x4k, respectively. The definition of the convolution operation is as follows:

$$A_{w\times h\times 2k}^{cls} = [\varphi(x)]_{cls} * [\varphi(z)]_{cls}$$
$$A_{w\times h\times 4k}^{reg} = [\varphi(x)]_{reg} * [\varphi(z)]_{reg} \quad (12)$$

The convolution operation is represented by the symbol *. The ultimate output of the classification branch consists of a feature map that includes 2k channels, with K denoting the number of anchors. Each of the k groups within this feature map is associated with a score map consisting of two channels, which reflect the scores for foreground and background classifications. Conversely, the regression branch produces a feature map with 4,000 channels. These channels are not only divided into K groups but are also organized into sets of four, which correspond to the dimensions and central positions of each anchor.

C. OTB2015 experiments

In the context of the OTB2015 datasets, Precision and Success are two key metrics employed to evaluate the performance of tracking algorithms. The evaluation of tracker robustness is conducted through the One Pass Evaluation (OPE) methodology.

1) Accuracy

Utilizing the Euclidean distance computation, we assess the proportion of video frames that fall below a specified threshold. The subsequent section presents a structured outline of the formula:

$$s = \sqrt{(x_u - x_r)^2 + (y_u - y_r)^2} \quad (13)$$

The center point of the ground truth is denoted by (x_r, y_r) in the equation, and the center point of the anticipated bounding box is represented by (x_u, y_u). Better tracking performance is indicated by a smaller value of s. We may create a curve by changing the threshold, and higher values signify the tracker's improved performance.

2) Success

The Overlap Score (OS) can be utilized to quantify the effectiveness of target tracking. It is computed as follows:

$$OS = \frac{|bounding\ \ box \cap ground\ \ truth\ \ box|}{|bounding\ \ box \cup ground\ \ truth\ \ box|} \quad (14)$$

where $|\bullet|$ stands for the quantity of pixels in the area. Object tracking is considered successful when the OS (Object Similarity) value of any given frame exceeds a predetermined threshold; conversely, it is classified as unsuccessful if the value does not meet this criterion. Typically, the threshold is established at 0.5.

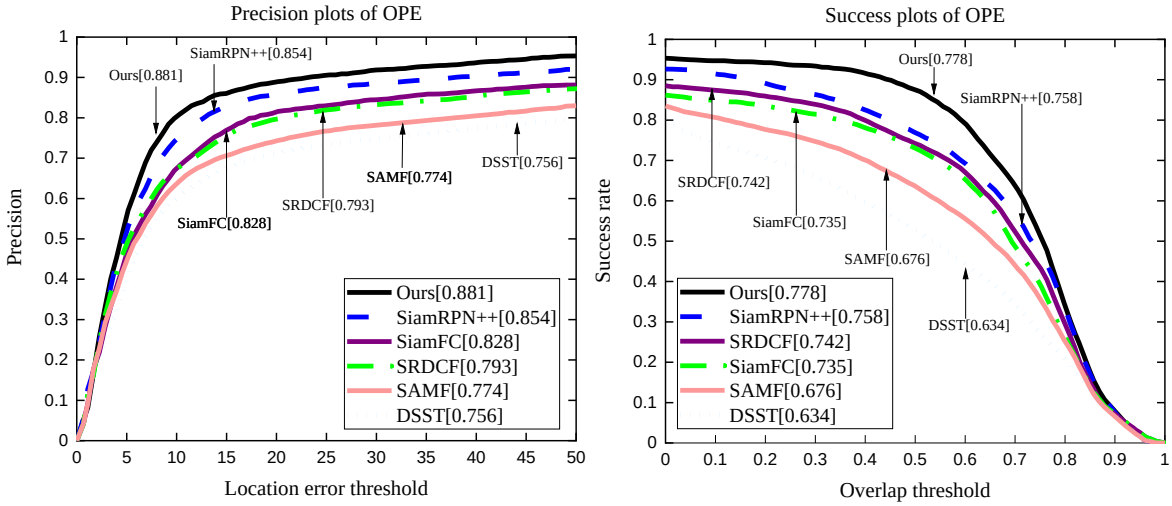
3) Single Trip Assessment

In order to establish the ground truth target location, the One Pass Evaluation for testing tracking utilizes solely the initial frame of the video sequence. Subsequently, the algorithm is executed to assess its accuracy and success rate. Experimental evaluations are conducted on the OTB2015 dataset to assess the performance of the proposed algorithm in comparison to leading tracking algorithms, specifically DSST, SAMF[22], SRDCF[23], SiamFC, and SiamRPN++. These trackers are based on Siamese networks, whereas DSST, SAMF, and SRDCF are correlation filter-based trackers. Figure 6 presents a comparative analysis of the results obtained from our proposed methodology in relation to alternative tracking algorithms. Figure 6(a) presents the precision plot, wherein "Ours" denotes the algorithm proposed in this study. The score located in the upper right corner indicates the tracker's performance when the center inaccuracy of the pixel distance is set to 20. In Figure 6(b), the success plot is illustrated, with the score in the top right corner representing the area under the curve. The results of the tests indicate that the proposed algorithm surpasses other tracking algorithms in terms of both precision and success rate. Additionally, Table I offers a comprehensive comparison of the performance disparities among various tracking algorithms based on these two metrics.

The approach presented in this research outperforms SiamRPN++ in terms of precision and success rate, by 3.16% and 2.64%, respectively. The gains in precision and success rate over SRDCF are 12.1% and 4.85%, respectively. Our method reaches a tracking speed of 65 frames per second, which is 10 frames faster than SRDCF, 38 frames faster than SAMF, and 43 frames faster than DSST. The technique still satisfies the requirements for real-time tracking, despite the fact that the addition of the coordinated attention mechanism and the merging of shallow and deep features under the LBP framework have increased the computational cost of the algorithm and caused a modest decrease in tracking speed. We tested in 11 unconstrained environments: "low resolution," "background clutter," "out of view," "out-of-plane rotation," "in-plane rotation," "fast motion," "motion blur," "deformation," "occlusion," "scale variation," and "illumination variation" in order to learn more about the algorithm's tracking performance in various settings. Figures 7 and 8, respectively, present comprehensive precision and success rate data.

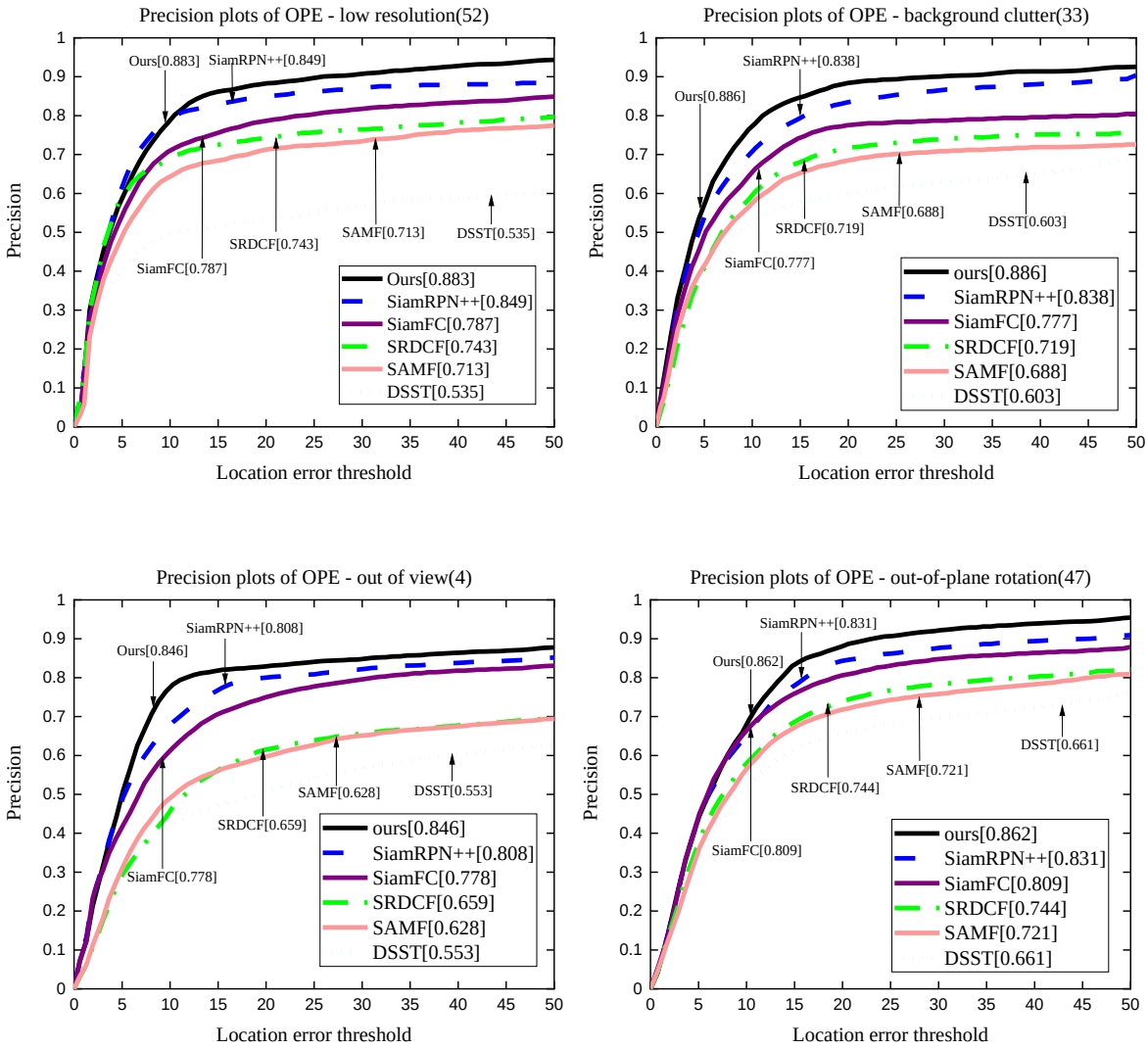
TABLE I
PRECISION AND SUCCESSION SCORES OF TRACKER. "IMPROVE" INDICATES OUR TRACKER'S IMPROVEMENT OVER OTHER, AND "SPEED" INDICATES THE TRACKING SPEED

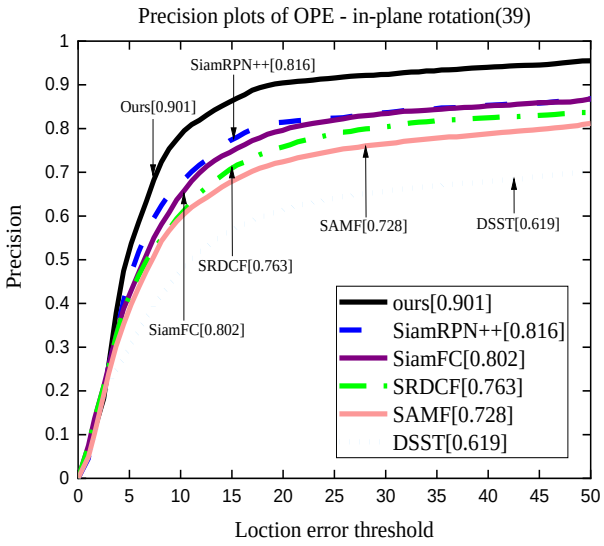
Tracking	Precision score	Success score	Improve(%)		Speed(FPS)
			Pre. (%)	Succ. (%)	
SiamRPN++	0.854	0.758	3.16	2.64	160
SiamFC	0.828	0.735	6.4	5.85	86
SRDCF	0.793	0.742	12.10	4.85	55
SAMF	0.774	0.676	13.82	15.09	27
DSST	0.756	0.634	16.53	22.71	22
Ours	0.881	0.778	-	-	65

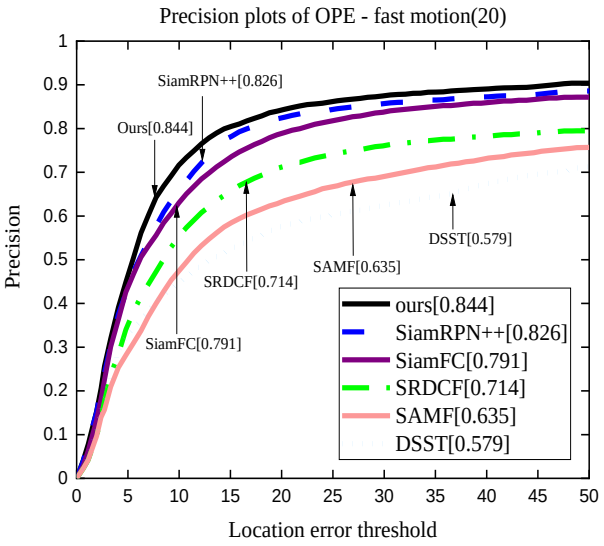


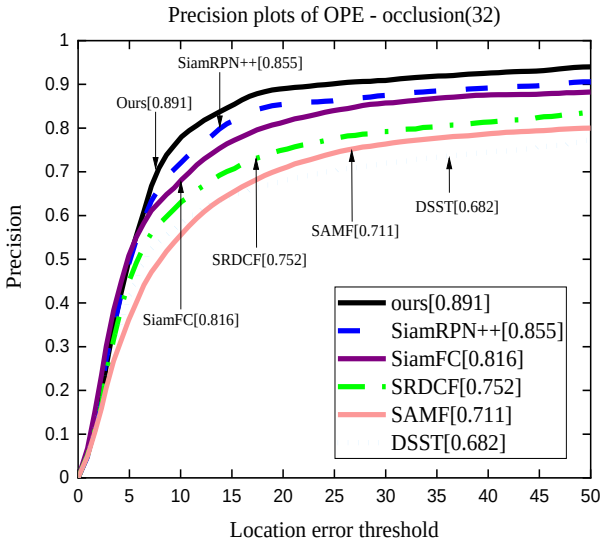
(a)Precision (b)Succession

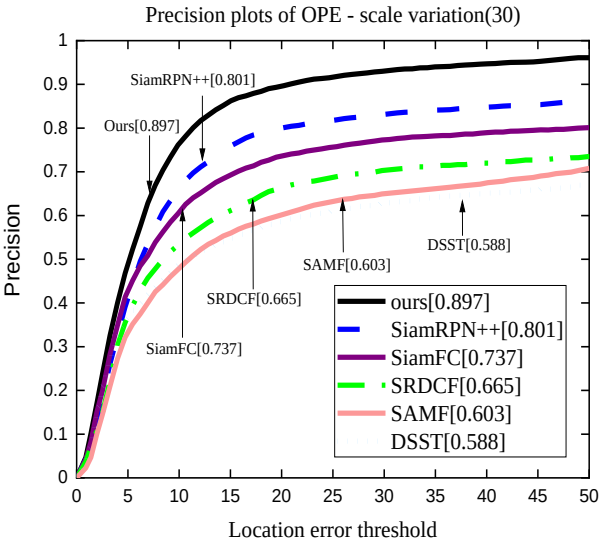
Fig. 6 presents the analysis of precision and success rate on the OTB2015 dataset using OPE

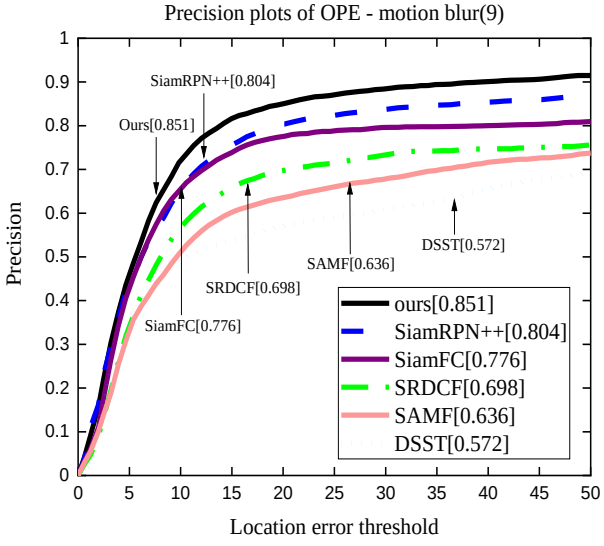


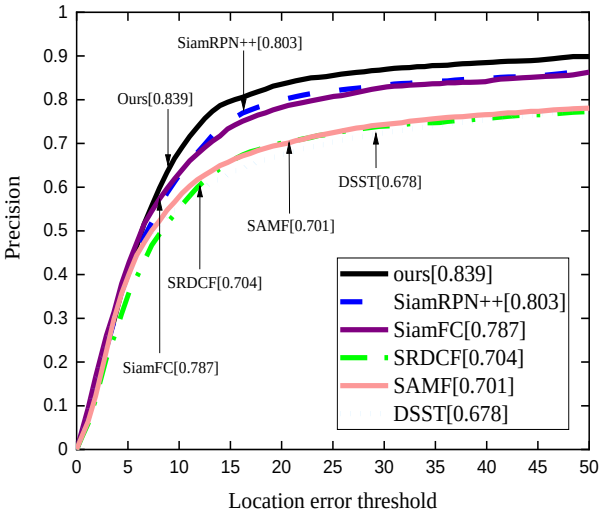

Precision plots of OPE - in-plane rotation(39)
SiamRPN++[0.816]
Ours[0.901]
SAMF[0.728]
DSST[0.619]
SRDCF[0.763]
SiamFC[0.802]
ours[0.901]
SiamRPN++[0.816]
SiamFC[0.802]
SRDCF[0.763]
SAMF[0.728]
DSST[0.619]
Precision
Loction error threshold


Precision plots of OPE - fast motion(20)
SiamRPN++[0.826]
Ours[0.844]
SRDCF[0.714]
SAMF[0.635]
DSST[0.579]
SiamFC[0.791]
ours[0.844]
SiamRPN++[0.826]
SiamFC[0.791]
SRDCF[0.714]
SAMF[0.635]
DSST[0.579]
Precision
Location error threshold


Precision plots of OPE - occlusion(32)
SiamRPN++[0.855]
Ours[0.891]
SAMF[0.711]
DSST[0.682]
SRDCF[0.752]
SiamFC[0.816]
ours[0.891]
SiamRPN++[0.855]
SiamFC[0.816]
SRDCF[0.752]
SAMF[0.711]
DSST[0.682]
Precision
Location error threshold


Precision plots of OPE - scale variation(30)
SiamRPN++[0.801]
Ours[0.897]
SAMF[0.603]
DSST[0.588]
SRDCF[0.665]
SiamFC[0.737]
ours[0.897]
SiamRPN++[0.801]
SiamFC[0.737]
SRDCF[0.665]
SAMF[0.603]
DSST[0.588]
Precision
Location error threshold


Precision plots of OPE - motion blur(9)
SiamRPN++[0.804]
Ours[0.851]
SAMF[0.636]
DSST[0.572]
SRDCF[0.698]
SiamFC[0.776]
ours[0.851]
SiamRPN++[0.804]
SiamFC[0.776]
SRDCF[0.698]
SAMF[0.636]
DSST[0.572]
Precision
Location error threshold


Precision plots of OPE- deformation(27)
SiamRPN++[0.803]
Ours[0.839]
SAMF[0.701]
DSST[0.678]
SRDCF[0.704]
SiamFC[0.787]
ours[0.839]
SiamRPN++[0.803]
SiamFC[0.787]
SRDCF[0.704]
SAMF[0.701]
DSST[0.678]
Precision
Location error threshold

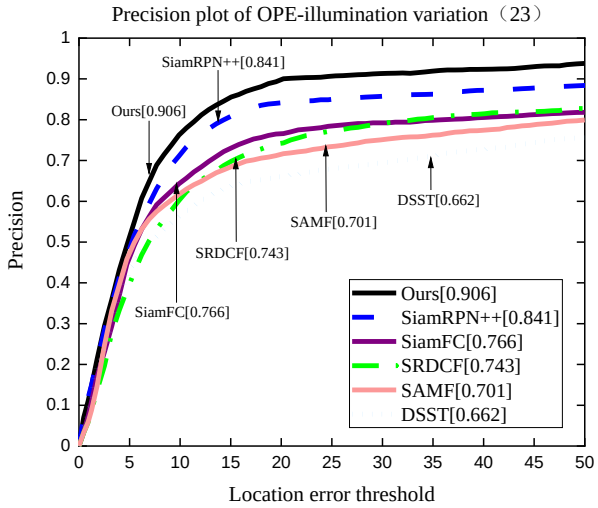
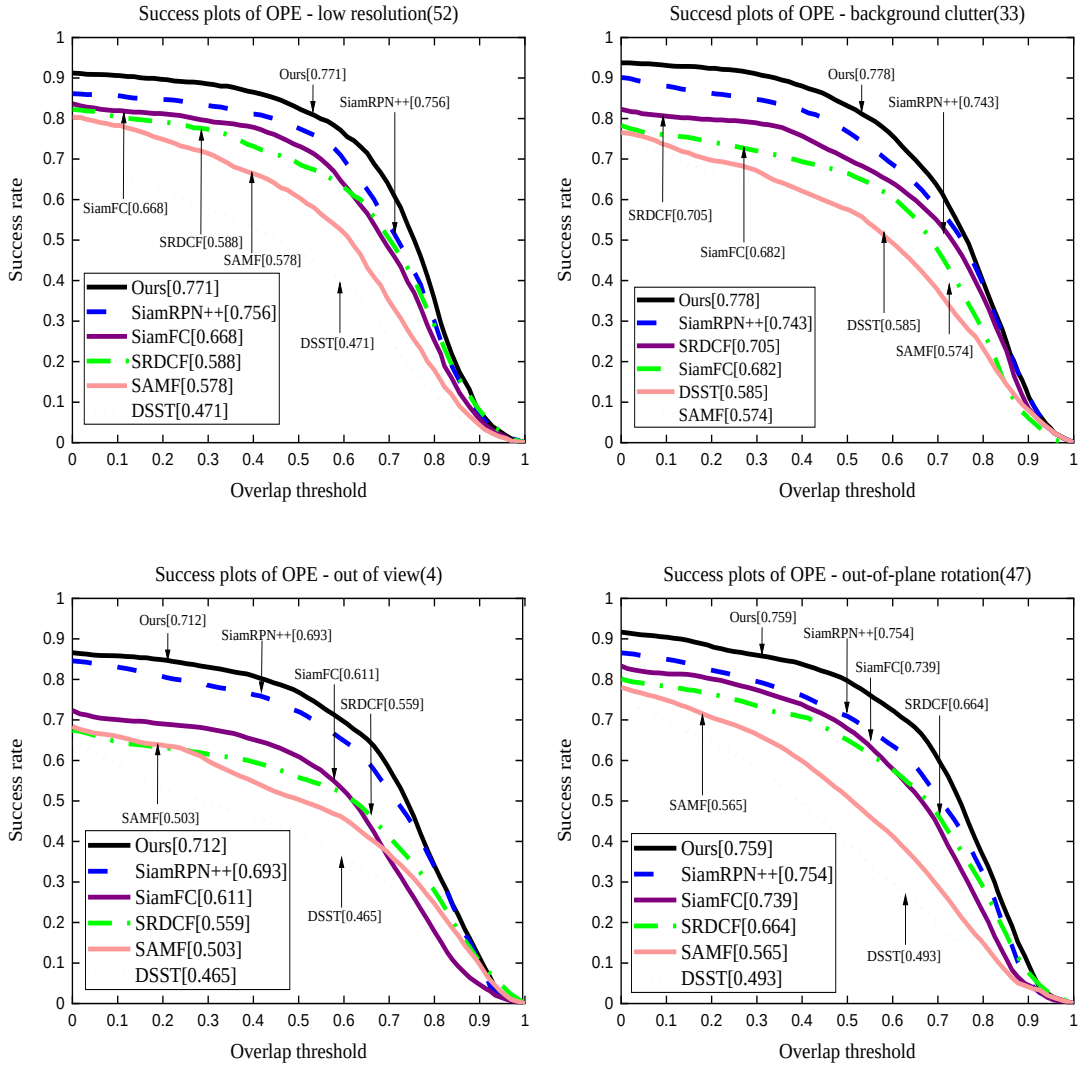
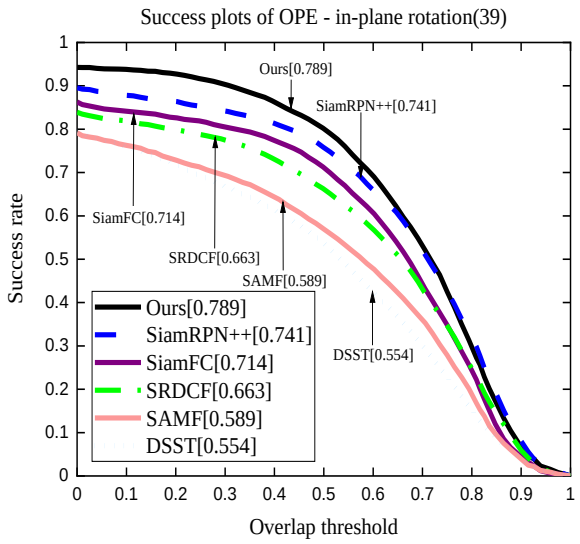
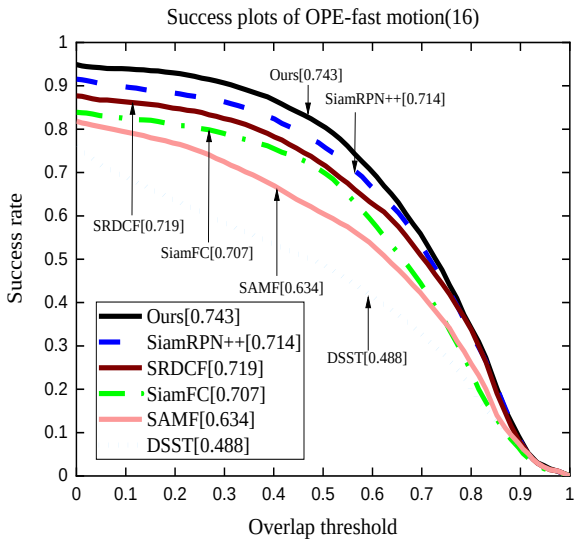
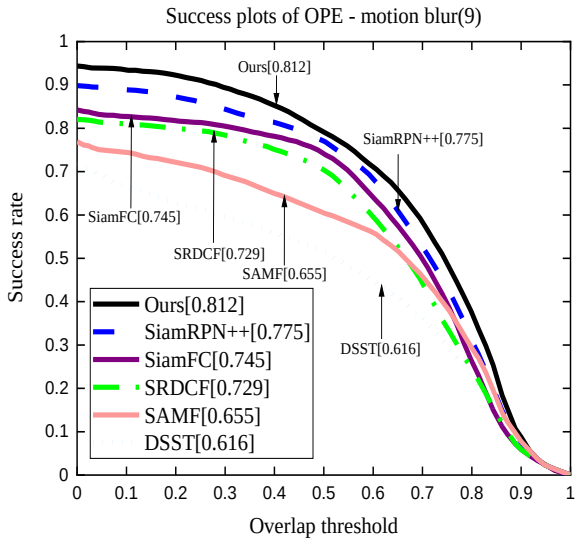


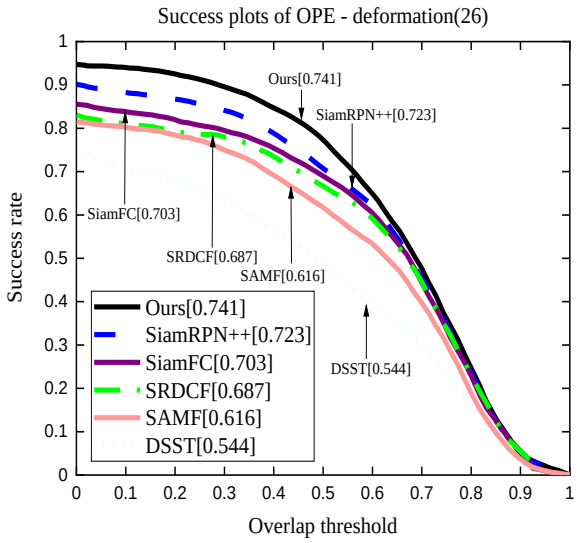
Fig. 7 illustrates the precision plots on OTB2015 for 11 tracking scenarios including "Low Resolution," "Background Clutter," "Out of View," "Out-of-Plane Rotation," "In-Plane Rotation," "Fast Motion," "Motion Blur," "Deformation," "Occlusion," "Scale Variation," and "Illumination Variation."

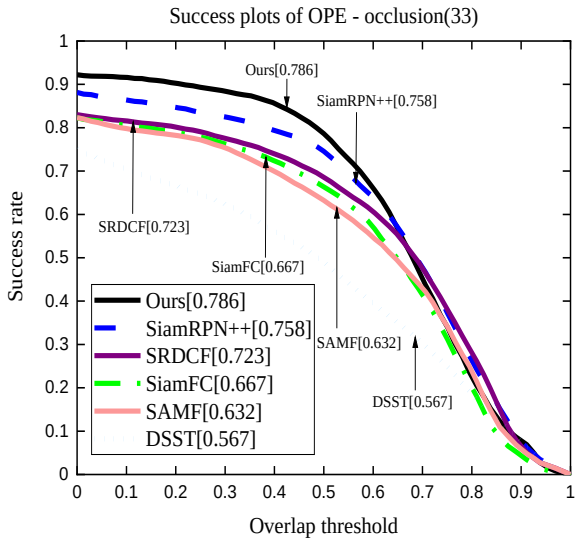


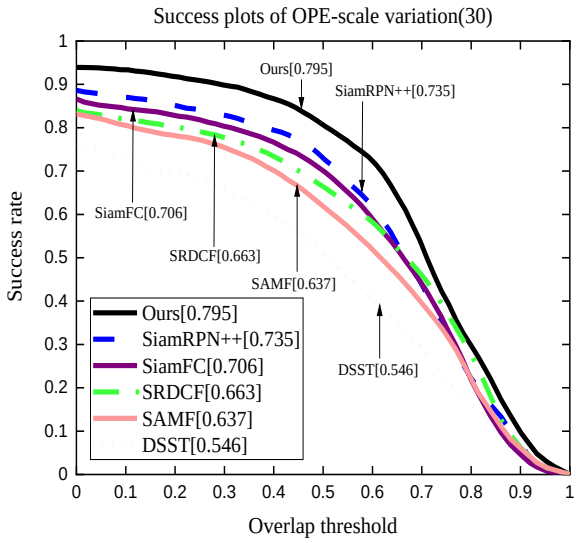

Success plots of OPE - in-plane rotation(39)
Success rate
Overlap threshold
Ours[0.789]
SiamRPN++[0.741]
SiamFC[0.714]
SRDCF[0.663]
SAMF[0.589]
DSST[0.554]
Ours[0.789]
SiamRPN++[0.741]
SiamFC[0.714]
SRDCF[0.663]
SAMF[0.589]
DSST[0.554]


Success plots of OPE-fast motion(16)
Success rate
Overlap threshold
Ours[0.743]
SiamRPN++[0.714]
SRDCF[0.719]
SiamFC[0.707]
SAMF[0.634]
DSST[0.488]
Ours[0.743]
SiamRPN++[0.714]
SRDCF[0.719]
SiamFC[0.707]
SAMF[0.634]
DSST[0.488]


Success plots of OPE - motion blur(9)
Success rate
Overlap threshold
Ours[0.812]
SiamRPN++[0.775]
SiamFC[0.745]
SRDCF[0.729]
SAMF[0.655]
DSST[0.616]
Ours[0.812]
SiamRPN++[0.775]
SiamFC[0.745]
SRDCF[0.729]
SAMF[0.655]
DSST[0.616]


Success plots of OPE - deformation(26)
Success rate
Overlap threshold
Ours[0.741]
SiamRPN++[0.723]
SiamFC[0.703]
SRDCF[0.687]
SAMF[0.616]
DSST[0.544]
Ours[0.741]
SiamRPN++[0.723]
SiamFC[0.703]
SRDCF[0.687]
SAMF[0.616]
DSST[0.544]


Success plots of OPE - occlusion(33)
Success rate
Overlap threshold
Ours[0.786]
SiamRPN++[0.758]
SRDCF[0.723]
SiamFC[0.667]
SAMF[0.632]
DSST[0.567]
Ours[0.786]
SiamRPN++[0.758]
SRDCF[0.723]
SiamFC[0.667]
SAMF[0.632]
DSST[0.567]


Success plots of OPE-scale variation(30)
Success rate
Overlap threshold
Ours[0.795]
SiamRPN++[0.735]
SiamFC[0.706]
SRDCF[0.663]
SAMF[0.637]
DSST[0.546]
Ours[0.795]
SiamRPN++[0.735]
SiamFC[0.706]
SRDCF[0.663]
SAMF[0.637]
DSST[0.546]

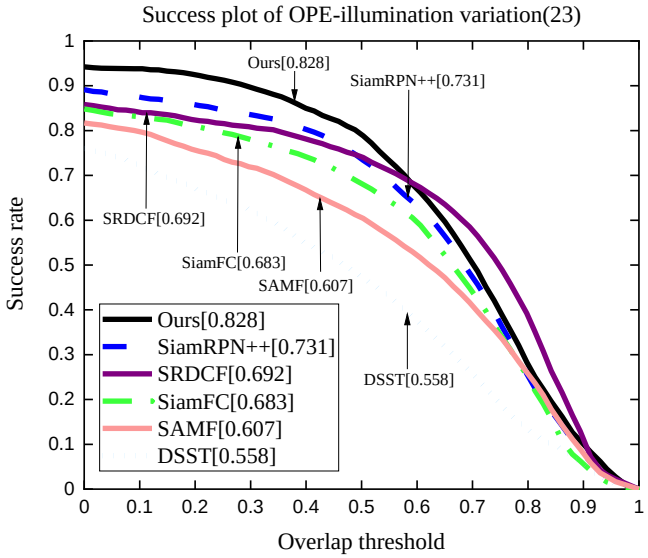


Fig. 8 depicts the success plots on OTB2015 for 11 tracking scenarios including "Low Resolution," "Background Clutter," "Out of View," "Out-of-Plane Rotation," "In-Plane Rotation," "Fast Motion," "Motion Blur," "Deformation," "Occlusion," "Scale Variation," and "Illumination Variation."

Coordinated attention is incorporated into the proposed approach, which improves target localization accuracy by giving the target image more weight and decreasing the weight of distractions during feature extraction. Furthermore, training with local binary pattern pictures improves the target's edge properties, reducing the effect of lighting fluctuations on tracking accuracy. Furthermore, the integration of shallow and deep features, leveraging the visual characteristics of shallow layers, significantly diminishes the discrepancy between the projected and actual target bounding boxes. The experimental results show that the suggested method consistently scores highest across all assessment metrics in the 11 unconstrained scenarios.Impressively, it performs exceptionally well in scenarios like illumination variation and in-plane rotation, with precision rates 7.73% and 10.42% higher than the second-ranked SiamRPN++ and 18.28% and 12.34% higher than the third-ranked SRDCF and SiamFC, respectively. Additionally, the algorithm continues to win in scenarios such as Scale variation and Motion blur, with success rates that beat those of the second-ranked SiamRPN++ by 8.16% and 4.77%, and the third-ranked SiamFC by 12.61% and 8.99%, respectively. Figure 9 shows tracking results from six chosen test videos in order to give a more comprehensible representation of the tracking performance.

D. VOT2018 experiments

Three assessment measures were used to assess the trackers in the VOT2018 testing: robustness, accuracy, and expected average overlap (EAO).

1) Robustness

The number of tracking failures is used to measure robustness; lesser values correspond to a tracker that is more robust.

2) Expected Average Overlap (EAO)

A tracker's accuracy and robustness are measured using EAO, where larger numbers denote greater performance.

Using the VOT2018 dataset, we evaluated our suggested approach against SiamRPN++, SiamFC, KCF, DSST[24], SAMF, ACT[21], SiamAN[21], ColorKCF[25], and TCNN[26]. Figure 10 presents a comparison plot of robustness and accuracy derived from the VOT2018 dataset, where robustness is indicated on the horizontal axis and accuracy on the vertical axis. The position of the tracker in relation to the top-right corner of the graph correlates positively with its performance. The data illustrated in the graph clearly indicates that the proposed method demonstrates superior performance in both accuracy and robustness.

The 10 trackers' comprehensive quantified results across a range of performance evaluation indicators are shown in Table II. Table II clearly shows that, with the exception of speed, the tracking strategy suggested in this research produces the highest test scores. Nonetheless, our suggested method's tracking speed satisfies real-time needs. According to these comparisons, SiamRPN++, which is rated second, is ranked lower in terms of EAO by 9.47%, Accuracy by 8.49%, Failures by 18.19%, and Overlap by 2.66%.

E. Ablation experiments

A series of ablation experiments was conducted to validate the impact of each functional module on tracking performance, as presented in Table III. In these experiments, the phrase "Without any modules" refers to the utilization of the ResNeSt backbone network in isolation for tracking purposes, while the symbol "+" signifies the incorporation of the corresponding functional module in conjunction with the ResNeSt network.

Table 3 indicates that the tracking precision and success rates decrease to 0.807 and 0.688, respectively, when solely utilizing the ResNeSt backbone network on the OTB2015 dataset. The incorporation of LBP feature extraction results in a modest improvement in tracking accuracy and success rates,

which rise to 0.818 and 0.703, respectively. While the performance of both levels of fusion operations is better than that of individual operations, multi-level fusion achieves higher performance. With 0.881 and 0.778 tracking precision and success rates, respectively, we top the rankings. Similar tendencies may be seen in these modules' performance changes across the two datasets.

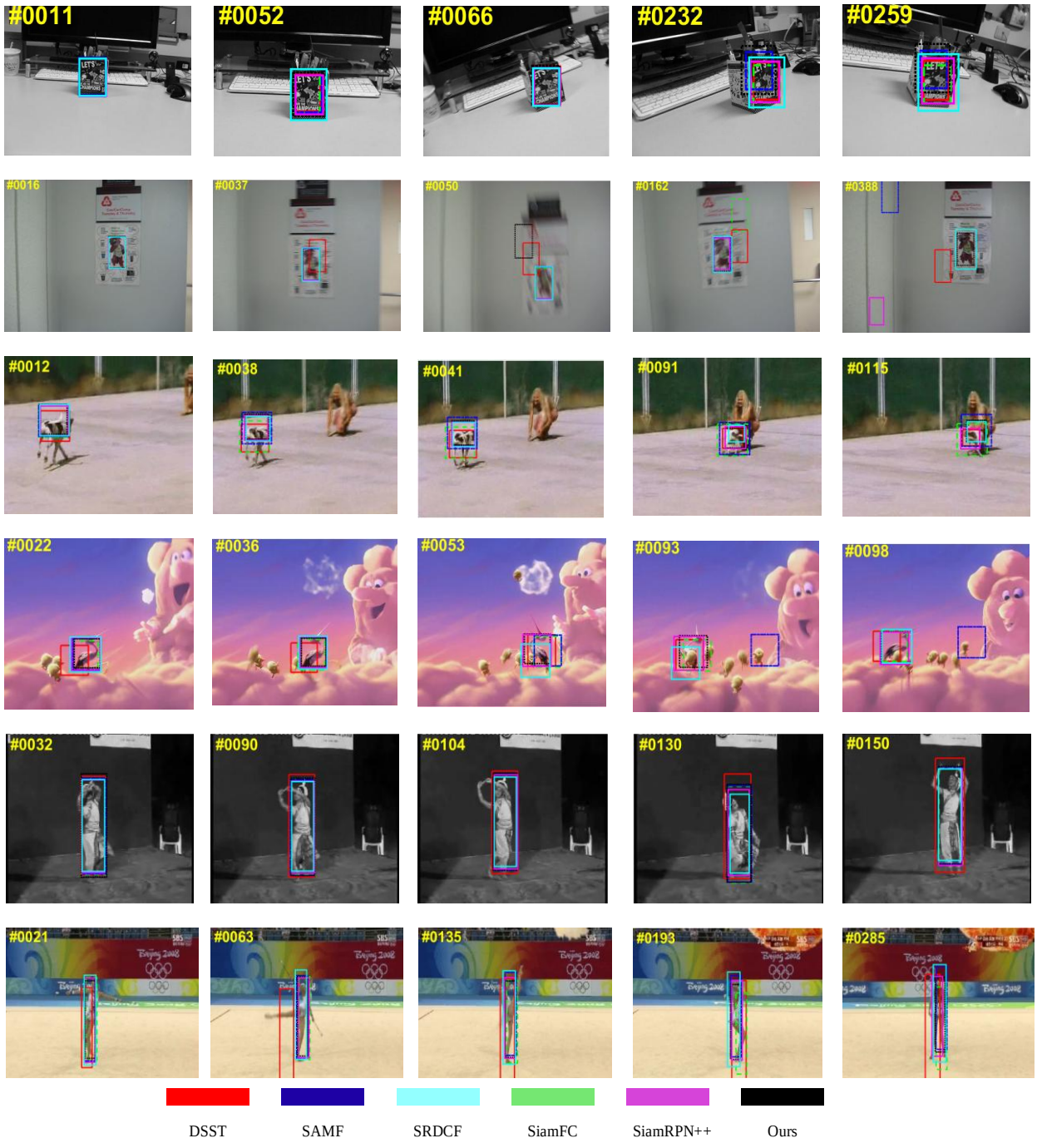


Fig. 9 illustrates the tracking results of DSST, SAMF, SRDCF, SiamFC, SiamRPN++, and our proposed algorithm on the OTB2015 dataset.

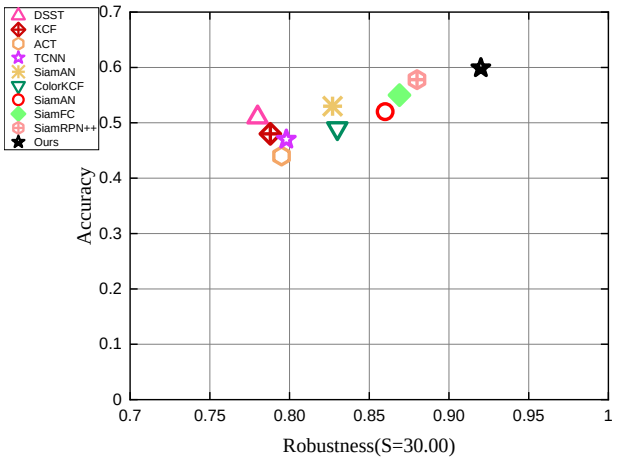


Fig. 10 The Robustness-Accuracy ranking of 10 trackers on the VOT2018 dataset. Trackers with better performance are positioned closer to the top-right corner of the plot.

TABLE II
DISPLAYS THE PERFORMANCE SCORES OF 10 TRACKERS ON THE VOT2018 DATASET ACROSS 5 EVALUATION METRICS.

Tracker	SiamRPN++	SiamFC	SiamAN	TCNN	SAMF	KCF	DSST	ColorKCF	ACT	Ours
EAO	0.3568	0.2599	0.2167	0.3457	0.1782	0.1781	0.1684	0.2156	0.1524	0.3906
Acc.	0.4902	0.4074	0.3911	0.4726	0.3398	0.3013	0.3154	0.3362	0.2755	0.5318
Fail.	21.8364	20.4823	30.0246	18.2324	36.4454	38.1412	45.0868	26.2237	41.1138	14.9161
Overlap	0.5789	0.5333	0.5221	0.5401	0.1988	0.1878	0.5187	0.4926	0.4244	0.5943
FPS	160	86	15	1	27	20	22	111	82	65

TABLE III
TEST RESULTS OF DIFFERENT MODULES IN BENCHMARK EXPERIMENTS

		Without any modules	+LBP	+ Attention mechanism	+ Shallow and deep feature fusion	Ours
OTB2015	Precision score	0.807	0.818	0.851	0.868	0.881
	Success score	0.688	0.703	0.708	0.717	0.778
VOT2018	Accuracy	0.4622	0.4806	0.4879	0.5104	0.5318
	Failures	17.6661	17.5274	17.1924	16.2824	14.9161
	Overlap	0.5022	0.5318	0.5694	0.5791	0.5943
	EAO	0.3478	0.3523	0.3585	0.3884	0.3906
	FPS	89	76	68	66	65

V. CONCLUSION

In this paper, we proposed a multi-feature object tracking algorithm integrating LBP and attention mechanisms to address the shortcomings of conventional Siamese-network-based object tracking methods, which rely on single-feature extraction and are sensitive to lighting variations, causing blurred object representation and position loss, leading to decreased tracking performance. First, local binary patterns were created by LBP feature extraction following the preprocessing of video frames. These patterns were subsequently used to train the SiamRPN network. By improving edge characteristics and local information in the photos, this step partially offset the impacts of illumination variations. Second, we added coordinated attention mechanisms after each convolutional layer to enhance feature extraction accuracy and consistency and to increase network depth by substituting ResNeSt for the conventional AlexNet backbone network. After going through the coordinated attention modules, we then fused the third and fifth levels of the network branches, making efficient use of both shallow and deep features. Ultimately, our approach employed the classification branch to predict both positive and negative samples within the current sequence, while the regression branch was utilized to assess the positional and scale information of the current output target, thereby facilitating the determination of the target's location. We evaluated our proposed methodology alongside several leading algorithms using the OTB2015 and VOT2018 datasets. Our tracking system achieved an accuracy rate of 88.1% and a success rate of 77.8% in the OPE analysis of the OTB2015 dataset. The experimental results indicate that our methodology is effective in mitigating challenges associated with scale variation, lighting conditions, and low image quality in panoramic data. It achieves high tracking ratings by displaying good visual effects, adapting well to small targets, target occlusion, and multi-target cross-motion, and retaining high tracking

accuracy while preserving real-time tracking performance.

REFERENCES

[1] Pace, Paul W., and John Sutherland. "Detection, recognition, identification, and tracking of military vehicles using biomimetic intelligence," Automatic Target Recognition XI. Vol. 4379. SPIE, 2001.

[2] Walker, Sean, et al. "Systems and methods for localizing, tracking and/or controlling medical instruments," U.S. Patent No. 11,504,187. 22 Nov. 2022.

[3] Onate, Johnny Mauricio Barreno, Darío José Mendoza Chipantasi, and Nancy del Rocio Velasco Erazo. "Tracking objects using artificial neural networks and wireless connection for robotics," Journal of Telecommunication, Electronic and Computer Engineering (JTEC) 9.1-3 (2017): 161-164.

[4] Brown, Matthew, et al. "Safe driving envelopes for path tracking in autonomous vehicles," Control Engineering Practice 61 (2017): 307-316.

[5] Bolme, David S., et al. "Visual object tracking using adaptive correlation filters," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010.

[6] Tao, Ran, Efstratios Gavves, and Arnold WM Smeulders. "Siamese instance search for tracking," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[7] Bertinetto, Luca, et al. "Fully-convolutional siamese networks for object tracking," Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part II 14. Springer International Publishing, 2016.

[8] Li, Bo, et al. "High performance visual tracking with siamese region proposal network," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.

[9] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks," Advances in Neural Information Processing Systems 28 (2015).

[10] Wang, Qiang, et al. "Fast online object tracking and segmentation: A unifying approach," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[11] Zhang, Zhipeng, and Houwen Peng. "Deeper and wider siamese networks for real-time visual tracking," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[12] Li, Bo, et al. "Siamrpn++: Evolution of siamese visual tracking with very deep networks," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

[13] Ojala, Timo, Matti Pietikainen, and David Harwood. "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," Proceedings of 12th International Conference on Pattern Recognition. Vol. 1. IEEE, 1994

[14] Ahonen, Timo, Abdenour Hadid, and Matti Pietikainen. "Face description with local binary patterns: Application to face recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence 28.12 (2006): 2037-2041.

[15] Yu, Yuechen, et al. "Deformable siamese attention networks for visual object tracking," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.

[16] Woo, Sanghyun, et al. "Cbam: Convolutional block attention module," Proceedings of the European Conference on Computer Vision (ECCV). 2018.

[17] Hou, Qibin, Daquan Zhou, and Jiashi Feng. "Coordinate attention for efficient mobile network design," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[18] Huang, Lianghua, Xin Zhao, and Kaiqi Huang. "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," IEEE Transactions on Pattern Analysis and Machine Intelligence 43.5 (2019): 1562-1577.

[19] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge," International Journal of Computer Vision 115 (2015): 211-252.

[20] Arun, K. S. "Transactions on pattern analysis and machine intelligence," IEEE, Vol. PAMI-9 5 (1987): 698-770.

[21] Kristan, Matej, et al. "The sixth visual object tracking vot2018 challenge results," Proceedings of the European Conference on Computer Vision (ECCV) Workshops. 2018.

[22] Li, Yang, and Jianke Zhu. "A scale adaptive kernel correlation filter tracker with feature integration," Computer Vision-ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II 13. Springer International Publishing, 2015.

[23] Danelljan, Martin, et al. "Learning spatially regularized correlation filters for visual tracking," Proceedings of the IEEE International Conference on Computer Vision. 2015.

[24] Danelljan, Martin, et al. "Accurate scale estimation for robust visual tracking," British Machine Vision Conference, Nottingham, September 1-5, 2014. Bmva Press, 2014.

[25] Senna, Pedro, Isabela Neves Drummond, and Guilherme Sousa Bastos. "Real-time ensemble-based tracker with kalman filter," 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI). IEEE, 2017.

[26] Pandey, Ashutosh, and DeLiang Wang. "TCNN: Temporal convolutional neural network for real-time speech enhancement in the time domain," ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.