Hindawi
Complexity
Volume 2021, Article ID 6690237, 16 pages
https://doi.org/10.1155/2021/6690237

*Research Article*

# Improved Hierarchical Convolutional Features for Robust Visual Object Tracking

**Jinping Sun** [1,2]

[1]*School of Information Engineering (School of Big Data), Xuzhou University of Technology, Xuzhou 221008, China*
[2]*School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221008, China*

Correspondence should be addressed to Jinping Sun; sjp@xzit.edu.cn

Received 19 November 2020; Revised 17 December 2020; Accepted 4 January 2021; Published 25 January 2021

Academic Editor: Heng Liu



The target and background will change continuously in the long-term tracking process, which brings great challenges to the accurate prediction of targets. The correlation filter algorithm based on manual features is difficult to meet the actual needs due to its limited feature representation ability. Thus, to improve the tracking performance and robustness, an improved hierarchical convolutional features model is proposed into a correlation filter framework for visual object tracking. First, the objective function is designed by lasso regression modeling, and a sparse, time-series low-rank filter is learned to increase the interpretability of the model. Second, the features of the last layer and the second pool layer of the convolutional neural network are extracted to realize the target position prediction from coarse to fine. In addition, using the filters learned from the first frame and the current frame to calculate the response maps, respectively, the target position is obtained by finding the maximum response value in the response map. The filter model is updated only when these two maximum responses meet the threshold condition. The proposed tracker is evaluated by simulation analysis on TC-128/OTB2015 benchmarks including more than 100 video sequences. Extensive experiments demonstrate that the proposed tracker achieves competitive performance against state-of-the-art trackers. The distance precision rate and overlap success rate of the proposed algorithm on OTB2015 are 0.829 and 0.695, respectively. The proposed algorithm effectively solves the long-term object tracking problem in complex scenes.

## 1. Introduction

Visual object tracking [1–4] is one of the most fundamental and challenging research problems in the computer vision area, which combines advanced technologies in several fields such as image processing, pattern recognition, and computer applications. The essence of video moving target tracking is to analyze and research the captured image sequence through image processing technology. First, the features such as the overall or partial edge, texture, shape, contrast, and brightness information of the specific target are extracted and analyzed. After the specific target is detected in the initial image frame of the video sequence, the real-time position of the specific target is dynamically estimated based on the extracted target features in the subsequent frames. Then, the corresponding relationship between the target position in the front and rear frames is established to obtain the motion trajectory of the target. Despite having achieved enormous improvements, visual object tracking remains more challenging to handle critical situations perfectly.

Among the mainstream tracking algorithms, the tracking algorithm based on the discriminative correlation filter (DCF) [5–7] framework has strong advantages and has been rapidly applied and developed. Bolme et al. [8] first adopted the correlation filter framework, which used the minimum output sum of square error (MOSSE) algorithm, and the tracking speed was greatly improved. However, the tracking accuracy of the MOSSE tracker could not meet the actual demand. To improve the tracking accuracy, Henriques proposed the circulant structure of tracking-by-detection with kernels (CSK) algorithm [9, 10], which used the diagonalization of the circulant matrix in the calculation process to simplify the calculation of nuclear regression, so the target tracking speed was greatly improved, and its

tracking accuracy was also higher. However, when the target scale continues to increase, the convolution calculation for extracting target features and training filters would increase, which would lead to a decrease in target tracking speed. The kernel correlation filter algorithm (KCF) [11] was a further improvement of the CSK algorithm which used the histogram of oriented gradients (HOGs) to track the target and improved the accuracy of tracking. The HOG features were extracted to detect the object, improving the accuracy of tracking. Galoogahi et al. [12] put forward a background-aware correlation filter (BACF) based on HOG features, which efficiently modulates the variety of an object in both foreground and background. Liu et al. [13] explored a patch-based tracking method with multi-CF models. The combination of multiple parts could effectively adjust the effects of noise. In 2014, Danelljan et al. proposed a DSST algorithm that used HOG features to construct a scale pyramid for target scale estimation based on MOSSE [14]. However, when the target scale continued to increase, the convolution calculation in training would increase, which would lead to a decrease in tracking speed. In [15, 16], the Kalman filter algorithm was used to predict the state of the target, determined whether the target was occluded, and marked it to predict the target that was still occluded later. In the long-term target tracking process, due to the changeable tracking environment, the target may have different deformations, severe occlusion, and other problems, which may cause tracking failure. How to quickly restore the target tracking function is the key to achieving long-term target tracking. Zhang et al. [17] established the descriptors for rotation and scale normalization and fused color and texture features to perform optimal similarity matching on the descriptors in the candidate of the front and rear frames for target tracking. To pay more attention emphasis on the target sample than on the background samples, Yuan et al. [18] designed a target-focusing convolutional regression model for the visual object tracking task. The target-focusing loss function could effectively balance the proportion of positive and negative samples and prevent overfitting the appearance model to the background samples. Ma et al. [19] trained an online random fern classifier to redetect objects in case of tracking failure. To deal with the shortcomings of one single feature to represent the target, some tracking methods based on multiple feature fusion were designed [20–23], which could improve the robustness of the algorithm to a certain extent.

In recent years, with the rapid development of deep learning in the field of machine vision, its accuracy has surpassed traditional image processing methods in face recognition and image detection. Unlike traditional methods, deep learning does not require the manual design of features but simulates the human visual perception system and abstract expressions based on the characteristics of the original image. Ma et al. [24] improved tracking accuracy and robustness by pretraining deep CNNs, extracting the last three layers of convolutional features, and learning adaptive correlation filters. Qi et al. [25] focused on a hierarchical CNN-based tracking framework (HDT), which took full advantage of different features and used an adaptive Hedge way to hedge these trackers into a stronger one. Valmadre et al. [26] decoded the DCF learner as a differentiable CNN layer and tracking target in an end-to-end way. Although those methods achieved some success, all of them are either limited by a larger computational cost or produce an unsatisfactory tracking performance.

To solve the abovementioned problems, further research on target tracking technology is necessary to improve the tracking efficiency and effect. In this paper, we mainly focus on the problem of long-term tracking in a complex environment, especially when the tracking object is occluded, illumination variation, deformation (DEF), and background clutter. To improve the algorithm's semantic description ability and precise positioning ability, the features of different layers of the convolutional neural network are merged to improve the accuracy of tracking. The objective function is modeled by lasso regression, and the sparse filter model is learned to improve the interpretability of the model. In the process of modeling, the low-rank constraint between different video frames is added to improve the temporal correlation of the filter to solve the problem of overfitting and unstable performance. The multilayer convolution features of candidate regions are extracted, and the response map of each layer is calculated by using the learned filter model. In this work, the coarse-to-fine localization strategy is used, using high-level features for rough positioning and low-level features for precise positioning. By finding the maximum response value in the response map, we can get central of the target. To solve the problem of target template drift in the process of target tracking, two template updating strategies are introduced to update the target template only when the two maximum responses $f_{\max}$ meet the threshold.

The main contributions of this work are summarized as follows:

(i) Through the lasso regression modeling of the objective function and the low-rank constraint between different frame filters, the interpretability and performance stability of the model are improved.

(ii) A coarse-to-fine target localization strategy is proposed by making full use of the rich semantic information in the high level and accurate position information in the low level of the deep convolution feature. The high-level feature is used to predict the target position for coarse-grained location, and then the location result is applied to the fine-grained location of low-level features to obtain the accurate target location.

(iii) A robust template updating method is designed: the filter $\omega_1$ learned from the first frame and the filter $\omega_t$ learned from the $t$-th frame are used to calculate the maximum response $f^1_{\max}$ and $f^{t+1}_{\max}$, respectively. When the two maximum response values meet the

preset threshold, the template is updated, which can solve the tracking failure in the following frame caused by incorrect template update.

The remainder of this study is organized as follows. Section 2 describes the extraction methods of different features, establishes a target feature model via the correlation filter framework, which includes the fundamental introduction about the KCF tracker, sparse and low-rank filter modeling, a coarse-to-fine target prediction model, and template update strategy. Section 3 verifies the effectiveness and robustness of the algorithm through experiments in two aspects, namely, quantitative and qualitative analyses, and describes the comparison with some correlative and representative trackers. Section 4 summarizes the brief conclusions about this work.

## 2. Methodology

Deep convolution features can effectively solve the tracking problem of severe deformation by extracting rich semantic information. The Conv5-4 convolutional features are selected to locate the target roughly, and the Block2-pool layer features are used to achieve accurate target location (see Section 2.1 for details). The traditional KCF algorithm uses the $L_2$ norm to design the objective function, which makes the learned model poor in interpretability. By forcing the sparse filter model when minimizing the objective function, the robustness of the algorithm can be enhanced. In this paper, we use the lasso regression to model the objective function and learn a sparse filter $\omega_t$. To solve the problem of overfitting and unstable performance, low-rank constraints between different video frames are added in the lasso regression modeling process of the objective function to improve the temporal correlation of the filter and enhance the robustness and stability of the algorithm (see Section 2.2 for details). A coarse-to-fine target localization strategy is designed (see Section 2.3 for details). Firstly, the position of the maximum response value $f_{\max}$ of the last layer is predicted which is used as a regular term of other layers to carry out the iteration layer by layer, and the response results of other layers are calculated. The position of the maximum response value is the predicted target position. A robust template updating method is designed: the filter $\omega_1$ learned from the first frame and the filter $\omega_t$ learned from the $t$-th frame are used to calculate the maximum response $f^1_{\max}$ and $f^{t+1}_{\max}$, respectively. When the two maximum response values meet the preset threshold $T_0$, the template is updated, which solves the problem that the template is not updated incorrectly and the failures of the subsequent tracking. The proposed algorithm model is shown in Figure 1.

*2.1. Feature Representation.* We use the hierarchical convolution features from pretrained VGGNet-19 [27] to encode object appearance for their expressive ability to solve the target tracking problem in complex scenes such as target deformation. The VGGNet-19 network has 19 layers, including 16 convolution layers and 3 full connection layers. VGGNet-19 has five convolution modules, Block1–Block5, and each module has a pooling layer. To make the feature graph have strong expression ability, the Relu function expressed as $\mathrm{Re}LU(x) = \max(0, x)$ is used to perform nonlinear processing operations after each convolution layer. Features of different convolutional layers have different expressive capabilities. The shallower the number of layers, the more detailed information contained in the feature map, but background clutter will be generated; the deeper the number of layers, the more semantic information contained in the feature map, but including the less detailed information. The pooling operation of the neural network reduces the spatial resolution of the image, improves the respective field, and makes the high-level features have scale and rotation invariance. To improve the ability of semantic description and precise positioning, some advanced tracking algorithms merge convolutional neural networks and manual features to improve the accuracy of tracking. However, manual features often contain a lot of background noise, which brings challenges to tracking and affects tracking performance. Therefore, it is considered to extract the complete edge information from the convolution neural network to locate the target accurately. The deeper the convolution neural network is, the more obvious the background suppression is. The visualization results of the five convolution modules in VGGNet-19 are shown in Figure 2. Conv1-2 is not selected because it is too close to the input layer (big noisy) and its receptive field is small. Compared with Conv2-2, the Block2-pool layer retains accurate location information while reducing spatial resolution. Therefore, the Block2-pool layer of the Block2 is used to extract features to achieve an accurate target location. As can be seen from Figure 2, although the output of the Conv3-4 convolution layer has a lot of position information, the edge information of the target is incomplete, which is easy to cause the target detection failure. Both Conv5-4 and Block5-pool have rich semantic information, but the feature resolution of the Block5-pool layer is half of that of the Conv5-4 layer. In the complex scene with background clutters, the rough position of the Block5-pool layer within the yellow bounding box is incorrect. In the contrast, the Conv5-4 convolutional features within the yellow bounding box are discriminative from background areas despite the dramatic background changes. The property of the Conv5-4 convolutional layer is suitable to deal with significant appearance changes and accurately locate the target at coarse-grained level. Therefore, considering the efficiency and complexity, the Block2-pool layer with strong spatial resolution and the Conv5-4 layer with strong semantics are selected to describe the appearance of the target.

The size of the feature map in different network layers is different for the pool operation applied in the convolutional neural networks. The deeper the layer is, the smaller the size of the feature map is. For example, the Block5-pool layer feature map size is $7 \times 7$, which is $(1/32)$ of the output image size $224 \times 224$. However, low spatial resolution is not enough to accurately locate the target. In this paper, bilinear interpolation is used to map each feature to a larger size to alleviate this problem. Let $f$ is the feature map before

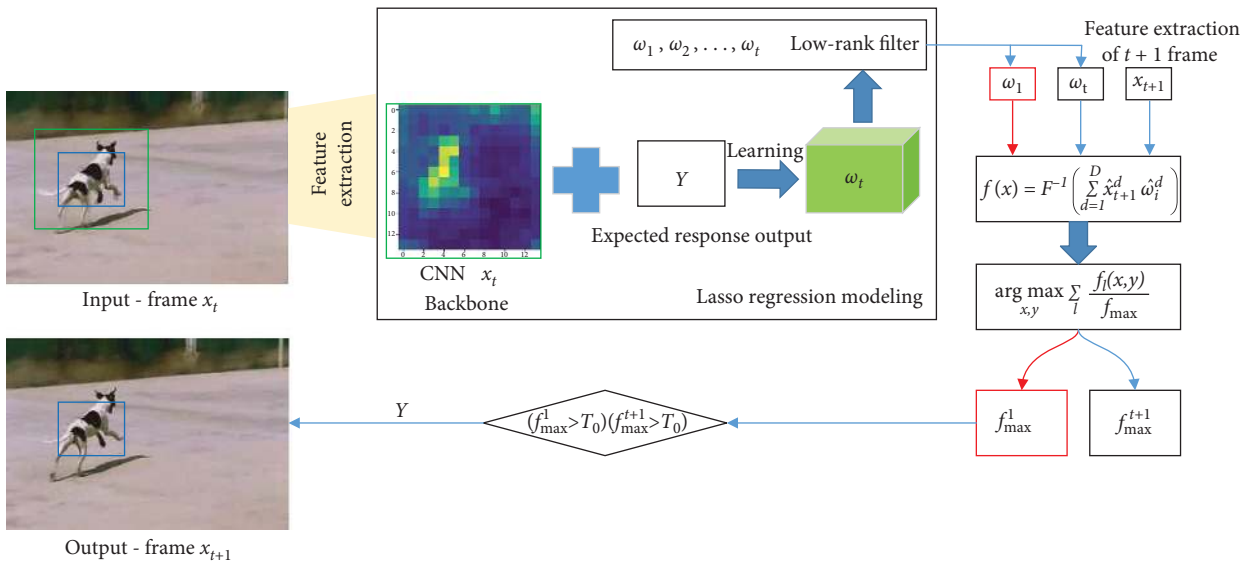



FIGURE 1: Algorithm model. In frame $t$, the blue box presents the prediction result of the previous frame and the green box presents the searching area with 1.5 padding. The convolution features are extracted from the original image within the green box. In frame $t+1$, features for the prediction are extracted around the location in the previous image. The response of the candidate region is calculated by the trained filter $\omega_i$, and the position with the maximum response value is the predicted target position.

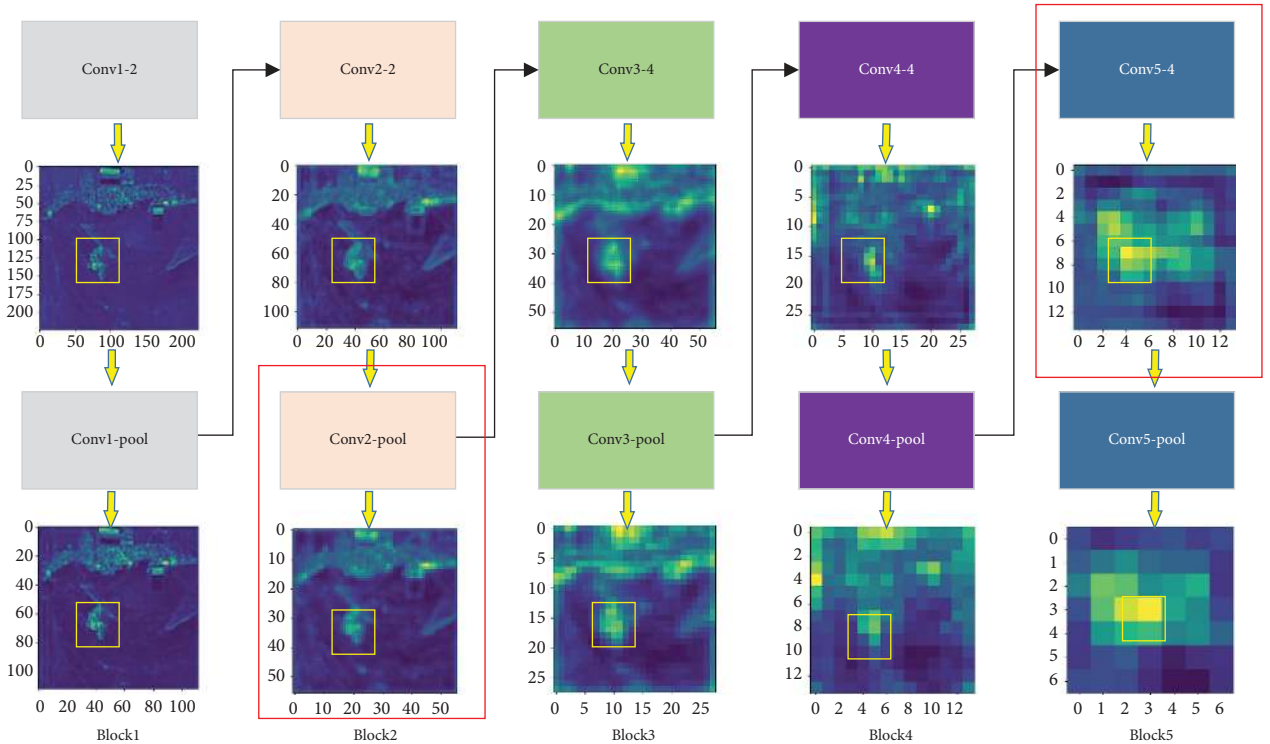



FIGURE 2: Visualization of convolutional layers on the challenging MotorRolling sequence. The yellow bounding boxes indicate the tracking results by our method. The red bounding boxes indicate the feature layer selected by our algorithm. Although the target is seriously deformed, the target position can still be identified by using the output features of the Conv5-4 layer. Compared with the features of the Conv5-pool layer, the resolution is too low to locate the accurate target position.

interpolation and $X$ is the upsampled feature map. Then, the eigenvector of the $i$-th position is expressed as follows:

$$X_i = \sum_j \beta_{i,j} f_j. \tag{1}$$

Among them, the interpolation coefficient is $\beta_{i,j}$, and its value depends on the eigenvector in the $(i, j)$ domain. Convolution feature is a multichannel feature, but not every channel feature is effective, and it may contribute less to target tracking. If such features are used for tracking, it may bring tracking uncertainty in the prediction phase. The feature redundancy can be reduced by the spatial feature selection and cross-channel method [28] to improve the tracking performance. Considering the low resolution of each layer of the deep convolution network, the application of spatial features to the deep convolution network can only achieve limited capacity improvement. A large number of redundant channels can be reduced by adding the channel feature regularization term into the objective function of the filter. To simplify the design of the objective function of the filter, redundant channels are reduced by calculating channel variance. For example, in the VGGNet-19 model, the Conv2-pool layer contains 256 channels, and the Conv5-4 layer contains 512 channels. Each layer selects the convolution features of the first 128 or 256 channels according to the variance size to remove the redundant feature channels and improve the accuracy of the tracking algorithm. The variance of each channel is calculated as follows:

$$\widehat{\sigma}^2 = \frac{1}{WH} \sum_{i=1}^{W} \sum_{j=1}^{H} \left(X_{i,j} - X_{\text{mean}}\right)^2, \tag{2}$$

where $W$ and $H$ represent the width and height of the feature map, respectively. $X_{i,j}$ represents the eigenvalue of the midpoint $(i, j)$ of a channel, and $X_{\text{mean}}$ represents the average characteristic of a channel.

*2.2. Discriminative Correlation Filter.* The typical correlation tracker can learn a discriminative classifier and complete the estimation of the target object by searching the maximum value of the relevant response maps. By exploiting the shifted samples, correlation filters can be efficiently trained with a substantially large number of training samples using the fast Fourier transform (FFT). The task of target tracking is to predict the position of the target in the subsequent frame after the target position in the initial frame is given. Figure 3 shows the result of cyclic shifting different pixels in the $X$-axis and $Y$-axis directions of the target sample.

*2.2.1. Single-Channel Ridge Regression Modeling.* The extracted feature map is $x_t \in R^{W \times H}$, which is a features tensor extracted from the $t$-th frame. $W$ and $H$ indicate the width and height of feature channels, respectively. We consider all the circularly shifted results of the feature $x_t$ along the $W$ and $H$ dimensions as training samples. Each shifted sample $x_{t,ij}$, $(i, j) \in \{0, 1, 2, \ldots, W-1\} \times \{0, 1, 2, \ldots, H-1\}$ has a Gaussian function $y_{ij}$ expressed as follows:

$$Y_{ij} = e^{-\left((i-(W/2))^2 + (j-(H/2))^2 / 2\sigma^2\right)}, \tag{3}$$

where $\sigma$ is the core bandwidth. The center position has the highest score, which is $Y_{(W/2)(H/2))} = 1$. When the position $(i, j)$ is gradually away from the target center, the score $y_{ij}$ decays rapidly from one to zero. The filter of the $t$-th frame named $\omega_t \in R^{W \times H}$ is learned using a pair of training samples. To obtain $\omega_t$, DCF formulates the objective as a regularised least square problem:

$$\omega_t = \arg\min_{\omega} \sum_{ij} \left\| x_t \omega_t - y_{ij} \right\|_2^2 + \lambda \left\| \omega_t \right\|_2^2, \tag{4}$$

where $\lambda$ is the regularization term to prevent overfitting. To simplify the description, the subscript $t$ is omitted, and the closed solution is obtained by deriving equation (4) and setting it to zero:

$$\omega = \left(X^H X + \lambda I\right)^{-1} X^T Y. \tag{5}$$

Among them, $X = [x_1, x_2, \ldots, x_n]^T$ and each row represents a vector. $Y$ is a column vector, and each element represents the expected output $y_{ij}$, $(i, j) \in \{0, 1, \ldots, W\} \times \{0, 1, \ldots, H\}$. $X^H$ represents the complex conjugate transpose matrix, that is $X^H = (X^*)^T$. $I$ is an identity matrix with the same size, and all elements of $I$ are 1. The circulant matrix can be diagonalized in the Fourier domain, and the result in the Fourier domain can be obtained by using this characteristic:

$$\widehat{\omega} = \frac{\widehat{x} \odot \widehat{y}^*}{\widehat{x}^* \odot \widehat{x} + \lambda I}. \tag{6}$$

The addition and division in (6) are carried out by element, $\odot$ means multiply by the element, is the corresponding Fourier representation, and $*$ is complex conjugate.

*2.2.2. Single-Channel Lasso Regression Modeling.* The $L_2$ norm is used to achieve the design of the objective function in formula (4) to balance the deviation problem in the evaluation process. The $L_2$ norm is the square root calculated from the sum of the squares of the elements in the vector $\omega$. To minimize the rule item $\|\omega\|^2$, each element made is very small, but not equal to 0. When each weight coefficient of $\omega$is not equal to 0, that is, all elements are activated, such a filter model is not sparse. In other words, to enhance the generalization ability of the model, the norm $L_2$ sacrifices the interpretability of the model. Especially when dealing with multichannel deep neural network features, most of the features in the training sample $x_t$ of the $\{x_t, Y\}$ are not closely related to the final output $Y$. By forcing a sparse model $\omega$ when minimizing the objective function, the interference to correct prediction can be reduced. To solve this problem, we use the lasso regression method to model the objective function. The sparse regularization operator $L_1$ can remove these uninformed features through learning, that is, reset the weights corresponding to these features to 0. A sparse filter is learned by modeling with lasso regression, which is represented as follows:


Figure 3: Cyclic shift of the target sample. Taking the target image as the base image, the method of cyclic offset can get some approximate negative samples as training samples.

$$\omega = \arg\min_{\omega} \|x\omega - y\|_2^2 + \lambda\|\omega\|_1. \tag{7}$$

*2.2.3. Low-Rank Constraint of the Filter.* In the long-term target tracking process, the appearance and background of the tracking target will change dynamically. The filters learned from the lasso regression model may have problems of overfitting and unstable performance. To solve this problem, low-rank constraints between different video frames are added in the lasso regression modeling of the objective function to improve the temporal correlation of the filter and enhance the robustness and stability of the algorithm. The time-series low-rank smoothing term is defined as follows:

$$\text{rank}(\omega_t) - \text{rank}(\omega_{t-1}). \tag{8}$$

Each column of $\omega_t = [\text{vec}(\omega_1), \ldots, \text{vec}(\omega_t)] \in R^{W\times H\times D}$ is a vectorized filter $\omega_i$. In the process of tracking, the rank $\text{rank}(\omega_t)$ of the $\omega_t$ calculated in real-time will affect the efficiency of the algorithm. By calculating the average value and adding a new increment each time, the difference between the ranks of two adjacent filters can be solved. Therefore, formula (8) is calculated in the following equivalent form [20]:

$$\begin{cases} d(\omega_t - \omega_{t-1}^{\text{mean}}), \\ \omega_{t-1}^{\text{mean}} = \sum_{i=1}^{t-1} \dfrac{\omega_i}{(t-1)}, \end{cases} \tag{9}$$

where $\omega_{t-1}^{\text{mean}}$ is the mean value of the filter learned before the $t-1$ frame and $d$ is a distance measurement function. Therefore, the regularization term represented by the $L_2$ norm in formula (7) achieves low-rank time series, which can be approximately described as follows:

$$\omega_t = \arg\min_{\omega} \left\|x\omega_t - y\right\|_2^2 + \lambda_1\left\|\omega_t\right\|_1 + \lambda_2\left\|\omega_t - \omega_{t-1}^{\text{mean}}\right\|_2^2. \tag{10}$$

It can be seen from formula (10) that the $\omega_{t-1}^{\text{mean}}$ calculation is performed in an incremental manner, and only one parameter $\omega_{t-1}^{\text{mean}}$ is needed to realize the time-series low-rank smoothing in the calculation process.

*2.2.4. Multichannel Low-Rank Modeling.* The Block2-pool layer with strong spatial resolution and the Conv5-4 layer with strong semantics are selected to describe the characteristics of the target. The extracted multichannel feature map of the $t$-th frame is $x_t \in R^{W\times H\times D}$, which is a tensor consisting of $d$-channel features extracted from the $t$-th frame. Then, $Y \in R^{W\times H}$ is the corresponding ideal Gaussian waveform of the response. The multichannel filters $\omega_t \in R^{W\times H\times D}$ of $t$-th frame are learned from a pair of training samples $\{x_t, Y\}$. The process of solving the filter is to minimize the objective function. The multichannel objective function of low-rank modeling is defined as follows:

$$\tilde{\omega}_t = \arg\min_{\omega_t} \left\|\sum_{d=1}^{D} \omega_t^d * x_t^d - Y\right\|_F^2 + \lambda_1 \sum_{d=1}^{D} \left\|\omega_t^d\right\|_{1,1} + \lambda_2 \sum_{d=1}^{D} \left\|\omega_t^d - w_{t-1}^d\right\|_F^2. \tag{11}$$

where $*$ is the circular convolution operator [9], $\omega_t^d \in R^{W\times H}$ is the corresponding discriminative filter of the $d$-th channel, $x_t^d \in R^{W\times H}$ represents the $d$-th channel feature, and $w_{t-1}^d$ represents the multichannel form of $\omega_{t-1}^{\text{mean}}$. $\|A\|_F$ is the Frobenius norm of matrix $A$ defined as the square root calculated from the sum of the squares of the elements in the matrix $A$, which represents the Euclidean distance between two matrices, expressed as follows:

$$\|A\|_F = \sqrt{\sum_{i=1}^{W}\sum_{j=1}^{H} \left|a_{i,j}\right|^2}, \tag{12}$$

where $a_{i,j}$ represents the element in the $i$-th row and $j$-th column of matrix $A$. The $L_1$ norm $\|A\|_{1,1}$ of matrix A can be expressed as $\|A\|_{1,1} = \sum_{i=1}^{W}\sum_{j=1}^{H} |a_{i,j}|$.

*2.2.5. Optimization of the Objective Function.* The objective function expressed by formula (11) is convex and can be optimized by the extended Lagrangian method [29]. By introducing relaxation variables $\omega' = \omega$, the Lagrangian function is constructed as follows:

$$\begin{aligned}\ell = &\left\| \sum_{d=1}^{D} \omega_t^d * x_t^d - Y \right\|_F^2 + \lambda_1 \sum_{d=1}^{D} \left\| \omega'^d_t \right\|_{1,1} \\ &+ \lambda_2 \sum_{d=1}^{D} \left\| \omega_t^d - w_{t-1}^d \right\|_F^2 + \frac{\alpha}{2} \sum_{d=1}^{D} \left\| \omega_t^d - \omega'^d_t + \frac{\Pi^k}{\alpha} \right\|_F^2,\end{aligned} \tag{13}$$

where $\alpha$ is the optimization parameter and $\Pi$ is the Lagrangian multiplier with the same dimension size as $x_t$. Iterative optimization of formula (13) is carried out by using alternating direction multiplier to ensure convergence [19], which is specifically expressed as follows:

$$\begin{cases} \omega = \arg\min_{\omega} \ell(\omega, \omega\prime, \Pi, \alpha), \\ \omega' = \arg\min_{\omega'} \ell(\omega, \omega\prime, \Pi, \alpha), \\ \Pi = \arg\min_{\Pi} \ell(\omega, \omega\prime, \Pi, \alpha). \end{cases} \tag{14}$$

(i) Solution of variables $\omega$

Given variables $\omega\prime$, $\Pi$, and $\alpha$, the solution of the variable $\omega$ is obtained by optimizing the corresponding objective function, which is expressed as follows:

$$\begin{aligned}\min &\left\| \sum_{d=1}^{D} \widehat{\omega}_t^d * \widehat{x}_t^d - \widehat{Y} \right\|_F^2 + \lambda_2 \sum_{d=1}^{D} \left\| \widehat{\omega}_t^d - \widehat{w}_{t-1}^d \right\|_F^2 \\ &+ \frac{\alpha}{2} \sum_{d=1}^{D} \left\| \widehat{\omega}_t^d - \widehat{\omega}_t'^d + \frac{\widehat{\Pi}^k}{\alpha} \right\|_F^2.\end{aligned} \tag{15}$$

For the sake of simplicity, the corresponding closed solution is obtained after omitting the subscript $t$ as follows [30]:

$$\widehat{\omega}_{i,j} = \left( I - \frac{\widehat{x}_{i,j} \widehat{x}_{i,j}^T}{\lambda_2 + \left(\alpha/2 + \widehat{x}_{i,j} \widehat{x}_{i,j}^T\right)} \right) \frac{\left(\widehat{x}_{i,j} \widehat{y}_{i,j} + \alpha \widehat{\omega}'_{i,j} - \alpha \widehat{\Pi}_{i,j} + \lambda_2 \widehat{w}_{i,j}\right)}{(\lambda_2 + \alpha)}. \tag{16}$$

Among them, the vectors $\omega'_{i,j}$, $\widehat{x}_{i,j}$, and $\widehat{w}_{i,j}$ are, respectively, composed of the elements in the $i$-th row and the $j$-th column in all channels of the vectors $\widehat{\omega}'$, $\widehat{x}$, and $\widehat{w}$.

(ii) Solution of variables $\omega'$

Given variables $\omega$, $\Pi$, and $\alpha$, the solution of the variable $\omega'$ is obtained by optimizing the corresponding objective function, which is expressed as follows:

$$\min \lambda_1 \sum_{d=1}^{D} \left\| \omega'^d_t \right\|_{1,1} + \frac{\alpha}{2} \sum_{d=1}^{D} \left\| \omega_t^d - \omega'^d_t + \frac{\Pi^k}{\alpha} \right\|_F^2. \tag{17}$$

The closed solution can be obtained by shrinking the threshold [20]:

$$\omega'^d_{i,j} = \operatorname{sign}\left( \omega_{i,j}^d + \frac{\Pi_{i,j}^d}{\alpha} \right) \max\left( 0, \left| \omega_{i,j}^d + \frac{\Pi_{i,j}^d}{\alpha} \right| - \frac{\lambda_1}{\alpha} \right), \tag{18}$$

where $\omega_{i,j}^d$ and $\Pi_{i,j}^d$ represent the elements in the $i$-th row and $j$-th column of $\omega$ and $\Pi$ of $d$-channel, respectively. sign is a symbolic function that determines the positive and negative problems of $\omega'^d_{i,j}$.

(iii) Solution of variables $\Pi$

Given variables $\omega$ and $\omega'$, the update mode of variables $\Pi$ and $\alpha$ is as follows:

$$\begin{cases} \Pi = \Pi + \alpha(\omega - \omega\prime), \\ \alpha = \min(\beta\alpha, \alpha_{\max}). \end{cases} \tag{19}$$

$\alpha_{\max}$ is the maximum penalty parameter to prevent singularity, and $\beta$ is the similarity measure between the slack variable $\omega'$ and the original variable $\omega$.

*2.3. Coarse-to-Fine Location Prediction.* The correlation filter $\omega_t$ learned from the $t$-th frame is used in the target estimation of the subsequent $(t+1)$-th frame. The features of the layer $l = \{2, 5\}$ of frame $t+1$ are extracted, and the filtering response in the frequency domain is calculated as follows:

$$f(x) = F^{-1}\left( \sum_{d=1}^{D} \widehat{x}_{t+1}^d \odot \widehat{\omega}_t^d \right), \tag{20}$$

where $\widehat{\ }$ represents the discrete Fourier transform, $\odot$ represents element-wise multiplication, and $F^{-1}$ represents the inverse transform of the fast Fourier. By searching for the position with the maximum response value $f_{\max}$ with size of $W \times H$, the target location of the $l$-th convolution layer can be estimated based on hierarchical prediction. The position is predicted based on the maximum response value $f_{\max}$ of the last layer and iterated layer by layer as the regular term to calculate the response results of the other layers. It is assumed that the response of the position $(x, y)$ in the layer $l$ is expressed as $f_l(x, y)$ and the maximum response position is expressed as $(x_c, y_c)$:

$$(x_c, y_c) = \arg\max_{x,y} f_l(x, y). \tag{21}$$

Given the response $f_l(x, y)$ and maximum response position $(x_c, y_c)$ of layer $l$, the position of layer $l - i$ can be predicted with the following formula:

$$\begin{cases} \arg\max\limits_{x,y} & f_{l-i}(x,y) + \gamma_l f_l(x,y), \\ \text{s.t.} & \sqrt{|x-x_c|^2 + |y-y_c|^2} \le r, \end{cases} \tag{22}$$

where $\gamma_l$ is the regularization term of layer $l$, which is propagated to the response maps of early layers. Specifically, with $(x_c, y_c)$ as the center and $\sqrt{|x-x_c|^2 + |y-y_c|^2}$ as the radius, we can get a circle $C$. Formula (22) indicates that the maximum response location $(x, y)$ is searched in the $r \times r$ neighboring regions centered at $(x_c, y_c)$ on the $(l-i)$-th correlation response map. Finally, using formula (22), the maximum response location is found on the Block2-pool layer as the final target location.

In practice, the tracking results are not sensitive to the parameter $r$ of the neighborhood search regions. The target location can be predicted by computing the weighted average of the response maps from different layers defined as follows:

$$\arg\max_{x,y} \sum_l \gamma_l f_l(x,y). \tag{23}$$

The last layer of the convolutional neural network contains rich semantic information, which is robust to target deformation. Therefore, we hope to assign larger regularization coefficients to deeper layers. It is found that the deeper the layers are, the lower the spatial resolution is, and the lower the maximum response value is. Taking advantage of the fact that the regularization term $\gamma_l$ of each layer is inversely proportional to the maximum response value $f_{\max}$, the regularization term is designed as follows:

$$\gamma_l \propto \frac{1}{f_{\max}}. \tag{24}$$

Combined with formulas (23) and (24), the target position can be located as follows:

$$\arg\max_{x,y} \sum_l \frac{f_l(x,y)}{f_{\max}}. \tag{25}$$

In Figure 4, we compare the weighted maximum response values from four different convolutional layers to locate the target in Dog sequence. By comparing the four filtering response curves in Figure 4, it can be found that the maximum response value of the Conv2-2 layer is the best. The reason is that Conv2-2 has a stronger spatial resolution than the Block2-pool layer, but it also leads to a low frame rate when using the Conv2-2 layer for tracking (see Figure 5 for details). The weighted maximum response values of Block2-pool using formula (25) help to track the target well over the entire sequence.

*2.4. Model Update.* In the long-term target tracking process, background interference may occur, which may cause the template to be updated incorrectly. The response of the filter on the background is greater than the threshold $T_0$, but at this time, the predicted position is wrong, and the newly learned template is also wrong. In this case, if the template is updated, it will cause tracking drift. By calculating the response value of each subsequent frame, it will be found that the maximum response value still meets the threshold condition, but the target is not located. At this time, the template is no longer accurate, and the maximum response position is the background. To solve the problem of target template update in complex scenes such as background interference, target deformation, and occlusion, an accurate template is used for prediction to obtain more accurate location information. For this reason, two template update judgment methods are proposed. The filter $\omega_1$ learned from the first frame and the filter $\omega_t$ learned from the $t$ frame are used to calculate the maximum response $f^1_{\max}$ and $f^{t+1}_{\max}$, respectively. When the two maximum response values meet the preset threshold value $T_0$, the template is updated (see Section 2.4 for details), which can solve the tracking failure in the following frame caused by incorrect template update.

To obtain a better approximation, we update the correlation filter $\omega_t$ in (26) using a moving average with a multichannel filter $\omega_{t-1}$ obtained in the optimization process in Section 2.2:

$$w_t = \gamma\omega_t + (1-\gamma)\omega^{\text{mean}}_{t-1}, \tag{26}$$

where $t$ is the frame index and $\gamma$ is a learning rate. Some existed trackers update object models [13, 16] at each frame without considering whether the detection is accurate or not. However, in the target tracking process, the target is likely to be severely occluded or completely occluded. If the target model continues to be updated in this case, it is equivalent to updating the background as the target, which may easily cause template drift. When there is more than one similar object in the scene, the tracker treats a similar target as the background. Only when the two maximum response values are greater than the preset threshold value $T_0$, the model is updated using equation (26). Otherwise, the model is not updated, and the previous filter will be used for position prediction in the subsequent frames. Therefore, the learned filter is robust to noisy updates that often cause rapid model degradation and obtains a long-term memory of target appearance.

## 3. Result Analysis and Discussion

To evaluate the proposed algorithm objectively and comprehensively, we run the proposed algorithm on two standard benchmark datasets: OTB2015 [31] and TC-128 [32]. OTB2015 is annotated with 11 attributes that cover various challenging factors, including scale variation (SV), illumination variation (IV), occlusion (OCC), motion blur (MB), deformation (DEF), fast motion (FM), out-of-plane rotation (OPR), out-of-view (OV), in-plane rotation (IPR), background clutters (BC), and low resolution (LR). The TC-128 benchmark contains 128 color video sequences with 11 annotated attributes, and it aims at analyzing the impact of color information on tracking. In this paper, the video sequence with more than 1000 frames belongs to long-term tracking. We use the benchmark protocols and the same parameters shown in Table 1 for all the sequences as well as

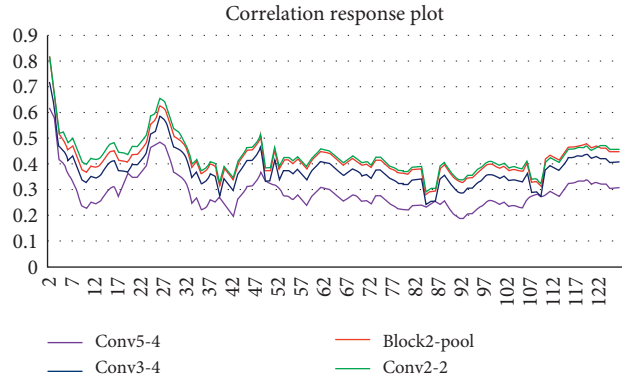



Figure 4: Frame-by-frame maximum response values of different convolutional layers on the Dog sequence. The response value of the last layer is the smallest, and the lower convolution layer has a higher spatial resolution achieving the higher the response value.

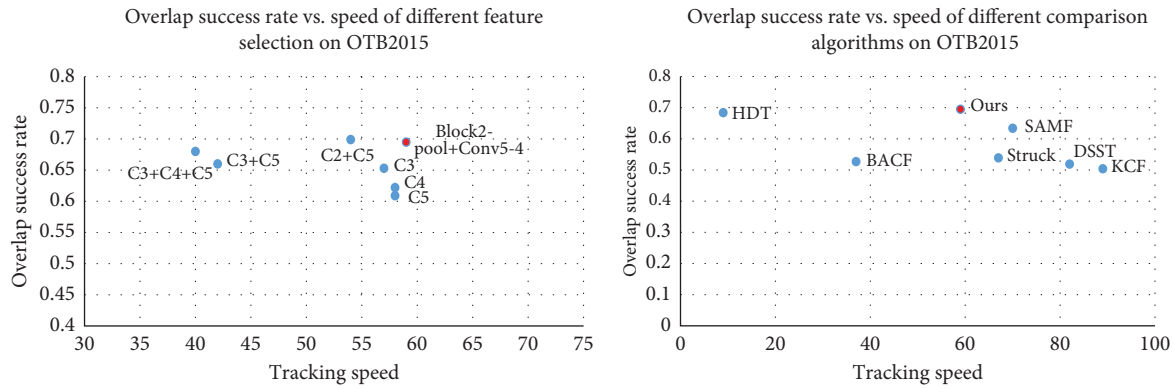



Figure 5: Comparison of OSR and tracking speed on the OTB2015 dataset. The horizontal axes represent the tracking speed, and the vertical axes represent the overlap success rate. (a) Comparison of the efficiency of selecting different features for tracking. (b) Comparison of the efficiency of different algorithms.

Table 1: Parameter settings.

| Parameter | Value |
|---|---|
| Regularization term $\lambda_1$ | $10^{-4}$ |
| Regularization term $\lambda_2$ | 4 |
| Learning rate $\gamma$ | 0.15 |
| $f_{\max}$ threshold $T_0$ | 0.3 |
| Gaussian kernel bandwidth $\sigma$ | 0.5 |

all the sensitivity analysis. For completeness, we also report the results in terms of distance precision rate, overlap success rate, and center location error in comparison with state-of-the-art algorithms: BACF [12], KCF [11], DSST [14], HDT [25], Struck [7], and SAMF [22]. We implement the proposed tracker in MATLAB R2018b on a computer with the configuration of an Intel Core i7-8550U 2.0 GHz CPU, 8 GB RAM, and a GeForce GTX GPU with MatConvNet toolbox.

*3.1. Evaluation Criterion.* To evaluate the tracking performance, the one-pass evaluation (OPE) is used as the evaluation index on the OTB2015 [31] dataset. The OPE strategy has two evaluation criteria, namely, distance precision rate (DPR) and overlap success rate (OSR). The distance precision rate represents the percentage of the center location errors between predicted position and ground-truth with different thresholds. The center position error refers to the Euclidean distance between the estimated position $(x\prime, y\prime)$ obtained by iteration and the true position $(x, y)$, which can be calculated using formula:

$$D = \sqrt{(x - x\prime)^2 + (y - y\prime)^2}. \tag{27}$$

As $D$ decreases, the accuracy and stability of the algorithm increase. Given the predicted bounding box $R_p$ estimated by the tracking algorithm and the ground-truth bounding box $R_g$, the overlap rate ($R$) can be computed as follows:

$$R = \frac{S\left(R_p \cap R_g\right)}{S\left(R_p \cup R_g\right)}, \tag{28}$$

where $\cap$ represents the intersection of the two bounding boxes, $\cup$ represents the union of the two bounding boxes, and $S(\cdot)$ represents the area of the two bounding boxes. As the overlap rate increases, the tracking success rate increases.

Overlap success rate represents the percentage of frames with an overlap rate greater than a given threshold. With different thresholds, a curve can be obtained, and the threshold is set to 0.3.

*3.2. Quantitative Analysis.* The algorithms will be evaluated and analyzed from four aspects on the OTB2015 dataset and two evaluation indicators, DPR and OSR on the TC-128 dataset.

*3.2.1. Experiment on OTB2015 Dataset.* In this section, the experimental results on the OTB2015 are given in terms of the tracking performance, center position error, DPR, and OSR, comparing with BACF [12], KCF [11], DSST [14], HDT [25], Struck [7], and SAMF [22] algorithm:

(1) The tracking performance

Figure 5(a) shows that the comparison of tracking results when different convolutional layers of VGGNet-19 are selected as features. Note that if one convolutional layer contains multiple sublayers, we use features from the last sublayer, e.g., *C*2 indicates the Conv2-2 layer in VGGNet-19. We use different layers (*C*5, *C*4, and *C*3) separately and merge different layers to express the comparison result of the speed and success rate. The VGG-C5-C2 strategy performs better than the VGG-C5-Block2-pool in terms of success rate and worse than the VGG-C5-C2 scheme in terms of speed. Since target tracking is a real-time task and requires relatively high processing speed, the VGG-C5-Block2-pool scheme achieves the most ideal comprehensive effect.

Figure 5(b) shows the success rate and tracking speed between the proposed tracker and other compared trackers on the OTB2015 dataset. It shows that the proposed tracker achieves favorable tracking accuracy with the highest tracking success rate. However, the tracking speed at 59 frames per second (FPS) is slower than other algorithms, such as KCF and DSST, so it still needs to be improved.

(2) Center position error (CPE)

In this section, we analyze the proposed tracker with BACF [12], KCF [11], DSST [14], HDT [25], Struck [7], and SAMF [22] in terms of CPE on OTB2015. CPE is an evaluation index of tracking target accuracy, which is the mean value of the Euclidean distance between the predicted center position and the actual target center position. Figure 6 shows the results of the CPE in sequences of the Singer1 with 351 frames and Car4 with 659 frames. The proposed algorithm performs well against the state-of-the-art trackers in CPE. The proposed algorithm obtains the low CPE value, with the maximum of only 9.69, in both Singer1 and Car4 sequence. The object in the video sequence of Singer1 has large illumination variations from frame 90. The CPE of BACF is much more than 20 pixels and even leads to tracking failures.

(3) Distance precision rate plot and overlap success rate plot

We use the success rate plot and precision plot calculated by the OPE evaluation standard to further analyze the tracking performance of the proposed algorithm and the other six comparison algorithms. The overlap success rate ($R$ defined in formula (28)) represents the size of the tracking success rate, and the range is 0 to 1. Figure 7 shows the comprehensive statistical results between the proposed algorithm and the comparison algorithm on OTB2015. Compared with the highest score of 0.832, the distance precision rate of the proposed algorithm is 0.829, which is in the second rank and is 16.76% higher than the traditional KCF algorithm. The proposed algorithm with 0.695 scores achieves top rank in the average success plots, which is 1.58% higher than that of the second-ranked HDT (0.684). Compared with the KCF tracker, which has a success ranking score of 0.504, the proposed algorithm obtains an improvement of over 27.48%. From the above analysis, it is concluded that the modeling method of lasso regression enhances the interpretability between different channel features and improves the accuracy of the algorithm.

Figure 7 shows the comparison of the average performance of the algorithm in different scenarios under the OPE evaluation index. The effect of the algorithm will be different if the attributes of the data set are different. To comprehensively and accurately analyze the performance of the proposed algorithm in various complex scenarios, Figure 8 shows the comparison of the tracking results of the proposed algorithm and the other compared tracking algorithms in IV, OCC, BC, and SV with different attributes. The proposed tracker achieves almost the best performance among all other compared algorithms, only performs slightly worse in video sequences with occlusion properties. The proposed algorithm has a significant improvement against the second-ranked algorithm under complex scenes with the attributes of illumination variation, occlusion, background clutter, and scale variation.

*3.2.2. Experiment on TC-128.* In this section, the TC-128 [32] dataset is used to validate the performance of the proposed tracker. The comparison with some state-of-the-art trackers, including BACF [12], KCF [11], DSST [14], HDT [25], Struck [7], and SAMF [22], is shown in Table 2. Besides, we use OTB2015 and TC-128 datasets to perform a quantitative comparison of DPR with 20 pixels and OSR with 0.5 pixels shown in Table 2. It shows that the proposed tracker obtains the highest performance with the DPR of 80.5 and the OSR of 66.5 on TC-128. The proposed tracker achieves the DPR of 82.9 and performs slightly worse than HDT with a DPR of 83.2 on OTB2015. Compared with KCF, the proposed tracker achieved significant improvements, which shows the benefits of using the Lasso regression

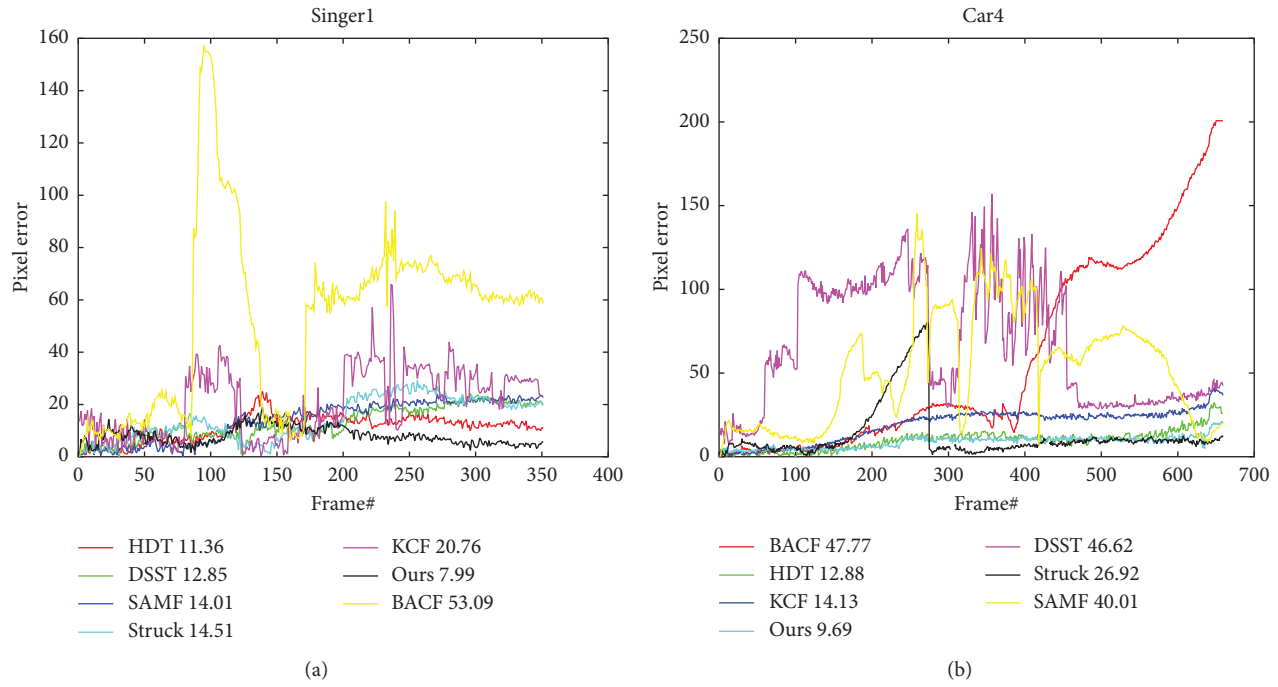



Figure 6: Comparison results of CPE in different test videos. (a) The comparison result of the CPE in Singer1. (b) The comparison result of the CPE in Car4.

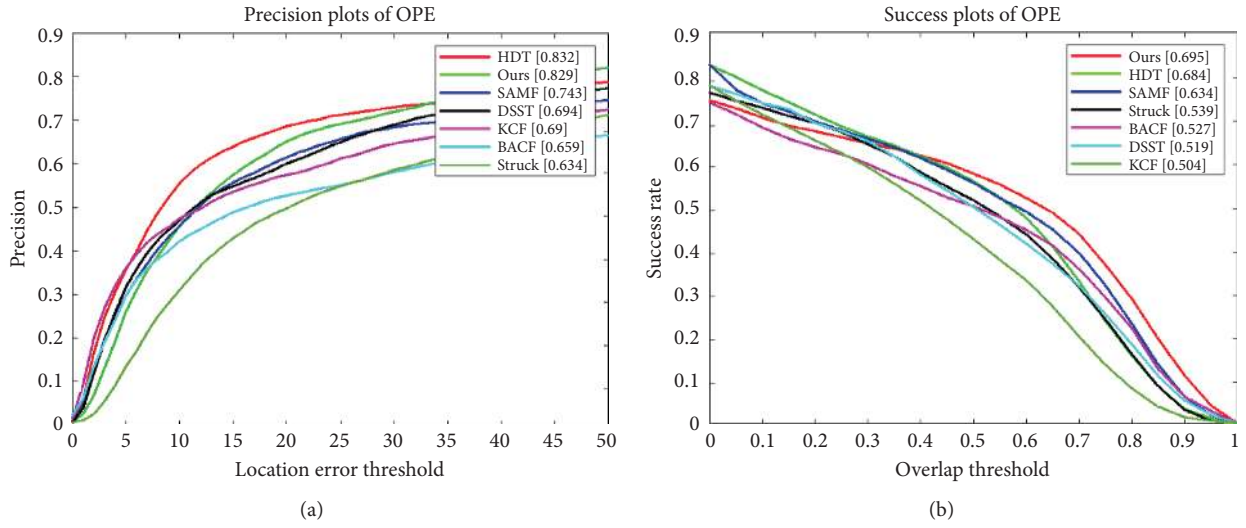



Figure 7: Distance precision and overlap success plots on the 100 benchmark sequences of OTB2015 [31] datasets using OPE. The legend of distance precision contains the threshold scores at 20 pixels, while the legend of overlap success contains area under the curve scores for each tracker.

modeling method and the multichannel feature selection scheme.

*3.3. Qualitative Analysis.* To visually illustrate the tracking accuracy of the proposed algorithm, Figure 9 shows the tracking results of other advanced tracking algorithms in the challenging video sequences Matrix, Dog, Singer1, and Girl2. The comparison algorithms include correlation filter tracker (KCF), multifeature tracker (SAMF), deep learning tracker (HDT), and representative tracker (Struck). The Matrix sequence as shown in Figure 9(a) has attributes of IV, SV, OCC, FM, IPR, OPR, and BC. The Dog sequence as shown in Figure 9(b) has attributes of SV, DEF, and OPR. The Singer1 sequence as shown in Figure 9(c) has attributes of IV, SV, OCC, and OPR. The Girl2 sequence as shown in Figure 9(d) has attributes of SV, OCC, DEF, MB, and OPR. Taking Figure 9(d) as an example, where the object is

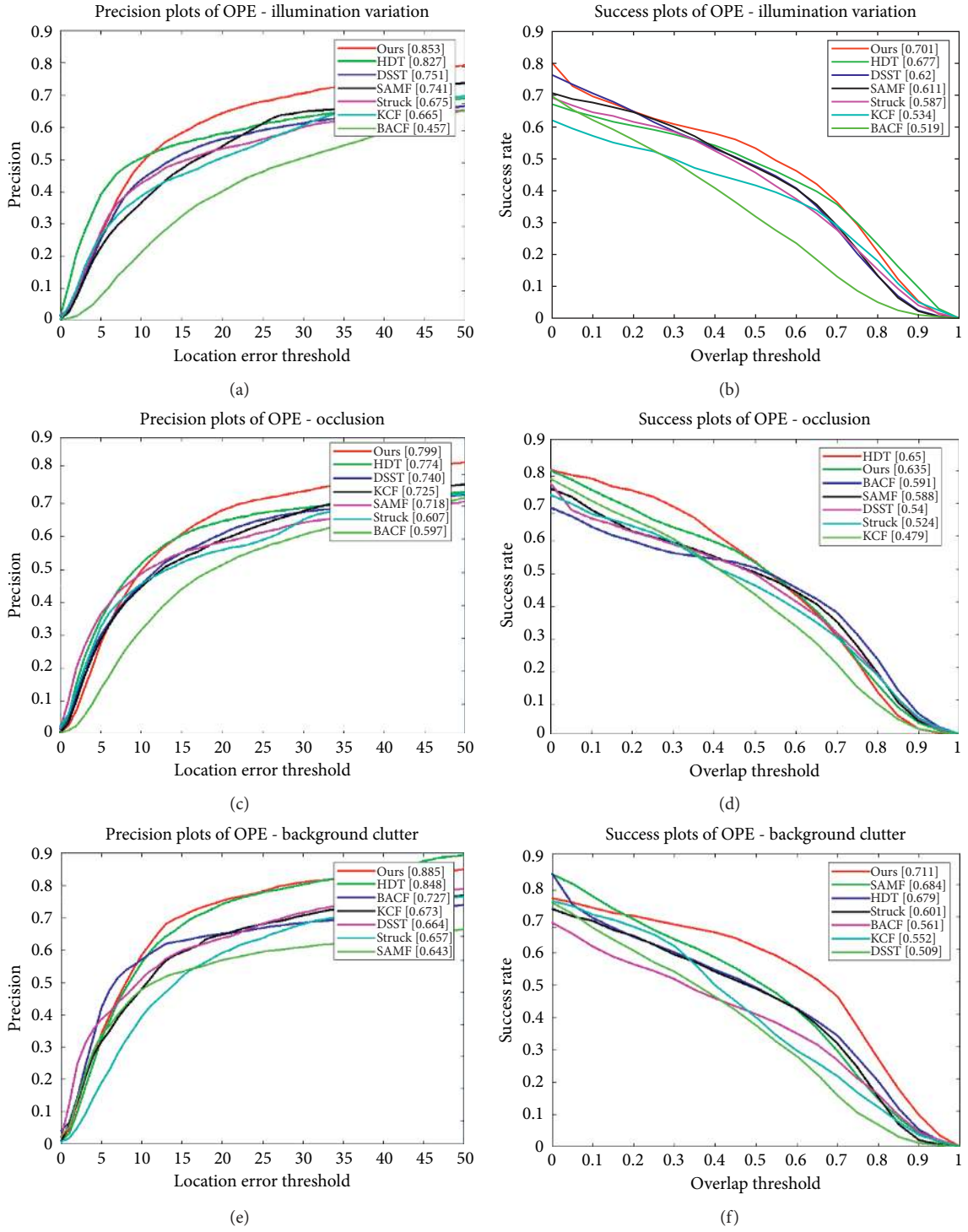
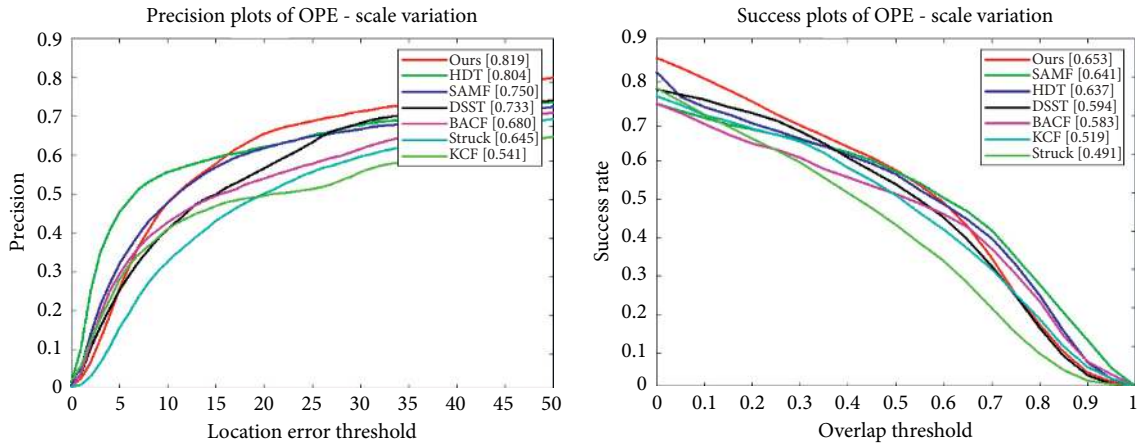



Figure 8: Continued.

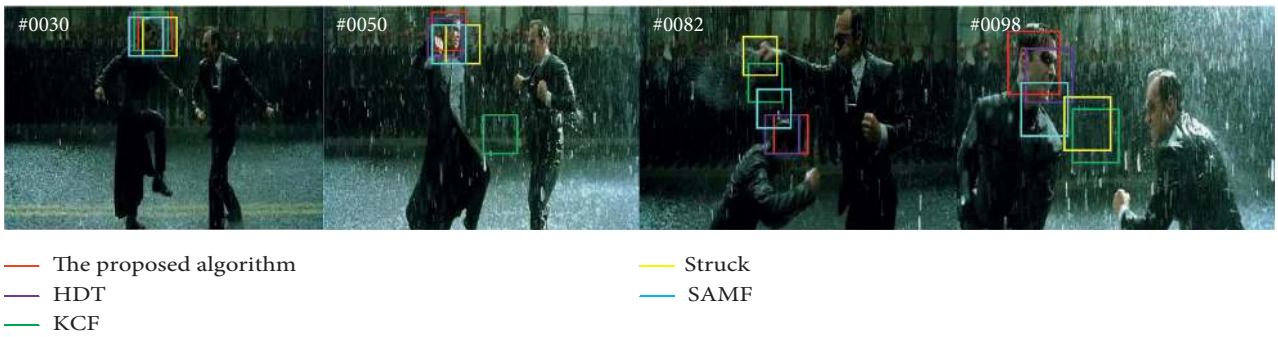



(g) (h)

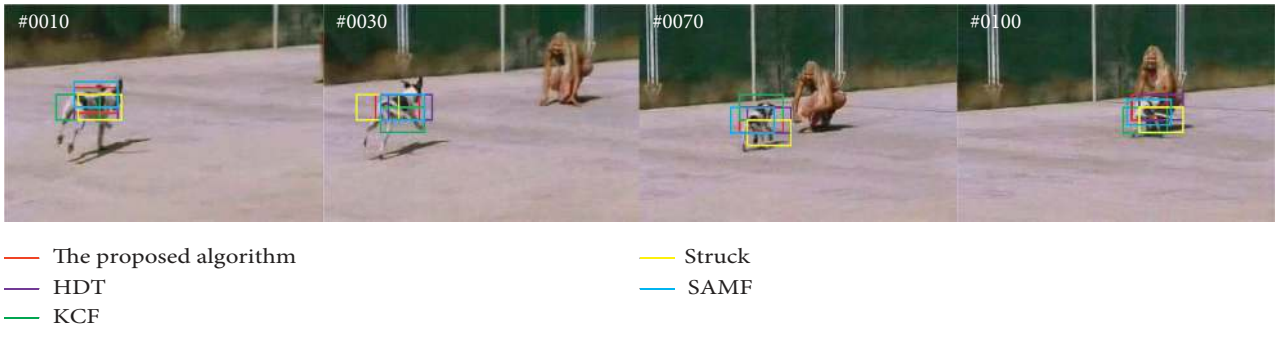
Figure 8: Distance precision and overlap success plots on different attribute scenarios.

Table 2: Comparisons with the state-of-art tracker in terms of DPR and OSR (%), including BACF, KCF, DSST, HDT, Struck, and SAMF on OTB2015 and TC-128 datasets. The first and second best values are highlighted in bold and italics.

| Dataset | Evaluation criterion | Ours | BACF | KCF | DSST | HDT | Struck | SAMF |
|---|---|---|---|---|---|---|---|---|
| OTB2015 | DPR | 82.9 | 65.9 | 69.0 | 69.4 | **83.2** | 63.4 | 74.3 |
| | OSR | **69.5** | 52.7 | 50.4 | 51.9 | *68.4* | 53.9 | 63.4 |
| TC-128 | DPR | **80.5** | 63.7 | 54.9 | 66.4 | *80.1* | 59.7 | 67.3 |
| | OSR | **66.5** | 49.7 | 49.4 | 46.8 | *56.4* | 50.9 | 53.4 |




(a)




(b)

Figure 9: Continued.

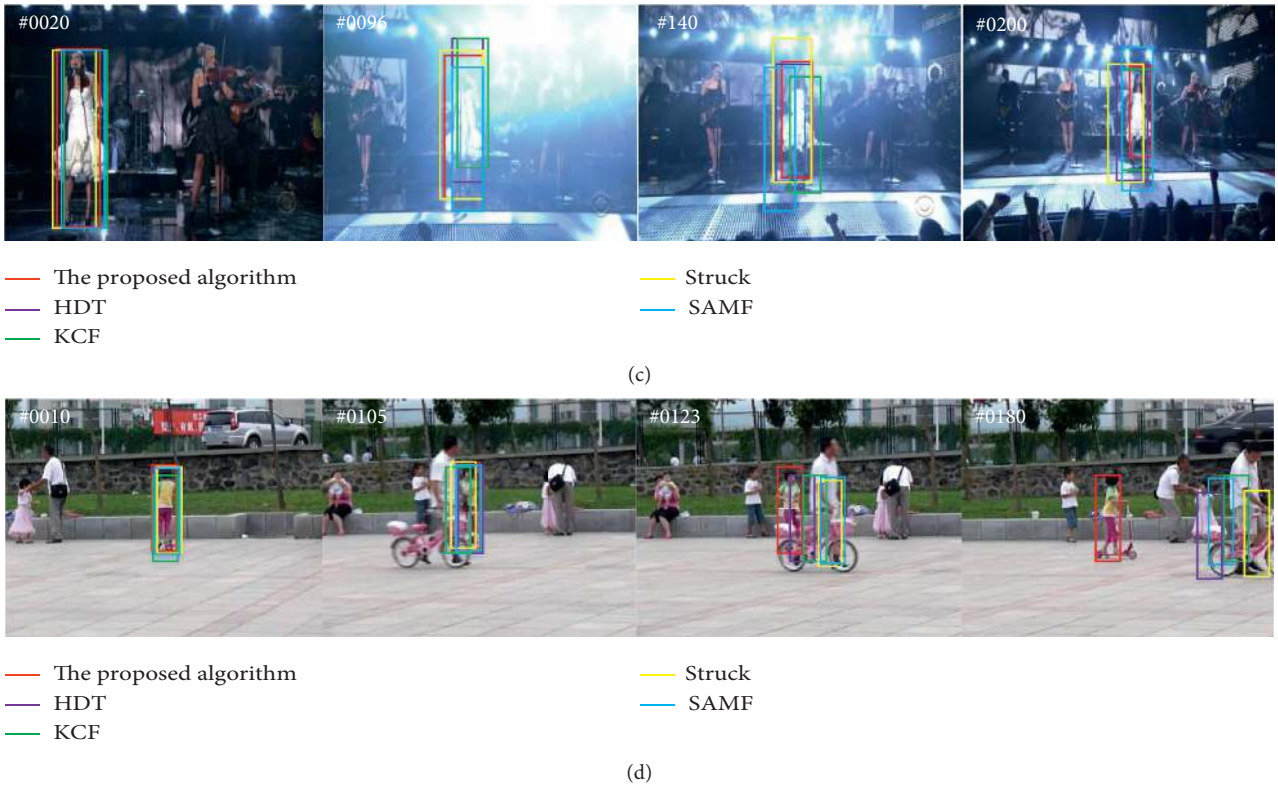



Figure 9: Qualitative comparison between the proposed tracker and other representative trackers (KCF [11], HDT [25], Struck [7], and SAMF [22]) on some visual object tracking sequences. (a) Matrix video sequence. (b) Dog video sequence. (c) Singer1 video sequence. (d) Girl2 video sequence. The proposed model provides consistent results in challenging scenarios, such as occlusions, illumination variation, fast motion, and background clutter.

occluded by other objects, other compared algorithms all fail to locate the object, while the proposed tracker can accurately locate the target due to its redetection function. The HDT performs well in sequences with deformation and fast motion (Matrix and Dog) but fails when the object is occluded (girls). It can be seen from Figure 9(b) that all algorithms have good results in the Dog video sequence, which shows that the algorithms can deal with the problem of slight deformation of the target. However, other algorithms cannot solve the problem of target tracking in complex scenes such as target deformation and background interference. In the Matrix video sequence containing background interference, the proposed algorithm can also effectively remove the background information and establish an effective model to smooth the filter.

## 4. Conclusion

To improve the tracking performance and robustness, an improved hierarchical convolutional features model is proposed into a correlation filter framework for visual object tracking. The objective function is designed through lasso regression modeling to obtain a sparse, time-series low-rank filter, which improves the time-series correlation of the filter and prevents algorithm overfitting and performance degradation. The proposed coarse-grained to fine-grained target location strategy makes full use of the complementary characteristics of different layers in the deep convolutional network and can achieve robust tracking in challenging videos. By extracting the features with rich semantic information in the last layer, coarse-grained positioning is performed, which effectively solves the problem of target deformation. By extracting low-level features with a high spatial resolution for fine-grained positioning, the positioning accuracy and precision can be improved. A robust template updating method is designed: the filter $\omega_1$ learned from the first frame and the filter $\omega_t$ learned from the $t$ frame are used to calculate the maximum response $f^1_{\max}$ and $f^{t+1}_{\max}$, respectively. When the two maximum response values meet the preset threshold value $T_0$, the template is updated, which solves the subsequent tracking failures caused by incorrect template updating. The experiment results with different attributes show the competitive performance of the proposed tracker.

## Data Availability

No data were used to support this study.

## Conflicts of Interest

The author declares that there are no conflicts of interest.

## Acknowledgments

This work was supported by the Ministry of Housing and Urban-Rural Development Science and Technology Planning Project (2016-R2-060), the National Key Research and Development Plan of China (nos. 2017YFC0804400 and 2017YFC0804401), the Major Project of Natural Science Research of the Jiangsu Higher Education Institutions of China (18KJA520012), the Xuzhou Science and Technology Plan Project (KC19197), and the School-Level Scientific Research Project of Xuzhou Institute of Technology (XKY20191070).

## References

[1] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, "Learning spatially regularized correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4310–4318, Santiago, Chile, December 2015.

[2] D. Yuan, X. Lu, D. Li, Y. Liang, and X. Zhang, "Particle filter re-detection for visual tracking via correlation filters," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 14277–14301, 2019.

[3] D. Yuan, W. Kang, and Z. He, "Robust visual tracking with correlation filters and metric learning," *Knowledge-Based Systems*, vol. 195, p. 105697, 2020.

[4] L. Chen, D. Jiang, H. Song et al., "A lightweight end-side user experience data collection system for quality evaluation of multimedia communications," *IEEE Access*, vol. 6, no. 1, pp. 15408–15419, 2018.

[5] F. Li, C. Tian, W. Zuo, L. Zhang, and M.-H. Yang, "Learning spatial-temporal regularized correlation filters for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4904–4913, Salt Lake City, UT, USA, June 2018.

[6] S. Li, Z. Qin, and H. Song, "A temporal-spatial method for group detection, locating and tracking," *IEEE Access*, vol. 4, pp. 4484–4494, 2016.

[7] S. Hare, S. Golodetz, A. Saffari et al., "Struck: structured output tracking with kernels," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 2096–2109, 2016.

[8] D. Bolme, J. Beveridge, B. Draper, and Y. Lui, "Visual object tracking using adaptive correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2544–2550, San Francisco, CA, USA, June 2010.

[9] J. Henriques, R. Caseiror, P. Martins, and P. Jorge, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of the 12th European Conference on Computer Vision*, pp. 702–715, Florence, Italy, October 2012.

[10] J. Henriques, J. Carreira, C. Rui, and B. Jorge, "Beyond hard negative mining: efficient detector learning via block-circulant decomposition," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2760–2767, Sydney, Australia, April 2013.

[11] J. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[12] H. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1369–1378, Venice, Italy, October 2017.

[13] T. Liu, G. Wang, and Q. Yang, "Real-time part-based visual tracking via adaptive correlation filters," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4902–4912, Boston, MA, USA, June 2015.

[14] M. Danelljan, F. S. Khan, and M. Felsberg, "Adaptive color attributes for real-time visual tracking," in *Proceedings of the Computer Vision and Pattern Recognition (CVPR)*, pp. 1090–1097, IEEE, Columbus, OH, USA, June 2014.

[15] G. H. Zhao, S. Zhuo, and X. L. Xu, "Multi-object tracking algorithm based on kalman filter," *Computer Science*, vol. 45, no. 8, pp. 253–257, 2018.

[16] Z. L. Zhang and Y. X. Wang, "SiamRPN target tracking method based on kalman filter," *Intelligent Computer And Applications*, vol. 10, no. 3, pp. 44–50, 2020.

[17] J. Zhang, H. M. Huang, and J. M. Wang, "An improved TLD real-time target tracking algorithm based on CN algorithm," *Computer Engineering & Science*, vol. 42, no. 7, pp. 1215–1225, 2020.

[18] D. Yuan, N. Fan, and Z. He, "Learning target-focusing convolutional regression model for visual object tracking," *Knowledge-Based System*, vol. 194, p. 105526, 2020.

[19] C. Ma, X. K. Yang, C. Y. Zhang, and M. H. Yang, "Long-term correlation tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.

[20] T. Y. Xu, *Research on Correlation Filter Based Visual Object Tracking*, pp. 74–77, Jiangnan University, Wuxi, China, 2019.

[21] D. Yuan, X. Zhang, J. Liu, and D. Li, "A multiple feature fused model for visual object tracking via correlation filters," *Multimedia Tools and Applications*, vol. 78, no. 19, pp. 27271–27290, 2019.

[22] Y. Li and J. Zhu, "A scale adaptive kernel correlation filter tracker with feature integration," in *Proceedings of the European Conference On Computer Vision*, pp. 254–265, Zurich, Switzerland, September 2014.

[23] P. Zhao, Y. N. Zhang, T. Yang, X. W. Zhang, and Y. H. Yang, "A novel multi-object detection method in complex scene using synthetic aperture imaging," *Pattern Recognition*, vol. 45, no. 4, pp. 1637–1658, 2012.

[24] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Robust visual tracking via hierarchical convolutional features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 11, pp. 2709–2723, 2019.

[25] Y. Qi, S. Zhang, L. Qin et al., "Hedged deep tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4303–4311, Las Vegas, NV, USA, June 2016.

[26] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "Endto-end representation learning for correlation filter based tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2085–2813, Honolulu, HI, USA, July 2017.

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, May 2015.

[28] T. Y. Xu, Z. H. Feng, X. J. Wu, and J. Kittle, "Joint group feature selection and discriminative filter learning for robust visual object tracking," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7949–7959, Seoul, South Korea, October 2019.

[29] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," 2010, http://arxiv.org/abs/1009.5055.

[30] K. B. Petersen and M. S. Pedersen, *The Matrix Cookbook*, Technical University of Denmark, Lyngby, Denmark, 2008.
[31] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1834–1848, 2015.
[32] P. Liang, E. Blasch, and H. Ling, "Encoding color information for visual tracking: algorithms and benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5630–5644, Dec. 2015.