Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6653954, 11 pages
https://doi.org/10.1155/2021/6653954

Research Article

Single-Object Tracking Algorithm Based on Two-Step Spatiotemporal Deep Feature Fusion in a Complex Surveillance Scenario

Yanyan Chen[1] **and Rui Sheng**[2]

[1]*Jiuzhou Polytechnic, Xuzhou 221116, China*
[2]*Southwest China Institute of Electronic Technology, Chengdu 610036, China*

Correspondence should be addressed to Yanyan Chen; chenyanyan@jzp.edu.cn

Received 3 November 2020; Revised 18 December 2020; Accepted 26 December 2020; Published 5 January 2021

Academic Editor: Yi-Zhang Jiang

Object tracking has been one of the most active research directions in the field of computer vision. In this paper, an effective single-object tracking algorithm based on two-step spatiotemporal feature fusion is proposed, which combines deep learning detection with the kernelized correlation filtering (KCF) tracking algorithm. Deep learning detection is adopted to obtain more accurate spatial position and scale information and reduce the cumulative error. In addition, the improved KCF algorithm is adopted to track and calculate the temporal information correlation of gradient features between video frames, so as to reduce the probability of missing detection and ensure the running speed. In the process of tracking, the spatiotemporal information is fused through feature analysis. A large number of experiment results show that our proposed algorithm has more tracking performance than the traditional KCF algorithm and can efficiently continuously detect and track objects in different complex scenes, which is suitable for engineering application.

1. Introduction

With the rapid development of computer vision technology, video-based object tracking algorithms have become a research hotspot in research institutes and universities at home and abroad [1]. Object tracking technology usually builds a robust model based on the object and its background information in the video to predict the shape, size, position, trajectory, and other motion states of the object in the video, which can achieve more advanced tasks, such as the behavior prediction, scene understanding, and situation awareness [2]. Object tracking currently has a wide range of application fields, including video surveillance [3], unmanned driving [4], military guidance [5], UAV reconnaissance, intelligent transportation, and human-computer interaction [6]. It has important research value.

In recent years, many effective object tracking algorithms have been proposed. Generally speaking, object tracking algorithms are divided into generative tracking algorithms and discriminative tracking algorithms according to different judgment methods [7]. The current main research direction is focused on discriminative tracking algorithms and has gradually occupied a dominant position in the field of visual object tracking and has achieved a series of excellent research models. Different from the generative-based tracking algorithm, the discriminative-based tracking algorithm does not ignore the background information, but regards the object tracking as a two-classification problem, where the object area of the current frame can be tracked by designing a classifier to distinguish the object and the background area [8].

The Struck tracking algorithm proposed by Sam et al. [9] in 2011 directly outputs the tracking results by introducing an output feature space mapping and uses a support vector machine to train the classifier, which improves the tracking accuracy and further accelerates the tracking speed of the

algorithm. Kalal et al. proposed a tracking learning detection (TLD) algorithm on the basis of online learning, which has a better tracking effect for long-term tracking under complex background [10]. Bolme et al. proposed the minimum output sum of squared error (MOSSE) tracking algorithm and introduced correlation filtering into the object tracking algorithm for the first time, but the used grayscale features are too simple to adapt all scenarios [11]. Therefore, there are many algorithms to improve on it since then. Henriques et al. introduced the kernel function mapping into the original MOSSE algorithm and proposed a circulant structure of tracking by detection with kernels (CSK) and adopted the cycle shifting method for dense sampling [12]. However, the CSK tracking algorithm did not improve the selection of features but still used the image gray features, which makes the feature characterization ability of the object not strong. On the basis of the CSK algorithm, Henriques et al. [13] used multichannel HOG features instead of single-channel gray features and proposed the kernelized correlation filtering (KCF) tracking algorithm and enhanced the robustness of the existing tracking algorithm. Moreover, the KCF algorithm uses a circulant matrix for sampling, which reduces the complexity of the algorithm and improves the speed of tracking. However, the KCF algorithm has a poor tracking effect on scale variations [14]. In order to solve these problems, Li and Zhu [15] proposed the scale adaptive kernel correlation filter (SAMF) tracking algorithm, which introduced the concept of scale pooling for the first time. The tracking effect of objects with scale changes is better than the KCF algorithm. The detection is performed on images of several scales, so the tracking speed of the SAMF algorithm is very slow, which cannot meet the real-time requirements. In 2017, Danelljan et al. [16] proposed the context aware correlation filtering (CALF) algorithm, where the filter was trained by strengthening background information, so that the CALF algorithm can maintain better performance for object tracking with complex background. On the basis of the SRDCF tracking algorithm, the spatial-temporal regularized correlation filter (STRCF) was proposed, in which a temporal regularization term is introduced into the SRDCF algorithm and can effectively suppress the boundary effect [17].

With the continuous development of neural networks and deep learning, the deep features learned by machines can better extract the most essential image information. Therefore, some scholars have proposed a series of object tracking algorithms based on deep features. The hierarchical convolutional features (HCF) tracking algorithm used three convolutional layers in the VGG network to obtain image deep features, and three different templates are obtained through training [18]; then, the obtained three confidence maps are weighted and fused to obtain the object position [19]. Similarly, Danelljan et al. used deep features to replace the original SRDCF algorithm and proposed the DeepSRDCF tracking algorithm, which greatly improved the tracking accuracy of the object tracking algorithm. The deep model tracking algorithms proposed above all use the image deep features extracted by the convolutional neural network for object tracking. In addition, the fully convolutional network (FCT) tracking algorithm uses the regression network based on deep learning to predict the object position so as to accurately track the object. In 2018, Zhong et al. [20] proposed the unveiling the power of deep tracking (UPDT) algorithm on the basis of the ECO algorithm. By analyzing the impact of deep features and shallow features on tracking accuracy, a novel feature fusion strategy was proposed to improve the tracking performance of the algorithm. Xue and Wang [21] proposed a SiamRPN algorithm and Siamese network structure based on RPN, giving up the use of traditional multiscale training and online tracking, thereby improving the tracking speed to a certain extent. In CVPR2019, Wang et al. proposed an accurate tracking by overlap maximization (ATOM) algorithm, which introduced the idea of IoUNet object detection and the object classification module so as to have more powerful discrimination ability for the tracker [22].

It can be seen from the above analysis that the traditional algorithms have high tracking speed, but their anti-interference ability is still insufficient. The tracking algorithms based on a deep model can be adapted to most complex scenes, but they consume a lot of hardware resources and have poor real-time tracking performance. In this paper, an object tracking model based on two-step spatiotemporal information fusion is proposed, which uses deep learning detection to obtain more accurate spatial position and scale information, reducing the cumulative error. In addition, the algorithm uses KCF to track and calculate the temporal information correlation of gradient features between video frames, so as to reduce the probability of missing detection and ensure the running speed. In the process of tracking, the detection is run after a certain number of image frames, and the spatiotemporal information is fused through feature analysis. Under the condition of ensuring the tracking speed and accuracy, it can also detect the new object in the complex video in time and track continuously for a long time.

2. Problem Description for Object Tracking

In this paper, we mainly study single-object tracking in a complex video. As shown in Figure 1, the basic framework of the single-object tracking algorithm mainly includes four parts: feature model, motion model, observation model, and online updating mechanism. Each part has its own special role. In other words, the four aspects are mutually reinforcing and indispensable parts of an integral whole. The feature model is designed to use image processing technology to obtain information that can characterize the appearance of the object and serve the construction of the observation model. The features suitable for object tracking are gray feature, color feature, histogram of oriented gradient feature, deep feature, etc.; the motion model mainly provides a set of candidate states that the object may appear in the current frame based on the context information of the object; the role of the observation model is to predict the state of the object on the basis of the candidate state provided by the feature model and the motion model; the online updating mechanism allows the observation model to adapt

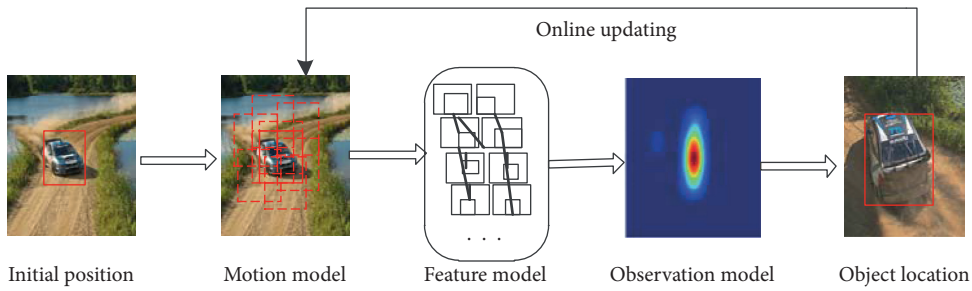

Figure 1: Basic framework of the single-object tracking algorithm.

the changes of the object and background and ensures that the observation model does not degenerate.

There are many interference factors in the video tracking task, and it faces a series of difficulties in practical tracking applications, such as appearance change, illumination variation, partial occlusion, and complex background. In object appearance changes, it refers to the change of the tracked object's appearance or the shooting angle of the camera during the movement, as shown in Figure 2(a). The illumination variation refers to the change of video imaging gray due to changes in the light source or the surrounding environment, as shown in Figure 2(b). Scale changes refer to the change of the pixel size of the object in the video due to the movement of the object or the change of the distance, as shown in Figure 2(c). Partial occlusion or object losing refers to an interference phenomenon where the object is affected by the background or moved out of the field of view, resulting in an incomplete appearance or completely out of the field of view, as shown in Figure 2(d). The complex background refers to a large number of interference factors (such as a large number of similar objects) in the background, which causes interference to the object observation model. In addition, there are other interference factors such as fast movement, small objects, and blurring during the tracking process. These interference factors limit the performance of the tracking model to varying degrees, resulting in a decrease in the overall accuracy. With the development of object tracking technology, although some problems have been solved, such as the use of HOG features to effectively solve the problem of illumination changes in tracking tasks, there are still many problems need to be solved in the actual application process. In this paper, we mainly focus on solving the problem of partial occlusion and object recapture in the process of object tracking.

3. Our Proposed Tracking Algorithms

Object detection and tracking based on spatiotemporal information fusion is mainly divided into three parts: object detection based on deep spatial information, KCF tracking based on temporal information, and fusion of spatiotemporal information. Firstly, the You Only Look Once (YOLO-V3) detector is used to detect the object. And then, the KCF tracking model is used to track the object in a complex surveillance video [23]. After tracking a certain number of frames, the YOLO-V3 detection mechanism is adopted again to compare the confidence of the old tracking bounding box and the new detection bounding box. Through the spatiotemporal information fusion strategy, the appropriate bounding box is obtained to continue tracking. If a new object is detected in the field of view, the new object is tracked at the same time. The overall detection and tracking system is shown in Figure 3.

3.1. Object Detection Based on Deep Spatial Information. In this paper, we use the framework of the YOLO-V3 deep model to realize the object detection, and we also redesign the bounding box selective search method to improve the detection accuracy of the object spatial information. Firstly, the input image features are fully extracted by the basic network through iterative convolution operation, and then further feature extraction and analysis are carried out through the additional network. The object position offset is predicted and classified by using a convolution predictor. Finally, the redundancy is removed by the nonmaximum suppression method. The basic network uses the improved VGG structure as the feature extraction network. Two convolution layers are used at the end of the network to replace the two fully connected layers of the original VGG network, and eight additional networks are added to further improve the feature extraction ability. It is widely known that different depth feature maps have different receptive fields and different responses to different scale objects. The network structure is shown in Figure 4.

The detection of multiscale objects is divided into 3 steps: default boxes with the different aspect ratio and same area are generated on different scale feature maps; after training a large number of samples, the convolution predictor uses the abstract features in the default box as an input to predict the offset of the default bounding box; nonmaximum suppression is used to remove redundant bounding boxes with low confidence.

The default bounding box generation method is improved as follows. Firstly, assuming that it is necessary to make predictions on a total of m feature maps, the area (scale) s_k of the default bounding box on the first $k-$th feature map can be written as follows:

$$s_k = s_{\min} + \frac{(s_{\max} - s_{\min})}{(m-1)}(k-1), \quad k \in [1, m], \tag{1}$$

where $m = 6$, the minimum area is 0.2, and the maximum area is 0.95. In this paper, the K-means clustering algorithm is used to process the aspect ratio of all suspected objects in

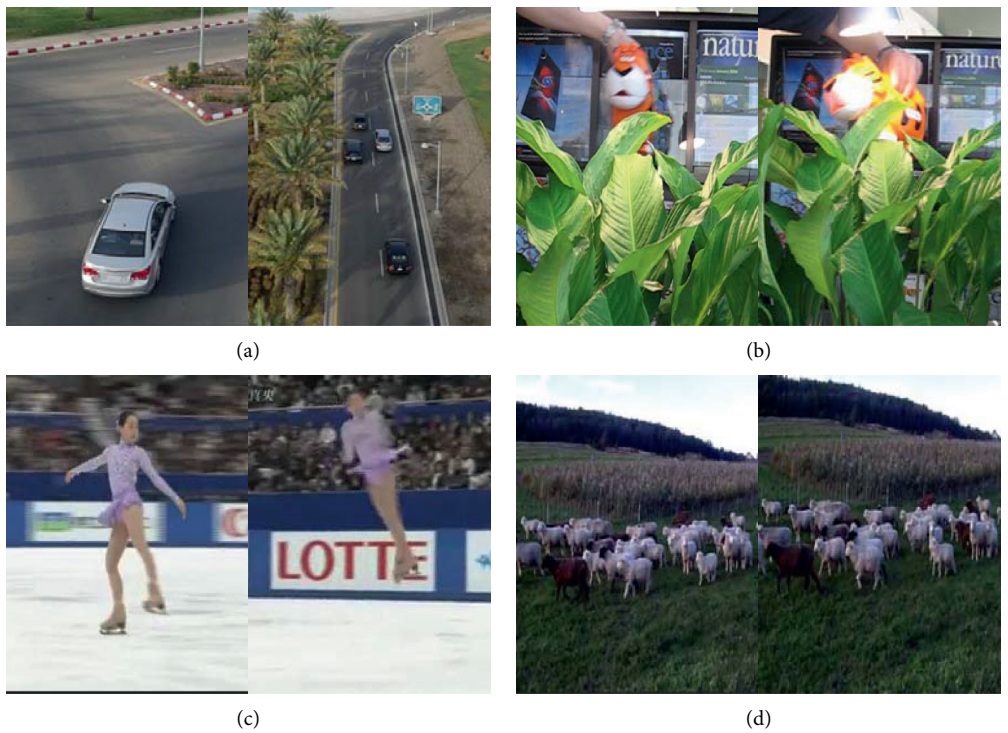

Figure 2: Samples for different interference factors. (a) Scale change. (b) Illumination variation. (c) Appearance change. (d) Partial occlusion.

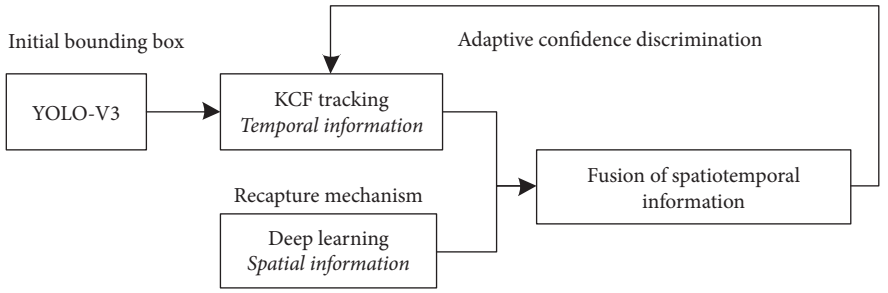

Figure 3: Overall detection and tracking framework for our model.

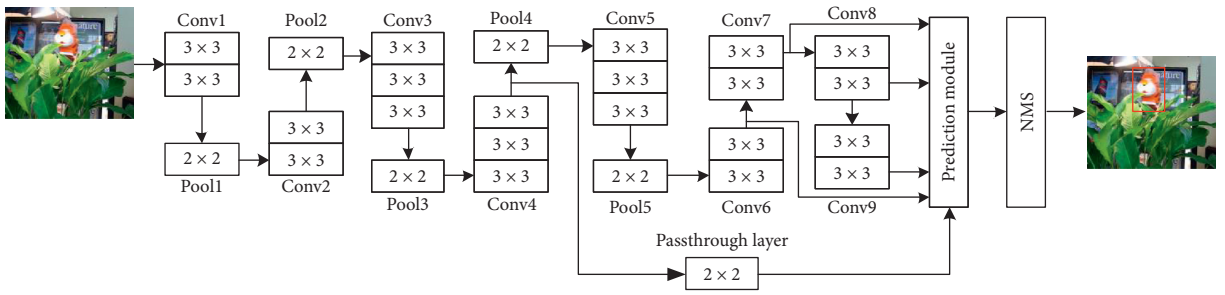

Figure 4: The network structure for object detection.

the dataset, and 5 cluster centers are obtained. Therefore, the new aspect ratio is denoted as $a_r \in \{1, 1.8, 2.5, 3.8, .5\}$, which provides a better initial bounding box for object detection.

YOLO-V3 convolutional network is used to obtain the coordinate offset from the fixed default bounding box to the actual benchmark value and the category score and obtain the loss function through the normalization and weighting

of the category score and the coordinate offset. Therefore, the loss function can be described as follows:

$$L\left(x_{ij}^{k}, c, l, g\right) = \frac{1}{N}\left[L_{\text{conf}}\left(x_{ij}^{k}, c\right) + \alpha L_{\text{loc}}\left(x_{ij}^{k}, l, g\right)\right], \quad (2)$$

where $x_{ij}^{k} = 1$ means that the candidate bounding box i matches the object real bounding box j with category p successfully, and otherwise $x_{ij}^{k} = 0$ means the match fails; N is the number of candidate bounding box that can be matched with the true value; L_{loc} is the position loss function smooth $L1$ loss; and α is set to 1. The network parameters can be optimized according to the result of the loss function.

3.2. KCF Tracking Based on Temporal Information. KCF algorithm is a classical discriminative-based object tracking algorithm, which has good performance in tracking speed and tracking accuracy. In the tracking process, the object bounding box of the KCF algorithm has been set, and the size of the object scale has not changed from beginning to end. However, the object size often changes in the tracking video sequence, which will lead to the drift of the bounding box in the tracking process of the tracker, even resulting in tracking failure. In addition, the KCF algorithm cannot deal with the occlusion of the object in the tracking process, which will lead to the feature extraction error when training the filter model. When the object moves rapidly, some object features cannot be extracted because of the fixed size of the searching box, where the quality of the detection model will be reduced and the tracking failure will be caused when updating the model. In order to solve the problem of tracking failure caused by the KCF algorithm in the above situations, some scholars improved the KCF algorithm and proposed some novel yet effective object tracking algorithms based on deep learning detection, and a large number of experiment results show that the improved algorithm has better accuracy and robustness than the original KCF algorithm.

As for complex monitoring applications, the real-time performance of object tracking is very important. We select KCF as the basic tracking algorithm, which has a greater advantage in speed. In addition, considering the characteristics of large changes in object scale, a multiscale adaptive module is added in KCF. HOG features are adopted to train the classifier and transform it into a ridge regression model so as to establish the mapping relationship between the input sample variable x and the output response y. The ridge regression objective function can be rewritten as follows:

$$\min_{\omega} \sum_{i} \left(f\left(x_i\right) - y_i\right)^2 + \lambda \|\omega\|^2, \quad (3)$$

where $\lambda\,(\lambda \geq 0)$ is a regularization parameter. The regularization term is added to avoid the occurrence of overfitting in optimization. In order to minimize the gap between the sample label predicted by the regression model and the real label, a weight coefficient is assigned to each sample to obtain a closed solution formula for the regression parameters. Therefore, the analytical solution ω can be deduced and represented as

$$\omega = \left(X^T X + \lambda I\right)^{-1} X^T y. \quad (4)$$

Due to the time-consuming calculation of dense sampling in equation (3), cyclic shifting is used to construct training samples, and the problem domain is transformed into the discrete Fourier domain. The characteristics of the circulant matrix can avoid the process of matrix inversion and accelerate feature space learning. The circulant matrix can be diagonalized, and this can be described as follows:

$$X = F\,diag\left(\hat{x}\right)F^H. \quad (5)$$

In order to simplify the calculation, the features obtained by ridge regression with linear space are mapped to the nonlinear space through the kernel function, and a dual problem is solved in the nonlinear space. Through the mapping function $\phi(x)$, the classifier can be denoted as follows:

$$f\left(x_i\right) = \omega^T \phi\left(x_i\right). \quad (6)$$

Given $\omega = \sum_i a_i \phi(x_i)$, the solution of ω can be transformed into the solution of α. Therefore, on the basis of the kernel function $K = \phi(X)\phi(X)^T$, we can get the solution based on the ridge regression under the kernel function, namely,

$$\alpha = (K + \lambda I)^{-1} y. \quad (7)$$

Finally, we can get the response results of all test samples in the Fourier domain:

$$f\left(z\right) = \hat{k}^{xz} \Theta \hat{\alpha}. \quad (8)$$

The sample with the strongest response is selected as the object position in the current frame.

The overall framework of the tracking algorithm is shown in Figure 5. First, the object is initialized in the first frame, and the features of the object are extracted, and then the ridge regression model is trained to obtain the optimal filter parameters; then in the process of object tracking, the feature is extracted on the current frame, and convolution operation is performed with the filter template trained in the previous frame. We can get the response map, where the maximum correlation value is the object position.

In order to adapt the change of the object scale, a scale adaptive strategy is developed to ensure the stability of tracking. Taking the object position as the center, rectangular bounding boxes with different scales are selected as samples, and their HOG features are extracted, respectively. Therefore, we can get the respective sample responses R_0, R_{+1}, and R_{-1} after tracking the classifier and obtain the strongest response after comparison:

$$R = \max\left(R_0, R_{+1}, R_{-1}\right). \quad (9)$$

The rectangular bounding box corresponding to the sample with the strongest response is the current object scale, where the improved KCF can be used for multiscale adaptation selection, and the amount of calculation is small and efficient and feasible.

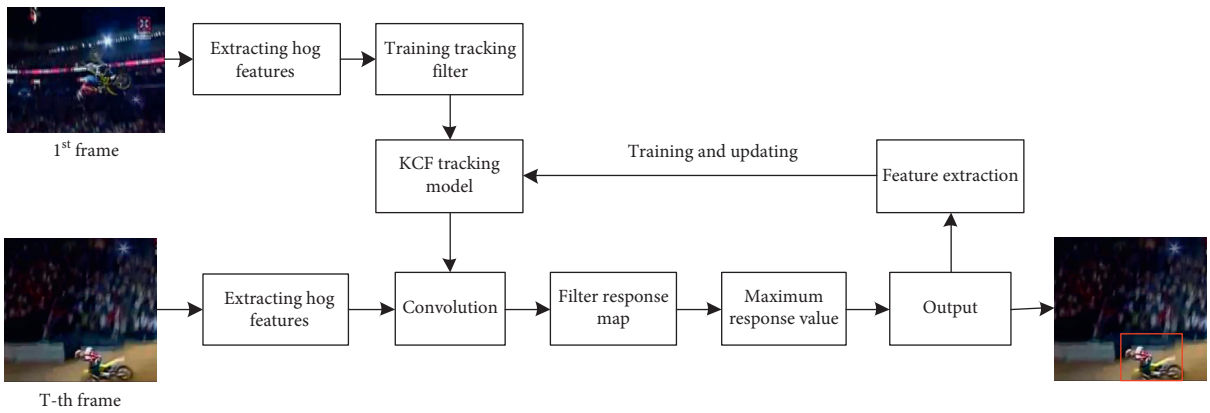

Figure 5: The overall framework of the tracking algorithm.

3.3. Object Detection and Tracking for Spatiotemporal Fusion. As we all know, using deep learning for object detection to extract single-frame image features has high accuracy, can identify and classify unknown objects, and has high robustness. However, the object detection does not combine the temporal information relationship between the consecutive frame in the video, which may lead to the missed detection and slow running speed. KCF tracking is achieved by extracting the characteristics of continuous frame images to train filters in ridge regression, where the calculation is small and the processing speed is also fast. However, it is easy to accumulate errors because of tracking drift and be easily affected by object occlusion and background interference. Therefore, the fusion of temporal information and spatial information can make full use of the advantages of deep learning and KCF, improve overall performance, and achieve more accurate and stable detection and tracking on the basis of robustness and real-time performance.

In the process of information fusion, the spatial position information of the object is determined by the deep learning-based object detection algorithm in the first frame, and then the position of the object in the first frame is used as the input of the KCF tracking algorithm, and the tracking algorithm is used to track the object in the following frames. After tracking a fixed number of frames, the detection mechanism is run to ensure the accuracy of continuous detection and tracking through the YOLO-V3 detection algorithm. The number of tracking frames between the two detection operations can be determined by experiment. Generally, it can be set to 50 frames. In addition, we can also use the confidence of the detection results as the basis of template refresh and recapture.

After running the redetection mechanism, it is not sure which one is better to track candidate bounding boxes or detect candidate bounding boxes obtained by the redetection module. Therefore, this paper designs a candidate frame selection strategy. Firstly, the overlap ratio between detection candidate bounding box S_i and tracking candidate bounding box K_j is calculated to judge whether the detected and tracked objects are the same. In this paper, the intersection over union (IOU) is used as the criterion of overlap ratio. The IOU of two candidate bounding boxes can be written as follows:

$$\mathrm{IOU} = \frac{S_i \cap K_j}{S_i \cap K_j}. \tag{10}$$

If $\forall K_j, IOU(S_i, K_j) < 0.4$, S_i will be regarded as a new object and output to achieve the initialization of the tracking algorithm. If $\exists K_j, IOU(S_i, K_j) < 0.4$, it is considered that the detection bounding box S_i and the tracking bounding box K_j have detected the same object; then the confidence level $\mathrm{conf}(S_i)$ of the bounding box of the detection algorithm is compared with the normalized response $\mathrm{conf}(K_j)$ of the bounding box of the tracking algorithm. Finally, the bounding box with higher confidence is taken as the output of the system.

4. Experimental Results and Analysis

4.1. Dataset and Verification Platform. In order to improve the accuracy and robustness of the detection and tracking algorithm in the video surveillance task, this experiment constructs a surveillance dataset with 321550 images. To facilitate performance analysis, all data are labeled frame by frame in scale and position and classified according to the interference state.

The improved detection and tracking model is divided into three parts: object detection based on deep spatial information, KCF tracking based on temporal information, and fusion of spatiotemporal information. The parameters of each part are consistent with the original model. During offline training, all convolution layers will be updated. After online updating, the parameters of the shallow convolution layer are fixed, and the last two convolution layers will be fine-tuned according to the test data. During the training, the YOLO-V3 model trained by Pascal VOC2007 [24] is used as the initial weight parameter to fine-tune the network, where the learning rate is set to 0.001 and the weight attenuation was 0.0005. 30000 iteration times in training were conducted on NVIDIA Geforce GTX 1080TI. The KCF module uses the peak-side-lobe ratio to select the optimal tracking point, and the threshold of normalized response is set to 0.65. If the regression response score is less than 0.65, it

is considered that the tracking is failed, and the improved YOLO-V3 detection network is used to recapture the optimal object.

In this paper, eight representative subsets from video surveillance are selected for verification, where characteristic for partial sequences is described in Table 1. For example, video 1 shows the similarity background, occlusion, and fast motion; video 2 shows the similarity background, fast motion, and rotation; video 3 and 4 show the occlusion, rotation, and attitude change; and video 5 shows the fast motion, illumination, and similarity background. The simulation platform is AMD Ryzen 5 3500U host with 3.1 GHz and 8 GB RAM.

In this paper, center error (CE) and overlap rate (OR) are used to compare and analyze the experimental results [19]. The former is the relative number of frames whose center position error is less than a certain threshold, and the latter is the percentage of frames whose overlap rate of the object bounding box exceeds the threshold. In this paper, the position error of 20 and the overlap rate of 0.6 are selected as the threshold of tracking success. Because of the different thresholds, there are great differences in quantitative analysis. Therefore, precision plot and success plot are used to quantitatively analyze the performance of the comparison algorithms.

4.2. Ablation Analysis. Our proposed method in this paper is an improved tracking method based on KCF to achieve the effect of scale adaptation. In order to illustrate the effectiveness, the comparison experiment in this paper selects tracking methods with adaptive scale capabilities for comparison, such as KCF, SAMF, DSST, CFNet [23], SiamRPN [24], and DKCF [25], where precision refers to the error between the tracking point and the labeled point. It can be known that the result of KCF only updates the position of the object (x, y), and the size of the object remains unchanged, so the adaptability to the change of the object scale is relatively poor; SAMF is also a modified algorithm on the basis of KCF, and the object feature adds color features (color name, CN), which means that HOG features and CN features are combined. In addition, multiscales {1 0.985 0.99 0.995 1.005 1.01 1.01 1.015} are added to the scale pooling, and the optimal scale is cyclically selected at the expense of tracking speed; DSST uses two mutually independent filters for scale calculation and object positioning, where 17 scale change factors and 33 interpolated scale change factors are established for scale evaluation and object positioning. SiamFC is an object tracking algorithm based on a fully convolution Siamese network, where multiscale object fusion is implemented through a pyramid strategy to improve tracking accuracy; our proposed algorithm is a detect-before-track model that uses deep neural networks in template updating and scale adaptation. The results of object detection and tracking under the influence of different environments are shown in Table 2, and the precision plot and success plot of detection and tracking in 8 different video sequences are shown in Figure 6. It can be seen from Table 2 and Figure 6 that compared with video 1, the tracking success rates of videos 2, 3, 4, and 5 have different degrees of decline. It can be seen that occlusion, scale change, motion blur, and illumination have an impact on the detection and tracking effect, of which occlusion and illumination changes have a greater impact. Different degrees of motion blur have different effects on detection and tracking. When the object overlap rate threshold is set to 0.6, the average detection and tracking accuracy is 76.17%, and the average speed can reach 18 FPS. The slower speed of video 2 is caused by the appearance of new objects in the field of view. The object scale in video 4 is larger, so the detection and tracking time is longer.

Table 1: Characteristic for partial sequences.

Sequences	Characteristic
Benchmark video 1	Similarity background, occlusion, fast motion
Benchmark video 2	Similarity background, fast motion, rotation
Benchmark video 3	Occlusion, rotation
Benchmark video 4	Fast motion, attitude change
Benchmark video 5	Fast motion, illumination, similarity background
Benchmark video 6	Occlusion, similarity background, blurry
Benchmark video 7	Occlusion, scale change, angle of view
Benchmark video 8	Occlusion, rotation, illumination

Video 2 under the condition of object occlusion and video 5 under the condition of illumination changes are selected for comparative experiments. Our proposed tracking algorithm is compared with a single tracking algorithm and detection algorithm. Video 2 has the phenomenon of object occlusion. The experimental results are shown in Table 2 and Figure 6. In terms of center error and overlap rate, the fusion algorithm is obviously better. Deep learning detection algorithm may not be able to detect the object with too small scale, resulting in low recall rate. In the long-term detection and tracking, the correlation filtering tracking algorithm will accumulate errors, resulting in poor accuracy. Especially for the occluded object, the tracking drift phenomenon is easy to occur. These reasons make the center error and overlap rate of the single detection or tracking algorithm not high. The fusion algorithm ensures a high recall rate through KCF tracking and corrects the cumulative error by YOLO-V3 detection. After the object is occluded, it can still recapture the object again and keep tracking, which solves the object lost problem in object detection and tracking.

There is illumination change in video 5. The experimental results are shown in Table 3 and Figure 7. Due to the influence of illumination change, it is difficult to distinguish the illumination and shade between the object edge and the background, which makes the object bounding box cannot be determined for detection and tracking. Even if the object position can be detected and

Table 2: The quantitative analysis for testing sequences.

Indexes	Video 1	Video 2	Video 3	Video 4	Video 5	Video 6	Video 7	Video 8
CLE	10.6	29.6	31.1	23.8	18.9	21.0	17.5	8.7
OR	0.88	0.72	0.62	0.81	0.68	0.71	0.58	0.73
FPS	17.4	12.0	20.9	16.8	14.1	19.1	18.5	17.2

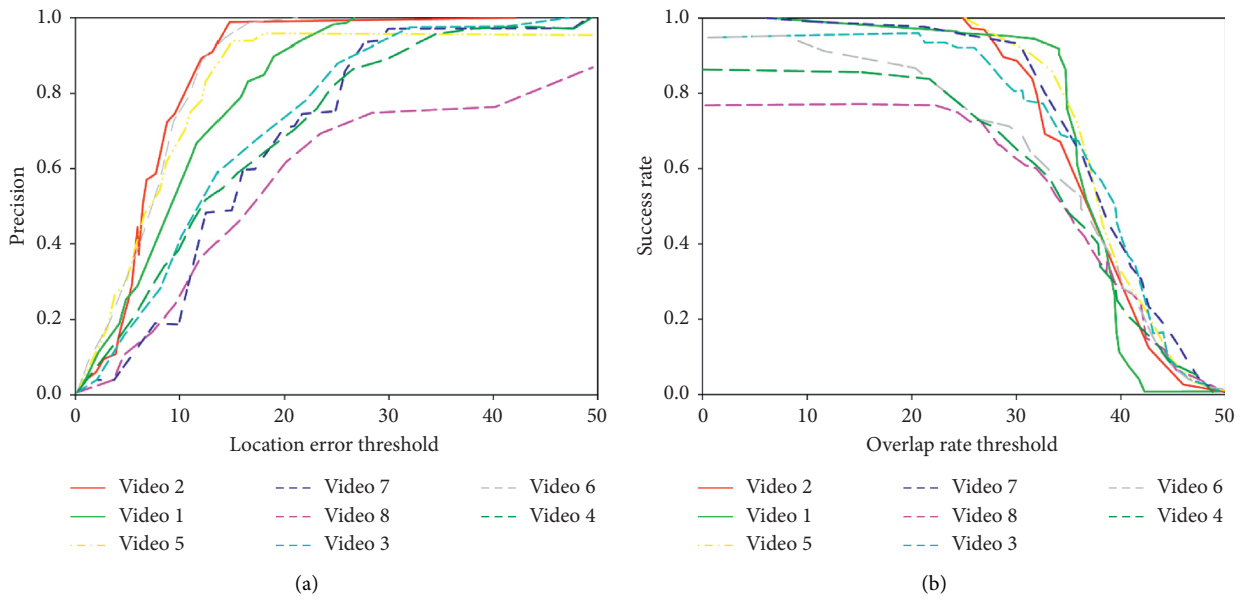

Figure 6: The tracking results for different testing sequences. (a) Precision plot and (b) success plot.

Table 3: Detection and tracking results for different modules.

Index (average)	YOLO-V3	KCF	Fusion
CLE (pixel)	16.9	14.6	10.5
OR	0.816	0.752	0.861
Frame rate (FPS)	8.7	**46.0**	16.1

tracked, the judgment of the object scale is not accurate. Therefore, the accuracy of center position error is higher, but the overlap rate is lower in the KCF tracking algorithm. YOLO-V3 detection algorithm has strong robustness, but it has the phenomenon of missing detection. Therefore, simulation results show that our proposed fusion algorithm has better detection and tracking performance in the complex environment.

4.3. Comparative Experiment and Analysis. In this paper, we select different detection and tracking algorithms to conduct comparative experiments on single-object videos, where the SSD and YOLO-V3 algorithms that are widely used are selected in the spatial dimension, and the classic single-object tracking algorithms DSST, KCF, and SAMF are selected in the temporal dimension. The experiment is divided into two parts. The first part is a comparison of a single spatial detection algorithm or a temporal tracking algorithm with our proposed algorithm; the second part is a comparison of different detection and tracking algorithm combinations based on the fusion strategy. Table 4 shows the comparison results of a single algorithm. If the detection algorithm is compared separately, the detection accuracy of the YOLO-V3 algorithm is higher. Overall, the success rate of a single algorithm is much lower than the YOLO-V3 + KCF fusion algorithm. This is because the detection algorithm is affected by the complex background, resulting in a large number of missed detections; the temporal algorithm will be affected by motion blur, and the accumulated error will cause the tracking drift, making the IOU between tracking result and ground truth less than 0.6.

Table 5 compares the fusion effects of different algorithms. It can be seen from Table 5 that the YOLO-V3 + KCF algorithm has the best effect. Because the KCF algorithm in Table 4 has a better effect in the tracking algorithm, the overall effect of the YOLO-V3 + KCF is also better than SSD + DSST and SSD + SAMF. Because the tracking algorithm uses temporal information to eliminate the missing detection of the detection algorithm, and the detection algorithm corrects the drift of the tracking result by accurately detecting a single object, the success rate of the fusion algorithm detection is more than that of the single algorithm in Table 4.

Figure 8 shows the qualitative results of different comparison algorithms. Table 6 is a quantitative comparison for different sequences. In Figure 8(a), there are factors such as object scale changes, illumination changes, and background interference. In the whole tracking process, only DKCF, SiamRPN, and our proposed algorithm have better tracking results. However, due to the

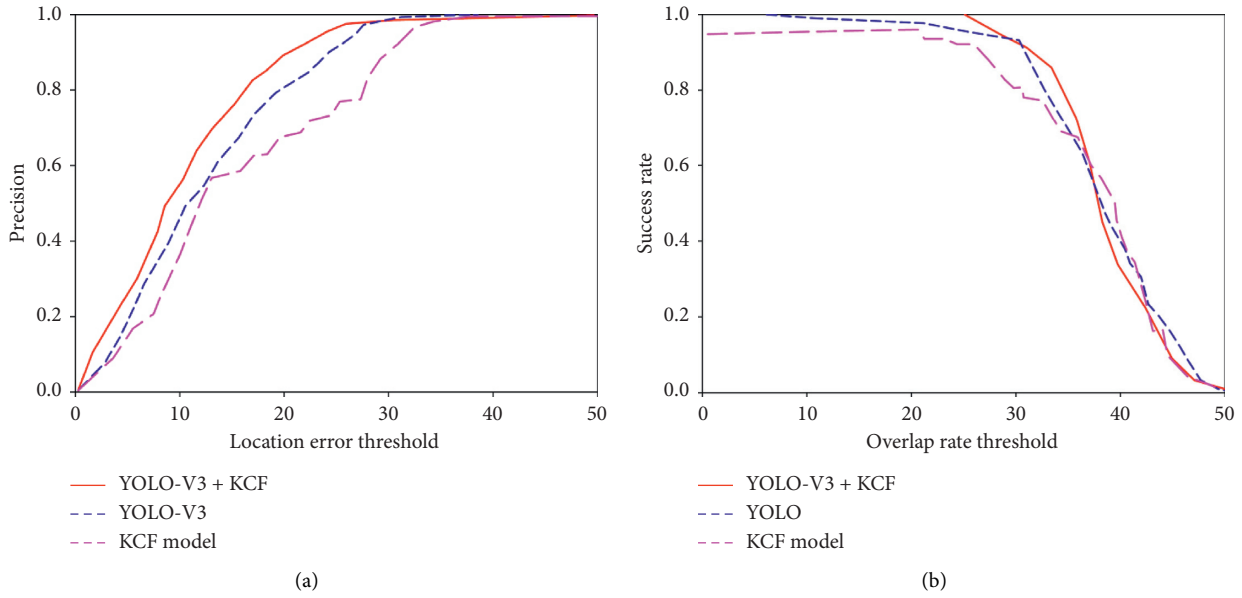

Figure 7: Ablation analysis for different modules in video 5. (a) Precision plot and (b) success plot.

Table 4: Comparison results for the single tracking algorithm.

Modules	Object detection		Object tracking			Fusion
Model	SSD	YOLO-V3	DSST	SAMF	KCF	YOLO-V3 + KCF
CLE (pixel)	21.1	16.9	17.27	13.28	14.6	10.1
OR	0.524	0.816	0.711	0.625	0.752	0.841
Frame rate (FPS)	6.5	8.7	34.2	26.1	46.0	16.1

Table 5: Tracking performance of different module combinations.

Index (average)	YOLO-V3 + KCF	YOLO-V3 + DSST	YOLO-V3+SAMF	SSD + KCF	SSD + KCF	SSD + KCF
CLE (pixel)	10.5	13.6	16.5	15.2	18.3	17.5
OR	0.861	*0.782*	0..771	0.837	0.825	0.776
Frame rate (FPS)	16.1	13..2	9.1	12.2	10.9	8.7

continuous change of object scale, the KCF tracking template-introduced background interference information gradually accumulates, and finally there is a large tracking deviation (such as the 640th frame). Our proposed algorithm can automatically adjust the tracking bounding box size according to the object scale change, thereby reducing the background interference information, so it can always estimate the location and the scale of the object; the object in video 7 has dramatic changes in illumination and scale (frames 65, 110, and 351 in Figure 8(b)). In the whole tracking process, only our proposed algorithm and SiamRPN can complete the tracking of the entire video, and other methods cannot adapt to drastic changes in illumination and scale; the object in video 6 has a certain scale and posture change, where KCF, SAMF, DSST, CFNet, SiamRPN, and our

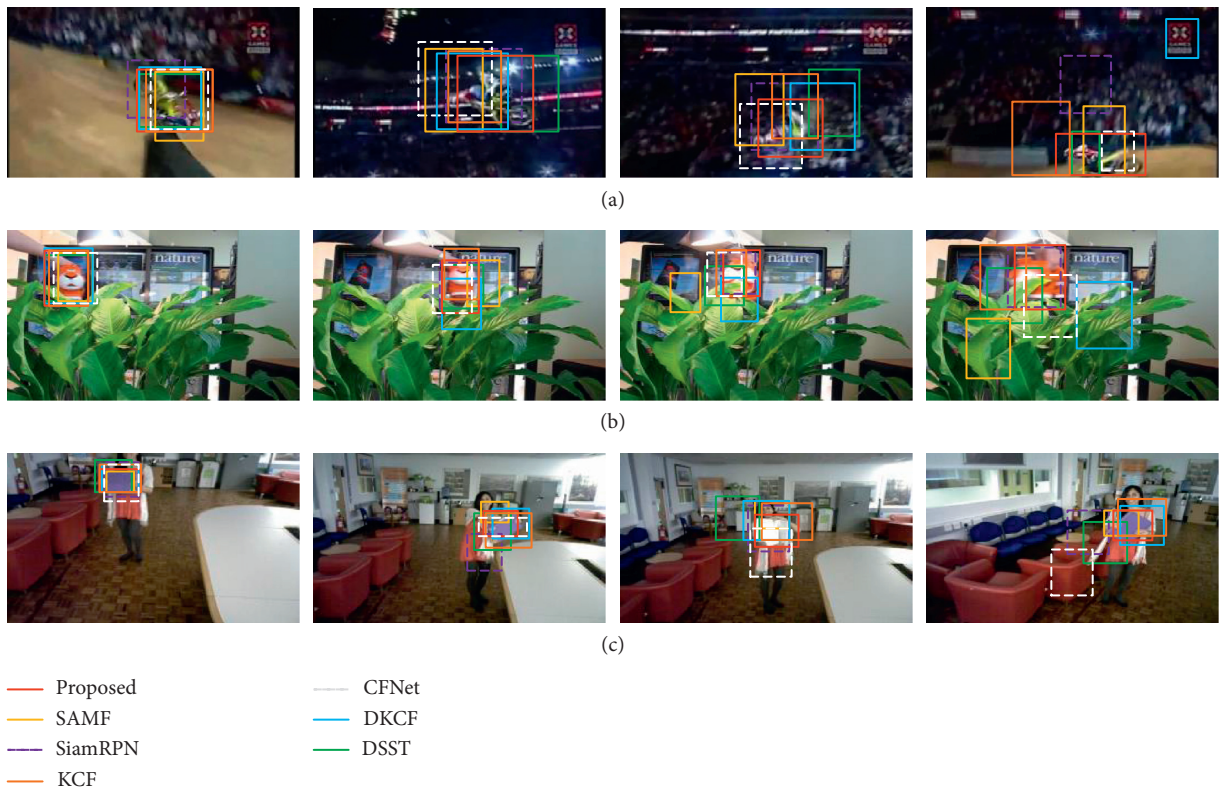

Figure 8: Qualitative comparison of different tracking algorithms in different scenarios. (a) Video 8, (b) video 7, and (c) video 6.

Table 6: Quantitative comparison for different sequences.

Sequences	Success rate							Center location error						
	KCF	SAMF	DSST	DKCF	CFNet	SiamRPN	Our	KCF	SAMF	DSST	DKCF	CFNet	SiamRPN	Our
Video 8	0.78	0.56	0.62	0.70	0.75	0.79	0.86	25.3	42.3	24.4	24.9	17.0	14.4	9.1
Video 7	0.60	0.48	0.65	0.61	0.78	0.81	0.81	21.7	28.6	9.2	37.4	21.2	10.8	7.5
Video 6	0.79	0.67	0.58	0.82	0.67	0.79	0.79	12.7	7.1	6.3	12.6	16.8	15.6	9.3
Video 5	0.61	0.52	0.54	0.77	0.47	0.40	0.42	14.6	25.0	11.4	17.1	15.2	2.9	8.1
Video 4	0.73	0.68	0.71	0.75	0.75	0.88	0.82	27.3	22.3	14.4	24.9	17.0	14.9	10.1
Video 3	0.68	0.70	0.72	0.76	0.79	0.83	0.87	31.7	28.3	9.2	37.4	21.2	10.8	8.4
Video 2	0.59	0.57	0.62	0.65	0.75	0.81	0.85	12.5	9.1	6.3	12.6	16.8	15.6	8.9
Video 1	0.68	0.62	0.66	0.72	0.61	0.79	0.79	74.6	23.0	21.4	27.1	15.2	19.9	18.2

proposed algorithm have better tracking performance, but our OR and CLE are the highest.

5. Conclusion

In a complex surveillance video, object detection and tracking usually suffers from various environmental interference, especially scale changes, occlusion, illumination changes, and motion blur. This paper proposes an object detection and tracking model based on spatiotemporal information fusion, which uses deep learning to detect and extract spatial information, improve detection accuracy, and avoid object position drift, and then, an improved KCF tracking is used to track temporal information so as to avoid missed detection; finally, the spatiotemporal information fusion strategy is designed to make detection information and tracking information complementation. The results show that our proposed algorithm can efficiently continuously detect and track objects in different complex scenes. To a certain extent, it can cope with the influence of the abovementioned environmental interference factors, has both robustness and stable performance. However, the detection and tracking effect with too small scale is slightly worse, so the next step will be to make improvements on it.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] C. Wu, H. Sun, H. Wang et al., "Online multi-object tracking via combining discriminative correlation filters with making decision," *IEEE Access*, vol. 6, pp. 43499–43512, 2018.

[2] Y. Qi, C. Wu, D. Chen, and Y. Lu, Superpixel tracking based on sparse representation," *Journal of Electronics and Information Technology*, vol. 37, no. 3, pp. 529–535, 2015.

[3] G. Yuan and M. Xue, "Visual tracking based on sparse dense structure representation and online robust dictionary learning," *Journal of Electronics & Information Technology*, vol. 37, no. 3, pp. 536–542, 2015.

[4] H. Luo, B.-K. Zhong, and F.-S. Kong, "Tracking using weighted block compressed sensing and location prediction," *Journal of Electronics & Information Technology*, vol. 37, no. 5, pp. 1160–1166, 2015.

[5] Z.-Q. Hou, A.-Q. Huang, W.-S. Yu, and X. Liu, "Visual object tracking method based on local patch model and model update," *Journal of Electronics & Information Technology*, vol. 37, no. 6, pp. 1357–1364, 2015.

[6] M. Xue, H. Zhu, and G.-L. Yuan, "Robust visual tracking based on online discrimination dictionary learning," *Journal of Electronics & Information Technology*, vol. 37, no. 7, pp. 1654–1659, 2015.

[7] L. Matthews, T. Ishikawa, S. Baker et al., "The template update problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 810–815, 2004.

[8] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International Journal of Computer Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[9] 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.

[10] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 1409–1422, 2010.

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

[12] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of the 12th European conference on Computer Vision - Volume Part IV - ECCV 2012*, vol. 75, no. 1, pp. 702–715, Florence, Italy, October 2012.

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

[14] M. Danelljan, G. Hager, F. Khan, and M. Felsberg, "Accurate scale estimation for robust visual tracking," in *Proceedings of the British Machine Vision Conference*, pp. 590–604, BMVA Press, Nottingham, England, September 2014.

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

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

[17] L. Bertinetto, J. Valmadre, S. Golodetz, O. Miksik, and P. H. S. Torr, "Staple: complementary learners for real-time tracking," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1401–1409, Las Vegas, NV, USA, June 2016.

[18] 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.

[19] M. Mueller, N. Smith, and B. Ghanem, "Context-aware correlation filter tracking," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1396–1404, Honolulu, HI, USA, July 2017.

[20] W. Zhong, H. Lu, and M. H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society*, vol. 23, no. 5, pp. 2356–2368, 2014.

[21] Y.-Z. Xue and T. Wang, "Object tracking based on cost-sensitive Adaboost algorithm," *Chinese Journal of Graphic Arts*, vol. 21, no. 5, pp. 544–555, 2016.

[22] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu, "DCFnet: discriminant correlation filters network for visual tracking," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[23] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5000–5008, Honolulu, HI, USA, July 2017.

[24] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance tracking with Siamese region proposal network," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8971–8980, Salt Lake City, UT, USA, June 2018:.

[25] B. Uzkent and Y. W. Seo, "EnKCF: ensemble of kernelized correlation filters for high-speed object tracking," in *Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision*, pp. 77–89, Lake Tahoe, NV, USA, March 2018.