Advances in Computer, Signals and Systems (2023)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2023.070809
ISSN 2371-8838 Vol. 7 Num. 8

Research on video target tracking algorithm based on deep neural network

Chen Rui[1,a,*]

[1]Wenzhou Business College, Wenzhou, Zhejiang, China
[a]390372911@qq.com
[*]Corresponding author

Keywords: Visual object tracking, Real-time tracking, Deep neural network model, Robustness

Abstract: In the field of computer vision applications, visual object tracking is a widely researched and hot-topic area, finding extensive practical applications in many key visual domains and demonstrating promising real-world performance. However, due to various factors such as lighting variations, scale changes, background clutter, low resolution, and other interferences, visual object tracking requires improvements on multiple fronts. In this paper, a video object tracking algorithm based on deep neural networks is proposed while ensuring real-time tracking. Addressing the limitation of traditional visual object tracking algorithms based on correlation filtering theory, which rely on shallow handcrafted features, this algorithm first leverages a deep neural network model to extract deep features of the target to be tracked. Given that different convolutional layers encode different information in their deep feature representations, these distinct layer features are subsequently fused to enhance representation capability. Furthermore, a kernel correlation-based approach is employed to boost the tracking speed of the visual object tracking algorithm. The experimental results demonstrate that the method proposed in this paper achieves a balance between target tracking accuracy and speed, enhancing the robustness of visual object tracking algorithms in complex and noisy backgrounds.

1. Introduction

In recent years, with the deep implementation of innovation-driven strategies, a new generation of information technologies represented by cloud computing, big data, the Internet of Things, and artificial intelligence has been advancing rapidly. This progress has significantly propelled the continuous application of computer vision technology. Many technology-driven and innovative unicorn companies have emerged, driving the rapid development of computer software and hardware technology[1]. This growth has been supported by increasing computational power, ongoing algorithm enhancements, growing datasets, and expanding applications. These factors have provided substantial technological, human resource, and financial support for the rapid development of computer vision technology. Among these advancements, visual object tracking technology, as an important branch of computer vision, has achieved notable results in various practical application scenarios. For example, it enables intelligent monitoring in the field of surveillance, enhances the efficiency and convenience of human-computer interaction, and provides immersive experiences in the realm of virtual reality[2]. As a result, it has garnered widespread attention and research interest from the scientific community and is gradually becoming an emerging applied technology.

In 2016, Nam et al. introduced a multi-domain learning framework based on Convolutional Neural Networks (CNNs) called MDNet. They employed the Multi-Domain Network to extract sequence-independent information features. The shared layers at the front end extracted general target features using a large amount of annotated bounding box videos. Different target categories corresponded to respective binary classification tasks, facilitating online training and achieving very high recognition speed[3]. However, this approach was not suitable for real-time tracking. To address the high computational complexity of the MDNet algorithm and the lack of optimization for potential targets, Lchae et al. proposed a tracking algorithm based on MDNet and Fast R-CNN (Fast Region Convolutional Neural Network). They utilized Mask-RCNN's ROIAlign to obtain more accurate positional features, improving localization accuracy. They also introduced a multi-task loss term in the loss function to enable the model to more effectively discriminate between targets in different domains. This method achieved a 25-fold increase in tracking speed while maintaining the same accuracy as MDNet[4]. Yun et al. introduced an action-driven mechanism, which involved continuous sampling through different actions to ultimately locate the target. They used the action-driven mechanism to capture the motion information of the target object, ensuring the prioritized search for higher-quality candidate samples. They employed reinforcement learning on a partially labeled dataset to obtain an action-driven model, avoiding the issues of dense sampling in previous tracking algorithms and reducing annotation requirements for training data. Fan et al. utilized multiple Recurrent Neural Networks (RNNs) at different semantic levels to perform self-structure information modeling for target objects. This approach enhanced the discriminative power of the algorithm by distinguishing target objects from similar entities. They fused the features from multiple CNN and RNN layers, resulting in improved algorithm accuracy. Their method achieved high accuracy on the OTB100 dataset[5].

This paper proposes a novel video object tracking algorithm based on deep neural network technology. In comparison to traditional visual object tracking methods, this algorithm overcomes reliance on shallow handcrafted features by first extracting deep features of the target using a deep neural network model. Recognizing that different convolutional layers encode different information in these features, we fuse these multi-layer features to enhance the representation capability of the target. Additionally, we employ kernel-based correlation methods to improve the tracking speed of the visual object tracking algorithm. Experimental results demonstrate that our proposed approach strikes a balance between tracking accuracy and speed, enhancing the robustness of the visual object tracking algorithm in complex and noisy backgrounds.

2. ResNet convolutional neural network model

The ResNet network model, introduced in 2015, brought innovation to the field of deep learning by introducing the concept of residuals. It was the first to propose the idea of adding the output of a convolutional layer, denoted as $F(x)$, to the input of that layer, transforming it into the function $F(x) + x$. This innovation aimed to address the gradient vanishing problem that arises as neural networks become deeper. The network structure is depicted in the following diagram (Figure 1).

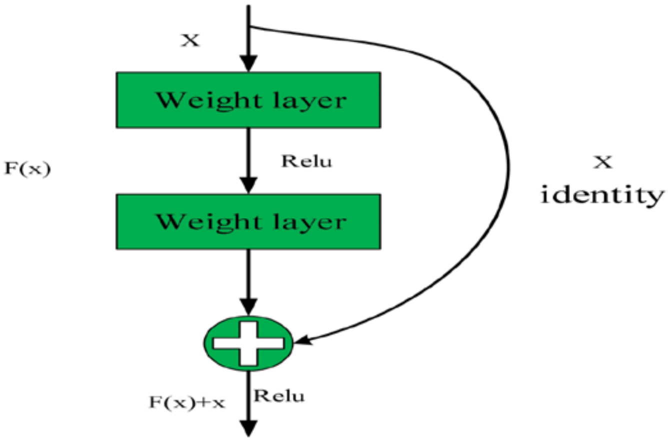

Figure 1: ResNet network structure.

3. Kernel correlation filtering algorithm based on multi-layer deep feature fusion

3.1. The depth features were extracted with pre-trained ResNet

The ResNet-50 deep convolutional neural network model has demonstrated excellent performance in various practical applications of object recognition. Therefore, in this paper, we adopt this network model structure, as shown in Figure 2. We pretrained this network model structure on the large-scale ImageNet dataset. After training, we transfer this model structure to the field of visual object tracking and utilize this pretrained network model for deep feature extraction in the context of object tracking. Typically, deep convolutional neural networks are applied in the domain of object recognition, where the ultimate goal is to classify and differentiate various objects in images [6]. In the process of object recognition, deep features from the final output layer of the convolutional neural network are commonly sampled to represent the encoded information of the target. These features obtained in the final layer are particularly conducive to the classification of different object categories. In contrast, in the field of visual object tracking, the objective is to accurately locate a tracked object in subsequent video frames. When using the output features from the last layer of a convolutional neural network to represent the encoded information of the target for tracking, there may be some bias, resulting in only marginal improvements in tracking performance. Given that there are significant differences between the deep convolutional features utilized in object recognition and those used in object tracking, it is crucial to consider a different approach. Unlike object recognition, when extracting deep features for tracking targets using deep convolutional neural networks in the field of visual object tracking, it is essential not only to rely on the deep output features from the final layer to handle various deformations of the tracked target but also to assign greater importance to the features obtained from earlier and intermediate convolutional layers for precise tracking target localization. This is because, during the convolutional process, the early convolutional layers offer higher-resolution target feature maps with richer original spatial information, which is advantageous for accurate target localization [7].

In the later convolutional layers, the resolution of the feature maps obtained for tracking targets is lower, but they contain richer semantic information, which is advantageous for distinguishing between different categories of targets. This approach ensures that the output features from each convolutional layer can all play a role, ultimately leading to significant improvements in both target tracking accuracy and speed. Following this design philosophy, this paper extracts deep features from the conv2-3, conv3-4, conv4-6, and conv5-3 layers of the ResNet-50 network model, as well as the final output features of this network model. These selected layers are used to adaptively learn corresponding correlation filter templates. Once these correlation filter templates are obtained, they are applied to the respective convolutional layer features through correlation filtering operations to generate target position response maps for each convolutional layer. Finally, multiple response maps are fused to determine the precise location of the tracked target[8]. However, due to the pooling (downsampling) operations within convolutional neural networks, as the depth of convolutional layers increases, the spatial resolution of the extracted deep feature maps gradually decreases. The resulting target feature maps have smaller dimensions, making it difficult to achieve accurate target localization. Therefore, a bilinear interpolation strategy is employed to upsample the extracted target depth feature maps, enlarging the dimensions of the target feature maps. This strategy is particularly beneficial for improving the accuracy of target localization during tracking.

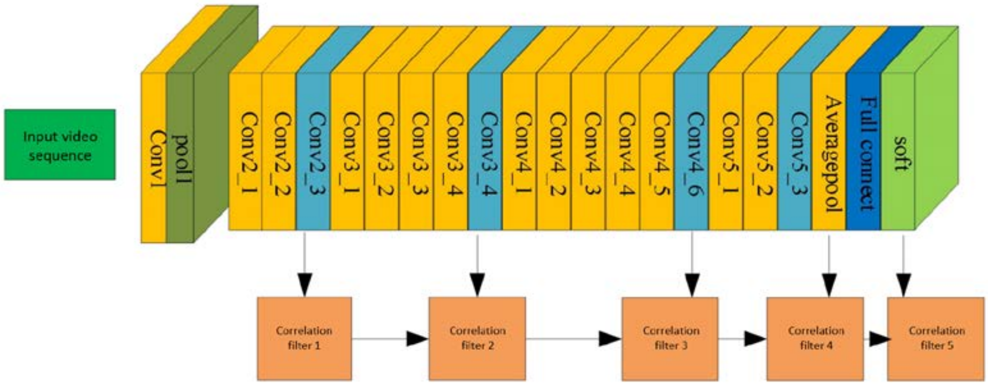

Figure 2: Network model structure for extracting depth features.

3.2. Kernel correlation filtering combined with depth features

In general, when it comes to visual object tracking, a commonly employed approach is to use a discriminative object tracking classifier based on correlation filtering. This classifier is trained to distinguish between the tracked target and the background. It involves applying filter templates to regions in subsequent video frames through correlation filtering operations, resulting in a feature response map[9]. On this response map, searching for the maximum value of the correlation filter response corresponds to the estimated location of the tracked target, thereby providing an estimate of the exact target position. In the algorithm proposed in this paper, different convolutional layers and the final layer's output are utilized to represent the target. The deep feature extraction network used in this paper has been pretrained on other large-scale datasets. To adapt this network for target feature extraction in the context of visual object tracking, it needs to be fine-tuned using tracking target samples. However, in the field of visual object tracking, only the first frame of a video can serve as a training sample, leading to a scarcity of training samples. To address this, the output features of each convolutional layer are upsampled, and the upsampled deep feature maps are subjected to cyclic shifts to generate additional tracking target samples for training the corresponding correlation filter templates. Each cyclically shifted sample is associated with a Gaussian function as a regression label, following a two-dimensional Gaussian distribution. This approach effectively reduces the problem of sample blurriness. Ultimately, the filter template 'w' of the same size as the upsampled feature maps is learned by minimizing the following expression.

$$w^{*} = \arg\min \sum_{m,n} \| w \bullet x_{m,n} - y(m,n) \|^{2} + \lambda \| w \|_{2}^{2} \tag{1}$$

Where $\lambda \geq 0$ is the regularization parameter, and *y(m,n)* represents the label of the image pixel at position *(m,n)*, following a two-dimensional Gaussian distribution. Since time-domain convolution can be computationally slow, the above objective function can be transformed into the frequency domain using fast Fourier transform for accelerated computation. The optimal solution for solving the filter is obtained as the minimum of the objective function:

$$W^d = \frac{Y \odot \bar{X}^d}{\sum X^i \odot \bar{X}^i + \lambda} \tag{2}$$

The expression represents the Fourier transform of *y(m, n)*, $\bar{X}$ represents the complex conjugate of *X*, and $\odot$ represents element-wise multiplication. Using equation (2), we can obtain the respective templates for the correlation filters in each convolutional layer. Once the correlation filter templates for each convolutional layer are obtained, given a candidate region in the next frame of a video image, the depth features *S* in that region can be extracted. The response map for the l-th convolutional layer after correlation filtering can be calculated using the following equation:

$$f_1 = F^{-1}\left(\sum_{d=1}^{D} W^d \odot \bar{S}^d\right) \tag{3}$$

Where F^{-1} represents the inverse Fourier transform, the maximum value of the correlation response map f_l corresponds to the exact location of the target to be tracked.

3.3. Accurately estimate the target position

In this study, we utilize the output from various convolutional layers of the ResNet-50 deep convolutional neural network model, such as conv2-3, conv3-4, conv4-6, conv5-3, as well as the final output features, to serve as the depth features for tracking the target. For each layer, an independent set of correlation filters is constructed, and the feature maps obtained from each layer are convolved with their respective filters to generate a collection of corresponding correlation filter response maps. When provided with a set of correlation filter response feature maps, layer-wise inference is required to determine the target's position within each convolutional layer's feature map. We use $(\hat{m}, \hat{n}) = \arg\max_{m,n} f_l(m,n)$ to represent the optimal position of the target in the previous convolutional layer, which can be obtained by maximizing the following equation:

$$\arg\max_{m,n} f_{l-1}(m,n) + \eta f_l(m,n) \tag{4}$$

Here, η represents a regularization parameter, and the response from the subsequent layer influences the response from the previous layer. In order to infer and search for the target's position within the previous convolutional layer, it is necessary to consider the vicinity of the maximum response in the subsequent convolutional layer. This involves weighting the response from the subsequent layer by the regularization parameter η and then backpropagating it to the previous layer. This mechanism is used to affect the correlation response mapping in the previous layer.

3.4. Correlation filter update

During the target tracking process, the target being tracked often undergoes significant deformations. To adapt the learned filter templates to these variations in the target, it is necessary to update the filters. In order to obtain a more robust approximation, we use the following equations to update the numerator A and denominator B in equation (2) for the filter W.

$$A_t = (1-\eta)A_{t-1} + \eta Y \odot \bar{X}_t \tag{5}$$

$$B_t = (1-\eta)B_{t-1} + \eta \sum X_t \odot \bar{X}_t \tag{6}$$

$$W_t = \frac{A_t}{B_t + \lambda} \tag{7}$$

Here, t represents the index of video frames, η represents the learning rate, and as the tracked target continues to change, the filter templates are constantly updated, further enhancing the accuracy of target tracking.

4. Experimental results and analysis

4.1. Experimental parameter design

The current experiment was conducted on the MatlabR2014b platform, utilizing a GeForce GT730 graphics card. The parameter λ in (1) was set to 10^-5, and the learning rate in (5) was configured as 0.001. We employed a convolutional neural network model pretrained on the ImageNet dataset and then transferred this trained model to the domain of visual object tracking. Using this model, we extracted depth features from the target to be tracked and subsequently fused the output features from each layer.

4.2. Performance evaluation criteria

The experiments were primarily conducted using video sequences from OTB50 and OTB100 datasets, following the relevant parameters and settings outlined in the OTB50 experiments. The selected video sequences encompass various interference factors commonly encountered during the visual object tracking process. These interference factors serve as a robustness measure for evaluating the proposed algorithm [10]. Additionally, mainstream tracking evaluation standards were employed to assess the accuracy and speed of the proposed algorithm. A series of tracking examples were presented to showcase the algorithm's tracking performance. Specifically, precision plot refers to the percentage of video frames in which the center position error is less than 20 pixels relative to the total number of frames in the video sequence. Success plot, on the other hand, represents the percentage of successfully tracked video frames, where success is defined by an overlap ratio exceeding a certain threshold, relative to the total number of frames in the video sequence.

1) Center position error

The center position error is defined as the average Euclidean distance, measured in pixels, between the tracked center position obtained by the proposed tracking algorithm ($c_t(x)$, $c_t(y)$) and the manually annotated true center position ($c_g(x)$, $c_g(y)$) in the first frame of the video sequence. When this distance falls below a certain threshold, typically set at 20 pixels, the target tracking is considered successful. This threshold is commonly used for comparing the tracking performance of different algorithms [11]. The Euclidean distance is defined as $D = \sqrt{(c_g(x) - c_t(x))^2 + (c_g(y) - c_t(y))^2}$.

2) Overlap rate

Overlap ratio refers to the ratio of the intersection between the target region B_T obtained by the tracking algorithm and the ground truth target region B_G, divided by their union. When this ratio exceeds a certain threshold, it indicates a successful frame in the video target tracking. Typically, the area under the success plot curve, which represents the percentage of successfully tracked frames, is used as an alternative measure to evaluate the performance of target tracking algorithms [12].

4.3. Quantitative comparative analysis

To further validate the robustness of the target tracking algorithm proposed in this study, we compared it with several other algorithms using precision plots and success plots. These comparative algorithms include MOSSE, KCF, SRDCF, BACF, CCOT_HOG, ECO_DEEP, STRCF, and DCF. The algorithm proposed in this paper replaces traditional shallow handcrafted features with deep features extracted using a deep convolutional neural network to represent the encoding information of the tracked target. Simultaneously, it performs multi-level fusion of the obtained deep features from various convolutional layers and the final output features, enriching the representation of the target. As a result, it significantly improves visual object tracking accuracy compared to other tracking algorithms. The precision plots and success plots for the final visual object tracking results are shown in Figure 3.

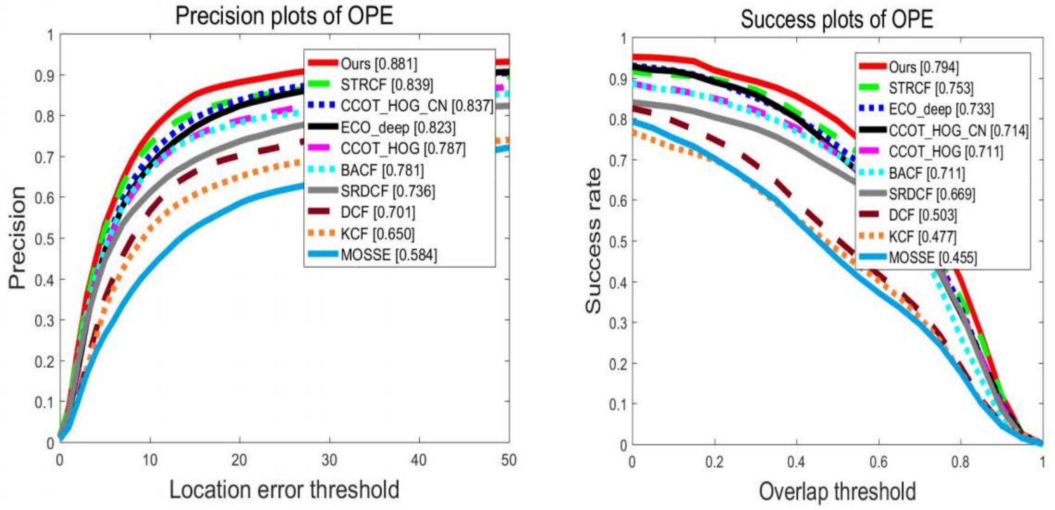

Figure 3: Tracking algorithm in OTB100 video sequence tracking comparison graph

From Figure 3, it can be observed that the improved target tracking algorithm in this study achieves an AUC score of 0.794 on the success rate curve and a score of 0.881 on the precision curve. In terms of tracking accuracy, success rate, and robustness, the proposed algorithm exhibits excellent tracking performance. Compared to the high-speed correlation filter tracking algorithm KCF, the proposed algorithm shows a 23.1% improvement in precision and a 31.7% improvement in success rate. In comparison to the SRDCF algorithm, which addresses boundary effects, the proposed algorithm demonstrates a 14.5% improvement in precision and a 12.5% improvement in success rate. Overall, the target tracking algorithm presented in this paper outperforms other algorithms in both precision and success rate plots, achieving superior tracking results.

Table 1: The average operation speed of different tracking algorithms

Ours	ECO	SRDCF	DCF	KCF	Mosse	STRCF
89	24	31	323	424	545	45

Table 1 presents the average processing speed of different visual object tracking algorithms, measured in FPS (Frames Per Second). During the visual object tracking process, it is generally considered that achieving 25 FPS (tracking 25 frames of video image sequences per second) is sufficient for real-time tracking. Therefore, the target tracking algorithm proposed in this paper meets real-time requirements. However, compared to high-speed algorithms like KCF and MOSSE, which are based on correlation filters, the algorithm presented in this paper utilizes deep features. While it enhances tracking accuracy, it is slower in terms of speed due to the computationally intensive nature of convolution operations, making it relatively more time-consuming than other correlation filter-based algorithms.

4.4. Attribute comparative analysis

To further validate the robustness of the algorithm proposed in this paper to various interfering factors in tracking video sequences, the following comparative performance chart is provided with respect to other algorithms when dealing with different interference factors. From the chart below, it can be observed that the target tracking algorithm presented in this paper exhibits good robustness to various interference factors and can effectively address common issues that arise during video sequence tracking.

1) Deformation

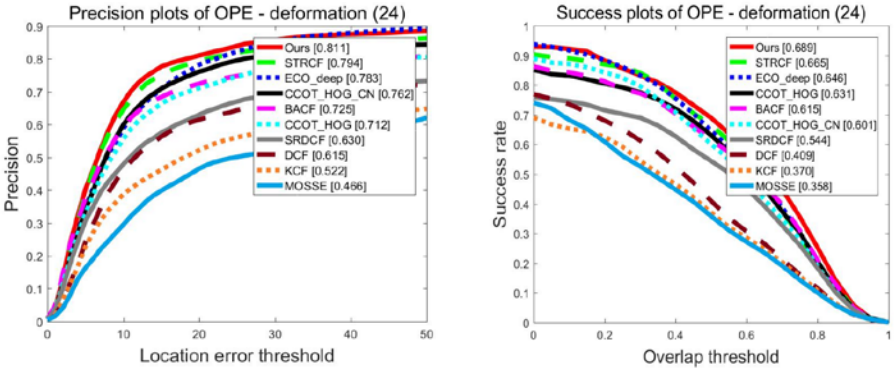

Figure 4: Video sequence tracking graph with deformation attribute

From the chart (Figure 4) above, it is evident that the algorithm proposed in this paper exhibits superior tracking performance on video sequences with deformation attributes compared to other algorithms. In particular, when compared to the STRCF algorithm developed by Harbin Institute of Technology, our algorithm demonstrates a 1.7% improvement in precision and a 2.4% improvement in success rate. It can successfully track targets even in cases of occlusion and is capable of adapting well to significant appearance changes in the tracked target.

2) Illumination change

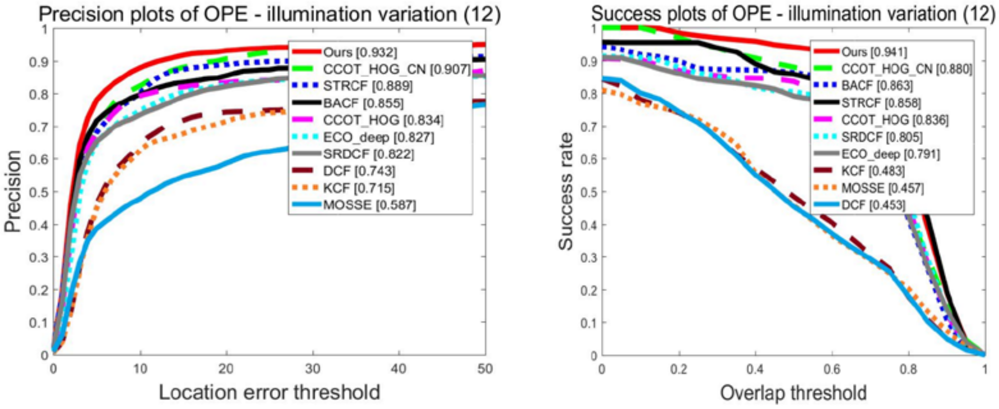

Figure 5: Video sequence tracking graph with illumination change attribute

From the chart (Figure 5) above, it is evident that the algorithm proposed in this paper exhibits excellent tracking performance on video sequences with lighting variations. It achieves accuracy and success rates exceeding 90% in both precision and success rate plots. In terms of precision, it outperforms the CCOT_HOG_CN algorithm by 2.5%, and in terms of success rate, it demonstrates a 6.1% improvement.

3) Rapid motion

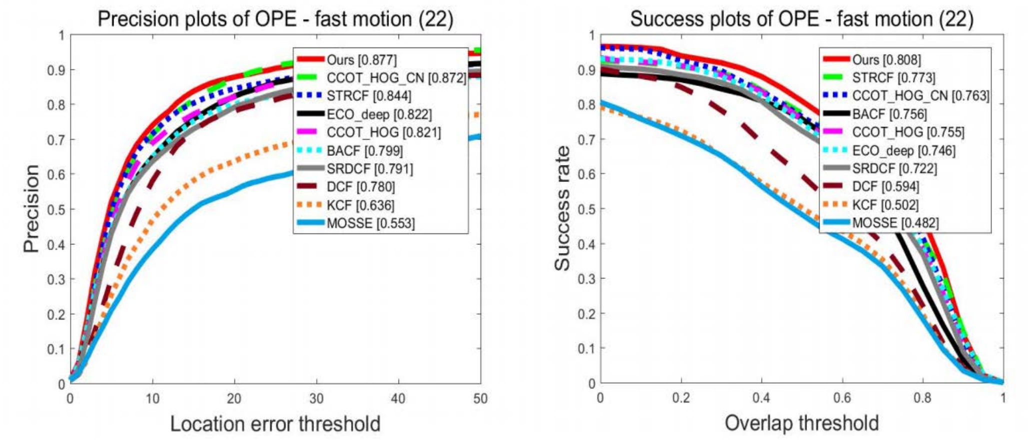

Figure 6: Video sequence tracking graph with illumination change attribute

From the chart (Figure 6) above, it is evident that the algorithm proposed in this paper exhibits superior tracking performance on video sequences with fast motion attributes. It outperforms the ECO_deep algorithm by 5.5% in terms of precision and demonstrates a 6.2% improvement in success rate.

4) Motion blur

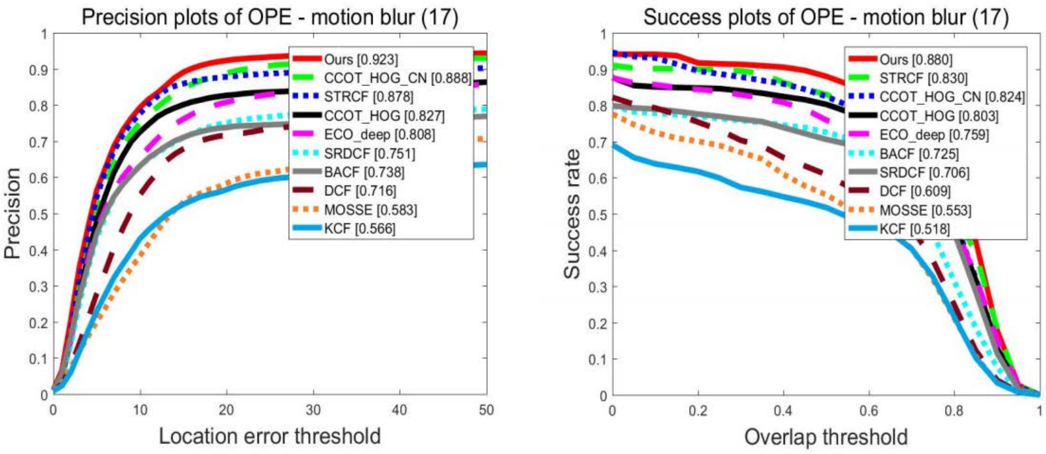

Figure 7: Video sequence tracking graph with motion fuzzy attribute

From the chart (Figure 7) above, it can be observed that the algorithm proposed in this paper demonstrates excellent tracking performance on video sequences with motion blur attributes. It outperforms the CCOT_HOG_CN algorithm by 4.4% in terms of precision and achieves a 5% improvement in success rate compared to the STRCF algorithm developed by Harbin Institute of Technology.

4.5. Qualitative comparative analysis

To demonstrate the tracking performance of the algorithm proposed in this paper, a set of video sequences with significant interference factors were selected from visual object tracking datasets such as OTB50 and OTB100. These sequences were used to evaluate the tracking performance of our algorithm. Simultaneously, the tracking results were compared with those of different tracking algorithms. The tracking results are depicted in Figure 8.

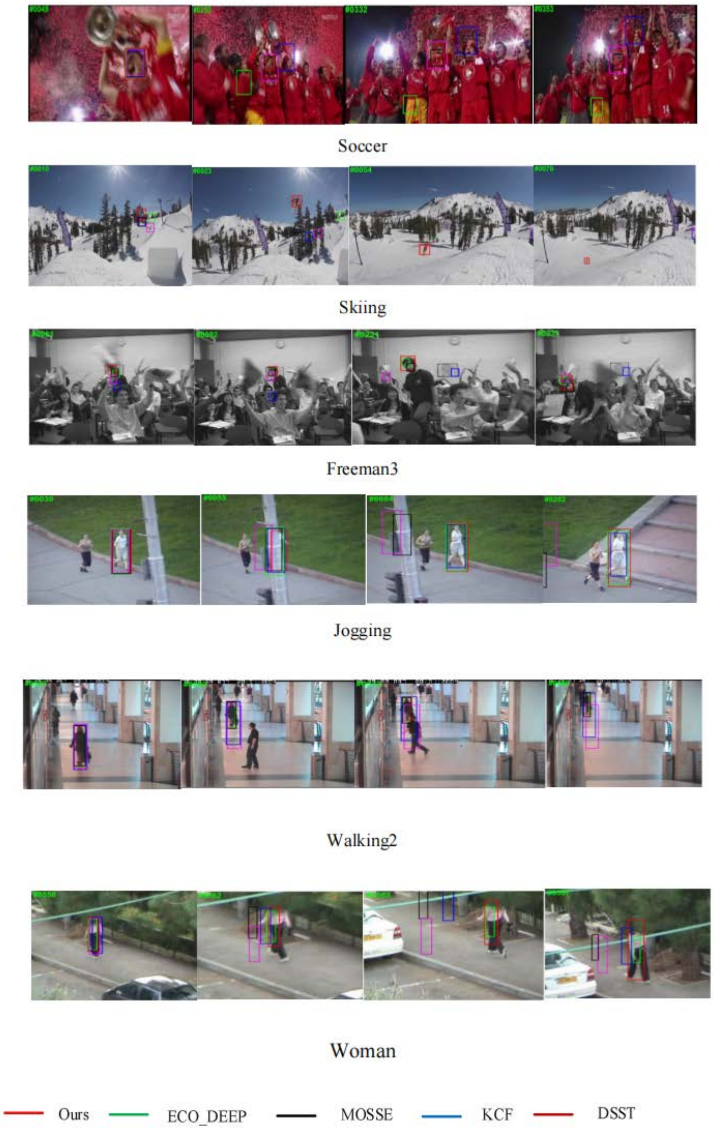

Figure 8: Algorithms track results in different video sequences

From Figure 8, it is evident that the algorithm proposed in this paper achieves precise tracking of the target by leveraging deep features, ensuring the algorithm's efficiency. In the "skiing" video sequence, where the target undergoes rapid motion, rotation, scale variations, and abrupt changes in motion direction, significant interference is introduced for the tracking task. Only the algorithm presented in this paper manages to successfully track the target continuously, while other algorithms fail in tracking. In the "soccer" video sequence, where interference factors include background clutter, rapid deformations, motion blur, and out-of-plane rotation, the tracking performance of our algorithm surpasses that of the other four algorithms.

5. Conclusions

Traditional visual object tracking algorithms often rely on shallow handcrafted features, which may not effectively represent the characteristics of the target being tracked. This directly impacts the accuracy and precision of visual object tracking. Therefore, this paper addresses the limitations of traditional handcrafted features in kernel-based correlation filter visual object tracking algorithms. Within the framework of kernel correlation filtering principles and with a focus on maintaining real-time performance, we automatically extract deep convolutional features from a pre-trained deep convolutional neural network to replace the deficiencies of traditional shallow handcrafted features. Subsequently, these deep convolutional features extracted from different convolutional layers undergo learning via kernel correlation filters to obtain distinct response maps. Finally, multiple response maps are weighted and fused to accurately estimate the precise location of the tracked target. This approach allows the tracking algorithm to strike a balance between tracking accuracy and speed, enhancing the robustness of visual object tracking algorithms in complex and interference-rich backgrounds.

References

[1] Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters [J]. In Proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 37(3):583-596.

[2] Bhat G, Johnander J, Danelljan M, et al. Unveiling the Power of Deep Tracking [J]. In IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 36(7):1442-1468.

[3] Wu Y, Lim J, Yang M H.Object tracking benchamark [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 37(9): 1834-1848.

[4] Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. Foundations & Trends in Machine Learning, 2019, 3(1): 1-3.

[5] Wang Q, Zhang L, Bertinetto L, et al. Fast Online Object Tracking and Segmentation: A Unifying Approach [J]. 2018, 50(8):11-13.

[6] He K M, Zhang X Y, Re S Q, et al. Deep residual learning for image recognition [J]. Proc of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, 72(14):770-774.

[7] Henriques J F, Caseiro R, Martins P, et al. High-speed tracking with kernelized correlation filters [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3):583-596.

[8] Avidan S. Support vector tracking [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(8): 1064-1072.

[9] Ross D A, Lim J, Lin R S, et al. Incremental learning for robust visual tracking[J]. International Journal of Computer Vision, 2019, 77(1): 125-141.

[10] Hua Y. Occlusion and motion reasoning for long-term tracking[C]// European Conference on Computer Vision. 2020: 172-187.

[11] Ma C, Yang X K, Zhang C Y, et al. Long-term correlation tracking[C]// Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2015: 5388-5396.

[12] Li B, Wu W, Wang Q, et al. Siam RPN++: Evolution of Siamese Visual Tracking with Very Deep Networks [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 28(8): 10-13.