

Article

Fusion of Multiple Attention Mechanisms and Background Feature Adaptive Update Strategies in Siamese Networks for Single-Object Tracking

Wenliang Feng ^{1,2}, Fanbao Meng ^{1,*}, Chuan Yu ¹ and Anqing You ¹

¹ Institute of Applied Electronics, Chinese Academy of Engineering Physics, Mianyang 621025, China; m18784035322_1@163.com (W.F.); yuchuan@263.com (C.Y.); anqingyou@163.com (A.Y.)

² Graduate School, China Academy of Engineering Physics, Mianyang 621999, China

* Correspondence: fanbao_meng@126.com

Abstract: Single-object tracking algorithms based on Siamese full convolutional networks have attracted much attention from researchers owing to their improvement in precision and speed. Since this tracking model only learns a similarity model offline, it is not able to obtain more useful feature discrimination information to adapt to the various variations of targets in complex scenes. To improve the performance of this tracking model, we propose a Siamese network tracking algorithm that incorporates multiple attention mechanisms and an adaptive updating strategy for background features. First, a backbone feature extraction network is proposed that utilizes a small convolutional kernel to fuse jump-layer connectivity features, thereby improving the feature representation capability of the network. Second, an adaptive update strategy for background features is proposed to improve the model's ability to discriminate between the object and background features. Third, the fusion of multiple attention mechanisms is proposed so that the model learns to focus on the channel, spatial, and coordinate features. Fourth, the response fusion operation is proposed after the inter-correlation operation to enrich the output response of the model. Finally, our algorithm is trained using the GOT-10K dataset and evaluated by testing on the object tracking benchmark datasets OTB100 and VOT2018. The test results show that compared with other algorithms, our algorithm can effectively cope with the problem of degradation of the tracking performance in complex environments, and it can further improve the tracking precision and precision under the premise of ensuring the tracking speed.

Keywords: fusion of multiple attention mechanisms; background adaptive updating strategy; Siamese network; single-object tracking algorithm



Citation: Feng, W.; Meng, F.; Yu, C.; You, A. Fusion of Multiple Attention Mechanisms and Background Feature Adaptive Update Strategies in Siamese Networks for Single-Object Tracking. *Appl. Sci.* **2024**, *14*, 8199. <https://doi.org/10.3390/app14188199>

Academic Editor: Pedro Couto

Received: 6 August 2024

Revised: 30 August 2024

Accepted: 11 September 2024

Published: 12 September 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid development of computer vision technology, visual object tracking has become a research direction with important applications value in the field of computer vision. Visual object tracking has a wide range of application scenarios that can be applied to human–computer interaction, video surveillance, urban transportation, automatic driving, and other fields, offering considerable convenience in our daily lives [1–4]. The number of tracking objects can be divided into single-object tracking and multi-object tracking. Visual single-object tracking is studied, and its process is as follows: First, the tracking frame of the object to be tracked is given in the first frame of the image of the video sequence, and then the position and size of the tracked object are automatically given in the subsequent frames of the video sequence, as shown in Figure 1. After many years of dedicated research and accumulation by many researchers, although single-object tracking algorithms have made some progress, they still face many challenges in complex scenarios, such as changes in the shape and size of the object, changes in the ambient illumination in which the object is located, similar background interference, and object occlusion [5,6]. All of these uncontrolled situations can degrade the algorithm's tracking performance, which can lead

to tracking drift and failure. Simultaneously, single-object tracking algorithms have high requirements for robustness, precision, precision, and speed in practical applications.

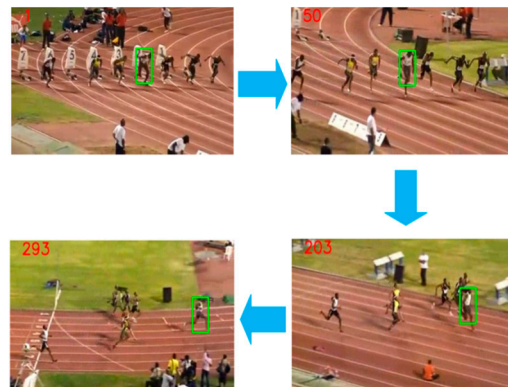


Figure 1. Effect diagram of visual single-object tracking.

Visual single-object tracking algorithms can be categorized into generative, correlation, and deep learning-based object tracking algorithms. The visual single-object tracking algorithm based on the generative model [7–10] uses the current frame to model the object region and searches for the region that is most similar to the model in the next frame to predict the object's position. Although it can effectively deal with the situation of object loss during the tracking process, it ignores the relationship between the background and the object, which leads to the degradation of the tracking precision in the face of a complex and changeable scene. A visual single-object tracking algorithm based on a correlation filter [11–17] aims to learn a correlation filter and obtain a dense response map by performing a correlation operation on the image, where the position of the maximum value in the response map is the predicted position of the object. Although it has achieved good results in the field of object tracking, owing to the lack of consideration for the changes in the size of the object's appearance, it is easily disturbed by external environments when confronted with occlusion. However, it has achieved good results in the field of object tracking, owing to the lack of consideration for the change of the object's appearance. When facing complex scenes such as occlusion, deformation, etc., it is easily interfered with by the external environment, which leads to a reduction in tracking precision and success rate. Compared with the above two types of tracking algorithms, the deep learning-based visual single-object tracking algorithm has greatly improved tracking performance and has become the mainstream object tracking algorithm nowadays.

Visual single-object tracking may require tracking arbitrary objects in nature; thus, it is not possible to collect data from all objects and train a tracker suitable for all objects in the research of single-object tracking algorithms. With the proposal of Siamese networks [18] and their successful application in computer vision tasks such as face recognition and image matching [19], scholars have proposed Siamese network-based visual single-object tracking algorithms [20–26] and attracted great attention in the field of object tracking owing to their faster speed and higher tracking performance. A single-object tracking algorithm based on Siamese fully convolutional networks uses an offline dataset to train a similarity matching model, thus transforming the object tracking problem into a similarity matching problem. Although the single-object tracking algorithm based on Siamese networks has achieved good tracking results, it also has high precision and speed and can be applied to some specific scenes. However, since the model uses the object features provided in the first frame of the image as the baseline template throughout the tracking process, the baseline template is not updated, causing the model to be unable to obtain more information conducive to feature discrimination, which leads to the model's inability to adapt to the various changes of the target in the complex environment. The backbone feature extraction network of the model uses AlexNet [27], which can cause the model to fail to extract the deep semantic features of the object because of the small number of layers

in the feature extraction layer of AlexNet. Therefore, the algorithm suffers from tracking performance degradation and tracking failure when facing various challenges in complex scenes. To solve the problems of tracking performance degradation and tracking failure of the Siamese full convolutional network-based single-object tracking algorithm in complex scenes for the above reasons, and at the same time, in order to further improve the tracking precision and precision of the algorithm so that the performance of the algorithm meets the requirements of practical applications, we propose a single-object tracking algorithm for Siamese networks that integrates a multiple-attention mechanism and an adaptive updating strategy of the background features. The structure of the algorithm is illustrated in Figure 2.

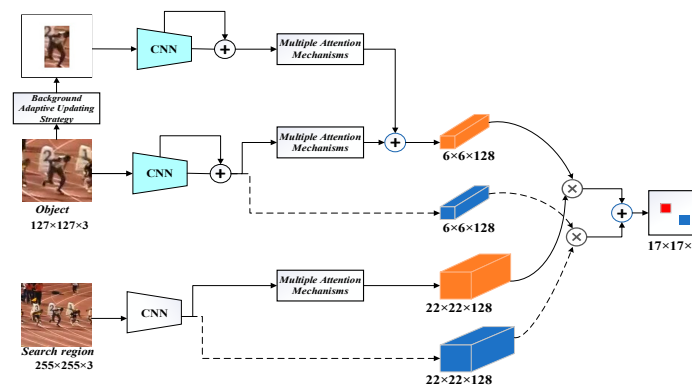


Figure 2. Structure of Siamese network single-object tracking algorithm combining a multiple attention mechanism and background feature adaptive updating strategy.

The framework of our tracking algorithm is shown in Figure 2. It contains a goal branch and a search branch. The target branch consists of a main branch and a background adaptive update branch; each of them contains a small convolutional and hopping layer connected feature fusion network and multiple attention modules. However, the search branch only uses a small convolutional kernel network and multiple attention modules. Finally, the target branch and the search branch output a response map through a convolutional inter-correlation operation.

In this work, we propose a new tracking algorithm based on the Siamese network framework. The main contributions are as follows:

- (1) We designed a backbone feature extraction network with a small convolutional kernel and hopping layer connection feature fusion. It can not only effectively extract the deep semantic features of the object, but also fuse the mid-level features with the deep features through the hopping layer connection, which is conducive to the enhancement of the feature expression ability of the network. Meanwhile, utilizing 3×3 and 1×1 small convolution kernels for convolution operation can increase the depth of the network while reducing the network parameters.
- (2) A background feature adaptive updating strategy was proposed in our algorithm and applied to the processing of the object template. First, after the object template is processed through the background feature adaptive updating strategy, it highlights the object features and weakens the background features so that the object features form a sharp contrast with the background features, so that the algorithm can fully distinguish the object and background features in the object template. Then, the object features obtained after processing through the background feature adaptive updating strategy, and then through the backbone feature extraction network and multiple attention mechanisms, are linearly fused with another object feature that has not been processed through the background feature adaptive updating strategy. Then through the backbone feature extraction network and multiple attention mechanisms, the algorithm's ability to discriminate between the object features and the background features can be effectively enhanced.

- (3) The fusion of multiple attention mechanisms is proposed in our algorithm, which consists of an improved hybrid attention mechanism connected in parallel with the CA attention mechanism. Applying it to the object template branch and the search region branch enables the algorithm to locate and recognize the tracking object more precisely, and makes the algorithm adaptively learn to focus on the channel, spatial, and coordinate feature information of the tracking object, thus improving the attention ability and precision of the algorithm.
- (4) In our algorithm, we propose a response fusion operation, a mutual correlation operation, where the original object features are convolved with the original search region features, and the object features after multiple attention mechanisms are convolved with the search region features after multiple attention mechanisms, and the results of the two convolutional mutual correlations are linearly fused to obtain the final response map in order to enrich the output response of the algorithm.

Our algorithm is compared with other Siamese network framework tracking algorithms, such as SiamFC [20], SiamRPN [22], etc. (1) SiamFC and SiamRPN use shallow AlexNet for their backbone feature extraction network, while our algorithm uses a deep backbone feature extraction network designed by us. (2) SiamFC and SiamRPN do not use any other strategies to distinguish between object and background features, while our algorithm uses our designed adaptive updating strategy for background features to distinguish between object and background features. (3) SiamFC and SiamRPN do not use any attention mechanism to enhance the algorithm's attention to object features, while our algorithm uses a fusion of multiple attention mechanisms designed by us to enhance the algorithm's attention to object features. (4) The internal structure of SiamFC and SiamRPN is double-branching without feature fusion, while the internal structure of our algorithm employs multi-branching feature fusion. (5) The output response of SiamFC and SiamRPN is a single response, while our algorithm employs a response fusion operation after correlating with each other, which enriches the algorithm's output response and ensures that the algorithm's output results are not degraded.

The remainder of this paper is organized as follows. Section 1 introduces the related algorithms and the proposed algorithm. Section 2 presents research work on the proposed algorithm. Section 3 presents the training and testing of the proposed algorithm. Section 4 presents the test results for the proposed algorithm. Section 5 concludes the paper.

2. Research Work

This section belongs to the research work of our algorithm and can be divided into four parts. Section 2.1 is the Siamese network framework; Section 2.2 is an adaptive updating strategy for background features; Section 2.3 is a fusion of multiple attentional mechanisms; and Section 2.4 is a response fusion operation after convolutional inter-correlation.

2.1. Siamese Network Framework

The Siamese network framework of our tracking algorithm is shown in Figure 2. It consists of two asymmetric branches: the target branch and the search branch. In the search branch, we design and apply a small convolutional backbone feature extraction network for feature extraction. In the target branch, we design and apply a backbone feature extraction network that uses a combination of small convolutional kernels and jump-layer connections for feature extraction.

First, we will design a small convolutional kernel backbone feature extraction network and apply it to the search branch. Given the wide range of applications of convolutional neural networks in computer vision [28,29], the newly designed backbone feature extraction network consists of 14 convolutional layers, 3 maximum pooling layers, 13 batch normalization layers, and 4 activation layers. We define convolutional layers 1, 2, and 3 plus 3 normalization layers plus 1 activation layer plus 1 maximum pooling layer as conv1; convolutional layers 4, 5, and 6 plus 3 normalization layers plus 1 activation layer plus 1 maximum pooling layer as conv2; convolutional layers 7, 8, and 9 plus 3 normalization

layers plus 1 activation layer plus 1 maximum pooling layer as conv3; convolutional layers 10, 11, 12, 13 plus 3 normalization layers plus 1 activation layer is defined as conv4; convolution layer 15 is defined as conv5; the specific information of the backbone feature extraction network is shown in Table 1. Compared with AlexNet, the newly designed backbone feature extraction network uses a 1×1 small convolution kernel to downsize the network channel several times, which not only reduces the number of parameters of the network, but also ensures the computational speed while increasing the depth of the network, so that the network can further obtain the deep features of the target. In addition, using a 1×1 small convolutional kernel can increase the nonlinearity, mix the cross-channel information, and improve the generalization ability of the network.

Table 1. Backbone features extract network specific information.

Definition Layer	Layer	Kernel Size	Stride/Channel	Template	Search
Input			/3	127×127	255×255
conv1	CONV1-BN	3×3	1/64	125×125	253×253
	CONV2-BN	3×3	1/128	123×123	251×251
	CONV3-BN-ReLu	1×1	1/64	123×123	251×251
	MaxPool	2×2	2/64	61×61	125×125
conv2	CONV4-BN	3×3	1/128	59×59	123×123
	CONV5-BN	1×1	1/64	59×59	123×123
	CONV6-BN-ReLu	3×3	1/128	57×57	121×121
	MaxPool	2×2	2/128	28×28	60×60
conv3	CONV7-BN	3×3	1/256	26×26	58×58
	CONV8-BN	1×1	1/128	26×26	58×58
	CONV9-BN-ReLu	3×3	1/256	24×24	56×56
	MaxPool	2×2	2/256	12×12	28×28
conv4	CONV10-BN	3×3	1/512	10×10	26×26
	CONV11-BN	1×1	1/256	10×10	26×26
	CONV12-BN	3×3	1/512	8×8	24×24
	CONV13-BN-ReLu	1×1	1/256	8×8	24×24
conv5	CONV14	3×3	1/256	6×6	22×22

In the table above, CONV*-BN represents the normalization layer after the convolutional layer, CONV*-BN-ReLu represents the activation layer after the normalization layer after the convolutional layer, and Maxpool represents the maximum pooling layer.

Secondly, on the basis of the small convolutional kernel backbone feature extraction network, we add the hopping layer connection feature fusion and apply it to the target branch. After the object template feature extraction branch conv3, bilinear interpolation is used for downsampling, and the intermediate features after downsampling are fused with the deep features after conv5, and the fused features can take into account the semantic discriminative ability of the deep features and the spatial structural details of the shallow features, so as to express the target more comprehensively, and the calculation formula is as follows:

$$f = \text{conv5}(\text{conv4}(\text{conv3}(\text{conv2}(\text{conv1}(z)))))) + \lambda \alpha(\text{conv3}(\text{conv2}(\text{conv1}(z)))), \quad (1)$$

where z denotes the input object template, α denotes downsampling, f denotes the fused object features, and $f \in R^{W \times H \times C}$, λ denotes the scale factor of the mid-level features after downsampling.

2.2. Adaptive Updating Strategy for Background Features

According to the working principle of human visual system, there always exists differentiation between the object and the surrounding background, and the greater the differentiation between them, the more the human visual system can distinguish the object and the background, and vice versa, the smaller the differentiation between them, the more the human visual system can't distinguish the object and the background, according to this principle, we propose an adaptive updating strategy of the background features to be integrated into the Siamese network single-object tracking algorithm. First, preprocessing is performed in the first frame image, and the object template image is produced by cropping and scaling, containing the object region image and the surrounding background region image, and the object template image is divided into two branches: (1) One branch enters into the backbone feature extraction network for feature extraction, and after the completion of the feature extraction, it is divided into two branches; one branch enters into the network that incorporates a variety of attention mechanisms, and the other branch maintains the features after the backbone feature extraction network. (2) The other branch enters the background adaptive updating strategy for background feature adaptive updating.

The flow of the background adaptive updating strategy is shown below:

In the first step, the object template image is cropped in the first frame image by

$$z = \sqrt{(w + 2p) \times (h + 2p)}, \quad (2)$$

and then the object template image is scaled by

$$Sz \times Sz = A^2, \quad (3)$$

where $p = (w + h)/4$ is the context of our target image patch, z denotes the size of the cropped object template image, w and h denote the width and height of the object in the original image, respectively, and S denotes a type of transformation of the image, where the scaling transformation is used, $A = 127$, which means that the cropped object template image is scaled to 127×127 .

In the second step, the width and height of the object region image in the scaled object template image were calculated as

$$z_h = \frac{A \times h}{z}, \quad z_w = \frac{A \times w}{z}, \quad (4)$$

and then the left, top, right, and bottom coordinates of the object region image in the scaled object template image were calculated as

$$\begin{aligned} l &= \frac{A - z_w}{2}, \quad t = \frac{A - z_h}{2} \\ r &= \frac{A + z_w}{2}, \quad b = \frac{A + z_h}{2} \end{aligned} \quad (5)$$

where z_h , z_w denote the width and height of the object region image in the scaled object template image, and l , t , r , and b denote the left, top, right, and bottom coordinates of the object region image in the scaled object template image.

In the third step, the object region image is cropped into the object template image, and the pixel mean value of the object region image is calculated as

$$P_{om} = \frac{1}{3} \sum_{i=0}^{i=2} \left(\frac{1}{m \times n} \sum_{m,n=0,0}^{m,n=z_w,z_h} P_{mn}^i \right), \quad (6)$$

where m, n denotes the width and height of the object region image, P_{mn} denotes the pixel value corresponding to (m, n) coordinates, and i denotes the color channel number.

In the fourth step, the size of the surrounding background region image was calculated in the object template image using Equation (7), and the pixel value of the object region image was set to 0. The mean pixel value of the surrounding background region image was calculated using Equation (8).

$$b_s = A^2 - (r - l) \times (b - t), \quad (7)$$

$$P_{bm} = \frac{1}{3} \sum_{i=0}^{i=2} \left(\frac{1}{b_s} \sum_{m,n=0,0}^{m,n=A,A} P_{mn}^i \right), \quad (8)$$

where b_s denotes the size of the surrounding background region image, m, n denotes the width and height of the object template image, P_{mn} denotes the pixel value corresponding to the (m, n) coordinate, and i denotes the color channel number.

In the fifth step, the larger the difference between the average value of the target pixels and the average value of the background pixels, the higher the differentiation between the target and the background. Conversely, the smaller the difference between the two, the lower the differentiation between the target and the background. Therefore, we define a contrast ratio to represent the difference between the target and the surrounding background, as shown in Equation (9); the larger the contrast, the easier it is to distinguish the target. Conversely, the smaller the contrast, the less easy it is to distinguish the target.

$$contrast = \frac{|P_{om} - P_{bm}|}{P_{om} + P_{bm}}, \quad (9)$$

In the sixth step, the background is adaptively updated: if $P_{om} > 127$ and $contrast < 0.5$, it means that the object area image is not significantly differentiated from the surrounding background area image; then, the pixel value of the surrounding background area in the object template image is set to 0, in order to enhance the model's ability to differentiate between the object area and the background area; if $P_{om} < 127$ and $contrast < 0.5$, it means that the object area image is not significantly differentiated from the image of the surrounding background region, then the pixel value of the surrounding background region in the object template image is set to 255 to enhance the model's ability to differentiate between the object region and the background region; in all other cases, it means that the object region image is significantly differentiated from the image of the surrounding background region, and there is no need to perform any processing for the surrounding background region as shown in Equation (10).

$$background = \begin{cases} 0, & \text{if } P_{om} > 127 \text{ and } contrast < 0.5 \\ 255, & \text{if } P_{om} < 127 \text{ and } contrast < 0.5 \\ \text{preserve}, & \text{Other} \end{cases}, \quad (10)$$

After the above six steps, the background adaptive update is completed, and after the update is completed, it enters the main feature extraction network for feature extraction, and after the completion of feature extraction, it enters the network that fuses multiple attention mechanisms, and then linearly fuses the branch features with another branch feature that carries out main feature extraction and fuses multiple attention mechanisms, and the linear fusion is as shown in Equation (11). This not only makes full use of the information of the object template image in the first frame image but also enhances the model's ability to discriminate between the object and background.

$$f_M = \lambda_1 \times \delta(f) + \lambda_2 \times \delta(f_b), \quad (11)$$

where $f_M \in R^{W \times H \times C}$, δ denotes the fusion of multiple attentional mechanisms, f denotes the object feature that has not been processed by the background adaptive updating strategy,

$f \in R^{W \times H \times C}$, λ_1 denotes the proportion coefficient of the object feature that has not been processed by the background adaptive updating strategy, f_b denotes the object feature that has been processed by the background adaptive updating strategy, $f_b \in R^{W \times H \times C}$, λ_2 denotes the proportion coefficient of the object feature that has been processed by the background adaptive updating strategy.

2.3. Fusion of Multiple Attention Mechanisms

People can perceive a large amount of information in the world by seeing a variety of things with their eyes. At the same time, they can also keep themselves free from the interference of the huge amount of information because people can select important information and ignore the unimportant information, which is called the attention mechanism. With an in-depth study of the attention mechanism, many scholars have attempted to apply it to computer vision tasks, such as image classification, object detection, and video behavior analysis [30]. Through many studies and experiments, it has been proven that the attention mechanism can effectively improve the performance of the algorithm.

Our goal in introducing attention mechanisms into visual single-object tracking algorithms is to allow the algorithms to mimic human visual and cognitive systems, essentially allowing the network to learn to focus its attention on areas of interest and ignore areas of disinterest. The most common attention mechanisms in the field of computer vision include channel attention [31], spatial attention [32], hybrid attention [33], and coordinate attention [34]. The channel attention mechanism is used for the network to calculate the importance of each channel, the essence of which is to construct an importance model between different channel features, so that the network can adaptively focus on the channel features with higher importance; in short, to select the channel features that are more useful for the task. The spatial attention mechanism allows the network to focus on the pixel points in the height and width directions within the feature layer to find the most important object region in the network for processing, and to allow the network to adaptively focus on the location of the object, which essentially allows the object region to become more heavily weighted. The hybrid attention mechanism is a combination of spatial and channel attention mechanisms, which not only allows the network to adaptively pay attention to the weight of each channel; it will also allow the network to adaptively pay attention to the weight of each pixel point. The structure of the hybrid attention mechanism is to connect the channel attention mechanism and the spatial attention mechanism in tandem; the features are inputted into the line channel attention mechanism first, and then after the output results are obtained, the features are input into the spatial attention mechanism, and the final results are obtained after the previous results are inputted into the spatial attention mechanism. Coordinate attention is an attention mechanism used to enhance the deep learning model's understanding of the spatial structure of input data. The core idea of coordinate attention is to introduce coordinate information such that the model can better understand the relationship between different locations.

The hybrid attention mechanism is a model that combines the channel attention mechanism and the spatial attention mechanism. The first half is the channel attention mechanism, and the second half is the spatial attention mechanism, which aims to enhance the network's ability to pay attention to images. Typically, the channel attention mechanism in the hybrid attention mechanism is designed based on the SENet model [35], and the implementation can be divided into two parallel steps, which perform the global maximum pooling and global average pooling operations on the individual feature layers of the input network, respectively, and then process the results of the global maximum pooling and global average pooling by using the shared full connectivity layer, respectively, and then sum the processing results of the two shared full connectivity layers. The connection layer processing results are summed up, the summed results in the sigmoid function, and the input feature layer of each channel weight can be obtained. The weights range from 0 to 1; the larger the weights of a certain channel, indicating that the channel is more important, and finally multiply the weights by the original input feature layer, the completion of the

hybrid attention mechanism in the operation of the channel attention mechanism. While Wang et al. in the ECA-Net [36] paper argued that the SENet model will bring some side effects to the prediction of the channel attention mechanism in the hybrid attention mechanism, and capturing the dependencies between all channels is inefficient and useless, the idea of the ECA-Net is to remove the fully_connected layer in the SENet and use the 1D convolution instead, because the 1D convolutional neural network has good cross-channel information acquisition ability. Thus, we adopt the ECA-Net model as the basis to redesign the hybrid attention mechanism in the channel attention mechanism to replace the hybrid attention mechanism in the channel attention mechanism designed by the SENet model. We introduce the single-object tracking algorithm to input the single feature layer of the network, perform the global average pooling operation, and subsequently, the result of the global average pooling is processed by using the 1D convolution. The result of the 1D convolution is processed by a sigmoid function, which can obtain the weights of each channel of the input feature layer, which ranges from 0 to 1, and the weights are multiplied by the original input feature layer. The input of the spatial attention mechanism in the hybrid attention mechanism is the output of the channel attention in the hybrid attention mechanism, which inputs a single feature layer of the network, takes the maximum value and the average value in the channel of each feature point, then stacks the two results, and then reduces the dimensionality of the channel by using the convolution of the channel as 1. Then, the results are obtained by a sigmoid function, which can obtain the input feature layer of each feature point, and the weight of each feature point of the input feature layer can be obtained by a sigmoid function, which ranges from 0 to 1. Finally, the weight is multiplied by the original input feature layer, completing the operation of the spatial attention mechanism in the hybrid attention mechanism. After the above two steps. The improved hybrid attention mechanism operation is completed, and the structure of the improved hybrid attention mechanism is shown in Figure 3.

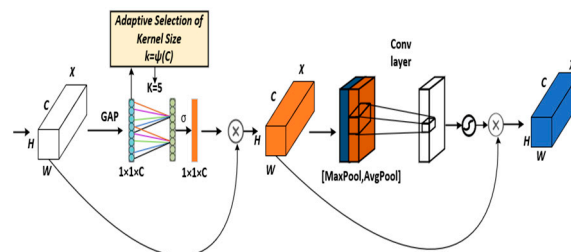


Figure 3. Improved structure of the hybrid attention mechanism.

Coordinate attention is used to encode precise position information into the neural network, and the core idea is to introduce coordinate information such that the model can better understand the relationship between different positions, as shown in Figure 4.

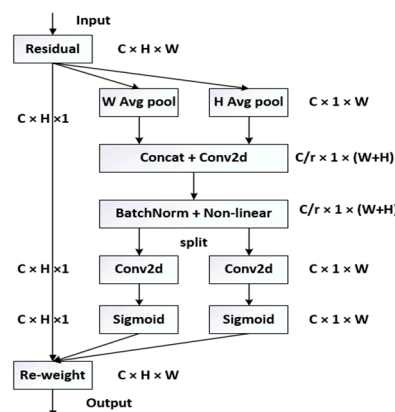


Figure 4. Structure of coordinate attention.

To enable the network to capture more accurate positional information about the object, we propose connecting the improved hybrid attention mechanism in parallel with coordinate attention, which involves modeling the feature relationship between channel, spatial, and positional information and fusing their results to introduce them into the single-object tracking algorithm, the structure of fusing multiple attention, which is shown in Figure 5. This captures not only cross-channel information but also direction-aware and position-aware information, which can help the algorithm localize and identify the object of interest more accurately. At the same time, it also enables the network to adaptively learn to focus on channel, spatial, coordinate, and other feature information, thus enhancing the feature expression, attention capability, and precision of the entire network.

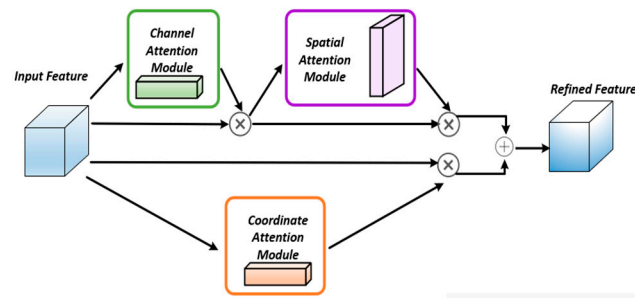


Figure 5. Structure of the mechanism for fusing multiple attention.

Multiple Attention Mechanism: Input feature x enters two different branches. Input feature x enters branch 1, and firstly captures channel feature information through the CANet channel attention mechanism, and then captures spatial feature information through the spatial attention mechanism; input feature x enters branch 2, and then captures position feature information through the coordinate attention; and then the channel, spatial, and position feature information are added together, so that the model learns to pay attention to the channel, spatial, and coordinate feature information of the tracked target adaptively, thus improving the ability of the whole model to pay attention to feature information, so as to improve the attention ability of the whole model. The realization process of the multiple attention mechanism is shown in Equation (12).

$$x^M = \{x \times ECA(x)\} \times SAM\{x \times ECA(x)\} + \{x \times CA(x)\}, \quad (12)$$

where x denotes input features, x^M denotes output features after multiple attention mechanisms, ECA denotes ECANet channel attention mechanism, SAM denotes spatial attention mechanism, CA denotes coordinate attention, $\times$ denotes dot product, and $+$ denotes dot plus.

2.4. Response Fusion Operations after Convolutional Inter-Correlation

In order to enrich the output response of the algorithm, considering that the algorithm can pay attention to both local and global responses during the execution process, we perform convolutional inter-correlation operation between the original object feature branch and the original search region feature branch, and at the same time, we perform convolutional inter-correlation operation between the object feature branch after multiple attentional mechanisms and the search region feature branch after multiple attentional mechanisms, and then the results of the two convolutional inter-correlations are fused linearly to get the final response map, and the response fusion formula is shown in (13).

$$f = k_1 \times f_1 + k_2 \times f_2, \quad (13)$$

where f_1 represents the response of the original object feature branch and the original search region feature branch after the convolutional inter-correlation operation, k_1 is the proportion coefficient occupied by the original feature branch, f_2 represents the response of

the object feature branch signification and the search region feature branch after multiple attention mechanisms after the convolutional inter-correlation operation, and k_2 represents the proportion coefficient occupied by the features after multiple attention mechanisms.

3. Training and Testing of Algorithms

To train the algorithm, we chose the GOT-10k [37] datasets as the training datasets of the algorithm, which have a rich variety of objects and a wide range of scenarios, so it is very suitable for the algorithm to be trained on the GOT-10k datasets, which can enhance the generalization ability of the model.

Before training the network, we need to preprocess the images in the GOT-10k datasets. For preprocessing of the object template region image, an object square region is cropped out with the center of the object as the cropping center, and the cropping operation uses Equation (2). After cropping is completed, the object square region is scaled to a 127×127 object template, and the scaling operation uses Equation (3).

For the preprocessing of the search region image, to maintain consistency with the scaling scale of z , we crop the object search region image; after the cropping is completed, a random deflation operation is performed on the cropped region, which helps in the data enhancement, and three random deflation coefficients are selected; which are $\{1.0375^{-1}, 1.0375^0, \text{ and } 1.0375^1\}$, and finally scaled to 255×255 , as shown in Equation (14).

$$x = \frac{255 \times z}{127} \times scale_i, \quad (14)$$

where x denotes the cropped search region image, z denotes the side length of the cropped object square region, and $scale_i$ denotes one of the three random deflation coefficients.

The loss function is defined as shown in Equation (15).

$$\ell(y, v) = \log(1 + \exp(-yv)), \quad (15)$$

where y represents the true value of similarity and $y \in \{-1, +1\}$ represents the positive and negative example pairs. v represents the similarity between the object template features and search region.

Define the error function as shown in Equation (16).

$$L(y, v) = \frac{1}{|D|} \sum_{u \in D} \ln(1 + \exp(-y(u), v(u))), \quad (16)$$

where $y(u)$ represents the true value of similarity corresponding to position u , $v(u)$ is the response value corresponding to position u in the response map, $u \in D$, and D is the set of position indices in the response map, which is determined by the Euclidean distance between position u and the center position c of the response map as shown in Equation (17).

$$y(u) = \begin{cases} +1 & k\|u - c\| \leq R \\ -1 & \text{Other} \end{cases}, \quad (17)$$

where R is the set threshold, R is taken as 16, k is the network step size, the network structure shrinks the original image eight times, and k is taken as 8.

The model is optimized using the stochastic gradient descent method, and the optimized parameters of the convolutional network θ can be obtained, as shown in Equation (18).

$$\operatorname{argmin}_{\theta} \frac{1}{|N|} \sum_{z, x, y} L(y, f(z, x; \theta)) \quad (18)$$

where y represents the true value of similarity, θ is a network parameter, and N represents the number of samples.

The Siamese network single-object tracking algorithm incorporating multiple attention mechanisms and a background feature adaptive updating strategy was developed using

Python under the PyTorch framework. The hardware platform comprised a dual Intel® Xeon® E5-2696 CPU (Santa Clara, CA, USA) and an NVIDIA RTX3080TI 16 G GPU (Santa Clara, CA, USA). The algorithm was trained on the GOT-10k dataset, and its performance was tested and verified using the object tracking benchmark datasets OTB and VOT, compared, and analyzed with other algorithms. The algorithm testing results are presented in Section 4.

Training setup: The learning rate was adjusted using exponential decay; the learning rate was decayed from 10^{-2} to 10^{-5} ; the number of times the datasets were trained through the network was 50; the batch size was set to 16; the weight decay was set to 5×10^{-4} ; the momentum was set to 0.9; and the window impact rate was set to 0.176.

Tracking test: During the object tracking process, the center of the tracking result from the previous frame is considered as the center of the search area in the current frame image. The displacement of the maximum response value in the current frame relative to the center of the response map was then calculated. This value was multiplied by the network step size to determine the displacement of the object between frames, thereby obtaining the center position of the object in the current frame.

4. Algorithm Test Results

This section presents a demonstration and analysis of the test results of our proposed tracking algorithm with other algorithms on the visual object tracking benchmark datasets OTB and VOT, which can be divided into two parts. Section 4.1 demonstrates and analyzes the test results for the OTB datasets. Section 4.2 demonstrates and analyzes the test results on the VOT datasets.

4.1. OTB Dataset Test Results and Analysis

The OTB dataset [6] is one of the most widely used benchmark datasets for visual object tracking. The OTB datasets include OTB-2013, OTB-50, and OTB-100. The OTB-100 dataset consists of 100 video sequences containing 11 complex situations such as deformation, occlusion, fast motion, and illumination changes. Therefore, we validated our proposed algorithm on the OTB-100 dataset. We evaluated the Siamese network single-object tracking algorithm that incorporates multiple attention mechanisms with adaptive updating strategies for background features by two evaluation metrics, precision and success rate, and plotted the precision and success rate graphs for these two metrics to compare with other algorithms. At the same time, we view the tracking effect of our proposed algorithm and compare it with other algorithms. The specific meanings of the two evaluation metrics, precision and success rate, respectively, are the ratio of the number of successful frames to the total number of frames for which the Euclidean distance between the tracking result and the labeled center position of the datasets is less than a certain threshold, representing the precision. The ratio of the number of frames to the total number of frames for which the tracking frame coverage is greater than a certain threshold represents the success rate, in which the coverage rate is defined as the ratio of the area of the intersection portion to the area of the concatenation portion between the predicted tracking frames and labeled tracking frames.

As shown in Figure 6, our proposed algorithm achieved the leading level in the comparison experiments with SRDCF [15], SAMF [38], MEEM [39], CSK [13], DSST [40], SiamFC [20], SiamRPN [22], and SiamFC++ [23] on the OTB-100 dataset, with a tracking precision of 0.818 and a tracking success rate of 0.610. Our proposed algorithm ran at an average speed of more than 60 fps in the experiments. Compared with the benchmark framework algorithm SiamFC, our proposed algorithm improves the precision by 7% and the success rate by 5.3%, which indicates that the Siamese network single-object tracking algorithm integrating multiple attentional mechanisms and an adaptive updating strategy of background features can effectively cope with the problem of degradation of tracking performance during the process of tracking an object in complex environments, and further improve the tracking precision and precision.

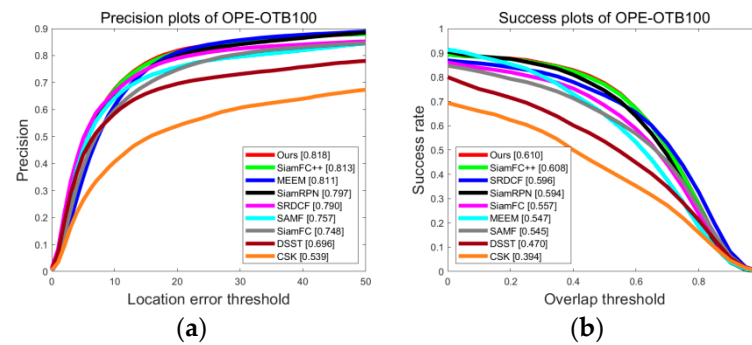


Figure 6. Precision and success rate of this algorithm and other algorithms in the OTB-100 dataset. Figure (a) represents the precision of this algorithm and other algorithms in the OTB-100 dataset. Figure (b) represents the success rate of this algorithm and other algorithms in the OTB-100 dataset.

To further verify and analyze the performance of our proposed tracking algorithm in different complex situations, we plot the precision and success rate plots of our proposed tracking algorithm in 11 different complex situations in the OTB-100 dataset to measure the performance of our proposed tracking algorithm in different complex situations. The precision plots for the 11 different complex situations are shown in Figure 7, and the 11 success rate plots for different complex situations are shown in Figure 8. In Figure 7, our proposed tracking algorithm is ranked first in terms of precision in five different complex cases, whereas in the remaining six different complex cases, three were ranked second in terms of precision and three were ranked third in terms of precision, which demonstrates that our proposed tracking algorithm can effectively cope with the single-object tracking problem in different complex cases. In Figure 8, the success rates of the remaining eight different complex cases are ranked first, except for the success rates of low-resolution scale change and plane rotation, which are ranked second, third, and third, further indicating that our proposed tracking algorithm is robust and can cope with almost all complex cases in object tracking. The network of our proposed algorithm is a 14-layer deep neural network, which can enable the extraction of deep features of the target, so that the robustness of the tracking algorithm can be improved to a certain extent; secondly, the introduction of the background adaptive updating strategy can enhance the ability of the algorithm to distinguish between the target and the background features, which solves the problem of the degradation of the tracking performance when the target's appearance in the tracking scene is deformed and the ambient illumination is changed, and enhances the robustness of the tracking; finally, during the tracking process, the target may encounter occlusion, scale changes and other factors, which lead to large changes in the target appearance representation model, making the tracking of moving targets difficult. In order to improve the model's adaptability and separability to complex backgrounds, we introduce a variety of attention mechanisms. The attention mechanism allows the algorithmic model to dynamically focus on specific regions of the input data, thus improving processing efficiency and precision of results. In target tracking, the attention mechanism can help the model better understand the image content and distinguish between different targets, especially in the case of overlapping or occluded targets. By introducing the attention mechanism, discriminative features can be effectively extracted and important features in the target can be focused on, thus improving the precision and robustness of target tracking. Firstly, the network of our proposed algorithm is a 14-layer deep neural network, which can extract the depth features of the target, so that the robustness of the tracking algorithm can be improved to a certain extent. Secondly, the introduction of the background adaptive updating strategy can enhance the algorithm's ability to distinguish between the features of the target and the background, which solves the problem of the degradation of the tracking performance in tracking scenes when the target's appearance is deformed and the ambient illumination is changing and enhances the robustness of the tracking. Finally, during the tracking process, the target may encounter

occlusion, scale changes, and other factors, which lead to large changes in the target appearance representation model, making it difficult to track a moving target. In order to improve the model's adaptability and separability to complex backgrounds, multiple attention mechanisms are introduced. The attention mechanisms allow the algorithmic model to dynamically focus on specific regions of the input data, thus improving processing efficiency and precision of results. In target tracking, the attention mechanism can help the model better understand the image content and distinguish between different targets, especially in the case of overlapping or occluded targets. By introducing the attention mechanism, discriminative features can be effectively extracted and important features in the target can be focused on, thus improving the precision and robustness of target tracking.

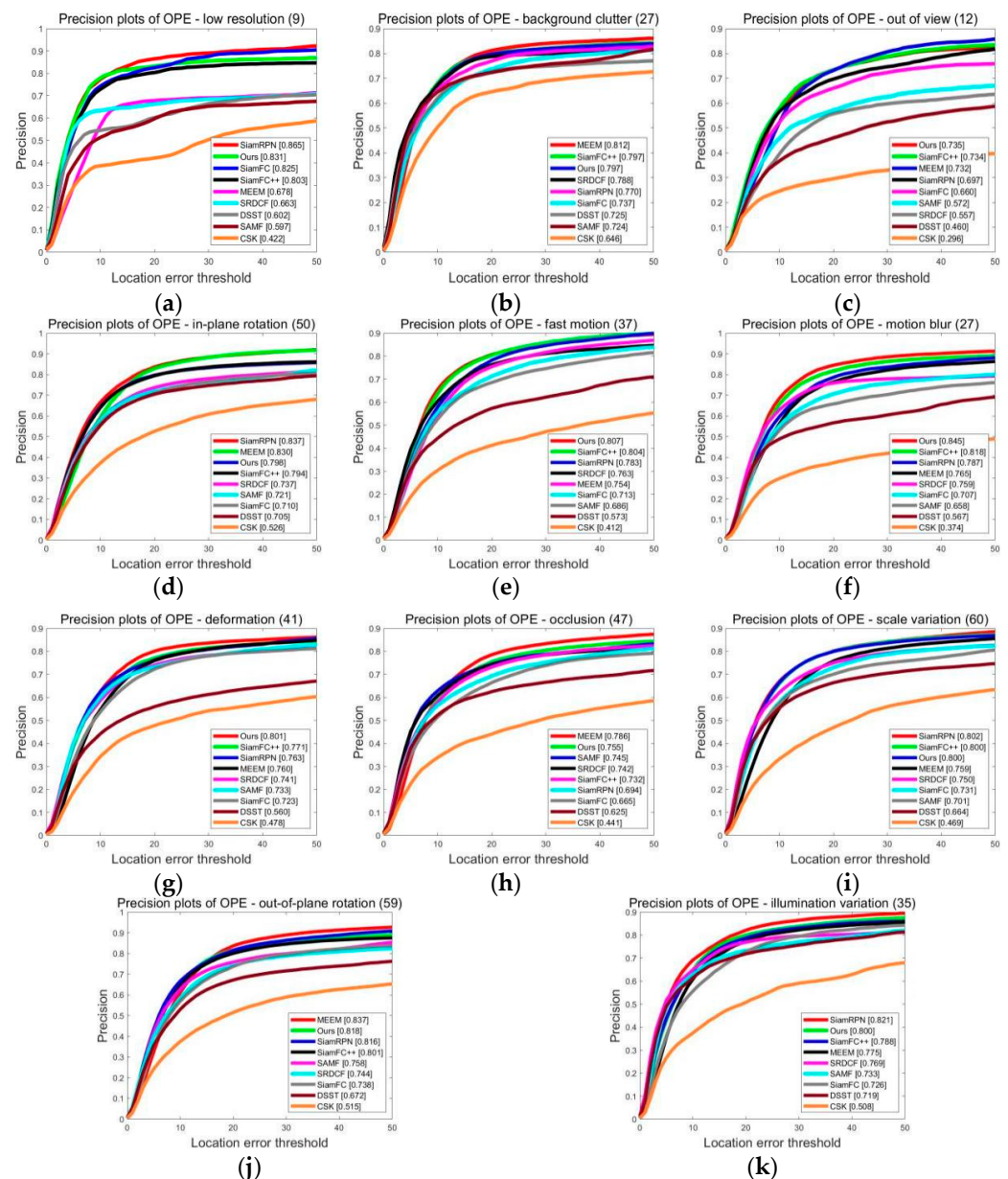


Figure 7. Comparison of the precision of this algorithm with other algorithms for 11 different complex cases in the OTB-100 dataset. Figure (a) represents the precision of this algorithm with respect to other algorithms at low resolution properties. Figure (b) represents the precision of this algorithm versus

other algorithms with respect to the background blurring property. Figure (c) represents the precision of this algorithm versus other algorithms for out-of-field-of-view attributes. Figure (d) represents the precision of this algorithm with respect to other algorithms for the in-plane rotation property. Figure (e) represents the precision of this algorithm with respect to other algorithms for fast motion properties. Figure (f) represents the precision of this algorithm with respect to other algorithms under motion blur property. Figure (g) represents the precision of this algorithm with respect to other algorithms under the deformation property. Figure (h) represents the precision of this algorithm with respect to other algorithms with respect to the occlusion property. Figure (i) shows the precision of this algorithm compared to other algorithms in terms of scale variation property. Figure (j) represents the precision of this algorithm with respect to other algorithms for the out-of-plane rotation property. Figure (k) represents the precision of this algorithm with respect to other algorithms under the light variation property.

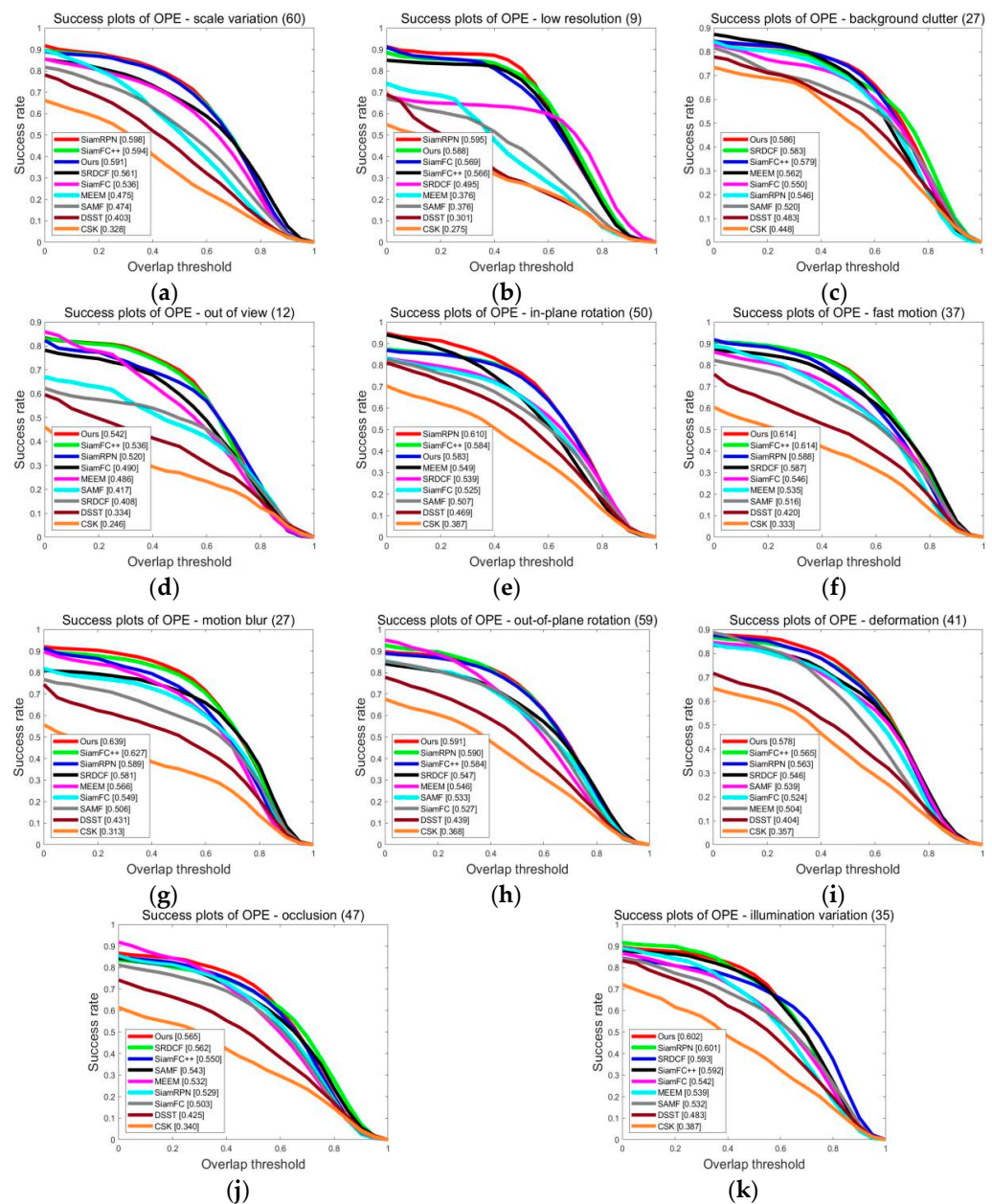


Figure 8. Comparison of the success rate of this algorithm with other algorithms for 11 different complexities in the OTB-100 dataset. Figure (a) shows the success rate of this algorithm versus other

algorithms for the scale variation property. Figure (b) represents the success rate of this algorithm compared to other algorithms for low-resolution attributes. Figure (c) represents the success rate of this algorithm as compared to other algorithms under the background blurring property. Figure (d) represents the success rate of this algorithm compared to other algorithms for out-of-view attributes. Figure (e) represents the success rate of this algorithm compared to other algorithms under the in-plane rotation property. Figure (f) represents the success rate of this algorithm with respect to other algorithms with respect to fast motion properties. Figure (g) represents the success rate of this algorithm compared to other algorithms under motion blur attribute. Figure (h) represents the success rate of this algorithm compared to other algorithms for out-of-plane rotation property. Figure (i) represents the success rate of this algorithm compared to other algorithms under deformation property. Figure (j) represents the success rate of this algorithm as compared to other algorithms under the occlusion property. Figure (k) represents the success rate of this algorithm as compared to other algorithms under light varying attributes.

We selected nine of the most representative video sequences from the OTB-100 dataset to compare the tracking results of our proposed tracking algorithm with those of other algorithms. These nine video sequences contain almost all the 11 complex situations. These nine video sequences are Biker, Bird1, Dragon Baby, freeman4, ironman, Jump, Kitesurf, Motor rolling, and soccer. To better show the tracking results of the algorithms, we chose the top seven algorithms in terms of performance in the experiments to show the tracking results, which are ours, SiamFC++, SiamRPN, SiamFC, SRDCF, MEEM, and SAMF, as shown in Figure 9.

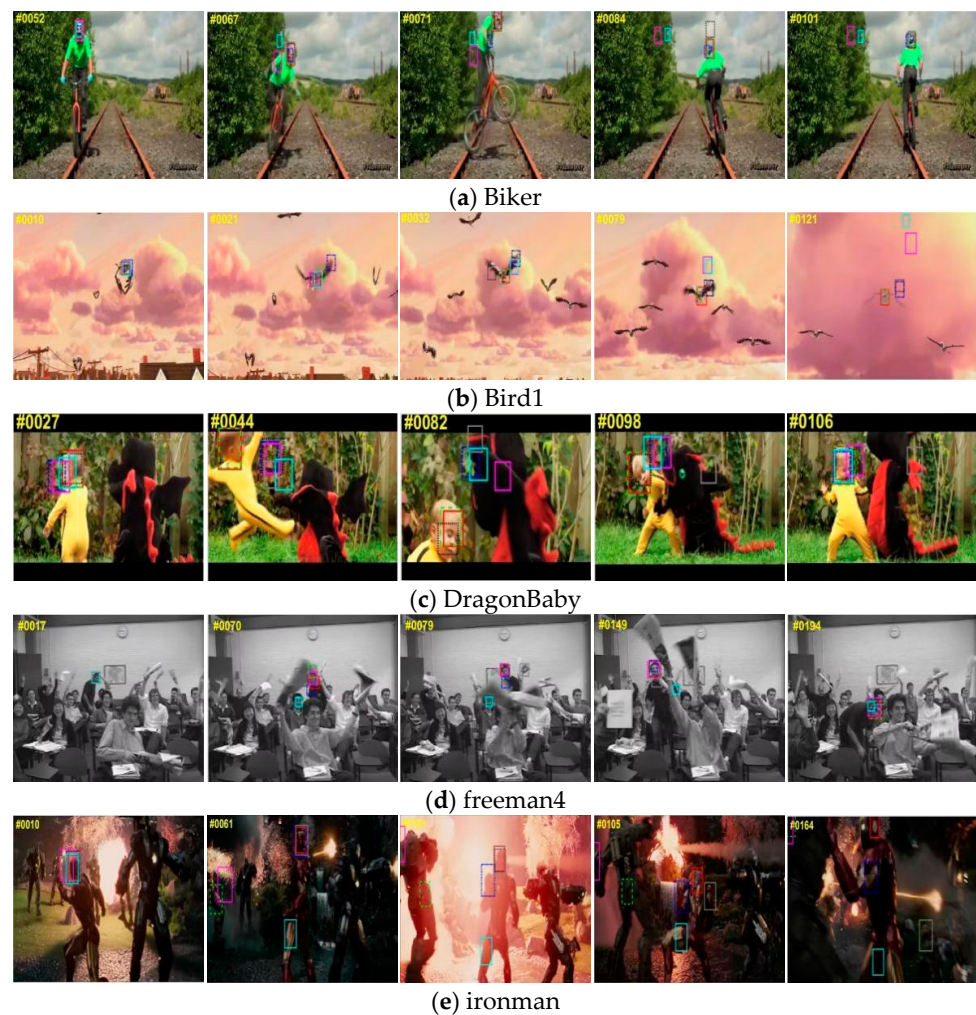


Figure 9. Cont.

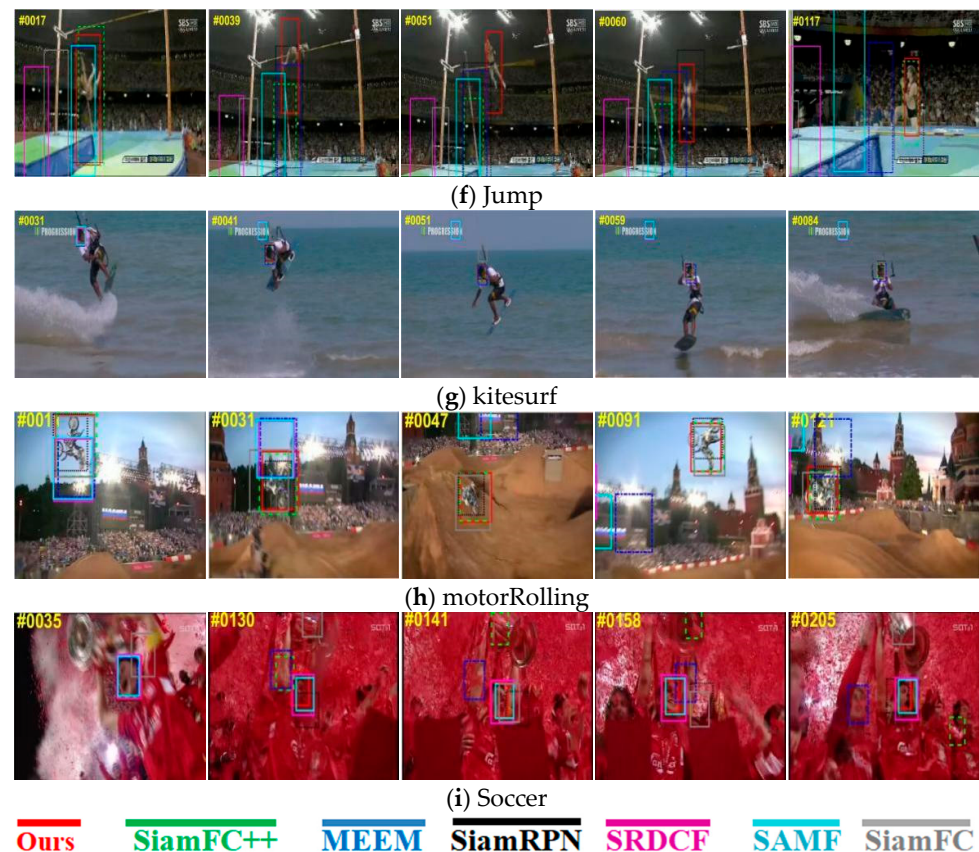


Figure 9. Tracking results of this algorithm with other algorithms for 9 video sequences in the OTB-100 dataset.

From Figure 9, we can see that our proposed tracking algorithm can effectively cope with the problem of tracking performance degradation during object tracking in complex environments with high stability. From the two video sequences of ironman and jump, we can see that our proposed tracking algorithm can always track the object stably, whereas the remaining five algorithms will fail to track. From the seven video sequences from Biker, Bird1, Dragon Baby, freeman4, Kitesurf, Motor rolling, and soccer, we can see that our proposed algorithm can track the object stably, while some of the remaining five algorithms track stably, and some of them fail to track. Our proposed algorithm's object tracking bounding box is more accurate than the object tracking bounding box of the algorithm that can track stably among the remaining five algorithms.

The tracking algorithm tracks only the objects in the video sequence that are bounded by a boundary; the initial position of the target is given in the first frame of the video sequence and the tracking of the subsequent frames is done by the tracking algorithm. In the Biker video sequence, Biker's face is tracked. In the Bird1 video sequence, a bird is tracked. In the Baby Dragon video sequence, the tracking is of Baby Dragon's face. In the freeman4 video sequence, it is freeman4's face that is tracked. In the "Kitesurfing" video sequence, the tracked is the face of a "kitesurfer". In the "Motor rolling" video sequence, the tracking is of bikes and cyclists. In the Triathlon video sequence, tracked is the face of Superman. In the jumping video, tracked are high jumpers.

Running speed is an important evaluation index of target tracking algorithms. This algorithm has an average running speed of 60 FPS when testing the OTB100 dataset on a computer with a CPU of Intel® Xeon® E5-2696 and a GPU of NVIDIA RTX3080TI 16 G and compares it with the running speeds of the other six tracking algorithms, and the results are shown in Table 2. From the table, it can be seen that this algorithm has a large advantage in running speed compared with MEEM and SAMF tracking algorithms, which is due to the fact that this algorithm trains a similarity matching model offline through

the training dataset, and there is no online updating process; there is a large gap in the running speed compared with CSK tracking algorithm, which is due to the large amount of computation in the twin network; and there is a large gap in the running speed compared with the benchmark algorithm framework SiamFC, the running speed of this algorithm decreases to a certain extent due to the deeper backbone feature extraction network, but it can still meet the requirement of real-time tracking.

Table 2. Tracking speed comparison.

Tracking Algorithms	Ours	SiamFC	SiamFC++	SiamRPN	MEEM	SAMF	CSK
Average Running Speed (FPS)	60	86	90	160	6	7	362

To verify the effectiveness of the four modules proposed in our algorithm, namely, the deep backbone feature extraction network with a small convolutional kernel fused with jump-layer connected features, an adaptive updating strategy for background features, multiple attention mechanisms, and feature response fusion operation after convolutional inter-correlation. We designed three ablation experiments, Siam_SHD, Siam_SHD_BAUS, and Siam_SHD_BAUS_MAM, to evaluate the performance impact of these four components on our algorithms through ablation experiments using SiamFC and our algorithms as a reference benchmark. Siam_SHD uses a deep backbone of small convolution kernel fused with a hopping-layer connected feature extraction network to replace the tracking algorithm of the shallow AlexNet in SiamFC; Siam_SHD_BAUS adds a tracking algorithm with a background adaptive update strategy to Siam_SHD; Siam_SHD_BAUS_MAM adds a tracking algorithm with multiple attention mechanisms to Siam_SHD_BAUS; our algorithm is based on Siam_SHD_BAUS_MAM and adds the feature response fusion operation after convolutional inter-correlation. The three ablation experimental algorithms were trained using the GOT-10K dataset and tested on the OTB100 dataset; the test results are shown in Figure 10.

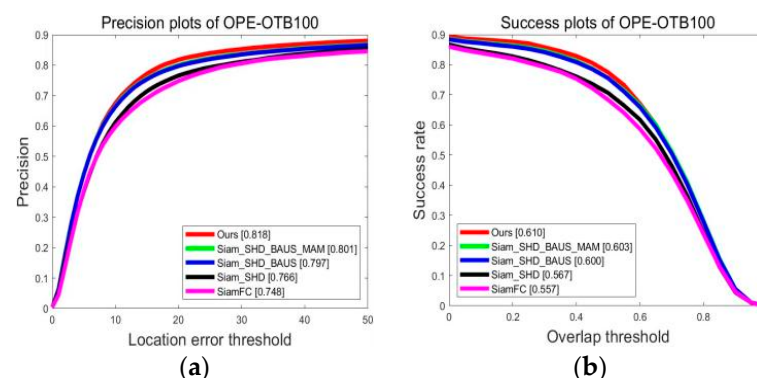


Figure 10. Ablation test results in the OTB-100 dataset. Figure (a) represents the precision of this algorithm for ablation experiments under the OTB100 dataset. Figure (b) shows the success rate of this algorithm for ablation experiments with the OTB100 dataset.

From Figure 10, it can be seen that using SiamFC as the reference frame, Siam_SHD improves the precision by 1.8% and the success rate by 1.0% on the basis of SiamFC; using Siam_SHD as the reference frame, Siam_SHD_BAUS improves the precision by 3.1% and the success rate by 3.3% on the basis of Siam_SHD; using Siam_SHD_BAUS as the reference benchmark, Siam_SHD_BAUS_MAM improves the precision by 0.3% and the success rate by 0.4% on top of Siam_SHD_BAUS; using Siam_SHD_BAUS_MAM as the reference benchmark, our algorithm improves the precision on top of Siam_SHD_BAUS_MAM by 1.7% and success rate improved by 0.7%; The results of the ablation experiments show that

all four modules are effective in our algorithm and all of them can improve the tracking performance of the algorithm.

4.2. VOT Dataset Test Results and Analysis

The VOT Challenge is an annual visual object-tracking competition. To further validate the tracking performance of our proposed single-object tracking algorithm, we evaluated it on the VOT2018 dataset. Among them, the VOT2018 dataset contains 60 challenging videos, and we quantitatively evaluate the performance of our proposed tracking algorithm by precision (A), robustness (R), and expected average overlap (EAO), and the algorithms involved in the comparison include siamFC [20], SRDCF [39], MEEM [39], DSST [40], and DCFNet [41], whose results are all from the literature [42]. The comparison results, as shown in Table 3, show that the Siamese network single-object tracking algorithm, which includes a multiple attention mechanism and an adaptive updating strategy for background features, outperforms the other algorithms involved in the comparison in all performance metrics.

Table 3. Comparison of the results of this algorithm with other algorithms on the VOT2018 dataset.

Tracking Algorithms	EAO	Precision	Robustness
Our	0.229	0.511	0.514
MEEM	0.192	0.463	0.534
siamFC	0.188	0.503	0.585
DCFNet	0.182	0.470	0.543
SRDCF	0.119	0.490	0.974
DSST	0.079	0.395	1.452

Among them, the higher the precision and average overlap expectation, the lower the robustness score, and the better the tracking performance of the algorithm. Conversely, the lower the precision and average overlap expectation, the higher the robustness score, and the worse the tracking performance of the algorithm.

5. Conclusions

To effectively address the problem of tracking performance degradation and tracking failure of single-object tracking algorithms based on Siamese full convolutional networks in complex scenarios, we propose a single-object tracking algorithm for Siamese networks that incorporates multiple attention mechanisms and adaptive updating strategies for background features. To this end, we make four contributions to the algorithm: (1) We designed a backbone feature extraction network using a small convolutional kernel to fuse jump-layer connected features, which can not only effectively extract the deep semantic features of the object, but also fuse the mid-level features with the deep features to enhance the model's characterization ability. Meanwhile, using a small convolutional kernel for convolution not only reduces the parameters of the network, but also increases the depth of the network while ensuring computational speed. (2) We also propose an adaptive updating strategy for background features and apply it to the processing of object templates, i.e., the linear combination of object features obtained from a processed object template with those obtained from another unprocessed object template through a backbone feature extraction network and multiple attention mechanism can improve the model's ability to distinguish between object and background features. (3) We also integrated the multiple attention mechanism in the algorithm, which can make the algorithm learn to pay attention to the channel, space, coordinates, and other feature information, and enhance the feature expression ability, attention ability, and precision of the entire algorithm. (4) We convolve the original object features with the original search region features in our algorithm, and at the same time convolve the object features after the attention mechanism with the search region features after the attention mechanism, and linearly fuse the interrelated results of the two convolutions to obtain the final response map to enrich the response of the model.

We validate the performance of our algorithm on the object tracking benchmark datasets OTB and VOT and compare it with other algorithms. The validation and comparison results show that our algorithm can effectively solve the problem of object tracking in complex scenes and further improve tracking precision and precision while ensuring tracking speed.

Author Contributions: Conceptualization, W.F. and F.M.; methodology, W.F. and C.Y.; software, W.F. and A.Y.; validation, W.F., F.M. and C.Y.; formal analysis, F.M.; investigation, C.Y.; resources, A.Y.; data curation, W.F.; writing—original draft preparation, W.F.; writing—review and editing, W.F.; visualization, W.F.; supervision, F.M.; project administration, F.M.; funding acquisition, F.M. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the China Academy of Engineering Physics under Grant No. 61426050303-2.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank the Institute of Applied Electronics, China Academy of Engineering Physics for supporting this research project.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the study's design; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Yilmaz, A.; Javed, O.; Shah, M. Object Tracking: A Survey. *ACM Comput. Surv.* **2006**, *38*, 13. [\[CrossRef\]](#)
2. Li, P.X.; Wang, D.; Wang, L.J.; Lu, H. Deep visual tracking: Review And experimental comparison. *Pattern Recognit.* **2018**, *76*, 323–338. [\[CrossRef\]](#)
3. Zheng, C.; Usagawa, T. A rapid webcam-based eye tracking method for human computer interaction. In Proceedings of the 2018 International Conference on Control, Automation and Information Sciences (ICCAIS), Hangzhou, China, 24–27 October 2018; pp. 133–136.
4. Sarcinelli, R.; Guidolini, R.; Cardoso, V.B.; Paixão, T.M.; Berriel, R.F.; Azevedo, P.; De Souza, A.F.; Badue, C.; Oliveira-Santos, T. Handling pedestrians in self-driving cars using image tracking and alternative path generation with Frenét frames. *Comput. Graph.* **2019**, *84*, 173–184. [\[CrossRef\]](#)
5. Wu, Y.; Lim, J.; Yang, M.-H. Online object tracking: A benchmark. In Proceedings of the 2013 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
6. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Wu, Y.; Cheng, J.; Wang, J.; Lu, H. Real-time visual tracking via Incremental Covariance Tensor Learning. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision (ICCV), Kyoto, Japan, 29 September–2 October 2009; pp. 1631–1638.
8. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.* **2002**, *50*, 174–188. [\[CrossRef\]](#)
9. Comaniciu, D.; Ramesh, V.; Meer, P. Real-time tracking of non-rigid objects using mean shift. In Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Hilton Head, SC, USA, 15 June 2000; pp. 142–149.
10. Comaniciu, D.; Ramesh, V.; Meer, P. Kernel-based object tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 564–577. [\[CrossRef\]](#)
11. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual Object Tracking using Adaptive Correlation Filters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, IEEE, 13–18 June 2010; pp. 2544–2550.
12. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [\[CrossRef\]](#)
13. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In Proceedings of the 12th European Conference on Computer Vision, Florence, Italy, 7–13 October 2012; Springer: Florence, Italy, 2012; pp. 702–715.
14. Ma, C.; Huang, J.B.; Yang, X.; Yang, M.H. Hierarchical Convolutional Features for Visual Tracking. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; IEEE: Santiago, Chile, 2015; pp. 3074–3082.

15. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the 2005 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4310–4318.
16. Li, Y.; Zhu, J. A scale adaptive kernel correlation filter tracker with feature integration. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–7 and 12 September 2014; Springer: Cham, Switzerland, 2014; pp. 254–265.
17. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H.S. Staple: Complementary learners for real-time tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1401–1409.
18. Bromley, J.; Bentz, J.W.; Bottou, L.; Guyon, I.; Lecun, Y.; Moore, C.; Sackinger, E.; Shah, R. Signature verification using a “Siamese” time delay neural network. *Int. J. Pattern Recognit. Artif. Intell.* **1993**, *7*, 669–688. [[CrossRef](#)]
19. Zagoruyko, S.; Komodakis, N. Learning to compare image patches via convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4353–4361.
20. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-convolutional siamese networks for object tracking. In Proceedings of the Computer Vision—ECCV 2016 Workshops (ECCV), Amsterdam, The Netherlands, 8–10 and 15–16 October 2016; pp. 850–865.
21. Abdelpakey, M.H.; Shehata, M.S.; Mohamed, M.M. DensSiam: End-to-end densely-siamese network with self-attention model for object tracking. In *International Symposium on Visual Computing*; Springer: Cham, Switzerland, 2018; pp. 463–473.
22. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High performance visual tracking with siamese region network. In Proceedings of the 2018 IEEE/CVF Conference of Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
23. Xu, Y.; Wang, Z.; Li, Z.; Yuan, Y.; Yu, G. Siamfc++: Towards robust and accurate visual tracking with object estimation guidelines. In Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI-20), New York, NY, USA, 7–12 February 2020; pp. 12549–12556.
24. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. SiamRPN++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
25. Li, Y.; Zhang, X. SiamVGG: Visual Tracking using Deeper Siamese Networks. *arXiv* **2019**, arXiv:arXiv.1902.02804. [[CrossRef](#)]
26. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H.S. Fast Online Object Tracking and Segmentation: A Unifying Approach. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 1328–1338. [[CrossRef](#)]
27. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H.S. End-to-end representation learning for correlation filter based tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5000–5008.
28. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
29. Parkhi, O.M.; Vedaldi, A.; Zisserman, A. Deep face recognition. In Proceedings of the British Machine Vision Conference (BMVC), Swansea, UK, 7–10 September 2015.
30. Chaudhari, S.; Polatkan, G.; Ramanath, R.; Mithal, V. An attentive survey of attention models. *arXiv* **2019**, arXiv:1904.02874. [[CrossRef](#)]
31. He, A.; Luo, C.; Tian, X.; Zeng, W. A twofold siamese network for realtime object tracking. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4834–4843.
32. Zhu, X.; Cheng, D.; Zhang, Z.; Lin, S.; Dai, J. An empirical study of spatial attention mechanisms in deep networks. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 6688–6697.
33. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional block attention module. In Proceedings of the Computer Vision—ECCV 2018 (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
34. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13713–13722.
35. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
36. Wang, Q.L.; Wu, B.G.; Zhu, P.F.; Li, P.H.; Zuo, W.M.; Hu, Q.H. ECA-Net: Efficient channel attention for deep convolutional neural network. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11531–11539.
37. Huang, L.; Zhao, X.; Huang, K. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *43*, 1562–1577. [[CrossRef](#)] [[PubMed](#)]
38. Sun, L.; Fan, B.; Peng, Y. Adaptive Scale Correlation Filter Tracker with Feature Fusion. In Proceedings of the 30th Chinese Conference on Control and Decision Making, Shenyang, China, 9–11 June 2018.
39. Zhang, J.; Ma, S.; Sclaroff, S. MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization. In Proceedings of the 13th European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014.

40. Danelljan, M.; Häger, G.; Khan, F.; Felsberg, M. Accurate Scale Estimation for Robust Visual Tracking. In Proceedings of the British Machine Vision Conference, Nottingham, UK, 1–5 September 2014.
41. Wang, Q.; Gao, J.; Xing, J.; Li, B.; Maybank, S. DCFNet: Discriminant Correlation Filters Network for Visual Tracking. *J. Comput. Sci. Technol.* **2024**, *39*, 691–714.
42. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pfugfelder, R.; Zajc, L.C.; Vojir, T.; Bhat, G.; Lukezic, A.; Eldesokey, A.; et al. The sixth visual object tracking VOT2018 challenge results. In Proceedings of the Computer Vision—ECCV 2018 Workshops (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–53.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.