sensors

MDPI

Article

Transformer Feature Enhancement Network with Template Update for Object Tracking

Xiuhua Hu [1,2,*], Huan Liu [1,2], Yan Hui [1,2], Xi Wu [1,2] and Jing Zhao [1,2]

[1] School of Computer Science and Engineering, Xi'an Technological University, Xi'an 710021, China; 15938373926@163.com (H.L.); yanxh_xatu@163.com (Y.H.); wuxi786931646@163.com (X.W.); coffeyzj@sina.cn (J.Z.)

[2] State and Provincial Joint Engineering Laboratory of Advanced Network, Monitoring and Control, Xi'an 710021, China

[*] Correspondence: huxhxatu@163.com

Abstract: This paper proposes a tracking method combining feature enhancement and template update, aiming to solve the problems of existing trackers lacking global information attention, weak feature characterization ability, and not being well adapted to the changing appearance of the target. Pre-extracted features are enhanced in context and on channels through a feature enhancement network consisting of channel attention and transformer architectures. The enhanced feature information is input into classification and regression networks to achieve the final target state estimation. At the same time, the template update strategy is introduced to update the sample template judiciously. Experimental results show that the proposed tracking method exhibits good tracking performance on the OTB100, LaSOT, and GOT-10k benchmark datasets.

Keywords: object tracking; transformer architectures; feature enhancement; template update strategy

check for updates

Citation: Hu, X.; Liu, H.; Hui, Y.; Wu, X.; Zhao, J. Transformer Feature Enhancement Network with Template Update for Object Tracking. *Sensors* **2022**, *22*, 5219. https://doi.org/10.3390/s22145219

Academic Editor: Qing Guo

Received: 3 June 2022
Accepted: 9 July 2022
Published: 12 July 2022

1. Introduction

As one of the challenging and significant research tasks in the field of computer vision, object tracking is the tracking of the position of the target in each frame by using the features of the initial frame in successive video frames in the subsequent frames. Although object tracking algorithms have achieved good results in recent years, the difficulties in the tracking process, such as illumination variations, target appearance variations, occlusion and disappearance, still make object tracking challenging research. We systematically analyzed the literature [1,2] for the current deep learning-based tracking methods, benchmark datasets, and evaluation metrics, which laid the research foundations and research direction for the proposal of this paper.

Currently, most attention is devoted to tracking using the matching ability of the Siamese network. SiamFC [3] uses a Siamese network architecture to extract the features of the upper and lower branch in parallel and determines the target's location through similarity calculation to balance the accuracy and real-time performance of the tracking problem. Many advanced trackers are algorithmic improvements based on this. SiamBAN [4] adopts a Siamese network structure and anchor-free strategy to remove the pre-defined anchor, thus reducing the model parameters holistically and improving the speed further. ULAST [5] extends the Siamese tracking method by investigating forward and backward video tracking to obtain self-supervised information, which is superior to previous unsupervised methods.

SiamRPN [6] begins with a coarse target localization and then uses the RPN network to process the features to obtain a more accurate target location. To acquire deeper semantic features. SiamRPN++ [7] uses a deep convolutional network and three RPN networks to fuse shallow and deep features, thus increasing the discriminant ability of the network. C-RPN [8] utilizes multiscale features to enhance the feature representation and improve the robustness of the tracking network. Due to the design concept of anchor in RPN, it increases the number of hyperparameters and computational burden to some extent.

DaSiamRPN [9] enriches hard negative sample data by constructing negative samples with semantic meaning. The discriminative learning ability of the network is improved. The idea of SiamDW [10] is like SiamRPN++ in that deeper and wider convolutional networks are introduced into the target tracking algorithm to improve the tracking performance of the algorithm. SiamMask [11] adds the idea of segmentation to the tracker and is a network that can be used to do both video target tracking and video target segmentation.

The attention mechanism can effectively suppress the influence of background interference and target variations on the tracker. To improve the feature representation without introducing too many uncontrollable hyperparameters, researchers have shifted the attention mechanism to the tracking domain. SASiam [12] constructed a two-branch network with semantic and appearance branches and added a channel attention mechanism SENet [13] to the semantic branch to weighted output feature maps. The two branches complement each other to improve feature representation ability. RASNet [14] developed an efficient tracker based on Siamese networks by combining residual attention, channel attention, and general attention. SiamAttn [15] enhanced the features extracted from the benchmark network on three attention modules to improve the robustness of the tracking algorithm.

Since self-attention can establish long-term dependencies between features. TrTr [16] used a transformer [17] combined with an online update method to improve the performance of the tracking algorithm. TransT [18] employed a transformer to design a tracking algorithm that allows the tracker to obtain better classification and regression results. UTT [19] is a transformer-based tracking algorithm that solves the tracking problem in different scenes using a paradigm. This tracker uses correlation between target features and tracked video frame features for localization of the tracked target. The Stark [20] algorithm models the global spatial–temporal feature relationships using the encoder–decoder transformer structure. The method converts object tracking into a straightforward bounding box prediction problem without employing any suggested or predefined anchors, simplifying the existing tracking pipeline. TransCenter [21] proposes the use of dense pixel-level multi-scale queries in a transformer dual decoder network. It can globally and robustly push the heat map at the center of the target, and the robustness of the extra-high tracking algorithm.

Some classical tracking algorithms consider channel information alone, lacking attention to context information and yielding inadequate feature representation capabilities. Inspired by references [16–21] and the advantage that the transformer model can effectively establish associations between feature information, this paper proposes a feature enhancement network with an improved transformer structure.

In the tracking process, the tracking algorithm should be able to adapt to the appearance change of the target in time. TA-Siam [22] designs a template adjustment module that adaptively adjusts the current template using the characteristics of subsequent frames, effectively overcoming the Siamese network model-shifting problem. DSiam [23] proposed a dynamic twin network that incorporates a "generalized morphological learning model" that learns the apparent changes in the target and background suppression online from previous frames.

GradNet [24] proposed a method to apply gradient information to template update, which mainly uses gradient information to update the current template online through forward and back propagation. MemTrack [25] designed a dynamic memory network for object tracking update templates. The network takes control mainly by an external memory block LSTM with attention mechanism and has good adaptability to changes in appearance. This method is too complicated, which increases the computational overhead and affects the real-time performance of the tracking algorithm.

The related research work provides useful ideas for this paper to design feature enhancement networks combining channel attention and transformer models based on the architecture of Siamese networks, as well as being conducive to the paper to design a reasonable and effective model update strategy.

In general, the main contributions of this work can be summarized as:

- In this paper, a feature enhancement module is designed to enhance the spatial–temporal and channel-wise saliency of the features extracted from the benchmark network, which can make the tracker automatically focus on the beneficial feature information and improve the feature characterization ability.
- In this paper, a template update strategy is introduced. By dynamically updating the target template, the impact of target appearance changes on the tracker is better mitigated.
- In this paper, the proposed tracking algorithm achieves state-of-the-art tracking performance on three challenging benchmarks, OTB100, LaSOT, and GOT-10k.

The remaining contents of this paper are organized as follows. In Section 2, we briefly describe the problems to be solved in the paper. In Section 3, our proposed tracking method is explained in detail. In Section 4, our tracking method is evaluated and analyzed on three challenging benchmarks. In Section 5, the whole paper is concluded briefly.

2. Problem Description

The fully convolutional Siamese network has two same network branches, which can be used to measure the similarity of two inputs and is suitable for the object tracking task. As a typical, full-convolution Siamese network tracking algorithm, SiamFC, the upper branch is the target template branch, and the lower branch is the search branch. The template image z and the search image x are sent to the upper and lower branch, respectively. The feature vectors f_z and f_x are taken from the input image by convolutional neural network φ (AlexNet). f_{map}, the feature response mapping, is obtained after performing the intercorrelation operation (*) on the two feature vectors.

$$f_z = \varphi(z) \tag{1}$$

$$f_x = \varphi(x) \tag{2}$$

where, f_z and f_x refer to the two feature maps obtained from the template image and the search image through the CNN convolutional network, and $\varphi(.)$ refers to the AlexNet convolutional neural network.

After obtaining two feature vectors by a fully convolutional Siamese network, we calculate the similarity of the two feature vectors. We use the sliding window to obtain the final response graph. The higher response is considered the location of the target. This method is simple and fast and specifically expressed as:

$$f_{map} = f_z * f_x + b \tag{3}$$

where b represents the bias, is the one-dimensional vector, and * represents the convolution operation.

This tracking algorithm uses a shallow network for feature extraction. Due to the small number of convolutional layers, it becomes difficult to extract deep semantic information. Therefore, it cannot take full advantage of the representation capability of a deep convolutional neural network. Meanwhile, the matching template is always the sample template of the initial frame during the tracking process. It cannot accurately adapt to the change of the target in real-time, which is not conducive to the robustness of the tracking algorithm. Moreover, the simple calculation of the distance between feature vectors to determine the similarity of targets has low accuracy.

To achieve better tracking performance, this paper uses deep ResNet50 to extract deeper semantic information. Combining shallow information, channel attention, and transformer architecture, a feature enhancement network to improve feature representation and introduce template updates to adapt to changes in target appearance was proposed. Simultaneously, according to the characteristics of the upper and lower branches of the

Siamese network, the tracking task is regarded as two sub-tasks of classification and regression. The classification branch separates the target from the background and the regression branch accurately obtains the bounding box of the target, and locates the tracking target accurately by combining the characteristics of both.

3. Transformer Feature Enhancement Network and Template Update for Object Tracking

Based on the SiamFC object tracking algorithm, an object tracking method combining feature enhancement and template update was proposed by dividing the tracking task into two branches, classification, and regression, according to the network structure characteristics of the Siamese network. Specifically, deep semantic features with ResNet50 to extract, and deep semantic features and shallow features for fusion after passing them through channel attention (attn), and the fused features were respectively fused through two encoders to construct the long-term dependence between features. Cross-attention in the transformer decoder was used to exchange information between the features of the two branches, highlight useful global context information and channel information, suppress the interference of similarity targets, and improve the ability of feature representation. In the meantime, a real-time template update strategy was introduced to alleviate the influence of target appearance changes and improve the robustness of the tracker. The overall framework is shown in Figure 1.

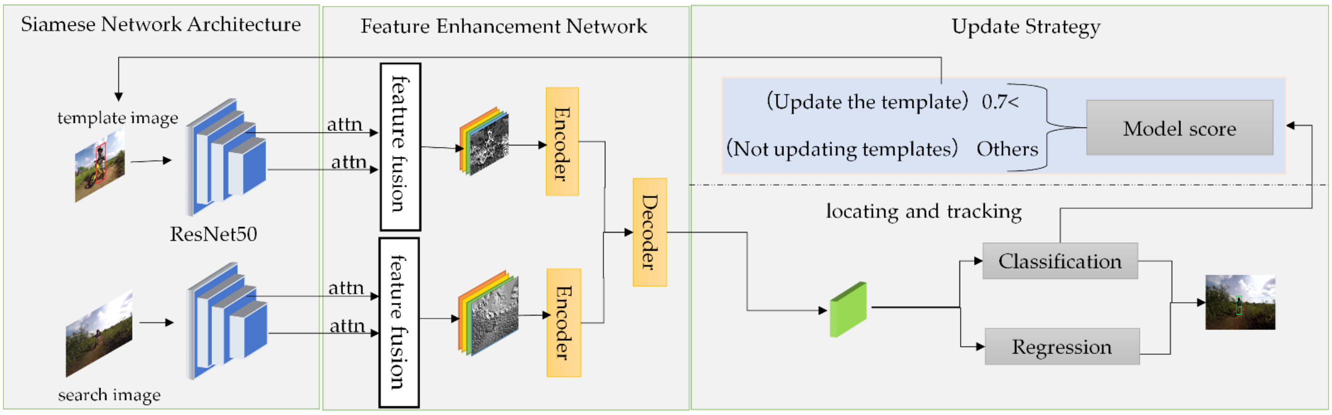


Figure 1. Overall framework of object tracking method based on transformer feature enhancement network with template update.

3.1. Input of The Benchmark Network

ResNet50 network is a benchmark network for feature extraction as used. Compared with the original AlexNet network, the benchmark network can extract deeper semantic information and improve the ability of target judgment and interpretation. For the input to the network, a pair of images selected from the video frames of the training dataset, namely, the template image and search image, feed into the Siamese network architecture and obtain the desired features through the benchmark network.

3.2. Feature Enhancement Process

We obtained feature maps with increased representational ability by constructing feature enhancement networks in order to retain more significant target features during tracking and strengthen the correlation between them. The feature enhancement network includes a feature fusion part based on a channel attention mechanism and a long-term dependency establishment part of the transformer.

3.2.1. Feature Fusion Based on Channel Attention Mechanism

Since the convolution operation only takes place in the local space, it is rare to gain enough information to extract the relationships between channels. The channel attention module calculates the weights of each channel of the input image. Then it fuses the shallow and deep features of the channel attention, focusing on the channel containing critical information and enhancing the feature representation ability.

Template images and search images are passed through the ResNet50 convolutional neural network to obtain the features of the last two stages. Then, they are passed through the channel attention mechanism separately to calculate the channel weight coefficients. The original feature maps are corrected using the resulting channel weight coefficients to obtain the enhanced attention feature maps. Finally, the features of the two stages are fused, where the feature fusion module is up-sampling with bilinear interpolation to unify the features dimension. The fusion process of shallow features and deep features is shown in Figure 2.

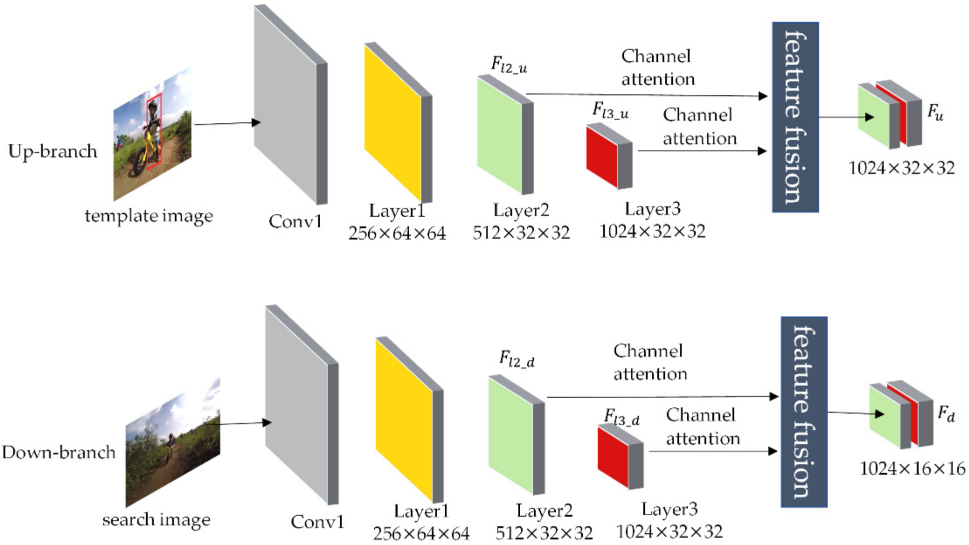


Figure 2. Diagram for fusion of shallow features and deep features.

The template image and the image go through the four stages of ResNet50. Then the feature vectors F_{l2_u}, F_{l3_u}, F_{l2_d}, and F_{l3_d} of Layer2 and Layer3 are extracted separately. The critical spatial information of the two-stage feature vectors are enhanced on the channel. The two-stage features are fused using the feature fusion module to obtain the feature vectors F_u and F_d of the upper and lower branch. The channel attention mechanism corrects the channel features, and the amended features retain the valuable features and eliminate the non-valuable ones. GAP represents global average pooling. A schematic diagram of channel attention is shown in Figure 3.

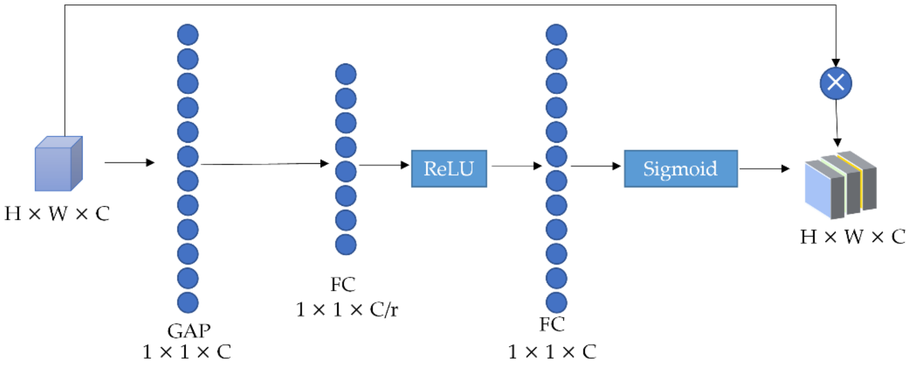


Figure 3. Schematic diagram of channel attention.

Firstly, the spatial dimension of the input F_{l2_u}, F_{l3_u}, F_{l2_d}, and F_{l3_d} feature maps are encoded as a global feature through global average pooling and the input image of size H × W × C to 1 × 1 × C channel descriptors by compression. The specific process of the upper and lower branch features through channel attention is defined as Formulas (4) and (5).

$$S_{lk\_u} = \frac{1}{\mathrm{H} \times \mathrm{W}} \sum\nolimits_{i=1}^{\mathrm{H}} \sum\nolimits_{j=1}^{\mathrm{W}} \varphi(F_{lk\_u}) \quad k = 2,3 \tag{4}$$

$$S_{lk\_d} = \frac{1}{\mathrm{H} \times \mathrm{W}} \sum\nolimits_{i=1}^{\mathrm{H}} \sum\nolimits_{j=1}^{\mathrm{W}} \varphi(F_{lk\_d}) \quad k = 2,3 \tag{5}$$

where, S_{lk_u}, S_{lk_d} denote the global information, H and W denote the height and width of the feature graph, and $\varphi(F_{lk\_u})$, $\varphi(F_{lk\_d})$ denote the feature vectors of the two stages of the upper and lower branch, and k denotes the index of layer2 and Layer3. The C/r dimensional vector comes from processing the result of the global average pooling through a fully connected layer. Then after performing a ReLU activation function, the C/r dimension vector is transformed back to a C dimensional vector by a fully connected layer. Finally, the final weight matrix is attained by a Sigmoid activation function. The process is defined as Formulas (6) and (7).

$$s_u = \sigma(W_2\delta(W_1 S_{lk\_u})) \quad k = 2,3 \tag{6}$$

$$s_d = \sigma(W_2\delta(W_1 S_{lk\_d})) \quad k = 2,3 \tag{7}$$

where, W_1, W_2 represent the weights of the two fully connected layers, δ and σ represent ReLU and Sigmoid activation functions respectively, s_u and s_d represent the weight matrix finally obtained. Finally, multiply the activation values of each channel learned by the original feature maps.

$$F_{uk} = s_u * F_{lk\_u} \quad k = 2,3 \tag{8}$$

$$F_{dk} = s_d * F_{lk\_d} \quad k = 2,3 \tag{9}$$

where, F_{uk} and F_{dk} represent channel feature maps combined with weight coefficients; s_u and s_d represent channel attention weights; F_{lk_u} and F_{lk_d} are feature maps of original Layer2 and Layer3. In the whole feature fusion process based on the channel attention mechanism, weight coefficients of each channel are obtained through learning, and corresponding weights are assigned to the original feature graph, thus making the model more discriminating for the features of each channel. After obtaining the two-stage channel feature map, the two fuses receive the feature maps F_u and F_d with the most representation ability of the upper and lower branch.

3.2.2. Transformer Long-Term Dependency Building Part

The traditional transformer includes an encoder and decoder. The encoder includes two sub-layers, a self-attention module, and forward propagation. The self-attention module serves to learn the internal relationship between features. It can effectively capture the internal correlation between data and features. The decoder consists of three sub-layers, a self-attention module, an interactive attention module, and forward propagation. The self-attention module and forward propagation have the same function as the modules in the encoder. The interactive attention module works to learn the relationship between the original feature and the target feature.

Self-attention gives query(Q), key(K), and value(V) based on the embedded feature vectors and calculates the similarity or association between them based on Q and K. In this paper, we selected to calculate the similarity using the dot product of two vectors, and the obtained scores as normalized. Then the weighted sum based on the normalized weight coefficients are applied to V. The calculation of self-attention can be defined as for Formula (10).

$$self - attn(Q, K, V) = softmax\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{10}$$

where Q, K and V are linear transformations derived from features.

To obtain salient features. The internal relationships between the template image feature and the search image feature as learned using the encoder on the upper and lower branch, The learned features are interacted between messages using a decoder without a self-attention mechanism. Among them, FFN is a feed-forward neural network. The principal structure of the transformer designed in this paper is shown in Figure 4.

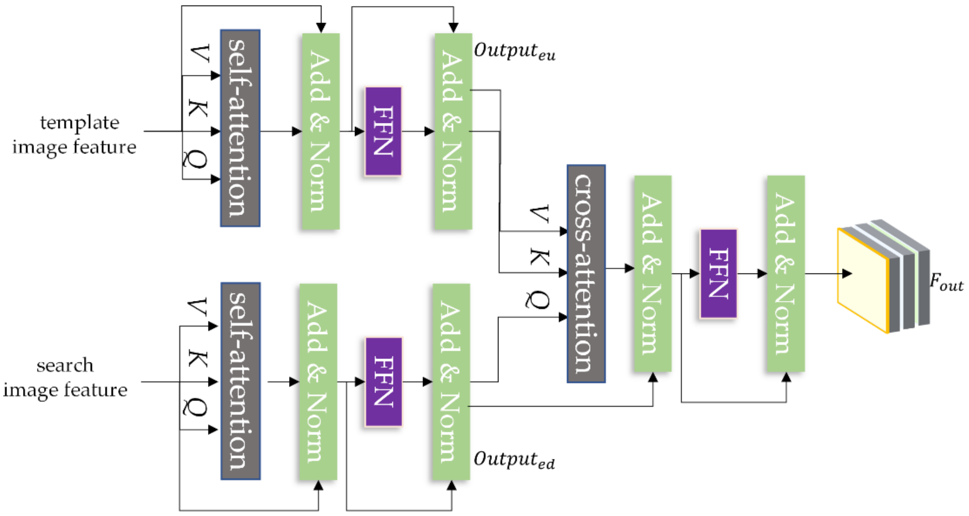


Figure 4. Principle diagram of transformer structure.

In this paper, we used a single-headed self-attention mechanism, whose internal operation can be expressed as.

$$Head(Q,K,V) = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{11}$$

where, W_i^Q, W_i^K, W_i^V respectively represent the weight matrix vectors of Q, K and V, which are the same as Q, K and V from self-attention.

The upper and lower branch encoders receive the channel enhancement feature vectors F_u and F_d. Before receiving the feature, the feature needs a dimensional conversion. The features need to convert into the feature vectors $\widetilde{F_u}$ and $\widetilde{F_d}$ required by the encoder. The self-attentiveness of the input template image feature is computed by Formula (10) while adding position encoding at the location of each feature.

$$Output_{eu} = \widetilde{F_u} + Head\left(\widetilde{F_u} + P_z, \widetilde{F_u} + P_z, \widetilde{F_u}\right) \tag{12}$$

$$Output_{ed} = \widetilde{F_d} + Head\left(\widetilde{F_d} + P_z, \widetilde{F_d} + P_z, \widetilde{F_d}\right) \tag{13}$$

where, P_z is the position encoder, $Output_{eu}$ and $Output_{ed}$ represent the output of the upper and lower branch encoder. The decoder is used to interact information on the output of the two branches of the encoder and the feature enhancement network to obtain a high-quality feature vector for classification and regression.

$$F_{out} = Output_{ed} + Head\left(Output_{e_d} + P_q, Output_{eu\_k} + P_k, Output_{eu\_v}\right) \tag{14}$$

where, $Output_{eu\_k} + P_k$, $Output_{eu\_v}$ are the K and V of the encoder branch, the information interaction is carried out in the cross-attention module, and F_{out} is the final output after the information interaction of the upper and lower branch features by the decoder, which is used for the subsequent positioning and tracking. The bounding box B^* of the target is determined by classification and regression networks.

3.3. Design of Update Strategy

In the tracking process, if the target disappears or is blocked, it is not recommended to update the template. In this case, judgment needs to be made on the obtained template to avoid updates leading to a deterioration of the tracking effect. In this paper, target confidence scores judgment by adding a target confidence score at the position of the classification branch. The classification branch has 1024 vectors of length 2, representing the foreground and background scores. The foreground score with the maximal confidence score is extracted and compared with the set threshold. Scores that exceed the set threshold α (>0.7) update their images to the position of the initial frame. This method can make full use of the temporal contextual information of the tracking process and alleviate the problem of changing target appearance. Where FC1, FC2 and FC3 denote fully connected layers, the classification branch-based template update strategy is shown in Figure 5.

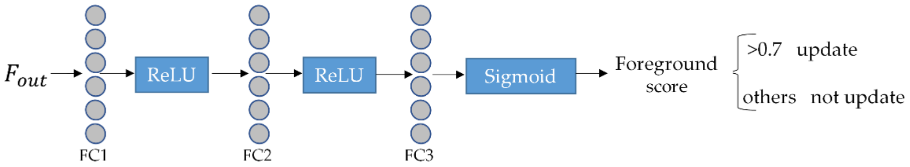


Figure 5. Principle diagram of template update strategy based on classification branch.

3.4. Algorithm Implementation

During the implementation of the whole tracking algorithm, offline training was carried out using the publicly available datasets LaSOT [26], GOT-10k [27], and COCO [28] to obtain the final network model. The target state of the template image was pre-set. The status of the search image was determined by network model tracking prediction. The shallow and deep features performed fusion after passing through the channel attention module. The fused upper and lower branch features were dimensionally adjusted and input to the two encoders to build the internal connection between the features. Finally, the output of the two encoders was used for information interaction with the decoder to achieve feature enhancement. The update strategy can effectively alleviate the influence of the change in target appearance. Updating the initial template from subsequent frames can enrich the time context information in the tracking process and make the model more robust. Finally, the location of the target was determined by classification and regression. The specific implementation process is as follows (Algorithm 1).

Algorithm 1 Procedure of the proposed method

```
$F_{l2\_u}, F_{l3\_u} \leftarrow \varphi(z)$; # Feed the template into CNN and the feature vectors of the last two stages are taken out
repeat
$F_{l2\_d}, F_{l3\_d} \leftarrow \varphi(x)$;
$s_u, s_d \leftarrow$ attn $(F_{l2\_u}, F_{l3\_u}, F_{l2\_d}, F_{l3\_d})$; # Feed the features to the channel attention network attn, get the weight matrix $s_u, , s_d$
$F_u(F_d) \leftarrow s_u(s_d) * F_{lk\_u}(F_{lk\_d})$ k $= 2, 3$; # Fuse the weight matrix and the original feature vector. Get the feature map , $F_u, F_d$
$Output_{eu}(Output_{ed}) \leftarrow \widetilde{F}_u\left(\widetilde{F}_d\right) +$ Encoder $\left(\widetilde{F}_u\left(\widetilde{F}_d\right)\right)$; # Feature map $F_u, F_d$ is processed into $\widetilde{F}_u, \widetilde{F}_d$ and input to Encoder
$F_{out} \leftarrow$ Decoder $(Output_{eu}, Output_{ed})$; # Fuse the Encoder information in the decoder. Get the final feature map $F_{out}$
$B^* \leftarrow$ Classification $(F_{out})$ and Regression $(F_{out})$; # Determining the location of targets through classification and regression networks
Foreground Score $\leftarrow$ Classification $(F_{out})$; # Foreground score is greater than 0.7 and the template is updated
until end of sequence
```

4. Experimental Results and Analysis

In order to validate the proposed tracking method, the model was initialized using the parameters pre-trained on ImageNet [29]. LaSOT [26], GOT-10k [27], and COCO [28] datasets were used for offline training of the whole network. During training, the template image was 127 × 127, the search image was 256 × 256, and batch size was 64, optimized using the AdamW [30] optimizer. The learning rate of the benchmark network was set as 10^{-5}, the learning rate of other modules was set as 10^{-4}, weight decay and momentum were 10^{-4} and 0.9, L_1 loss and L_{Giou} loss weight were 5 and 2, respectively.

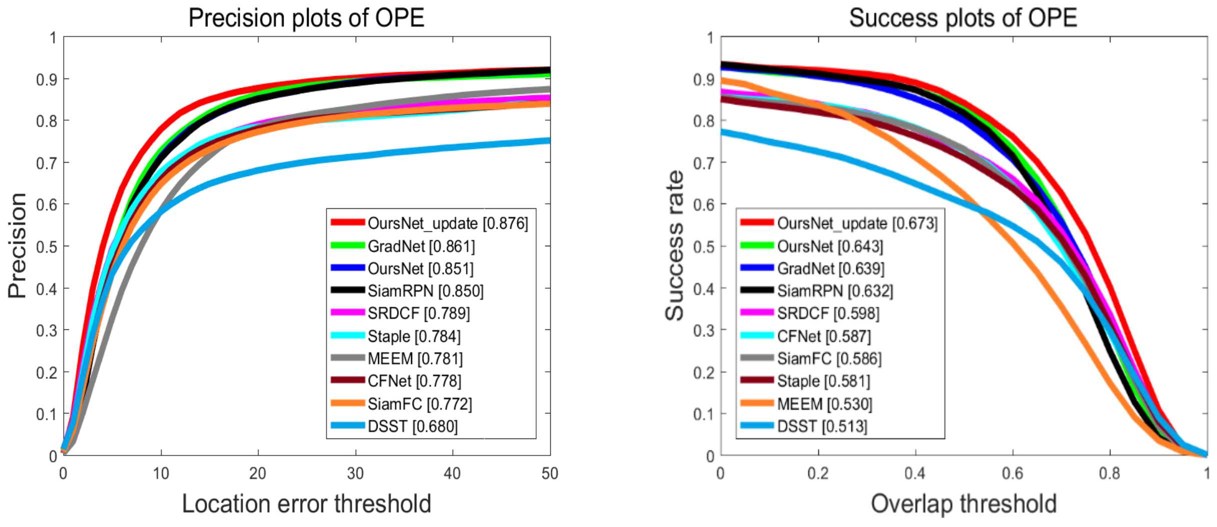
4.1. Quantitative Analysis

To quantitatively analyze the overall performance of the tracker under the interference of complex background and other factors, the proposed method was compared with typical tracking algorithms on OTB100 [31], GOT-10k [27], and LaSOT [26] test datasets.

4.1.1. Comparative Analysis with Typical Tracking Algorithms on OTB100 Dataset

The OTB100 dataset contains 11 tracking challenge factors, 98 videos, and 100 test scenarios, enabling fair and accurate evaluation and comparison of various tracking algorithms. In this paper, the compared method was evaluated with the one-pass evaluation (OPE), and two main metrics of precision plots and success rate plots were adopted to evaluate the performance.

The trackers compared on the OTB100 dataset included SiamFC [3], SiamRPN [6], CFNet [32], GradNet [24], SRDCF [33], Staple [34], DSST [35], and MEEM [36]. Figure 6 shows the precision and success rate plots for eight state-of-the-art (SOTA) trackers on the OTB100 dataset.



(**a**) (**b**)

Figure 6. Performance evaluation results of different algorithms on OTB100 dataset. (**a**) Precision plots; (**b**) success plots.

As can be seen from Figure 6, the performance of the proposed tracking algorithm was significantly better than that of SiamFC and other algorithms. Compared with SiamFC, the precision and success rate of the proposed method were improved by 10.4% and 8.7%, respectively, due to the addition of the transformer feature enhancement network and template update strategy to the base network SiamFC, indicating the effectiveness of the proposed method.

CFNet combines correlation filtering with CNN, using a correlation filtering layer to implement online update function, which allows the network to perform well with shallow features. The performance was still not as good as the state-of-the-art trackers due to the boundary effects that CF has difficulty handling.

SiamRPN algorithm uses shallow AlexNet for feature extraction, but it cannot extract deep semantic features and ignores the context information. The tracking algorithm proposed in this paper uses ResNet50 to extract deep semantic features. In addition, the proposed feature enhancement network could effectively improve the feature representation ability. Compared with SiamRPN, the precision and success rate of the proposed method in this paper are improved.

The accuracy of the GradNet algorithm exceeded that of the tracking algorithm proposed in this paper. However, the success rate was slightly lower than the algorithm proposed. It indicates that the proposed tracking algorithm could adapt the bounding box to the target size well when tracking the target.

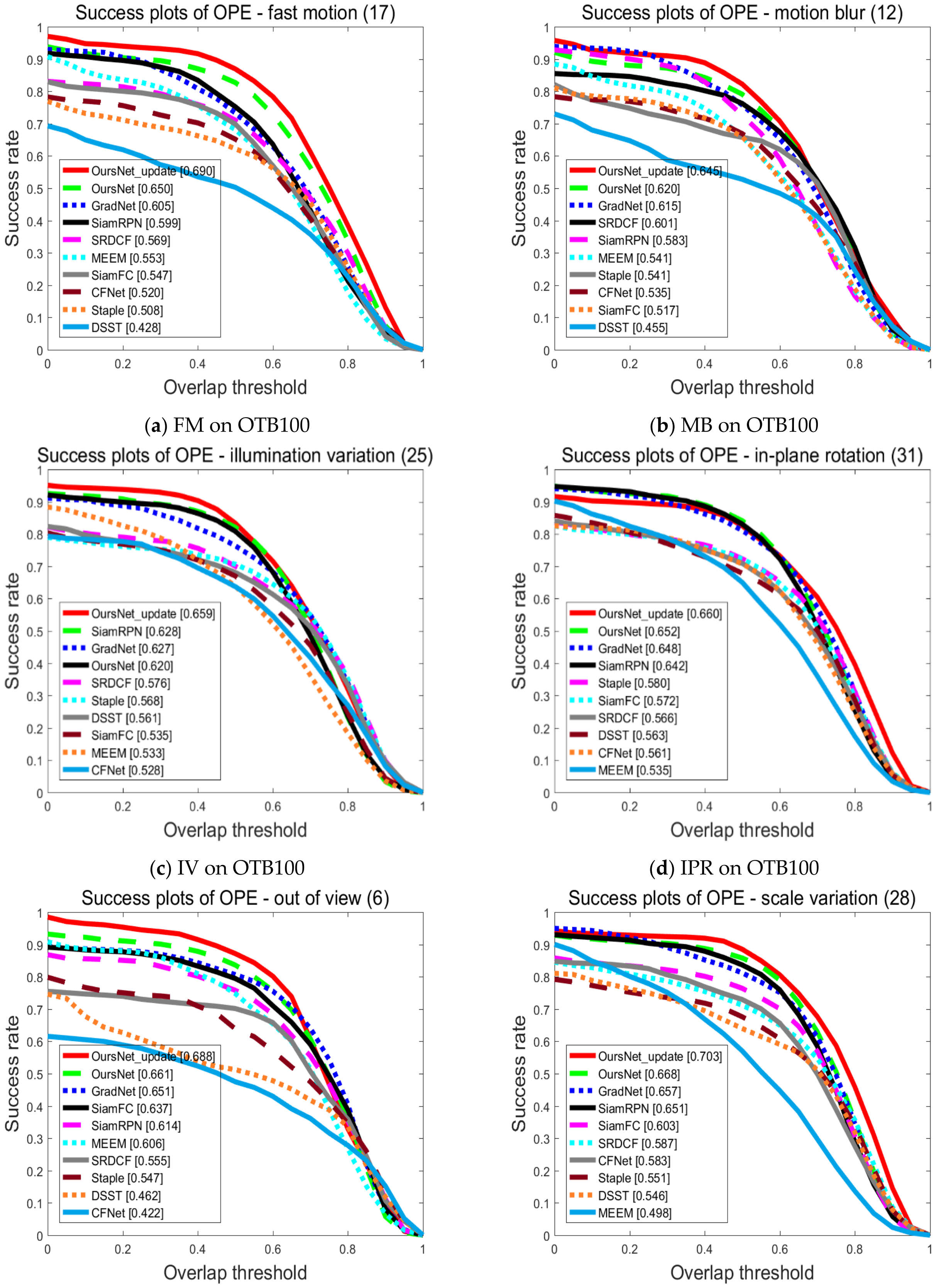
Experimental results show that this method is superior to some classical tracking algorithms in precision and success rate.

Figure 7a–k provides the detailed results of the success rate on OTB100 for eleven attributes, including fast motion (FM), background clutter (BC), motion blur (MB), deformation (DEF), illumination variation (IV), in-plane rotation (IPR), low resolution (LR), occlusion (OCC), out of view (OV), scale variation (SV), out-plane rotation (OPR). For example, in the case of fast motion, a high adaptive capability of the tracking network becomes necessary. The transformer in the feature enhancement network proposed in this paper could effectively build the long-term dependence relationship of features and accurately obtain feature information in the case of fast movement. In the motion blur case, the tracking network needs to be capable of accurate target tracking in the low-resolution video frame. The transformer feature enhancement network proposed in this paper can make up for the influence of motion blur on the tracker and improve the performance of the tracker.

Further to Figure 7a–f, in this paper, the proposed tracking algorithm was achieved by introducing the transformer feature enhancement network and template update strategy onto the basic network SiamFC, thus, making the tracker perform better on illumination changes, motion blur, fast motion, and target disappearance challenge factors. The robustness of the tracking algorithm was further improved. The transformer feature enhancement network enabled the tracking algorithm to obtain the most representative feature vector, and the template update strategy enabled the tracking algorithm to adapt to the appearance change of the target well.

From Figure 7g–i, the SiamRPN had good tracking performance under the factors of background interference and out-of-plane rotation. However, due to the shallow benchmark network used, the semantic features were difficult to acquire. In addition, in the tracking process, it was difficult to adapt to the situations when the target was moving fast or the appearance changed only depending on the feature information of the initial frame. At the same time, due to the design of the anchor frame, too many super parameters were introduced, so the speed was relatively slow.

As observed in Figure 7j,k GradNet outperformed other trackers due to the inclusion of gradient information and online template updates, under the low resolution and occlusion challenging factors of the OTB100 dataset. Because GradNet mainly solves the problem of updating templates online, it does not deal with the ability to represent features. Therefore, the performance was slightly lower than the tracking algorithm proposed in this paper under the conditions of motion blur and illumination variation.



(**a**) FM on OTB100

(**b**) MB on OTB100

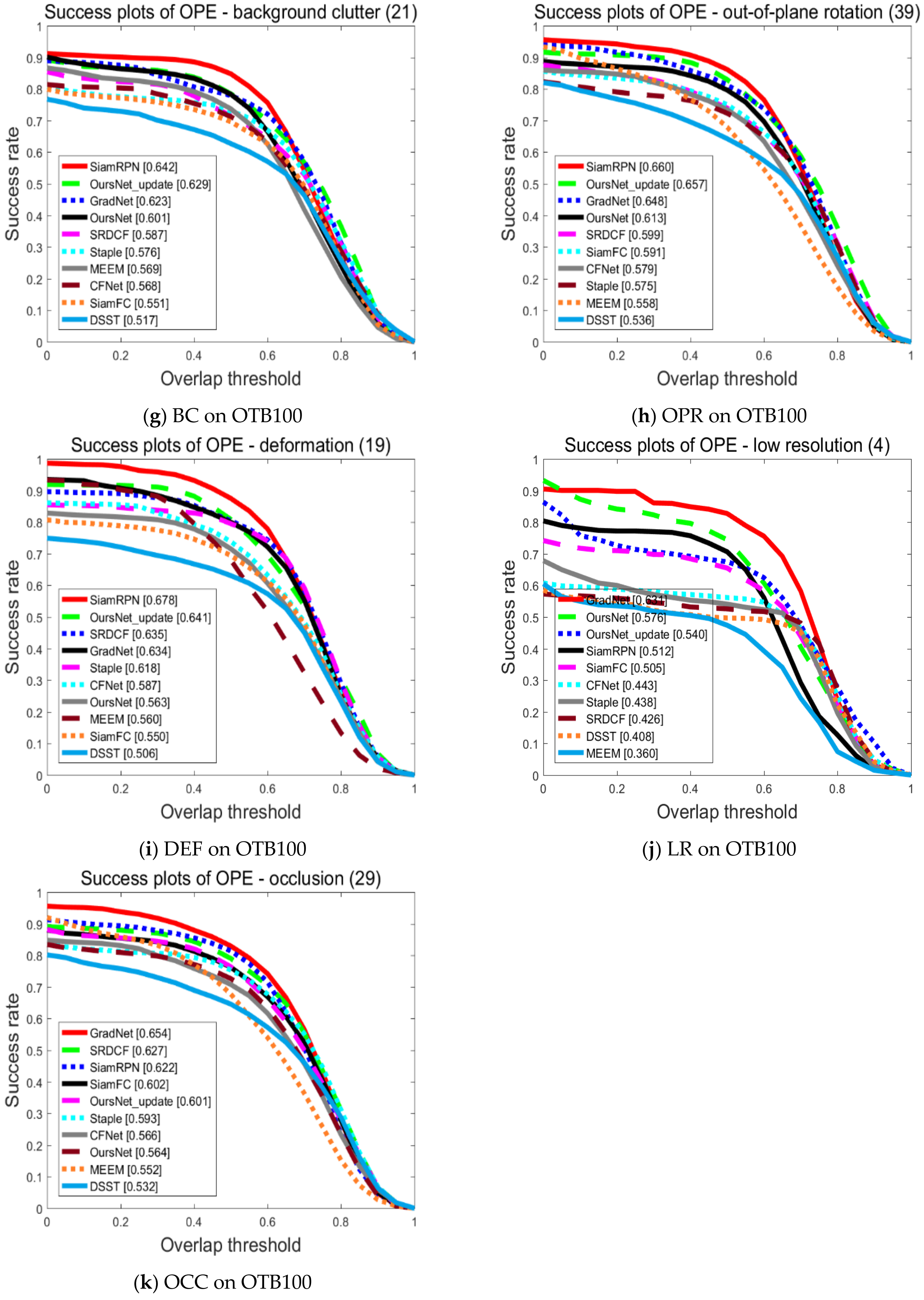
(**c**) IV on OTB100

(**d**) IPR on OTB100

(**e**) OV on OTB100

(**f**) SV on OTB100

Figure 7. *Cont.*



(**g**) BC on OTB100

(**h**) OPR on OTB100

(**i**) DEF on OTB100

(**j**) LR on OTB100

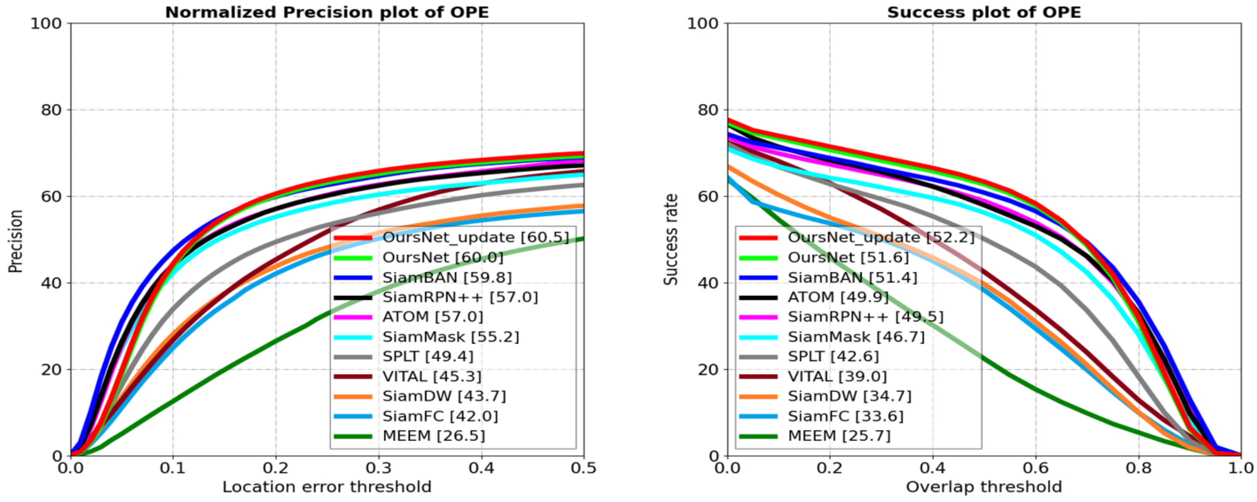
(**k**) OCC on OTB100

Figure 7. Performance evaluation results of different algorithms on OTB100 dataset for various attributes.

4.1.2. Comparative Analysis with Typical Tracking Algorithms on LaSOT Dataset

The LaSOT dataset contains 1400 videos with an average of 2512 frames per sequence. Each frame is carefully inspected and manually labeled, with results visually verified and corrected when needed, representing by far the largest single-target tracking dataset with dense annotation for depth tracking training and realistic evaluation of tracking algorithms. We used two evaluation metrics, the normalized precision plots P_{Norm} and the success plot AUC, for performance evaluation. AUC represents area under curve, P_{Norm} denotes normalized precision rate.

The trackers compared on the LaSOT dataset include SiamBAN [4], SiamRPN++ [7], ATOM [37], TrTr-online [16], TrTr-offline [16], SiamFC [3], SiamMask [11], SiamDW [10], VITAL [38], SPLT [39], MEEM [36], UpdateNet [40], DSiam [23], and ECO [41]. Figure 8 shows the normalized precision and success rate plots for nine SOTA trackers on the LaSOT dataset.



(**a**) (**b**)

Figure 8. Performance evaluation results of different algorithms on LaSOT dataset. (**a**) Normalized precision plots; (**b**) success plots.

We can see this in Figure 8. The proposed tracker achieved better performance in the long-term tracking dataset LaSOT attributed to the introduction of the proposed feature enhancement network and template update strategy. The tracking algorithm improved accuracy by 18.5% and the success rate by 18.6%, compared to the base SiamFC network.

The biggest contribution of SiamRPN++ is to change the network structure of twin trackers, using a deeper network instead of a shallow network. The main drawback is that there is no template update, and it is not possible to change the template adaptively when the target appearance changes. Furthermore, the custom anchor frame setting introduces a fuzzy similarity score, which severely hinders robustness and leads to a lower performance compared to advanced trackers.

SiamBAN is an anchor-free based object tracking algorithm that may not contain sufficient target information for accurate regression. Moreover, it lacks the flexibility to handle target changes because it does not consider the internal relationship between features and uses only the initial frame as a template. Therefore, the results are not very effective on the long-time dataset LaSOT.

SiamMask adds a segmentation branch alongside the classification branch and the regression branch, which causes the whole network to be difficult to train. Although ResNet50 is used as the network for extracting features, there is still much room for improvement in robustness.

The SiamDW tracking algorithm ignores changes in the target motion state, resulting in tracking failure when challenging factors such as rapid target motion and occlusion occur.

In this paper, the proposed feature enhancement network not only considered the key information on the channel, but also used the self-attention mechanism to establish long-term dependencies between features and improve the correlation of features, and the template update strategy could effectively adapt to the changes of target appearance. Therefore, the performance was slightly higher on long-time datasets LaSOT compared to trackers such as SiamBAN, SiamRPN++, SiamMask, and SiamDW.

Specifically, the compared performance evaluation results of the proposed method and typical template update trackers on LaSOT dataset are given in Table 1.

Table 1. Performance results of SOTA tracker with an update mechanism on the LaSOT dataset.

Methods	AUC	P_{Norm}
OurNet_update	52.2	**60.5**
OurNet	51.6	60.0
TrTr-online	**55.1**	-
TrTr-offline	46.3	-
UpdateNet-DaSiamPRN	47.5	56.0
UpdateNet-SiamFC	34.9	43.7
DSiam	30.3	40.5
ECO	32.4	33.8

The method used only the feature enhancement network proposed in this paper before introducing the template update strategy. The performance was over 5.3% compared to the TrTr-offline tracking algorithm that also uses the transformer architecture. This result shows that the features in the upper and lower branch were fused separately after passing through the channel attention. Then, the association between the features within each branch was performed by the encoder. Finally, the information from the upper and lower branch interacted through the decoder. The performance was improved compared to the decoder process where the upper branch is passed through one encoder and then inputs it to the lower branch.

The performance of TrTr-online tracking algorithm exceeded the tracking algorithm proposed in this paper. It is a combination of classification plus an online update method, which is slower. However, the tracking algorithm in this paper used a simpler linear update mechanism. The real-time performance of the tracking algorithm was preserved with a slight performance improvement.

UpdateNet-DaSiamPRN and UpdateNet-SiamFC were tested by using UpdateNet network on the original DaSiamRPN and the base network SiamFC. However, the performance was lower compared with the tracking algorithm proposed in this paper that combines the transformer and simple update mechanism strategies. This reflects the effectiveness of the tracker proposed in this paper.

The DSiam tracking algorithm learns an online update transformation in two separate branches to update the tracking model through a fast transformation module. However, the tracking method lacks the detection of tracking failures, resulting in low performance.

The ECO tracking algorithm updates the model every few frames to avoid model overfitting. However, its adaptability is reduced due to the underutilization of information during model updates.

As can be seen from Table 1, the tracking algorithm proposed in this paper significantly outperformed typical tracking algorithms such as DSiam and ECO in terms of performance.

4.1.3. Comparative Analysis with Typical Tracking Algorithms on GOT-10k Dataset

The GOT-10k dataset contains more than 10,000 real-world video clips and more than 560 categories, with more than 1.5 million manually marked frames, featuring rich scenes and difficult algorithm challenges. Average overlap (AO) and success rate (SR) are two evaluation metrics for the GOT-10k dataset, and $SR_{0.5}$ and $SR_{0.75}$ are two thresholds for the success rate. Therefore, the tracking algorithm proposed in this paper was tested on this dataset. The output results are uploaded to the official website to obtain the performance results of the test. The fairness and effectiveness of the tracking algorithm in this paper as demonstrated. Figure 9 shows the success rate plots for 18 SOTA trackers. It can be observed from the figure that the tracking algorithm proposed in this paper performed significantly better than the others on the GOT-10k dataset.

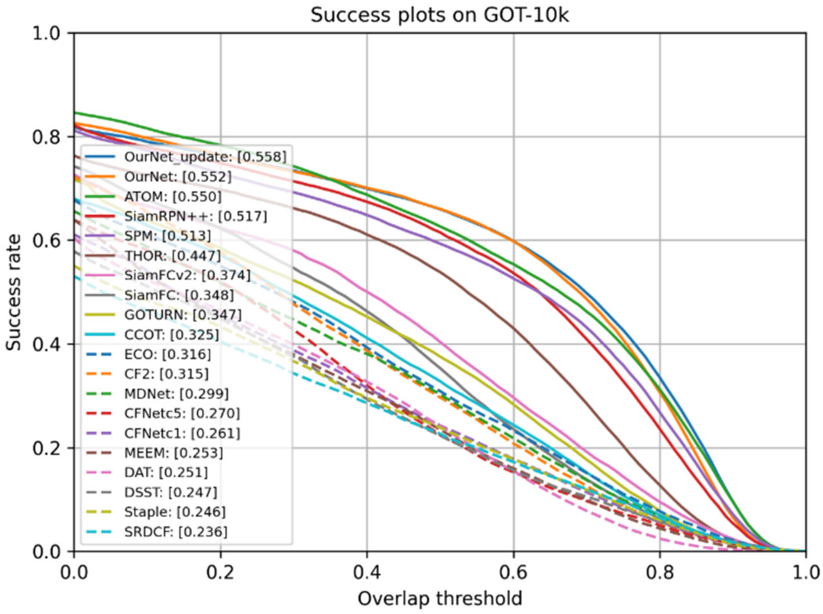


Figure 9. Comparison results of the GOT-10k benchmark.

Table 2 shows the performance details of the tracking algorithm proposed in this paper compared with typical tracking algorithms on the GOT-10k test dataset. The tracking algorithms compared include, SiamFC [3], SiamRPN [6], SiamRPN++ [7], MDNet [42], ATOM [37], ECO [41], THOR [43], SPM [44], GOTURN [45], Staple [34], and SRDCF [33].

Table 2. Performance comparison on GOT-10k benchmark.

Methods	AO	$SR_{0.5}$	$SR_{0.75}$
OurNet-update	**55.8**	**66**	**42.7**
OurNet	55.2	66	40.8
ATOM	55.0	63.4	40.2
SiamRPN++	51.7	61.6	32.5
SPM	51.3	59.3	35.9
SiamRPN	46.3	54.9	25.3
THOR	44.7	53.8	20.4
SiamFCv2	37.4	40.4	14.4
SiamFC	34.8	35.3	9.8
GOTURN	34.7	37.5	12.4
ECO	31.6	30.9	11.1
MDNet	29.9	30.3	9.9
Staple	24.6	23.9	8.9
SRDCF	23.6	22.7	9.4

ATOM refines multiple initial bounding boxes iteratively, which provides a significant improvement in tracking accuracy. However, the performance was slightly lower than the tracking algorithm proposed in this paper. Meanwhile, the method of target estimation in ATOM not only brings a heavy computational burden but also presents many additional hyperparameters that require careful tuning.

The SPM tracker distinguishes targets from backgrounds by introducing additional relational networks. However, the use of a matching mechanism only and the lack of online updates severely limit the performance of this tracker in dealing with similarity interferers.

THOR is a tracking algorithm based on dynamically updated target templates. The tracker uses greedy search in a local search window, resulting in a less than friendly performance in challenging benchmark tests.

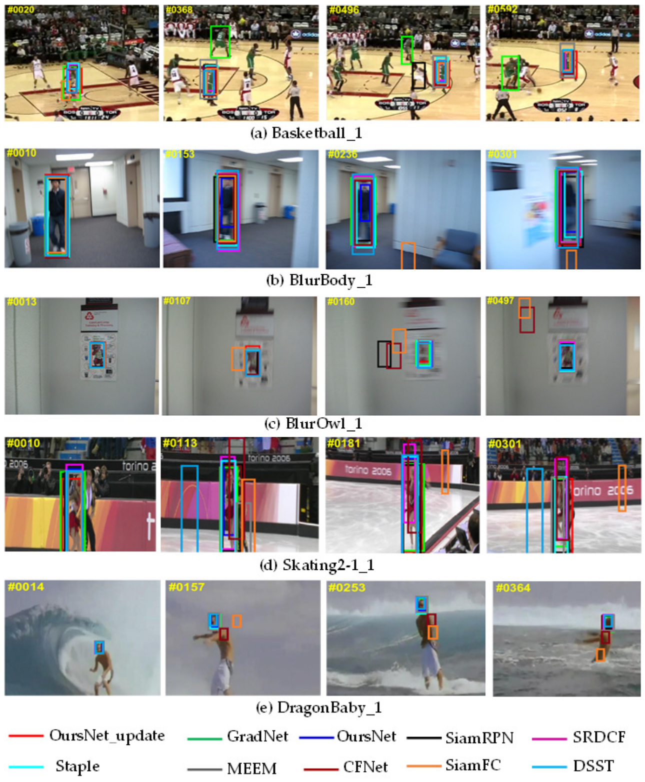
GOTURN uses a fully connected network to estimate corner coordinates directly, while objects in the background also have corner points. Furthermore, the relationship between the template and the test image is not effectively probing. Therefore, it is not easy to distinguish them from the corner points of the tracked target. To some extent, it affects the performance of the tracker.

The MDNet tracking algorithm does not incorporate any feature enhancement operation during feature extraction. Moreover, fine-tuning online makes it slower and requires time and effort to fine-tune. Therefore, the robustness of the tracking performance needs to be improved.

As can be seen in Table 2, the proposed tracking algorithm showed superior performance results in AO, $SR_{0.5}$, and $SR_{0.75}$ metrics. Compared to the base network SiamFC, AO, $SR_{0.5}$ and $SR_{0.75}$ were improved by 21, 30.7 and 32.9%, respectively. Compared with SiamRPN++, which introduces the RPN network, AO, $SR_{0.5}$, and $SR_{0.75}$ were improved by 4.1, 4.4 and 10.2%, respectively. The proposed tracking algorithm also outperformed the THOR and ECO tracking algorithms that use the same dynamic update template.

4.2. Qualitative Analysis

To further verify the model performance, the qualitative performance of the designed method and eight typical tracking algorithms were evaluated on the OTB benchmark, where the comparisons included SiamFC [3], SiamRPN [6], CFNet [32], GradNet [24], SRDCF [33], Staple [34], DSST [35], and MEEM [36]. The results of the visualization experiment in some moments of selected challenging test sequences are shown in Figure 10a–e.


(a) Basketball_1

(b) BlurBody_1

(c) BlurOwl_1

(d) Skating2-1_1

(e) DragonBaby_1

Figure 10. Sample tracking results of evaluated algorithms on different challenging sequences.

As shown in Figure 10a, the video sequence basketball_1 has background similarity interference and background clutter, which can have some performance impact on the tracking process. In frame 20, the bounding box of the SiamFC tracker shifts slightly. From frames 368, 496, and 592, we found that the GradNet and SiamRPN trackers failed to track. The tracking algorithm in this paper benefits from the fact that the proposed feature-enhanced network can track the target better than other trackers.

As observed in Figure 10b,c, in BlurBody_1 and BlurOwl_1 video sequences, the tracking algorithm in this paper fused the shallow and deep features of passing channel attention, while using the transformer to construct long-term dependencies. When there was target blurring, which led to the complete tracking failure of trackers such as SiamFC and CFNet, the tracker in this paper still demonstrated better robustness.

As viewed in Figure 10d, there was fast-moving and similar interference in the Skating2-1_1 video sequence. SiamFC and Staple trackers could not track due to the lack of ability to extract feature representations. In contrast, the tracker proposed in this paper and other trackers could track the target robustly.

As evidenced in Figure 10e, SiamFC and CFNet trackers failed to track the target accurately at 157 frames due to the interference factors of fast target change in the DragonBaby_1 video sequence. In this paper, we used a dynamic template updating mechanism, which could always accurately track the target position in the DragonBaby_1 video sequence where the target appearance changed or there was fast movement, reflecting the robustness of the tracking algorithm.

5. Conclusions

This paper proposes a tracking method combining feature enhancement and template update, which uses ResNet50 as a benchmark network for feature pre-extraction of template images and search images. The feature vectors obtained from it are fed into the proposed feature enhancement network, and the feature enhancement module consists of a channel attention mechanism and a reconfigured transformer. After passing the channel attention mechanism, the shallow features fuse with the deep features. Then, the internal relationships between features are established by encoders. A decoder is used to carry out the interaction of information with the output of the encoder. It enables the network to select the key information effectively and improve the feature representation.

Thanks to the introduction of a dynamic template update strategy that allows adaptive template updates when the appearance of the target changes and, simultaneously, enriches the temporal-context information between video frames, the robustness of the tracker is improved. The experimental comparison results show that the proposed tracking algorithm improved the tracking accuracy and success rate, and outperformed some existing classical trackers.

Author Contributions: Methodology, X.H.; validation, X.H.; writing—review and editing, X.H. and H.L.; software, H.L.; writing—original draft preparation, H.L.; investigation, Y.H.; formal analysis, X.W.; resources, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Natural Science Basic Research Project of Shaanxi Provincial Department of Science and Technology (No. 2022JQ-677), and the Natural Science Foundation of Shaanxi Provincial Department of Education (No. 18JK0383).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Most data generated of analyzed during this study are included in the submitted article. The datasets involved in this paper are all public datasets.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Marvasti-Zadeh, S.M.; Cheng, L.; Ghanei-Yakhdan, H.; Kasaei, S. Deep learning for visual tracking: A comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *23*, 3943–3968. [CrossRef]
2. Javed, S.; Danelljan, M.; Khan, F.S.; Khan, M.H.; Felsberg, M.; Matas, J. Visual Object Tracking with Discriminative Filters and Siamese Networks: A Survey and Outlook. *arXiv* **2021**, arXiv:2112.02838.
3. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 850–865.
4. Chen, Z.; Zhong, B.; Li, G.; Zhang, S.; Ji, R. Siamese box adaptive network for visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 14–19 June 2020; pp. 6668–6677.
5. Shen, Q.; Qiao, L.; Guo, J.; Li, P.; Li, X.; Li, B.; Feng, W.; Gan, W.; Wu, W.; Ouyang, W. Unsupervised learning of accurate siamese tracking. *arXiv* **2022**, arXiv:2204.01475.
6. Li, B.; Yan, J.; Wu, W.; Zhu, Z.; Hu, X. High performance visual tracking with siamese region proposal network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8971–8980.
7. 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 IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4282–4291.
8. Fan, H.; Ling, H. Siamese cascaded region proposal networks for real-time visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 7944–7953.
9. Zhu, Z.; Wang, Q.; Li, B.; Wu, W.; Yan, J.; Hu, W. Distractor-aware Siamese Networks for Visual Object Tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 101–117.
10. Zhang, Z.; Peng, H. Deeper and wider siamese networks for real-time visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4591–4600.
11. Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H. Fast online object tracking and segmentation: A unifying approach. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 1328–1338.
12. He, A.; Luo, C.; Tian, X.; Zeng, W. A twofold siamese network for real-time object tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 4834–4843.
13. 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, 18–22 June 2018; pp. 7132–7141.
14. Wang, Q.; Teng, Z.; Xing, J.; Gao, J.; Hu, W.; Maybank, S. Learning attentions: Residual attentional siamese network for high performance online visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4854–4863.
15. Yu, Y.; Xiong, Y.; Huang, W.; Scott, M.R. Deformable Siamese attention networks for visual object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 6728–6737.
16. Zhao, M.; Okada, K.; Inaba, M. TrTr: Visual tracking with transformer. *arXiv* **2021**, arXiv:2105.03817.
17. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 12 June 2017; pp. 5998–6008.
18. Chen, X.; Yan, B.; Zhu, J.; Wang, D.; Yang, X.; Lu, H. Transformer Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Virtual, 20–25 June 2021; pp. 8122–8131.
19. Ma, F.; Shou, M.Z.; Zhu, L.; Fan, H.; Xu, Y.; Yang, Y.; Yan, Z. Unified Transformer Tracker for Object Tracking. *arXiv* **2022**, arXiv:2203.15175.
20. Yan, B.; Peng, H.; Fu, J.; Wang, D.; Lu, H. Learning Spatio-Temporal Transformer for Visual Tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–17 October 2021; pp. 10428–10437.
21. Xu, Y.; Ban, Y.; Delorme, G.; Gan, C.; Rus, D.; Alameda-Pineda, X. Transcenter: Transformers with dense queries for multiple-object tracking. *arXiv* **2021**, arXiv:2103.15145.
22. Tang, C.; Qin, P.; Zhang, J. Robust template adjustment siamese network for Object Tracking. *Sensors* **2021**, *21*, 1466. [CrossRef] [PubMed]
23. Guo, Q.; Feng, W.; Zhou, C.; Huang, R.; Wan, L.; Wang, S. Learning dynamic siamese network for visual object tracking. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1763–1771.
24. Li, P.; Chen, B.; Ouyang, W.; Wang, D.; Yang, X.; Lu, H. Gradnet: Gradient-guided network for visual object tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 6162–6171.
25. Yang, T.; Chan, A.B. Learning dynamic memory networks for object tracking. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 20 March 2018; pp. 152–167.
26. Fan, H.; Lin, L.; Yang, F.; Chu, P.; Deng, G.; Yu, S.; Bai, H.; Xu, Y.; Liao, C.; Ling, H. LaSOT: A high-quality benchmark for large-scale single object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Bean, CA, USA, 16–20 June 2019; pp. 5374–5383.

27. 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.* **2021**, *43*, 1562–1577. [CrossRef] [PubMed]
28. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014; pp. 740–755.
29. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
30. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. *arXiv* **2017**, arXiv:1711.05101.
31. Wu, Y.; Lim, J.; Yang, M.-H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
32. Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; Torr, P.H. End-to-end representation learning for correlation filter based tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2805–2813.
33. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Learning spatially regularized correlation filters for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 4310–4318.
34. Bertinetto, L.; Valmadre, J.; Golodetz, S.; Miksik, O.; Torr, P.H. Staple: Complementary learners for real-time tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 1401–1409.
35. Danelljan, M.; Häger, G.; Khan, F.; Felsberg, M. Accurate scale estimation for robust visual tracking. In Proceedings of the British Machine Vision Conference (BMVC), Nottingham, UK, 1–5 September 2014; pp. 1–11.
36. Zhang, J.; Ma, S.; Sclaroff, S. MEEM: Robust tracking via multiple experts using entropy minimization. In Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland, 5–12 September 2014; pp. 188–203.
37. Danelljan, M.; Bhat, G.; Khan, F.S.; Felsberg, M. Atom: Accurate tracking by overlap maximization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4660–4669.
38. Song, Y.; Ma, C.; Wu, X.; Gong, L.; Bao, L.; Zuo, W.; Shen, C.; Lau, R.W.; Yang, M.H. Vital: Visual tracking via adversarial learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 8990–8999.
39. Yan, B.; Zhao, H.; Wang, D.; Lu, H.; Yang, X. 'Skimming-Perusal' Tracking: A framework for real-time and robust long-term tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 2385–2393.
40. Zhang, L.; Gonzalez-Garcia, A.; van de Weijer, J.; Danelljan, M.; Khan, F.S. Learning the model update for siamese trackers. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 4010–4019.
41. Danelljan, M.; Bhat, G.; Shahbaz Khan, F.; Felsberg, M. Eco: Efficient convolution operators for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6638–6646.
42. Nam, H.; Han, B. Learning multi-domain convolutional neural networks for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4293–4302.
43. Sauer, A.; Aljalbout, E.; Haddadin, S. Tracking Holistic Object Representations. *arXiv* **2019**, arXiv:1907.12920.
44. Wang, G.; Luo, C.; Xiong, Z.; Zeng, W. Spm-tracker: Series-parallel matching for real-time visual object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3643–3652.
45. Held, D.; Thrun, S.; Savarese, S. Learning to track at 100 fps with deep regression networks. In Proceedings of the European Conference Computer Vision (ECCV), Amsterdam, The Netherlands, 8–16 October 2016; pp. 749–765.