symmetry

MDPI

Article

Discriminative Siamese Tracker Based on Multi-Channel-Aware and Adaptive Hierarchical Deep Features

Huanlong Zhang, **Rui Duan, Anping Zheng *, Jie Zhang, Linwei Li and Fengxian Wang**

School of Electrical and Information Engineering, Zhengzhou University of Light Industry,
Zhengzhou 450002, China; hlzhang@zzuli.edu.cn (H.Z.); 331901060053@zzuli.edu.cn (R.D.);
2018007@zzuli.edu.cn (J.Z.); 2019028@zzuli.edu.cn (L.L.); 2019031@zzuli.edu.cn (F.W.)
* Correspondence: 331901050049@zzuli.edu.cn

Abstract: Most existing Siamese trackers mainly use a pre-trained convolutional neural network to extract target features. However, due to the weak discrimination of the target and background information of pre-trained depth features, the performance of the Siamese tracker can be significantly degraded when facing similar targets or changes in target appearance. This paper proposes a multi-channel-aware and adaptive hierarchical deep features module to enhance the discriminative ability of the tracker. Firstly, through the multi-channel-aware deep features module, the importance values of feature channels are obtained from both the target details and overall information, to identify more important feature channels. Secondly, by introducing the adaptive hierarchical deep features module, the importance of each feature layer can be determined according to the response value of each frame, so that the hierarchical features can be integrated to represent the target, which can better adapt to changes in the appearance of the target. Finally, the proposed two modules are integrated into the Siamese framework for target tracking. The Siamese network used in this paper is a two-input branch symmetric neural network with two input branches, and they share the same weights, which are widely used in the field of target tracking. Experiments on some Benchmarks show that the proposed Siamese tracker has several points of improvement compared to the baseline tracker.

Keywords: target features; siamese trackers; multi-channel aware; adaptive hierarchical features; visual tracking

check for updates

Citation: Zhang, H.; Duan, R.; Zheng, A.; Zhang, J.; Li, L.; Wang, F. Discriminative Siamese Tracker Based on Multi-Channel-Aware and Adaptive Hierarchical Deep Features. *Symmetry* **2021**, *13*, 2329. https://doi.org/10.3390/sym13122329

Academic Editor: José Carlos R. Alcantud

Received: 11 October 2021
Accepted: 17 November 2021
Published: 5 December 2021

1. Introduction

Object tracking is a basic research hotspot in the field of computer vision, and has many applications in daily life, such as autonomous driving [1], video surveillance [2], and human–computer interaction [3]. Usually, the information of the tracked object is given in the first frame, and the new position of the target in the subsequent frames is predicted by the designed tracker. Since only the first frame of the target information is given, prior knowledge is seriously insufficient. Therefore, when facing some complex scenes, such as background clutter, lighting changes, fast motion, and partial occlusion, the tracking effect will sharply drop. A number of models were proposed to extract target features in target tracking, such as manual-features [4], correlation-filters [5–7], regressors [8,9], and classifiers [10–12]. While most Siamese-based trackers use pre-trained deep models to extract features for the tracking task, they pay less attention to how learning more discriminative deep features.

Recent work in this area includes the design of loss functions to select appropriate feature channels [13], using memory networks to preserve the latest appearance models [14], attention mechanisms to enhance feature representation [15], and multi-layer feature fusion to represent targets [16]. For example, MemDTC [14], an algorithm that uses Memory networks to memorize the appearance of targets, achieves a good performance; however, due to the presence of memory banks, it occupies a large amount of device memory during tracking. This uses up the limited computational resources and leads to a decrease in tracking speed, which then does not meet the requirements of real-time performance. TADT [13] designed regression loss and sequencing loss, used the first-frame target information to train the network, and achieved a feature dimension reduction by back-propagating the gradient. Unlike them, this paper starts from the structural aspect of depth features, considers the contribution of different feature layers and different channels of the same feature layer in target modeling, and designs two modules to learn a depth feature with a stronger discriminative power to better represent the target and improve the performance of the Siamese tracker.

Most deep-learning-based trackers [17–20] take the target as a positive sample, and some randomly selected areas from around the target as negative samples, during training, and then use CNN networks to extract features from these samples to train and learn a classifier. Although some existing Siamese-based trackers achieved an excellent tracking performance, we note that pre-trained deep features are more effective for target recognition and do not perform well enough for target-tracking tasks. The use of pre-trained depth features in target tracking may pose two rather obvious problems. First, CNN is generally taken as an online classifier in the target recognition task, and only its last layer feature is used to represent the target, which is effective in the target recognition task. This is because the last layer of CNN contains the highest level of semantic information of the object and contributes the most to object recognition, which satisfies the requirements of the target recognition task. However, in the tracking task, where there is no need to classify the tracked targets but only to precisely localize them, the last layer of features is not sensitive to the intra-class distinction and position changes of the targets. Using only the last layer to represent the targets is not the best choice. Second, the object being tracked is arbitrary, and if the pre-trained deep features do not contain the class of the object being tracked, i.e., the model does not contain the feature information of the object being tracked; therefore, there will be lower efficiency when distinguishing the target from the background.

This paper proposes a novel scheme to learn target deep features via the multi-channel aware and the adaptive hierarchical deep features module to guide the generation of the most significant features of the target. This work is based on the following methods. With the use of two branches to learn the overall information and saliency information of the target respectively, the weight vector generated by the two branches can determine the importance of a channel for representing target objects. Instead of introducing new spatial dimensions, The proposed method use feature recalibration to add the obtained importance value to channels that are useful for target modeling. In addition, the semantic features at higher levels are robust to changes in the appearance of the target, while the detailed features at lower levels are more effective for localizing the target. The proposed method identifies the importance of each feature layer according to the ratio of the maximum response value, and the primary and secondary peak values, to integrate features to represent the target. At the same time, features from hierarchical layers of CNNs are used to represent targets, rather than only the last layer, and the fusion weights of hierarchical layers are adaptively updated in real time. Figure 1 shows the tracking results for our tracker compared with other similar trackers. As can be seen from the three video sequences, our tracker has a better performance in the face of complex scenes, such as deformation and background clutter.

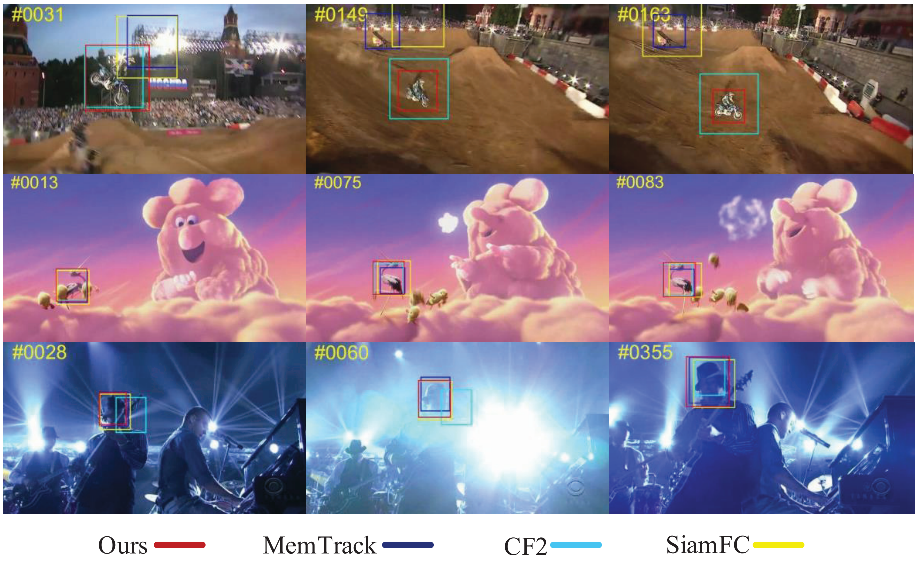


Figure 1. Comparison of our tracker with other trackers, including Siamese-based trackers SiamFC [21] and CF2 [22], attention-based tracker MemTrack [23] for MotorRolling (**top**), Bird2 (**middle**), and Shaking (**bottom**).

The Siamese network is a symmetric network. It was originally applied to the field of template matching. It contains two input branches that share the same network structure and internal parameters. SiamFC was introduced to the field of target tracking and achieved good results. We integrated the two methods with a Siamese network for an object-tracking task, and evaluated the proposed tracker on several benchmarks, including OTB-50 [24], OTB-100 [25], UAV123 [26], Temple Color-128 [27], and VOT2016 [28]. Extensive experiments have shown that the proposed tracker is more effective in terms of success rate and precision rate compared to trackers based on pre-trained deep features. The main contributions of the proposed method can be summarized as:

1. The proposed method designed a multi-channel-aware, deep-feature module to establish the interdependence between feature channels, which include two branches to learn the overall information and saliency information of the target, and adopted feature re-calibration to enhance the channel weights that play a positive role in target representation.

2. To effectively fuse the features of different layers, the proposed method uses adaptive hierarchical deep features to guide the generation of the most significant features of the target, which can obtain the contribution of different feature layers, then fuses the two feature layers according to their contribution, and this contribution value is adaptively updated.

3. We integrate the two methods with a Siamese network for object tracking and evaluate the proposed method on some benchmarks. The experimental results have shown that the proposed tracker is more effective than some other trackers.

2. Related Work

Visual object tracking has been developed for decades, and many tracking methods have been proposed. This section provides short outlines for some representative trackers related to our work, such as trackers using deep features, trackers based on the Siamese network, and trackers based on the deep feature and attention mechanism.

2.1. *Deep Features Based Tracker*

Thanks to the powerful appearance modeling abilities of deep features, the performance of the tracker can be significantly improved; therefore, the traditional manual features are gradually replaced. The DCF-based trackers also use deep features to improve performance, such as DeepSRDCF [29], C-COT [30], ECO [31]. To take advantage of deep features, CF2 [22] and FCNT [32] use shallow and deep features to fuse the representation targets for efficiency.

Although these trackers have outstanding feature representation power, there is a significant problem, as only limited training samples and the ground-truth visual appearance of the target in the first frame are available. In addition, we found that the previously mentioned tracker only utilizes the last layer of the CNN features; unlike their approach, our tracker uses multiple convolutional layers to model the target and the weights between multiple convolutional layers are adaptively updated.

2.2. Siamese Network Based Tracker

The Siamese network-based tracker [21,33–35] views tracking as a matching problem and learns a similarity metric network. The input to the Siamese tracker consists of two parts: the initial frame template of the tracked object and the search region of the current frame. They both use the same full convolutional network to extract target features, and finally use cross-correlation operations for template-matching to generate a response map. The position of the maximum value in the response map is the corresponding position of the target in the search area. SiamFC [21] is a tracking method based on an offline end-to-end training of the Siamese network. This aims to learn a similarity function for target matching. Since SiamFC mainly focuses on appearance features and ignores the high-level semantic information of the target, SA-Siam [35] improves this. It uses a dual Siamese network tracking scheme, in which one Siamese branch is responsible for apparent feature matching, and the other branch is responsible for semantic information matching, which combines the apparent features and semantic information to make the performance of the tracker more stable. Differing from the detection network used in some methods, GOTURN [33] uses a regression method based on the Siamese network to learn the relationship between the appearance and movement of the target. After entering the search area of the template, the Siamese network extract target feature, and then the regression network can compare the two image returns to the position of the target. SiamMCF [36] and DSiam [37] solve the similarity problem through multi-layer interconnection.

Although these Siamese networks have been pre-trained on some large datasets, they are more suitable for classification tasks and do not take full advantage of the semantic and object information associated with a particular target object. Therefore, there are certain problems in the modeling of target feature expression.

2.3. Deep Feature and Attention Based Tracker

Attention mechanism has been widely used in the field of computer vision, such as object detection [38], person search [39] and image segmentation [40]. Introducing the attention mechanism into target tracking can help the tracker pay more attention to the information of the target itself and reduce the influence of the unimportant parts when positioning. This strategy is applicable in most scenarios. With the development of attention mechanism in the field of tracking, some related trackers have been proposed. To acquire spatial and semantic features of thermal infrared targets, HSSNet [41] design a Siamese CNN with multiple hierarchical features. MLSSNet [42] proposes a multi-level similarity network to learn the global semantic features and local structural features of objects. RASNet [43] integrates three attention modules-channel attention, general attention and residual attention into one layer of the Siamese network, which alleviates the overfitting problem in deep network training and improves its discriminative ability and adaptability. MemTrack [23] and MemDTC [14] introduce attention mechanisms for spatial location and use a long short term memory-based (LSTM) controller to manage the read and write operations of feature maps in memory. IMG-Siam [44] introduces channel attention to better learn the matching models.

This paper proposes a multi-channel-aware deep-feature method, which includes two branch attention mechanisms. This multi-channel aware deep feature method works on two feature layers, and finally obtains a fusion of multi-layer and multi-channel attention features.

3. Proposed Method

By carefully designing feature extraction strategies, the matching accuracy can be improved. However, the tracking target is arbitrary, and it is impractical to design features that are suitable for any target. To deal with these problems, this paper proposes a novel scheme to learn target deep features via the multi-channel-aware and adaptive hierarchical deep features module to guide the generation of the most significant target features. The proposed method uses the features extracted by existing methods to improve the performance of the Siamese-based tracker.

In this section, we will introduce the details of the proposed tracking framework. As shown in Figure 2, the proposed tracking framework consists of a Siamese network for feature extraction and a feature-learning mechanism to enhance the target feature representation. Specifically, the feature-learning mechanism consists of two modules: one is responsible for learning the multi-channel-aware deep features and the other is responsible for fusing adaptive hierarchical deep features. The learning multi channel-aware deep-feature module has two branches with the same structure, which are responsible for the recalibration of the corresponding characteristic channels. The adaptive hierarchical deep-feature fusion module determines the weight of feature layers by combining the peak side lobe ratio (PSLR) with the peak point constraint. The whole system is trained end-to-end by inputting the image block containing the target into the framework. Our tracker is based on the SiamFC [21] framework. We will describe the proposed method in detail.

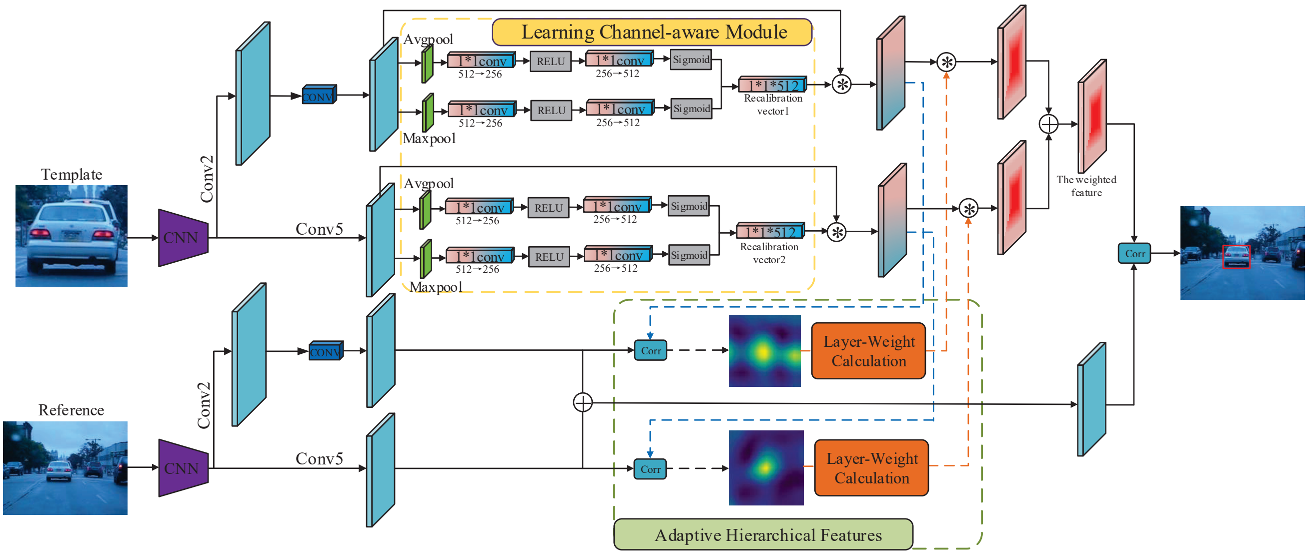


Figure 2. Overview of our network architecture for visual tracking.

3.1. Basic Siamese Network for Visual Tracking

As mentioned above, Siamese network are widely used to solve the problem of template matching, and their input contains two parts: the template image z and the search image x of the current frame. The template image for visual tracking is usually given in the first frame, while the search image of the current frame is cropped in a certain region around the target location estimated in the previous frame. Both parts of the input use the same CNNϕ_θ network to extract the depth features and then obtain the target response using cross-correlation. This can be represented as:

$$f_\theta(z,x) = \phi_\theta(z) \star \phi_\theta(x) + d \tag{1}$$

where $\star$ in the formula represents the cross-correlation operation of two features. The parameter d represents the bias; $\phi_\theta()$ denotes the method of extracting deep features. The position of the maximum response value of $f_\theta(z, x)$ represents the target position.

3.2. Multi-Channel Aware Deep Features

Each channel in the feature layer contains different target information. In template-matching process, the contribution from each channel is different. The proposed method designs two branches to focus on multi-channel-aware deep features. At first, the two branches are global average pooling and global maximum pooling. This is because global average pooling preserves the overall information about the target, and global maximum pooling preserves the salience information about the target, so the proposed method uses two pooling operations to obtain a cropped feature map that preserves the overall and detailed knowledge of the target. After the pooling operation, two feature vectors of $1 \times 1 \times 512$ are obtained, and 1×1 convolution operation is used to reduce them to $1 \times 1 \times 256$, then restore them to $1 \times 1 \times 512$. The non-linear expression ability of feature vectors can be increased by adding the ReLU function between the two lifting dimension operations. The traditional attention network uses the full-connection layer operation, but the full-connection layer is mainly put forward for the classification task. Before tracking, the target information is known from the first frame, so there is no need to carry out object classification. Moreover, the full-connection layer will destroy the spatial structure of the image. As the convolution operation will not destroy the spatial structure of the image, it helps to retain the local features of the image, which is more conducive to target positioning. The Sigmoid function is used to normalize the previously obtained 512 dimension feature vectors to between 0 and 1.

Hence, the proposed method obtains two pooling feature vectors f_{max}^{1*1*c} and f_{avg}^{1*1*c} for max and average branch, respectively, from CONV2 and CONV5. Finally, the two pooling feature vectors from two feature layers are respectively fused together to obtain vector $\phi_{\theta 2}(\cdot)^{1*1*c}$ and $\phi_{\theta 5}(\cdot)^{1*1*c}$ that can represent the weight of the channel. This two-weight vector is multiplied by the original feature to obtain a feature map weighted to the channel C_{M2}^{H*W*C} and C_{M5}^{H*W*C}. This process is called feature recalibration. The calculation process can be expressed as:

$$f_{max2}^{1*1*c} = CONV_2\Big(\mathrm{Re}LU\Big(CONV_1\Big(Pool_{max}\Big(F_{M2}^{H*W*C}\Big)\Big)\Big)\Big) \tag{2}$$

$$f_{avg2}^{1*1*c} = CONV_2\Big(\mathrm{Re}LU\Big(CONV_1\Big(Pool_{avg}\Big(F_{M2}^{H*W*C}\Big)\Big)\Big)\Big) \tag{3}$$

$$f_{max5}^{1*1*c} = CONV_2\Big(\mathrm{Re}LU\Big(CONV_1\Big(Pool_{max}\Big(F_{M5}^{H*W*C}\Big)\Big)\Big)\Big) \tag{4}$$

$$f_{avg5}^{1*1*c} = CONV_2\Big(\mathrm{Re}LU\Big(CONV_1\Big(Pool_{avg}\Big(F_{M5}^{H*W*C}\Big)\Big)\Big)\Big) \tag{5}$$

$$\phi_{\theta 2}(\cdot)^{1*1*c} = \varepsilon(f_{max2}^{1*1*c} \oplus f_{avg2}^{1*1*c}) \tag{6}$$

$$\phi_{\theta 5}(\cdot)^{1*1*c} = \varepsilon(f_{max5}^{1*1*c} \oplus f_{avg5}^{1*1*c}) \tag{7}$$

Finally, the multi-channel-aware deep features that are obtained can be expressed as:

$$C_{M2}^{H*W*C} = \phi_{\theta 2}(\cdot)^{1*1*c} \otimes F_{M2}^{H*W*C} \tag{8}$$

$$C_{M5}^{H*W*C} = \phi_{\theta 5}(\cdot)^{1*1*c} \otimes F_{M5}^{H*W*C} \tag{9}$$

where ε represents the sigmoid function $f(x) = \frac{1}{1-e^{-x}}$, $pool_{max}$ and $pool_{avg}$ are global max pooling and global average pooling, respectively; F_{M2}^{H*W*C} and F_{M5}^{H*W*C} are features from CONV2 and CONV5, C_{M2}^{H*W*C}; C_{M5}^{H*W*C} are the weighted features.

Figure 3 clearly show that, after the use of multi-channel aware methods, the obtained target features are more concentrated and the target can be more effectively distinguished from the background.

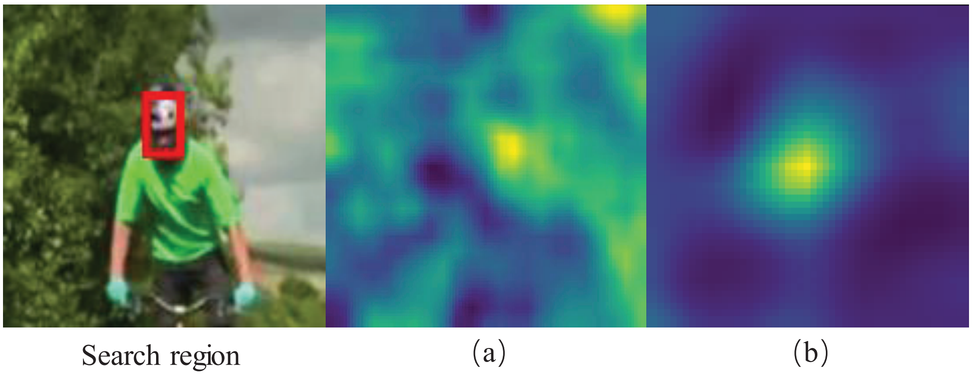


Figure 3. Comparison diagram of feature extraction before and after adding multi-channel aware features. (**a**) is before the addition and (**b**) is after the addition.

3.3. *Adaptive Hierarchical Deep Features*

Due to the characteristics of depth features, low-level features contain more target details due to their higher resolution, while high-level features encode more high-level semantic information despite their lower resolution. In the tracking stage, the fusion of high- and low-level features becomes an effective method to solve the problem of positioning accuracy. Therefore, the fusion of high and low-level features has become a research problem. Figure 4 contains two video sequences, showing the response values of different feature layers on the same video frame. Obviously, different feature layers contribute differently to the target response.

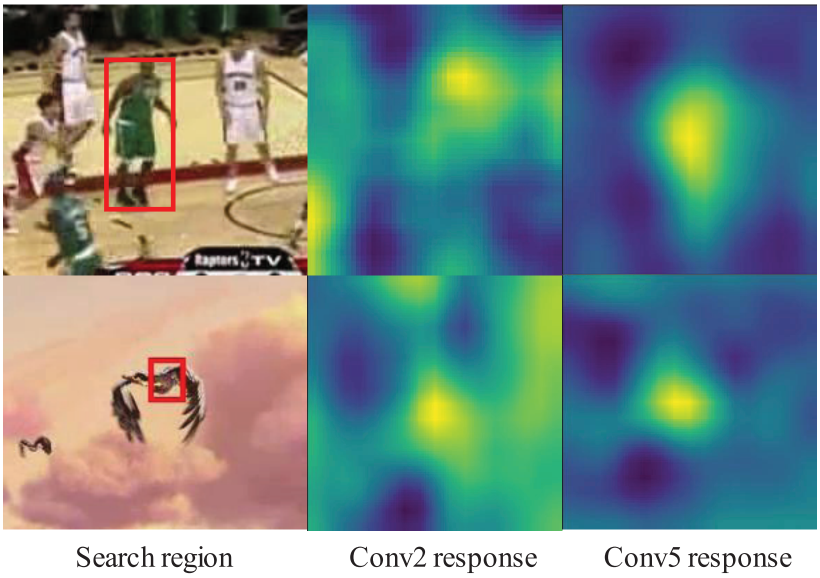


Figure 4. Response map at different feature layers.

The CNNs used in this paper total have five feature layers; after each convolution, the resolution will be lower. To ensure that the template features have a wealth of detailed information and high-level semantic information, the proposed method adopts the adaptive weighted fusion method to enhance the performance. The proposed method gives the CONV2 and CONV5 layer different reliability weights and the response of the reliability weight from the features of the layer itself, and the reliability weights are updated in real-time. In our method, the reliability weight estimation of the feature layer consists of two parts: (1) The layer max response learning reliability weight w_d^{max}, namely, the response peak in the feature layer and the template area. The larger the response value, the higher the reliability. (2) The layer interference detection reliability weight w_d^{ratio}, that is, the ratio of main lobe peak intensity to the peak intensity of the strongest side lobe. The lower the ratio, the higher the reliability. In the tracking stage, the two parts work together to determine the reliability of the feature layer, which can be expressed as:

$$w_d = w_d^{max} * w_d^{ratio} \tag{10}$$

and normalized s.t.$\sum_d W_d = 1$. The reliability measures are described in the following paragraphs.

3.3.1. Layer Response Learning Reliability

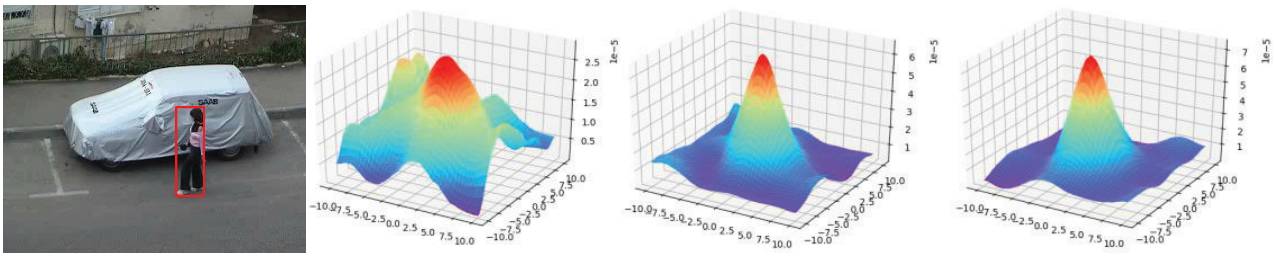
The ideal response peak should be the unique peak obtained by cross-correlation between the template and the search area, and its size should be close to 1. However, in the actual tracking process, due to the existence of a high level of background interference, the response map is high-noise in some frames with a low discrimination ability. Therefore, the feature layer's response weight can be obtained as follows:

$$w_{d2}^{max} = max\left(C_{M2}^{H*W*C} \star F2\right) \tag{11}$$

$$w_{d5}^{max} = max\left(C_{M5}^{H*W*C} \star F5\right) \tag{12}$$

where $\star$ is the cross-correlation, w_{d2}^{max} is the max response of CONV2 and w_{d5}^{max} is the max response of CONV5, $F2$ and $F5$ are the features of CONV2 and CONV5 in the search area.

Figure 5 shows the influence of the hierarchical deep features on the response map. It can be seen that the response map with hierarchical deep features has a higher peak value and a more concentrated response point.



sequence (a) (b) (c)

Figure 5. The response map of different feature layers in the same frame. (**a**) is the response map of Conv2, (**b**) is the response map of Conv5, and (**c**) is the response map after the weighted fusion of two feature layers using the layer reliability module.

3.3.2. Layer Interference Detection Reliability

The second part of the feature layer reliability reflects the ratio of the contribution of different feature layers to target localization. Unlike the similar method proposed by Bolme et al. [45] to detect target loss, our method detects the primary and secondary peaks in the response map and determines the interference strength of different feature layers by the ratio of these two peak points, $1 - \rho_d^{max2} / \rho_d^{max1}$. The smaller the ratio, the lower the interference. In this way, the influence of nearby, strong interfering objects on the target modeling can be reduced, and the final ratio can be lower than 0.5. PSLR weight can be expressed as:

$$w_{d2}^{ratio} = max\left(1 - \frac{\rho_{d2}^{max2}}{\rho_{d2}^{max1}}, 0.5\right) \tag{13}$$

$$w_{d5}^{ratio} = max\left(1 - \frac{\rho_{d5}^{max2}}{\rho_{d5}^{max1}}, 0.5\right) \tag{14}$$

Therefore, the adaptive hierarchical feature weight can be expressed as:

$$C_{M2}^{H*W*C} = \frac{w_{d2}^{max} \otimes w_{d2}^{ratio}}{w_{d2}^{max} \otimes w_{d2}^{ratio} + w_{d5}^{max} \otimes w_{d5}^{ratio}} \tag{15}$$

$$C_{M5}^{H*W*C} = \frac{w_{d5}^{max} \otimes w_{d5}^{ratio}}{w_{d2}^{max} \otimes w_{d2}^{ratio} + w_{d5}^{max} \otimes w_{d5}^{ratio}} \tag{16}$$

Figure 6 compares the responses of the same video sequence with different frames before and after using adaptive hierarchical deep features. We can clearly see that, in the video, when the appearance and position of the biker significantly changed, the tracking failed without the use of adaptive hierarchical deep features; at this time, the tracking box appeared in the same position and the response value was approximately the same. When using adaptive hierarchical deep features, although the response value drops sharply as the biker's appearance and position changes, the tracking can still be completed, and it gradually returns to normal and remains in the subsequent frames.

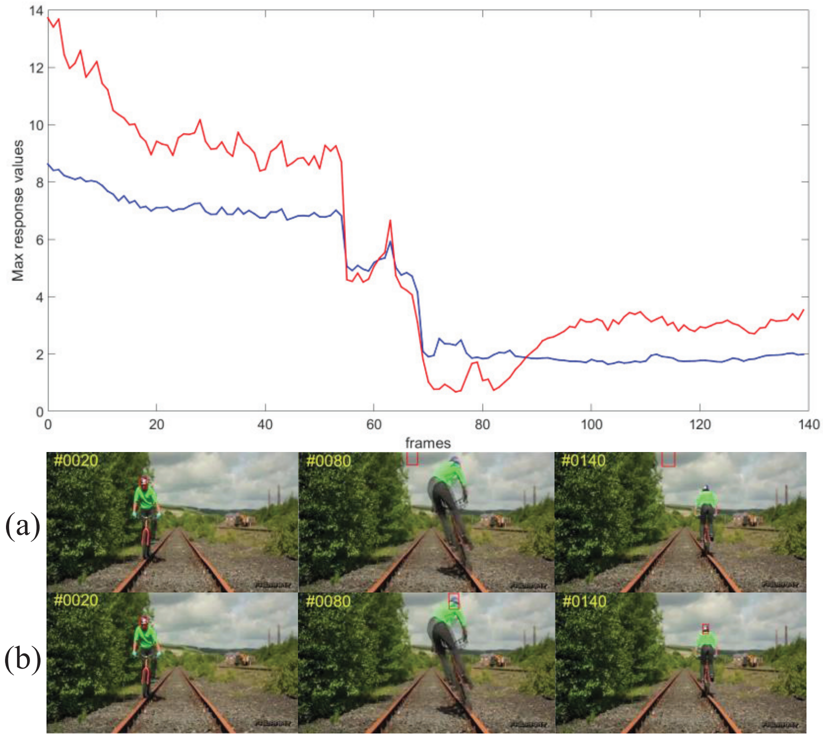


Figure 6. The line chart of different response values for each frame. Blue line is the response value of each frame before adding the layer reliability, and red line is the response value of each frame after adding. (**a**) it reflects the tracking result of the response value of the blue line, and (**b**) it reflects the tracking result of the response value of the red line.

4. Experimental Details

4.1. Training Detail

The proposed method used a GOT10K dataset [46] and ImageNet Large Scale Visual Recognition Challenge 2015 VID dataset [47] to train the model. In the training process, the SiamFC cropping strategy was used to crop the template image z and the search image x, respectively, and the target position was taken as the center position. Image pairs (z, x) were randomly selected from the training set and a logistic loss function was used in the following form:

$$L(f(z,x),g) = \frac{1}{|N|} \sum_{n \in N} log(1 + exp(-f(z,x)[n] \cdot g[n])) \tag{17}$$

where N is the possible locations of the target on the response map, $f(z,x)[n]$ is the response map score, and $g[n] \in \{1, -1\}$ is the ground truth coordinate. To ensure that more training samples were obtained, we randomly selected 10 image pairs from each video sequence and set the maximum interval between the template and the search images to 100 images; the batch size was set to 32. The Stochastic Gradient Descent(SGD) method was used to optimize the objective function. In the test stage, the same strategy as SiamFC was used for target positioning.

$$argmin_{\theta} \frac{1}{M} \sum_{i=1}^{M} L(f(z_i,x_i), g_i) \tag{18}$$

Based on experience, the momentum was set to 0.9, learning rate decay from 1×10^{-2} to 1×10^{-5}, weight decay rate 5×10^{-4}, and a total of 35 generations were used for training.

We implemented the proposed tracker with Python and PyTorch framework, on a PC with 16G memory, an Intel(R) Core i7-9700 CPU @3.0 GHz, and a NVIDIA GeForce RTX 2060 GPU.

4.2. Evaluation on OTB Benchmark

The OTB dataset is a public dataset to test the effectiveness of target-tracking algorithms, which is divided into OTB50 [24] and OTB100 [25], containing 50 and 100 video sequences, respectively.

On the OTB100, we compared several different categories of algorithms, including Siamese trackers SiamFC [21], attentional Siamese trackers MemTrack [23] and MemDTC [14], correlation-filter based trackers including KCF [6], Staple [48], DSST [49] and SRDCF [50], CNN and correlation-filter based trackers including CF2 [22], CREST [51], CSR-DCF [52].

As shown in Figure 7, on the OTB100 dataset, our tracker achieved excellent results in terms of both success rate and precision rate, with a success rate of 63.1% and an precision rate of 84.2%, which are 4.8% and 7.0% better than the baseline algorithm SiamFC, respectively. Compared with the attention Siamese tracker MemTrack, our tracker was 0.4% and 3.1% ahead in success rate and precision rate, respectively. However, compared to the attention memory tracker MemDTC, our tracker lagged behind in success rate and precision rate by 0.7% and 0.5%, which we speculate is due to the dynamic memory network introduced by MenDTC, which enables the target template to adapt to changes in target appearance during tracking. We also compared some CNN- and correlation-filter-based trackers such as SRDCF, CREST, CSR-DCF. The proposed tracker achieved a 3.1%, 1.1%, and 5.2% improvement in success rate and 5.0%, 0.8%, and 4.3% improvement in precision rate compared to these methods.

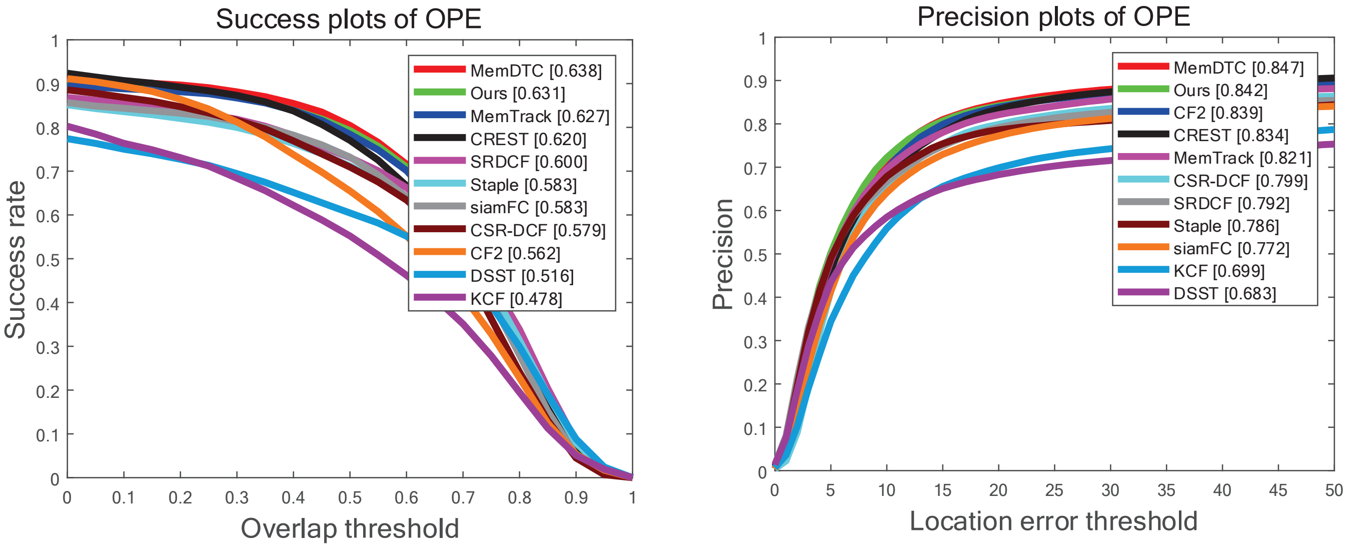


Figure 7. Success and precision rates on the OTB100 dataset.

Figure 8 shows the overall performance of the proposed tracker on OTB50. It can be seen that the proposed tracker had the best OTB50. Compared with the baseline algorithm SiamFC, the proposed tracker leads in two metrics, success rate and precision rate, by 8.7% and 13.8%, respectively. The proposed tracker also achieved a 4%, 3.2%, 6.4% and 8.4% improvement n success rate and an 8%, 3.6%, 9.7% and 10.8% improvement in precision compared tot the MemTrack, CREST, SRDCF, and CSR-DCF trackers, respectively. Unlike the OTB100 performance, our tracker has a better performance in terms of success rate and accuracy compared to MemDTC. Experiments on both datasets show that our tracker has excellent performance, proving the effectiveness of the proposed approach.

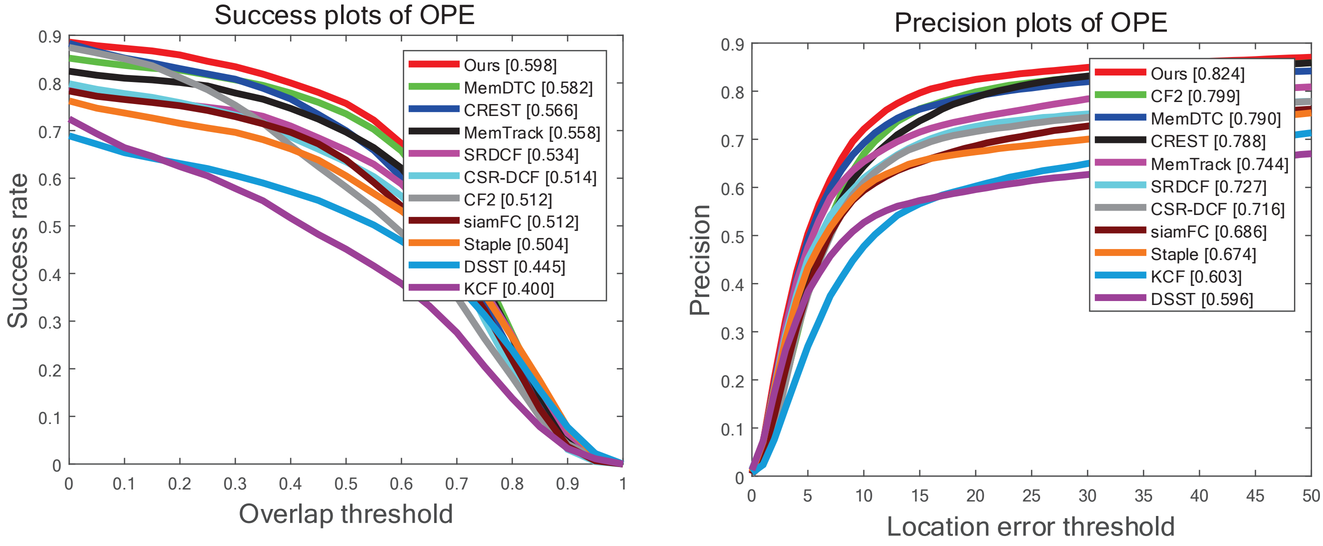


Figure 8. Success and precision rates on the OTB50 dataset.

Qualitative Analysis on OTB Benchmark

In order to analyze the proposed tracker in more depth, we performed another qualitative analysis. Figure 9 shows the effect comparison of different trackers on six typical video sequences. These trackers include a CF-based tracker DSST, attention-based tracker MemTrack, Siamese-based tracker SiamRPN and SiamFC, CNN's and CF-based tracker CF2.

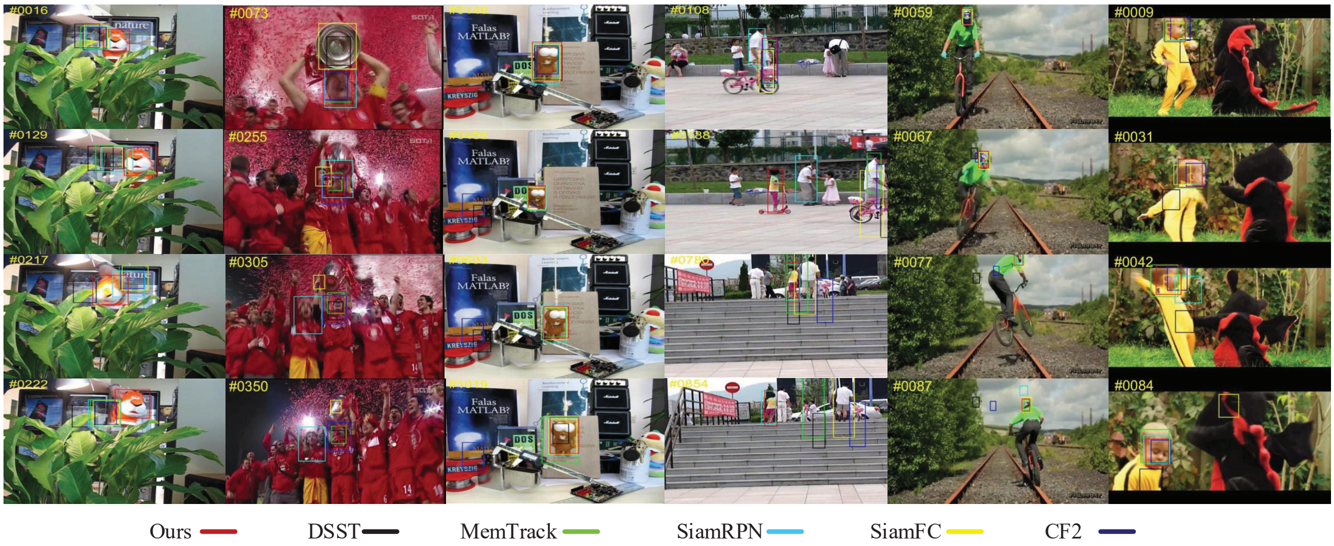


Figure 9. The qualitative results for six challenging sequences from the OTB100 benchmark, including tiger1, soccer, lemming, girl2, biker, and dragonbaby.

The following types of video sequence contain several common challenges that would be faced in visual target tracking, such as: scale variation (in Biker, Girl2), being obscured (in Biker, DragonBaby, Girl2, Lemming), being out of view (in Soccer), and background clutter (in DragonBaby). Figure 9 shows the tracking effectiveness of our tracker when facing these challenges. Due to the introduction of the multi-channel-aware module and the adaptive hierarchical depth feature module, our proposed tracker can adapt well to these challenges compared to other algorithms.

In addition, to validate the performance of our proposed tracker in more depth, we conducted experiments in the 11 challenges of the OTB100 dataset. Tables 1 and 2 present the results of the proposed tracker compared with other trackers in the 11 challenges. It can be seen that the proposed tracker is able to consistently maintain an excellent performance in challenging situations due to the introduction of learning multi-channel-aware and adaptive hierarchical depth feature modules. In Tables 1 and 2, SV represents scale variation, LR represents low resolution, OC represents occlusion, DF represents deformation, MB represents motion blur, FM represents fast motion, IR represents in-plane rotation, OR represents out-of-plane rotation, OV represents out-of-view, BC represents background clutter, and IV represents illumination variation.

As shown in Tables 1 and 2, some more details about the proposed algorithm can be seen in this paper. In general, the proposed algorithm performs well on all 11 challenges. In all 11 challenges, the algorithm in this paper performs better than the baseline algorithm SiamFC, which directly uses pre-trained deep features to model the target, while we learn multi-channel-aware deep feature and adaptive hierarchical deep features to obtain a more discriminative feature. CF2 also uses hierarchical deep features to model the target; however, the weight of each layer expressed on the target is directly given. In contrast, the hierarchical deep features of the proposed algorithm are derived from the performance of each frame, and this weight is adaptively updated. MemTrack and MemDTC preserve the most recent appearance information of the target by introducing a memory network, and these are similar to the proposed algorithms in LR, OC, and OV scenarios; however, there are still some gaps. It can be seen that the proposed algorithm performs slightly worse in both IR and IV scenes, which indicates that the algorithm has room for improvement in planar rotation and strong illumination change scenes.

Table 1. Precision score comparison of 11 challenges in the OPE experiment on the OTB100 dataset. The top three trackers are marked with red, green and blue, respectively.

Tracker	SV	LR	OC	DF	MB	FM	IR	OR	OV	BC	IV
Ours	0.857	0.872	0.783	0.732	0.872	0.840	0.794	0.803	0.829	0.837	0.823
CF2	0.790	0.831	0.749	0.721	0.801	0.798	0.813	0.741	0.671	0.766	0.794
MemDTC	0.772	0.866	0.754	0.692	0.749	0.765	0.756	0.765	0.808	0.710	0.759
MemTrack	0.768	0.807	0.705	0.588	0.748	0.751	0.726	0.723	0.744	0.717	0.762
CREST	0.749	0.819	0.715	0.720	0.777	0.749	0.807	0.763	0.681	0.795	0.867
SRDCF	0.688	0.655	0.680	0.640	0.722	0.745	0.651	0.655	0.573	0.723	0.718
CSR-DCF	0.660	0.682	0.643	0.710	0.722	0.729	0.675	0.647	0.686	0.661	0.669
SiamFC	0.682	0.847	0.655	0.571	0.662	0.692	0.614	0.646	0.672	0.635	0.652
Staple	0.611	0.631	0.654	0.653	0.638	0.613	0.622	0.614	0.658	0.648	0.681
KCF	0.553	0.560	0.591	0.565	0.540	0.540	0.572	0.585	0.441	0.623	0.657
DSST	0.544	0.567	0.569	0.502	0.480	0.448	0.579	0.538	0.411	0.659	0.656

Table 2. Success score comparison of 11 challenges in the OPE experiment on the OTB100 dataset. The top three trackers are marked with red, green and blue, respectively.

Tracker	SV	LR	OC	DF	MB	FM	IR	OR	OV	BC	IV
Ours	0.624	0.637	0.563	0.509	0.650	0.630	0.576	0.571	0.603	0.602	0.590
CF2	0.478	0.439	0.484	0.465	0.561	0.542	0.529	0.485	0.443	0.512	0.512
MemDTC	0.570	0.605	0.550	0.493	0.570	0.573	0.557	0.552	0.572	0.544	0.564
MemTrack	0.573	0.574	0.518	0.452	0.561	0.575	0.537	0.529	0.534	0.533	0.556
CREST	0.534	0.527	0.518	0.509	0.598	0.576	0.589	0.555	0.504	0.579	0.614
SRDCF	0.510	0.494	0.487	0.451	0.525	0.562	0.475	0.475	0.430	0.530	0.521
CSR-DCF	0.479	0.439	0.462	0.500	0.546	0.556	0.483	0.459	0.497	0.472	0.476
SiamFC	0.515	0.592	0.483	0.425	0.504	0.531	0.473	0.475	0.495	0.476	0.484
Staple	0.453	0.418	0.481	0.497	0.472	0.479	0.455	0.455	0.463	0.495	0.511
KCF	0.348	0.307	0.392	0.395	0.401	0.389	0.384	0.391	0.327	0.417	0.431
DSST	0.400	0.383	0.411	0.380	0.384	0.366	0.427	0.390	0.323	0.491	0.497

4.3. Evaluation on TC-128 Benchmark

The TC-128 [27] is a dataset for color information, which contains 128 video sequences to test the performance of the tracker. We compared this dataset with some other excellent trackers, including: ECO [31], CREST [51], HCFTstar [53], CF2 [22], CACF [54], KCF [6], DSST [49], LOT [55], CSK [56]. The results show that our tracker is in second place in both precision rate and success rate metrics. Figure 10 shows the performance of all algorithms.

As shown in Figure 10, the success rate and precision rate of the proposed tracker reach 54.5% and 73.8%, respectively, which are inferior to the 55.2% and 74% reached by ECO. The reason for this may be that ECO uses a combination of depth features and color features, while the TC-128 dataset is designed to obtain the color information of objects, so the extraction of color features is beneficial for target modeling. However, the complex feature extraction method of ECO leads to its tracking speed of only 8 FPS, which cannot meet the requirements of real-time tracking, while our tracker can reach a speed of 29FPS. Moreover, our tracker has a 5% and 4.6% higher success rate and precision rate than

CF2, which also uses multi-layer depth features. Meanwhile, trackers based on manual features, such as KCF,CSK and DSST, are much less effective than other trackers that use deep features.

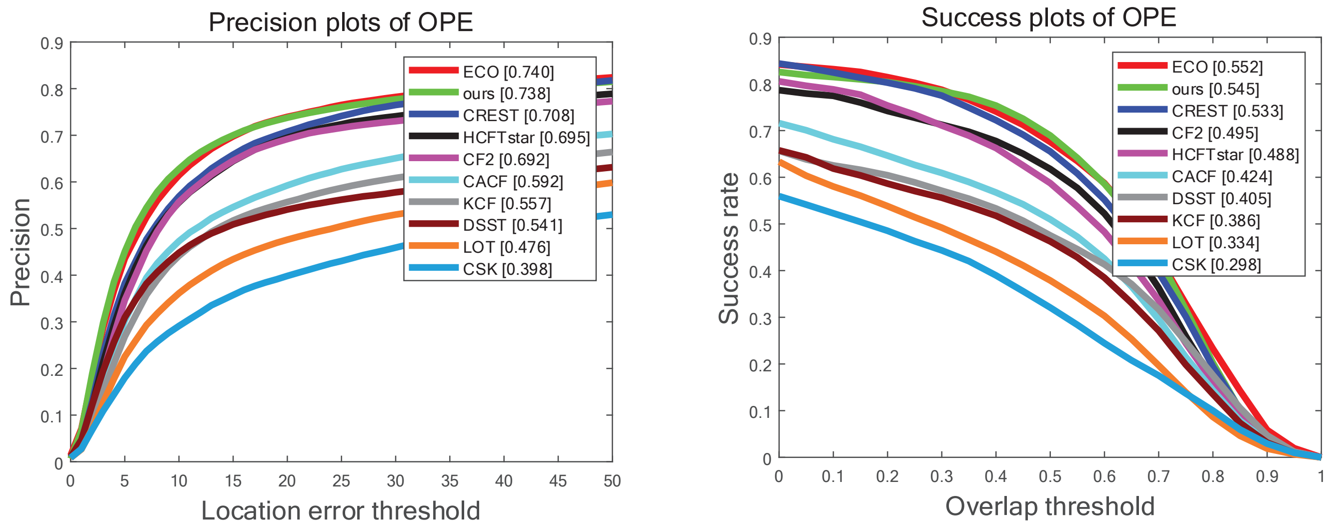


Figure 10. Success and precision rates on the TC-128 dataset.

Qualitative Analysis on TC-128 Benchmark

Similar to the OTB dataset, we also compared the results of 11 challenges on the TC-128 dataset, including Scale Variation (SV), Low Resolution (LR), Occlusion (OC), Deformation (DF), Motion Blur (MB), Fast Motion (FM), In-plane Rotation (IR), Out-of-plane Rotation (OR), Out-of-View (OV), Background Clutter (BC), and Illumination Variation (IV). The results are shown in Tables 3 and 4.

Table 3. Precision score comparison of 11 challenges in the OPE experiment on the TC-128 dataset. The top three trackers are marked with red, green and blue, respectively.

Tracker	**SV**	**LR**	**OC**	**DF**	**MB**	**FM**	**IR**	**OR**	**OV**	**BC**	**IV**
Eco	0.712	0.752	0.706	0.779	0.612	0.625	0.670	0.680	0.618	0.795	0.675
Ours	0.782	0.686	0.684	0.745	0.603	0.647	0.712	0.713	0.568	0.791	0.738
CREST	0.660	0.678	0.662	0.781	0.638	0.630	0.663	0.680	0.571	0.763	0.733
HCFTstar	0.681	0.577	0.608	0.773	0.618	0.627	0.623	0.681	0.511	0.756	0.733
CF2	0.688	0.583	0.622	0.802	0.635	0.634	0.635	0.673	0.492	0.744	0.721
CACF	0.567	0.499	0.524	0.664	0.530	0.506	0.552	0.549	0.388	0.677	0.632
KCF	0.529	0.449	0.478	0.652	0.486	0.490	0.510	0.524	0.374	0.625	0.581
DSST	0.538	0.405	0.488	0.502	0.449	0.431	0.501	0.512	0.384	0.552	0.583
LOT	0.451	0.448	0.443	0.542	0.381	0.426	0.431	0.458	0.361	0.514	0.400
CSK	0.380	0.348	0.343	0.351	0.299	0.282	0.358	0.366	0.217	0.427	0.370

Table 4. Success score comparison of 11 challenges in the OPE experiment on the TC-128 dataset. The top three trackers are marked with red, green and blue, respectively.

Tracker	SV	LR	OC	DF	MB	FM	IR	OR	OV	BC	IV
Eco	0.532	0.496	0.545	0.552	0.451	0.507	0.520	0.523	0.470	0.562	0.526
Ours	0.569	0.466	0.508	0.544	0.458	0.501	0.533	0.532	0.427	0.561	0.549
CREST	0.509	0.406	0.506	0.565	0.484	0.521	0.524	0.540	0.453	0.544	0.573
HCFTstar	0.457	0.342	0.449	0.533	0.431	0.479	0.461	0.490	0.398	0.516	0.522
CF2	0.486	0.323	0.473	0.557	0.446	0.499	0.481	0.503	0.382	0.501	0.526
CACF	0.379	0.278	0.389	0.481	0.391	0.407	0.403	0.417	0.317	0.458	0.465
KCF	0.340	0.238	0.344	0.457	0.342	0.376	0.350	0.375	0.297	0.422	0.414
DSST	0.402	0.269	0.371	0.370	0.345	0.363	0.387	0.394	0.297	0.396	0.454
LOT	0.333	0.230	0.320	0.360	0.294	0.330	0.334	0.340	0.282	0.346	0.318
CSK	0.281	0.205	0.270	0.248	0.240	0.269	0.283	0.289	0.205	0.294	0.301

From Tables 3 and 4, it is clear that the algorithm proposed in this paper performs well on these challenges. It also outperforms CF2, which also uses hierarchical depth features, in terms of overall performance. The CREST algorithm, which uses only one layer of deep features, performs worse than our algorithm, illustrating the benefits of using adaptive hierarchical depth features. However, it can be seen that the proposed algorithm generally performs well in the two challenges of Deformation and Motion Blur. The reason for this may be that the rapid deformation causes blurring in the object's appearance, meaning that the most significant features of the target may be affected. Therefore, the model does not learn more discriminative features and the ability to distinguish the background is reduced. In the follow-up, we will continue to study this problem and try to achieve an improvement.

4.4. Evaluation on UAV123 Benchmark

The UAV-123 [26] is a dataset consisting of low-altitude UAV capture videos, which is fundamentally different from the videos in mainstream tracking datasets, such as OTB50, VOT2014. It contains a total of 123 video sequences and over 110k frames. Unmanned aerial vehicles (UAVs) are increasingly used in daily life, so it is of practical significance to test the proposed algorithm on this dataset. We tested our algorithm on UAV123, using the same evaluation method as the OTB dataset, against nine other algorithms, including: SRDCF [50], CREST [51], CF2 [22], SiamRPN [34], DSST [4], Struck [57], ECO [31], TADT [13], KCF [6], and CSK [56], the comparison results are shown in Figure 11.

As shown in Figure 11, thanks to the proposed method, our tracker achieved a 53.9% success rate and 76.1% precision rate on UAV-123, higher than CF2 and SRDCF, which also used depth features, and similarly improved the performance ECO, TADT and CREST by 1.4% and 2.0%, 2.6% and 3.7%, 5.8% and 8.3%. As the UAV123 dataset contains many UAV aerial images, the targets being tracked in the images are generally small, so it is especially important to learn a more discriminative target feature. Compared with ECO, which uses a complex computational strategy for feature selection, the proposed algorithm in this paper can more accurately identify these small targets. Similar to ECO, TADT also works on feature reduction by designing a regression loss and ranking loss to learn more effective target features, respectively; however, the learned features are not as accurate as the features of the proposed algorithm when facing smaller targets, so the tracking effect is average.

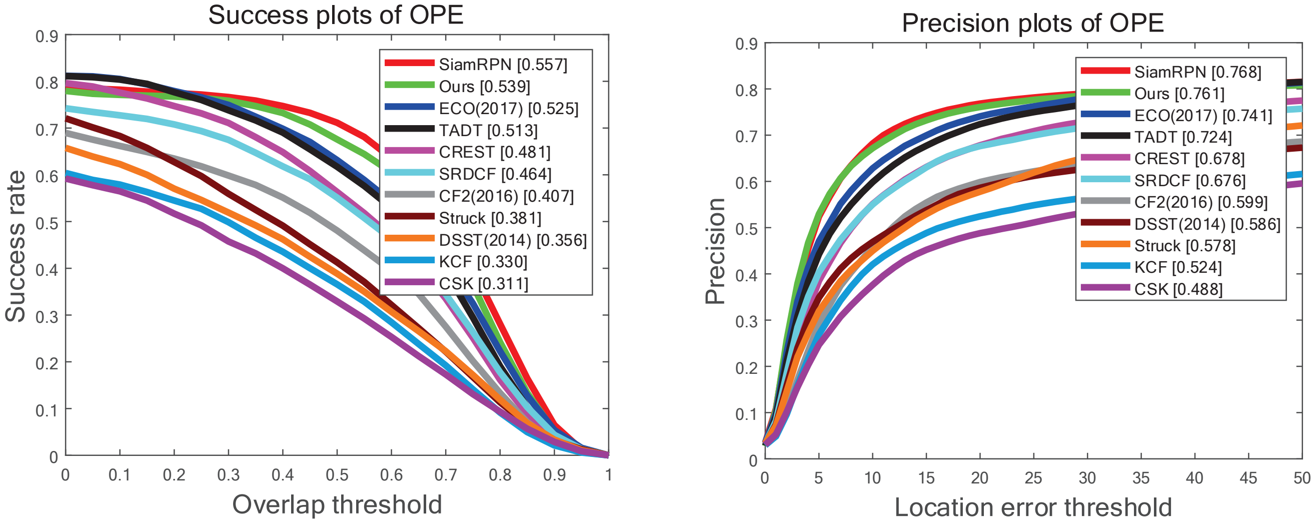


Figure 11. Success and precision rates on the UAV-123 dataset.

Using end-to-end training on a large-scale image dataset while introducing a region proposal network, SiamRPN achieves a higher precision rate than our tracker. However, as it uses ordinary depth features, its performance is weaker than the proposed tracker with a similar target interference.This can also be verified in the lemming and girl2 sequences in Figure 9. Similarly, trackers using manual features, e.g., KCF, Struck, and DSST, all perform worse than trackers using depth features.

4.5. Evaluation on VOT2016 Benchmark

The VOT2016 [28] is a very popular dataset in the field of target tracking, which automatically labels samples to annotate sample coordinates. It uses two metrics, accuracy and robustness, to evaluate the performance of the tracker, as these two have the weakest relationship of the several evaluation metrics used for target tracking to avoid interference. The Expect Average Overlap Rate (EAO) was introduced to rank the algorithms, which better reflects some issues when compared to the OTB dataset. We used VOT-2016 to evaluate our tracker, and compared this with some other trackers.

We selected 11 trackers, including TADT [13], Staple [48], SA-Siam [35], DeepSRDCF [29], MDNet [10], SRDCF [50], CF2 [22], DAT [58], SAMF [59], DSST [49], KCF [6]. To ensure a fair comparison, the results of the other algorithms were downloaded from the VOT-2016 official website. Figure 12 shows the EAO ranking results, and it can be seen that our tracker outperforms TADT, which is innovative in feature modeling, and is in the first position. Table 3 shows more detailed comparison information, including EAO score, OP score and Failures score, and our tracker is in the leading position in all three metrics.

From Table 5, we can see that the proposed algorithm achieves the highest performance for EAO, which indicates the robustness of the proposed algorithm. The proposed algorithm performs better than our baseline algorithm SiamFC on EAO, Overlap and Failure, which can reflect the effectiveness of the proposed multi-channel-aware, deep-feature and adaptive hierarchical deep features in this paper. The last column of Table 4 shows that the proposed algorithm has the lowest tracking failure rate, which means the prediction results have the smallest deviation from the groundtruth. The experimental results further demonstrate the effectiveness of the proposed method.

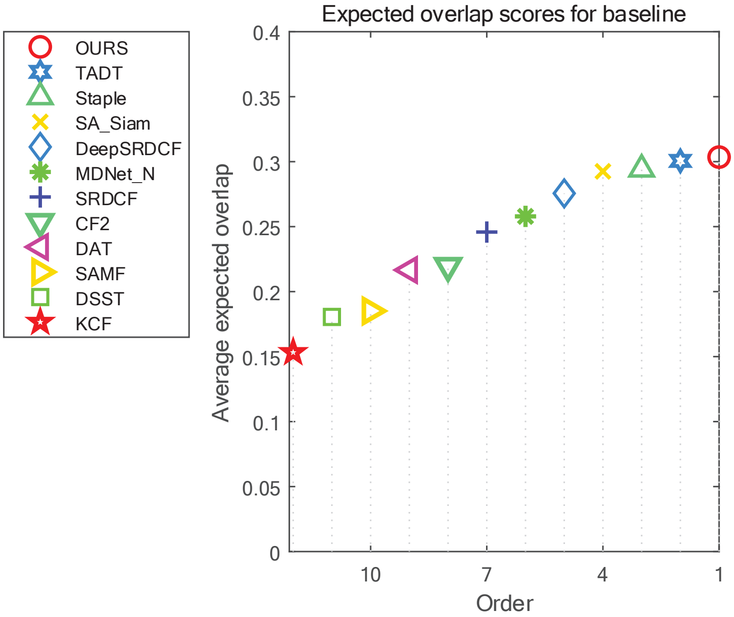


Figure 12. EAO score ranking of the compared trackers VOT2016 dataset.

Table 5. Overall performance on VOT2016 dataset; the top three trackers are marked with red, green and blue, respectively.

Tracker	EAO	Overlap	Failures
Ours	0.303	0.560	18.514
TADT	0.300	0.546	19.973
Staple	0.294	0.540	23.895
SA-Siam	0.292	0.539	19.560
DeepSRDCF	0.275	0.522	20.346
MDNet	0.257	0.538	21.081
SRDCF	0.245	0.525	28.316
CF2	0.219	0.436	23.856
DAT	0.216	0.458	28.353
SAMF	0.185	0.496	37.793
DSST	0.180	0.524	44.813
KCF	0.153	0.469	52.031

4.6. Ablation Studies

The baseline algorithm of the proposed method is SiamFC, to which we introduce a multi-channel-aware, deep-feature module and an adaptive hierarchical deep-feature module. To test the effectiveness of the proposed modules, we conducted ablation experiments to compare the performance of individual modules and the overall algorithm with the baseline tracker SiamFC.

We separately tested two modules on OTB100, and the results are shown in the figure below. It is easy to see that the effect of a single module is not as good as that of two modules acting at the same time. Figure 13 compares success rate and precision rate for these variations in the OTB100 benchmark, where WLR and WCR modules achieved a 63.0% and 83.3%, and 63.0% and 83.4% success rate and precision rate, respectively. Therefore, combining the two reliability modules can achieve the best performance. This is also a significant improvement compared to the baseline tracker SiamFC.

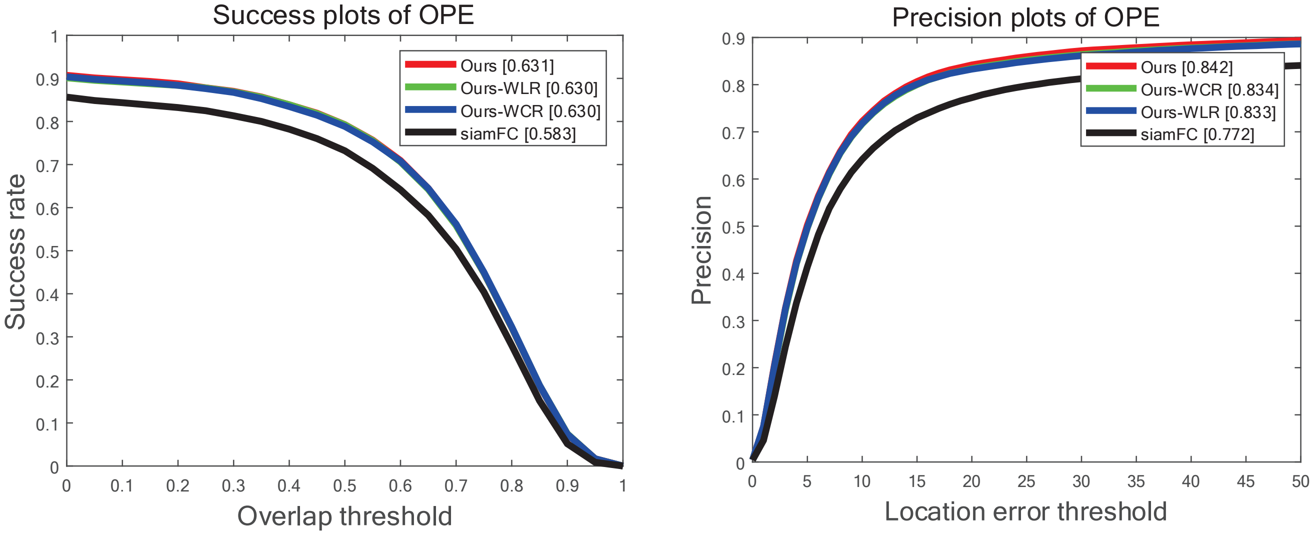


Figure 13. Comparison of the two modules when they act separately. Ours shows the effect when the two modules work together. Ours-WCR representative without multi-channel aware deep feature. Ours-WLR representative without using adaptive hierarchical deep features. SiamFC is our baseline algorithm.

5. Conclusions

This paper proposes a novel scheme to learn target deep aware features, including the learning of multi-channel-aware deep-feature and adaptive hierarchical deep features. The modified mechanism can focus on the modeling of the target appearance, effectively deal with changes in the target appearance, and suppress the interference of background information. The proposed learning multi-channel-aware deep-feature module can focus on important information in the channel, and the proposed adaptive hierarchical deep features module can obtain adaptive feature layer fusion weights. Finally, the two modules work together to enhance the discriminative abality of the tracker. We combine the proposed model with the Siamese framework and prove its effectiveness. In conclusion, this paper proposes a new approach to better utilize the feature modeling abilities of pre-trained neural networks, and a large number of experimental results on several datasets show that the proposed method has a good performance.

From a comparative analysis of different datasets, it can be seen that, compared with the method that uses only single-layer features to model the target, using layered depth features can yield a more discriminative target feature. Compared with the method that uses complex computational strategies for feature dimensionality reduction, our method will be much simpler computationally, and can achieve real-time performance. Compared with the memory network-based method, the proposed method does not have a complicated model update strategy and does not occupy too much memory, which is also beneficial for the efficient use of hardware resources. However, the proposed method performs poorly in some specific scenarios. In future research, we will analyze the reasons for this and try to solve the problems.

In future research, we plan to investigate the use of meta-learning [60] methods to generate an optimal set of initialization parameters, so that the network can be trained online using reliable target information in the first frame, allowing the network to converge faster and obtain a better set of weights for the feature layers and feature channels. A more interesting plan is to enhance the feature representation using the multi-headed attention mechanism proposed by transformer [61] to further improve the performance.

Author Contributions: Conceptualizatio: H.Z., R.D. and A.Z.; methodology: H.Z., R.D. and A.Z.; software: R.D., J.Z., L.L. and F.W.; validation: R.D., J.Z., L.L. and F.W.; analysis: H.Z., R.D. and A.Z.; investigation: H.Z. and A.Z.; resources: H.Z. and A.Z.; writing—original draft preparation: H.Z. and R.D.; writing—review and editing: H.Z., R.D. and A.Z.; visualization: J.Z., L.L. and F.W.; supervision: H.Z. and A.Z.; project administration: J.Z., L.L. and F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (61873246, 62072416, 62102373, 61702462), Program for Science & Technology Innovation Talents in Universities of Henan Province (21HASTIT028), Natural Science Foundation of Henan (202300410495), Zhongyuan Science and Technology Innovation Leadership Program (214200510026),The Science and Technology Research Project of Henan Province(212102310053).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: No conflict of interest exits in the submission of this manuscript, and this manuscript is approved by all authors for publication.

References

1. Grigorescu, S.; Trasnea, B.; Cocias, T.; Macesanu, G. A survey of deep learning techniques for autonomous driving. *J. Field Robot.* **2019**, *37*, 362–386. [CrossRef]
2. Funde, N.; Paranjape, P.; Ram, K.; Magde, P.; Dhabu, M. Object Detection and Tracking Approaches for Video Surveillance over Camera Network. In Proceedings of the 2019 5th International Conference on Advanced Computing and Communication Systems, Coimbatore, India, 15–16 March 2019; pp. 1171–1176. [CrossRef]
3. Baumgartl, H.; Sauter, D.; Schenk, C.; Atik, C.; Buettner, R. Vision-based Hand Gesture Recognition for Human-Computer Interaction using MobileNetV2. In Proceedings of the 2021 IEEE 45th Annual Computers, Software, and Applications Conference, Madrid, Spain, 12–16 July 2021; pp. 1667–1674. [CrossRef]
4. Bousetouane, F.; Dib, L.; Snoussi, H. Improved mean shift integrating texture and color features for robust real time object tracking. *Vis. Comput.* **2012**, *29*, 155–170. [CrossRef]
5. Fawad; Khan, M.J.; Rahman, M.; Amin, Y.; Tenhunen, H. Low-Rank Multi-Channel Features for Robust Visual Object Tracking. *Symmetry* **2019**, *11*, 1155. [CrossRef]
6. Jamil Khan, M.; Rahman, M.; Amin, Y.; Tenhunen, H. High-Speed Tracking with Kernelized Correlation Filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 583–596.
7. He, Z.; Fan, Y.; Zhuang, J.; Dong, Y.; Bai, H. Correlation Filters with Weighted Convolution Responses. In Proceedings of the IEEE International Conference on Computer Vision Workshop, Venice, Italy, 22–29 October 2017.
8. Sun, C.; Lu, H.; Yang, M.-H. Learning spatial-aware regressions for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
9. Sun, C.; Wang, D.; Lu, H.; Yang, M.-H. Correlation tracking via joint discrimination and reliability learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
10. 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.
11. Qi, Y.; Zhang, S.; Qin, L.; Yao, H.; Huang, Q.; Lim, J.; Yang, M.H. Hedged deep tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
12. Hong, S.; You, T.; Kwak, S.; Han, B. Online tracking by learning discriminative saliency map with convolutional neural network. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015.
13. Li, X.; Ma, C.; Wu, B.; He, Z.; Yang, M.H. Target-Aware Deep Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
14. Yang, T.; Chan, A.B. Visual tracking via dynamic memory networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**. [CrossRef]
15. Guo, D.; Shao, Y.; Cui, Y.; Wang, Z.; Zhang, L.; Shen, C. Graph Attention Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021.
16. Cao, Z.; Fu, C.; Ye, J.; Li, B.; Li, Y. HiFT: Hierarchical Feature Transformer for Aerial Tracking. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021.
17. Zhang, H.; Chen, J.; Nie, G.; Hu, S. Uncertain motion tracking based on convolutional net with semantics estimation and region proposals. *Pattern Recognit.* **2020**, *102*, 107232. [CrossRef]
18. Guo, W.; Gao, J.; Tian, Y.; Yu, F.; Feng, Z. SAFS: Object Tracking Algorithm Based on Self-Adaptive Feature Selection. *Sensors* **2021**, *21*, 4030. [CrossRef]

19. Wang, L.; Liu, T.; Wang, G.; Chan, K.L.; Yang, Q. Video tracking using learn-ed hierarchical features. *IEEE Trans. Image Process.* **2015**, *24*, 1424–1435. [CrossRef]
20. Rahman, M.M.; Fiaz, M.; Jung, S.K. Efficient Visual Tracking with Stacked Channel-Spatial Attention Learning. *IEEE Access* **2020**, *8*, 100857–100869. [CrossRef]
21. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H.S. Fully-convolutional siamese networks for object tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016.
22. Ma, C.; Huang, J.B.; Yang, X.; Yang, M.H. Hierarchical convolutional features for visual tracking. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015.
23. Yang, T.; Chan, A.B. Learning dynamic memory networks for object tracking. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
24. Wu, Y.; Lim, J.; Yang, M.H. Online object tracking: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013.
25. Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 1834–1848. [CrossRef] [PubMed]
26. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for UAV tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016.
27. Liang, P.; Blasch, E.; Ling, H. Encoding color information for visual tracking: Algorithms and benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5630–5644. [CrossRef] [PubMed]
28. Kristan, M.; Matas, J.; Leonardis, A.; Felsberg, M.; Cehovin, L.; Fernandez, G.; Vojir, T.; Hager, G.; Nebehay, G.; Pflugfeld, R.; et al. The visual object tracking vot2016 challenge results. In Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), Santiago, Chile, 7–13 December 2015.
29. Danelljan, M.; Hager, G.; Shahbaz Khan, F.; Felsberg, M. Convolutional Features for Correlation Filter Based Visual Tracking. In Proceedings of the IEEE International Conference on Computer Vision, Honolulu, HI, USA, 27–30 June 2016.
30. Danelljan, M.; Robinson, A.; Khan, F.S.; Felsberg, M. Beyond correlation filters: Learning continuous convolution operators for visual tracking. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016.
31. 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.
32. Wang, L.; Ouyang, W.; Wang, X.; Lu, H. Visual tracking with fully convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
33. Held, D.; Thrun, S.; Savarese, S. Learning to track at 100 fps with deep regression networks. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016.
34. 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, 18–23 June 2018.
35. 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, Salt Lake City, UT, USA, 18–23 June 2018.
36. Morimitsu, H. Multiple context features in siamese networks for visual object tracking. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
37. 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 Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
38. Wang, X.; Zhi, M. Summary of object detection based on convolutional neural network. In Proceedings of the Eleventh International Conference on Graphics and Image Processing (ICGIP 2019), Hangzhou, China, 3 January 2020. [CrossRef]
39. Chen, D.; Zhang, S.; Ouyang, W.; Yang, J.; Tai, Y. Person search via a mask-guided two-stream CNN model. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.
40. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019.
41. Li, X.; Liu, Q.; Fan, N.; He, Z.; Wang, H. Hierarchical Spatial-aware Siamese Network for Thermal Infrared Object Tracking. *Knowl. Based Syst.* **2018**, *166*, 71–81. [CrossRef]
42. Liu, Q.; Li, X.; He, Z.; Fan, N.; Yuan, D.; Wang, H. Learning Deep Multi-Level Similarity for Thermal Infrared Object Tracking. *IEEE Trans. Multimed.* **2019**, *23*, 2114–2126. [CrossRef]
43. 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 Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
44. Qin, X.; Fan, Z. Initial matting-guided visual tracking with siamese network. *IEEE Access* **2019**, *7*, 41669–41677. [CrossRef]
45. 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, 13–18 June 2010.
46. 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]

47. 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. Vision* **2015**, *115*, 211–252. [CrossRef]
48. 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, 27–30 June 2016.
49. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Discriminative scale space tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1561–1575. [CrossRef]
50. Li, F.; Tian, C.; Zuo, W.; Zhang, L.; Yang, M.H. Learning Spatial-Temporal Regularized Correlation Filters for Visual Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018.
51. Song, Y.; Ma, C.; Gong, L.; Zhang, J.; Lau, R.W.; Yang, M.H. Crest: Convolutional residual learning for visual tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
52. Lukežič, A.; Vojíř, T.; Zajc, L.; Matas, J.; Kristan, M. Discriminative Correlation Filter with Channel and Spatial Reliability. *Int. J. Comput. Vision* **2018**, *126*, 671–688. [CrossRef]
53. Ma, C.; Huang, J.-B.; Yang, X.; Yang, M.-H. Robust Visual Tracking via Hierarchical Convolutional Features. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 2709–2723. [CrossRef]
54. Mueller, M.; Smith, N.; Ghanem, B. Context-Aware Correlation Filter Tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
55. Oron, S.; Bar-Hillel, A.; Levi, D.; Avidan, S. Locally orderless tracking. *Int. J. Comput. Vision* **2015**, *111*, 213–228. [CrossRef]
56. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. In Proceedings of the European Conference on Computer Vision, Florence, Italy, 7–13 October2012.
57. Hare, S.; Golodetz, S.; Saffari, A.; Vineet, V.; Cheng, M.-M.; Hicks, S.L.; Torr, P.H.S. Struck: Structured Output Tracking with Kernels. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 2096–2109. [CrossRef] [PubMed]
58. Pu, S.; Song, Y.; Ma, C.; Zhang, H.; Yang, M.H. Deep Attentive Tracking via Reciprocative Learning. *arXiv* **2018**, arXiv:1810.03851.
59. 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, 6–12 September 2014.
60. Park, E.; Berg, A.C. Meta-tracker: Fast and Robust Online Adaptation for Visual Object Trackers. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018. [CrossRef]
61. 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 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.