applied sciences

Article

Fast and Accurate Visual Tracking with Group Convolution and Pixel-Level Correlation

Liduo Liu [1,2], Yongji Long [1,2], Guoning Li [1], Ting Nie [1], Chengcheng Zhang [1,2] and Bin He [1,*]

[1] Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China; liuliduo21@mails.ucas.ac.cn (L.L.); liguoning@ciomp.ac.cn (G.L.)
[2] University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: heb@ciomp.ac.cn

Abstract: Visual object trackers based on Siamese networks perform well in visual object tracking (VOT); however, degradation of the tracking accuracy occurs when the target has fast motion, large-scale changes, and occlusion. In this study, in order to solve this problem and enhance the inference speed of the tracker, fast and accurate visual tracking with a group convolution and pixel-level correlation based on a Siamese network is proposed. The algorithm incorporates multi-layer feature information on the basis of Siamese networks. We designed a multi-scale feature aggregated channel attention block (MCA) and a global-to-local-information-fused spatial attention block (GSA), which enhance the feature extraction capability of the network. The use of a pixel-level mutual correlation operation in the network to match the search region with the template region refines the bounding box and reduces background interference. Comparing our work with the latest algorithms, the precision and success rates on the UAV123, OTB100, LaSOT, and GOT10K datasets were improved, and our tracker was able to run at 40FPS, with a better performance in complex scenes such as those with occlusion, illumination changes, and fast-motion situations.

Keywords: feature fusion; pixel-level correlation; Siamese network; attention mechanism

1. Introduction

As one of the research contents in computer vision, visual object tracking has wide application prospects and value in security surveillance, intelligent transportation, autonomous driving, human–computer interaction, autonomous robotics, marine exploration, military target identification, and tracking. Visual object tracking was first carried out using correlation filtering for tracking, and with the development of deep learning, convolutional neural networks have gradually been widely used due to their powerful feature extraction capabilities. Visual object tracking is usually divided into three parts: using a backbone network to extract the target's features, then correlating the template features with the search, and finally utilizing a classification and regression sub-network to predict the center and bounding box of the target. Siamese networks are widely used in object tracking with this structure.

SiamFC [1] first introduced Siamese networks to object tracking. In SiamFC, the template features are correlated with the search features to find the region with the largest response and complete tracking and evaluation. Since then, many works have been carried out on Siamese networks in object tracking. SiamRPN [2] introduced the RPN (region proposal network) structure of object detection to tracking, constructing two branches—one for the regression of the target bounding box, and the other for the classification of the target—where the multi-scale anchor box improves the performance under object scale changes. SiamRPN++ [3] solved the problem of poor results in deep networks due to the destruction of translation invariance when the network is deepened, successfully using ResNet [4] and MobileNet [5] as the backbone networks. SiamFC++ [6] removes the anchor

check for updates

Citation: Liu, L.; Long, Y.; Li, G.; Nie, T.; Zhang, C.; He, B. Fast and Accurate Visual Tracking with Group Convolution and Pixel-Level Correlation. *Appl. Sci.* **2023**, *13*, 9746. https://doi.org/10.3390/app13179746

Academic Editor: Andrea Prati

Received: 25 July 2023
Revised: 12 August 2023
Accepted: 16 August 2023
Published: 29 August 2023

frame and changes the output prediction to an anchor-free style without presetting the anchor frame.

In recent years, transformer structures have boomed in various fields of computer vision. TransT [7] uses the structure of a transformer as the correlation operation, which improves accuracy. Zhao et al. [8] used a transformer structure as the backbone network and utilized a decoder to reconstruct the target appearance within the search region so that the template is close to the search frame, rather than the search frame being directly related to the template image. In this way, the robustness of the tracker is enhanced, even if the appearance of the target has changed. Gao et al. [9] proposed a one-and-a-half-stream structure that uses an adaptive token division method so that the search and template regions have self-attention and cross-attention, as in a two-stream structure, as well as advanced template interactions with the search region, as in a one-stream structure. This structure outperforms some two-stream and one-stream pipelines.

In object tracking, training datasets usually contain many videos and multiple forms of motion. Some annotations may be less accurate due to occlusion and present similarities; thus, some trackers use data processing methods to improve the performance. Yang et al. [10] analyzed the dataset distribution in a low-level feature space and proposed a sample squeezing method to eliminate redundant samples, making the dataset more abundant and informative and increasing the diversity of the dataset. Qi et al. [11] adaptively obtained a tight enclosing box; when the target is in deformation or rotation, the bounding box cannot tightly enclose the target. They also designed a classifier to determine whether the target is occluded or not, which helps to avoid the collection of occluded samples for tracker updates, and to improve accuracy.

However, there are still some challenges in practical applications. Target appearance changes, illumination variation, and occlusion can affect the effectiveness of tracking.

Generally, different features of the object are extracted in different stages of the network. As shown with HDT [12], combining these features from different layers improves the performance of the tracker. HDT uses an improved hedge algorithm to hedge weak trackers from each layer into a strong tracker. In this work, we consider feature fusion by using a 1 × 1 convolution to concatenate and fuse features from different stages in the Siamese backbone network, which can improve the algorithm accuracy. Meanwhile, in order to improve the detection speed, we use a group convolution for the dimensionality reduction. A group convolution [13] can exponentially reduce the number of parameters compared with a normal convolution, which can speed up the operation. In the correlation stage, we use a new matching method, namely a pixel-level correlation operation, in the network, which is able to obtain a correlation feature map with a smaller kernel size and a more diverse target representation, reducing the interference of background clutter and preserving the target boundary and scale information, which is beneficial to the subsequent prediction.

The main contributions of this work are as follows:

(1) Feature fusion: we use not only the last layer output feature map for prediction but also the feature map of layers 3, 4, and 5 for feature fusion to output the prediction;

(2) Pixel-level correlation: the template features are decomposed into spatial features and channel features, which are matched with the search features, instead of correlating channel-by-channel;

(3) Speed improvement: we use a group convolution for the dimensionality reduction, which reduces the number of parameters and the use of activation functions and normalization in the backbone to speed up the detection;

(4) New attention module: we designed two new attention modules, namely, a multi-scale feature aggregated channel attention block (MCA) and a global-to-local-information-fused spatial attention block (GSA), enabling the network to focus on certain parts of the features and reduce the attention on useless parts, thus improving the performance and accuracy of the model.

The rest of this paper is organized as follows. In Section 2, we present research on object tracking based on Siamese networks published in recent years. Section 3 outlines the core of our tracker, including four parts to improve accuracy, from the lightness to the robustness of the algorithm. Section 4 is the experimental section, which presents an ablation study and a comparison of the results of different trackers on different datasets to analyze the validity of our work. Finally, we conclude the paper in Section 5.

2. Related Work

This section introduces the development of object tracking and some object trackers that have been reported in recent years. Object tracking algorithms can be divided into two categories: one is based on correlation filtering, and the other is based on deep learning. The methods based on correlation filtering include MOSSE [14], KCF [15], and DSST [16]. Correlation filtering introduces the convolution theorem from the signal domain to object tracking and transforms the template matching problem into a correlation operation in the frequency domain. This method is fast in operation but has average accuracy in complex scenarios. In recent years, with the development of deep learning technology and the establishment of large-scale datasets, object tracking algorithms based on convolutional neural networks have gradually emerged, among which Siamese network-based visual object trackers are particularly remarkable. A Siamese network consists of two sub-networks with the same structure and shared parameters, which are initially used for picture similarity analysis and metric learning. SINT [17] and SiamFC [1] first introduced Siamese networks to the visual object tracking field. SiamFC inputs the template picture and search sample, obtains the template feature map and search feature map, and then slides the template feature map over the search feature map as part of the correlation operation. The point with the largest response on the search feature map is considered the prediction target. SiamFC, as a fully convolutional network, has a simple structure and high tracking speed, and many subsequent works have been based on it. SiamRPN [2] introduced the RPN structure from object detection to the tracking field. One branch judges whether the object is in the foreground or background, and the other branch predicts the bounding box of the target. However, these algorithms only use shallow networks, and the tracking effect worsens for deep networks. Through the use of SiamRPN++ [3], it was found that the accuracy of deep networks is reduced because the strict translational invariance is broken, but allowing the target to be shifted in a certain range near the center point during training can alleviate the impact, enabling the successful application of deep networks in tracking algorithms. SiamFC++ [6] uses an anchor-free prediction head that does not set any anchor parameters, eliminating the effect of preset hyperparameters on the generalization ability of the algorithm. There are also some transformer structures used in visual object tracking that have achieved good results.

Although these works achieved good results, the tracking accuracy decreases and the inference speed becomes slower in the face of occlusion, object scale changes, background clutter, and other situations. In this paper, we adopt feature fusion and some simplified methods for complex scenes to reduce the computational cost and improve accuracy at the same time, using pixel-level correlation to reduce the influence of background clutter and to refine the object bounding box.

3. Proposed Method

In this section, we describe the network framework in detail. As shown in Figure 1, our model mainly consists of a Siamese network backbone and two sub-network detection heads for the bounding box classification and regression. The Siamese backbone network is fine-tuned from ResNet50, inspired by the transformer structure, reducing the use of activation functions and normalization, and instead using channel attention [18] and spatial attention [19] modules in the classification and regression sub-networks to make the network more accurate in extracting features. Moreover, to improve the inference speed, a group convolution and 1×1 convolution are used for the dimensionality reduction in the feature fusion stage; both of them accelerate the computation speed and reduce the inference time. The cross-correlation operation no longer uses depth-wise correlation [3]; template features and search features are correlated in a pixel-level matching model, which can effectively reduce background clutter and allow the model to refine the object boundary ranges and focus more on the target.

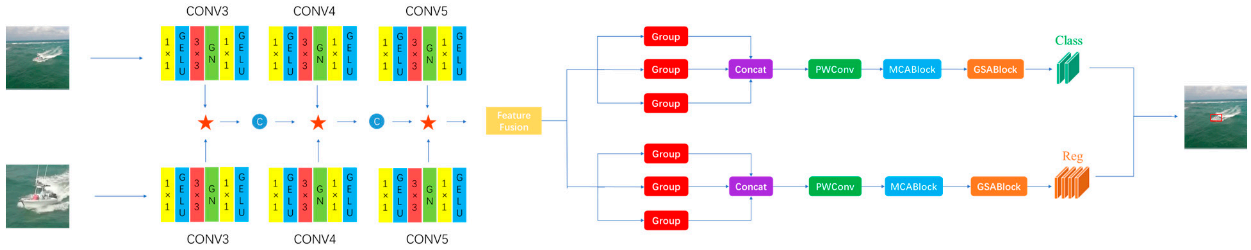


Figure 1. Illustration of our proposed framework. Section 3.1 presents Siamese backbone network, CNN1, CONV3, CONV4, CONV5 represent layer 3, 4, 5 of it. ★ represents the pixel-level correlation method, which is presented in Section 3.2. The feature fusion model is presented in Section 3.3. The classification and regression sub-network using a dual-attention mechanism, CNN2, is presented in Section 3.4.

3.1. Siamese Backbone Framework

Thus far, deep convolutional neural networks have been successfully applied in the field of object tracking. The deepening of these networks has led to improvements in the performance of trackers, such as ResNet [4], ResNeXt [13], and MobileNet [5], which have achieved a good performance. ResNet50, as a classical network, has good robustness and effectiveness and is usually used in trackers as a feature extraction backbone network while modifying the backbone network in order to cater to the accuracy requirements of the tracking task.

Ren et al. [20] proposed Flow Alignment FPN (FAFPN) to align feature maps of different resolutions to solve the semantic misalignment problem when fusing features of different layers. We set the steps of the conv4 and conv5 feature layers to 1 and remove the down-sampling operation so that the output resolution of the last three blocks is the same; meanwhile, to increase the receptive field, the use of a dilated convolution [21] to extract more features has been proven to be effective. Transformers [22], as excellent model architectures, are widely used in various vision tasks. Compared to convolutional neural networks, transformers usually use less activation functions and normalization operations with good results. Inspired by this, a similar method is applied in the backbone.

The original ResNet50 network uses a convolution of 7 × 7 with a 2-step size in the first layer, following a maximum pooling to complete a 4-fold down-sampling of the input image. The transformer divides the image into patches of the same size and feeds each patch into the network. We change the first layer of the network to a convolution of 4 × 4 with a 4-step length, with no overlap between convolutions. Compared with the previous one, the convolutional kernel with K = 4 and S = 4 has a smaller kernel size and a larger step size. The computation and parameter numbers are shown in Equations (1) and (2):

$$FLOPs_{old} : \left(\frac{N}{2}\right)^2 \times 7^2 \times 3 \times 64 = 2352N^2 \tag{1}$$

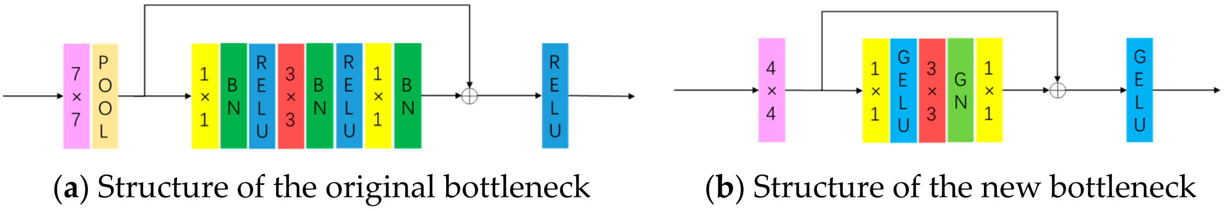
$$FLOPs_{new} : \left(\frac{N}{4}\right)^2 \times 4^2 \times 3 \times 64 = 192N^2, \tag{2}$$

where N denotes the input size, and 3 and 64 are the input and output channels in the first layer of the network, leading to a significant reduction in computation.

Another difference between transformers and CNNs is the use of activation functions and normalization. RELUs are widely used in various CNN networks as simple and efficient activation functions. GELUs, as a variant of RELUs, are used in the latest transformer structures, such as the Swin Transformer and BERT, and can effectively alleviate neuron death and avoid gradient disappearance. Therefore, we use GELUs [23] instead of RELUs.

Traditional convolutional neural networks use an activation function after each layer of convolution. In order to speed up the operation, we remove the activation function after the 3 × 3 convolution, only using it after the 1 × 1 convolution.

As for normalization, BN is the most common normalization method, which is widely used in various vision tasks. Meanwhile, the setting of the batch size affects the final result. Models with an insufficient batch size are not suitable for convergence, while there may be a reduction in the generalization ability of models with too large a batch size. Group normalization [24] can be used for the normalization of samples, and it has been used in many application scenarios. We use GN instead of BN and also reduce its use to improve the inference speed. The modified Resnet50 consists of a new bottleneck (see Figure 2), and the inference speed is about 5 FPS faster.



(**a**) Structure of the original bottleneck (**b**) Structure of the new bottleneck

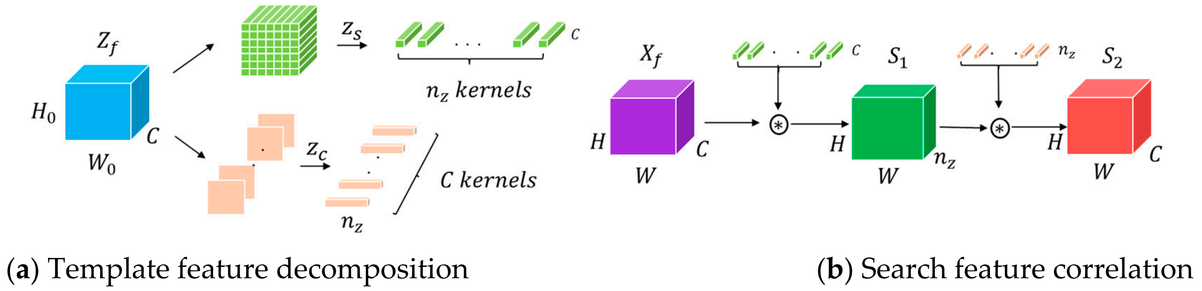
Figure 2. (**a**) original bottleneck using triple activation function and triple normalization. (**b**) new bottleneck using two activation function and one normalization.

3.2. Pixel to Global Correlation

Correlation is the most important part of object tracking, which combines template features with search features and then connects them to the output of the classification and regression sub-networks. Unlike depth-wise correlation [3], which correlates template features with search features channel by channel, in this work, we use pixel to global correlation [25], which decomposes template features and correlates every pixel with the search features to obtain a correlated feature map S. This correlation can effectively suppress background interference, improve the target response on the feature map, and further improve the accuracy of the target bounding box.

The process is shown in Figure 3, where the template features $Z_f \in R^{C\times H_0\times W_0}$ are first decomposed into spatial feature vectors $Z_s = \{Z_s^1, Z_s^2, \ldots, Z_s^{n_z}\}$, $Z_s^i \in R^{C\times 1\times 1}$ for each pixel.

$$n_z = H_0 \times W_0 \tag{3}$$



(**a**) Template feature decomposition (**b**) Search feature correlation

Figure 3. Illustration of pixel to global correlation, where Z_f is the template feature, and X_f is the search feature. (**a**) The template feature is decomposed into feature vectors Z_s and Z_c. Z_s converts the template feature into feature vectors according to each pixel position. Z_c converts the template feature maps of each channel into feature vectors. (**b**) Feature vectors Z_s and Z_c are successively correlated with the search feature X_f to obtain features S_1 and S_2. S_2 is the correlation feature map combining the template and search features.

Similarly, the template features are also converted into channel feature vectors, $Z_c = \{Z_c^1, Z_c^2, \ldots, Z_c^c\}$, $Z_c^i \in R^{n_z \times 1 \times 1}$, according to the channel dimension. The search features are first correlated with the spatial feature vectors Z_s to obtain feature map S_1 based on Equation (4):

$$S_1 = X_f * Z_s. \tag{4}$$

Then, feature map S_1 is correlated with the channel feature vectors X_f to obtain feature map S_2 based on Equation (5):

$$S_2 = S_1 * Z_c, \tag{5}$$

where $*$ represents the convolution process. Feature map S_2 is obtained after both the channel features and spatial features of the template are correlated. Then, the classification and regression sub-networks complete the target prediction.

Naive correlation [1] and depth-wise correlation [3] use whole template features as kernels to correlate the search features so that the adjacent sliding windows on the feature map produce similar responses, blurring the spatial information. As a refinement method, pixel to global correlation decomposes the template into 1$*$1 feature sub-kernels according to the space and channel to correlate the search region, which effectively reduces background interference and further improves the accuracy of the target bounding box, avoiding the blurring of features.

3.3. Feature Fusion

In order to make full use of the features extracted from the backbone network and the advantages of deep networks, features from different layers are used in our feature fusion, and at the same time, in order to speed up the inference, a group convolution [13] is used to first reduce the feature dimensions to simplify the number of parameters and then aggregate the features via a pointwise convolution.

Group convolutions [13] have been widely applied as efficient convolution methods. Their specific process is shown in Figure 4. $C_1 \times H \times W$ is used as the input, and the output is $C_2 \times H \times W$, which represents the channel, height, and width of the convolution. The input is divided into g groups, and each group uses a convolution with a kernel size of $k \times k$ and C_1/g channels. Compared with the number of parameters of an ordinary convolution, i.e., $k \times k \times C_1 \times C_2$, the number of parameters of the group convolution is $k \times k \times C_1 \times C_2/g$, which is $1/g$ of an ordinary convolution, greatly reducing the parameter redundancy. A group convolution is equivalent to decomposing the input and processing the data in parallel, which can speed up the operation. The number of parameters and FLOPs is calculated using Equations (6) and (7):

$$Params_{normal}: k \times k \times C_1 \times C_2, \quad FLOPs_{normal}: k \times k \times C_1 \times C_2 \times H \times W \tag{6}$$

$$Params_{group}: \frac{k \times k \times C_1 \times C_2}{g}, \quad FLOPs_{group}: \frac{k \times k \times C_1 \times C_2 \times H \times W}{g}. \tag{7}$$

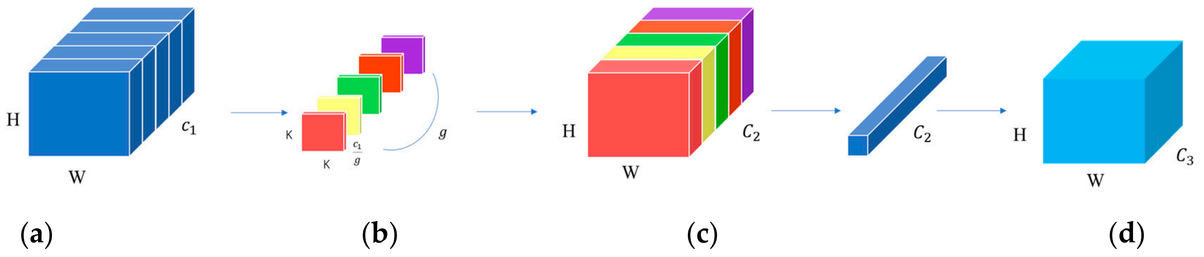


Figure 4. Feature fusion model with a group convolution and pointwise convolution: (**a**) denotes input features, (**b**) denotes group convolution, (**c**) denotes pointwise convolution, and (**d**) denotes output.

Generally speaking, during the tracking process, there may be problems such as illumination changes and scale variation, which require the tracking task to use as much feature information as possible. It is usually considered that in the shallow layer of a network, the network extracts the fine-grained information [26] of the object, such as its color and shape, to help locate the object's position, and as the network deepens, the network extracts the semantic information of the object. Fusing these features from different deep and shallow layers helps to track the target. After correlation, the features of the three stages are concatenated together, and the fusion of the features is implemented using a pointwise convolution [27], which achieves the fusion of cross-channel information quickly and efficiently.

3.4. Classification and Regression Sub-Network

The aim of an attention mechanism is to allow the model to learn how to allocate its own attention and weight the input signal. An attention mechanism scores each dimension of the input and then weights the features according to the score, increasing the weight of interesting parts and decreasing the weight of uninteresting parts, so that the network adaptively highlights the features that are important to the downstream model or task. In this work, two attention modules, namely, channel attention and spatial attention modules, are implemented in the classification and regression sub-network (CNN2), as shown in Figure 5. The features are first reduced in dimensionality via a group convolution [13]; then, a PW convolution [27] is used for feature fusion, and finally the dual channel and spatial attention module is followed.

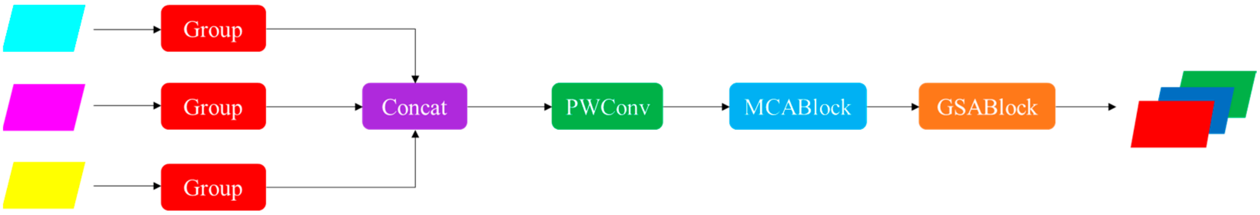


Figure 5. Illustration of the classification and regression sub-network (CNN2).

The multi-scale feature aggregated channel attention block (MCA) is a mechanism for tuning the network at the channel level, as shown in Figure 6. The input features are first divided into four parts, each of which is reduced to half of the original channel via a convolution layer. Two operations are performed independently: one directly uses global average pooling to make the features $1 \times 1 \times C$ in size, with a global perceptual field, aggregating the global features and squeezing information from the channels after the sigmoid activation to obtain the channel weights, which are then multiplied back to the divided features; the other uses an additional convolution layer and then performs the same operation as the former. The four parts adopt the same operation and concatenate together, completing the attention enhancement of the channel dimension, making the network automatically focus on the channels that are important.

The MCA block is based on Equations (8)–(10), where F is the input, S is the spilt operation, Cat is the concatenate operation, δ is the activation function, C_1 and C_2 represent the convolution layers, and GAP stands for global average pooling.

$$F_1 = C_1(S(F)) \tag{8}$$

$$F_{SE1} = Cat(\delta(GAP(F_1)) \times F_1, \delta(GAP(C_2(F_1))) \times C_2(F_1)) \tag{9}$$

$$F_{SE} = Cat(F_{SE1}, F_{SE2}, F_{SE3}, F_{SE4}) \tag{10}$$

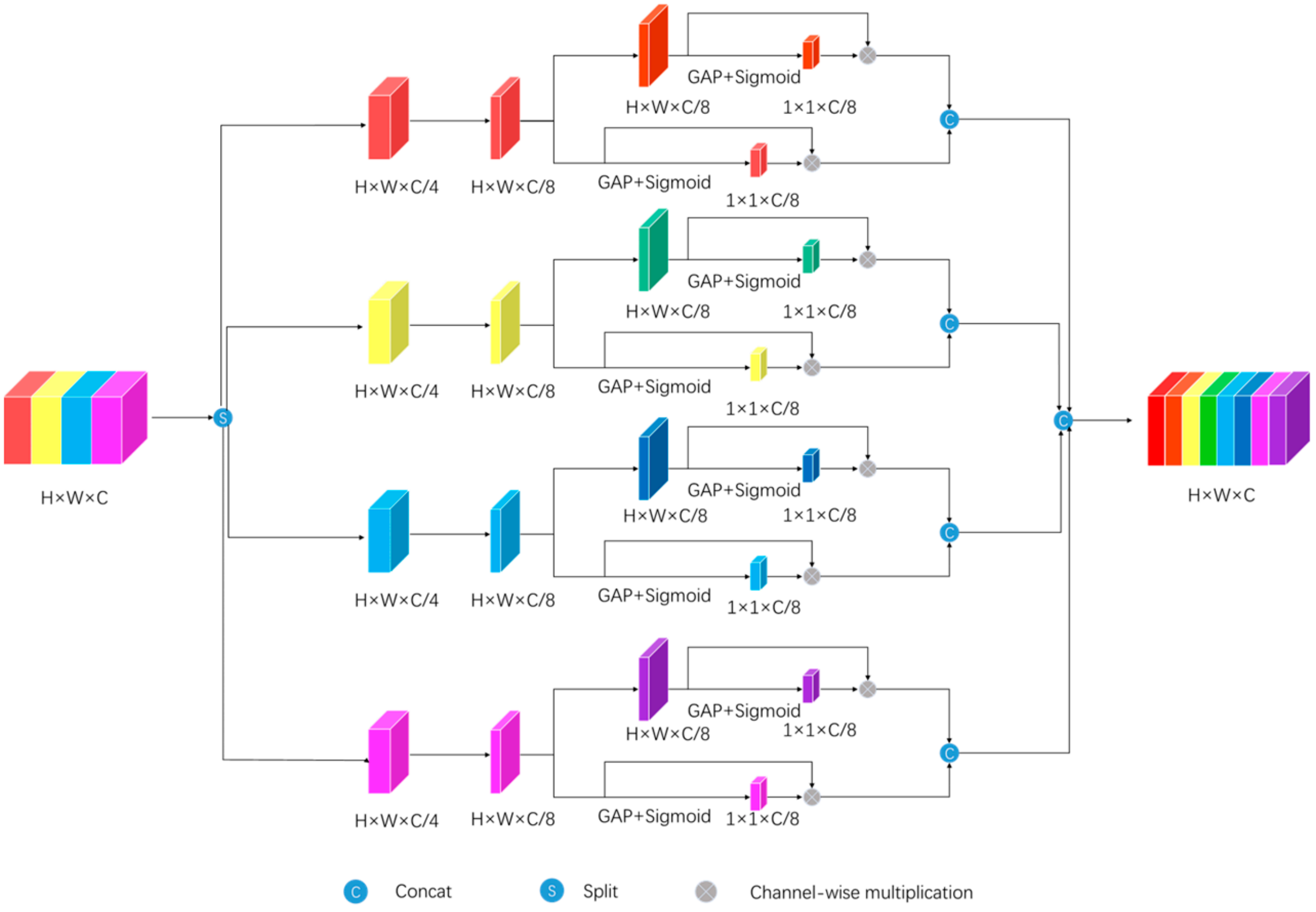


Figure 6. The multi-scale feature aggregated channel attention block (MCA).

The global-to-local-information-fused spatial attention block (GSA) is similar to the channel attention block in that it weights the network from the spatial dimension as shown in Figure 7. The same input features are divided into four parts, using two convolution layers, average pooling, and maximum pooling [28] for each feature point of the network along the channel direction to obtain four 1∗h∗w feature maps. The pooling map and convolution map are concatenated before another convolution layer to obtain weights in the spatial dimension, which are then multiplied back to the input. Two parts are then added to complete the attention enhancement of the spatial dimension, making the network focus on the more important regions. We employ the GSA block in Equations (11)–(13).

$$F_{SPA1} = F \times C_2(Cat(C_1(F), GAP(F))) \tag{11}$$

$$F_{SPA2} = F \times C_4(Cat(C_3(F), GMP(F))) \tag{12}$$

$$F_{SPA} = F_{SPA1} + F_{SPA2}, \tag{13}$$

where *GAP* and *GMP* represent average pooling and maximum pooling, *F* is the input feature, C_1, C_2, C_3, C_4 represent the convolution layers, and *Cat* is the concatenate operation.

After the template features are correlated with the search features (pixel-level correlation), they are fed into the classification and regression sub-networks (CNN2), which predict whether it is an object or background, along with the bounding box of the target. As shown in Figure 8, the two sub-networks use the same correlation module as the input and do not use separate correlation modules, which also reduces the amount of computation and speeds up the operation of the network. The algorithm finally runs at 40 FPS, which is nearly 9 FPS faster than SiamCAR.

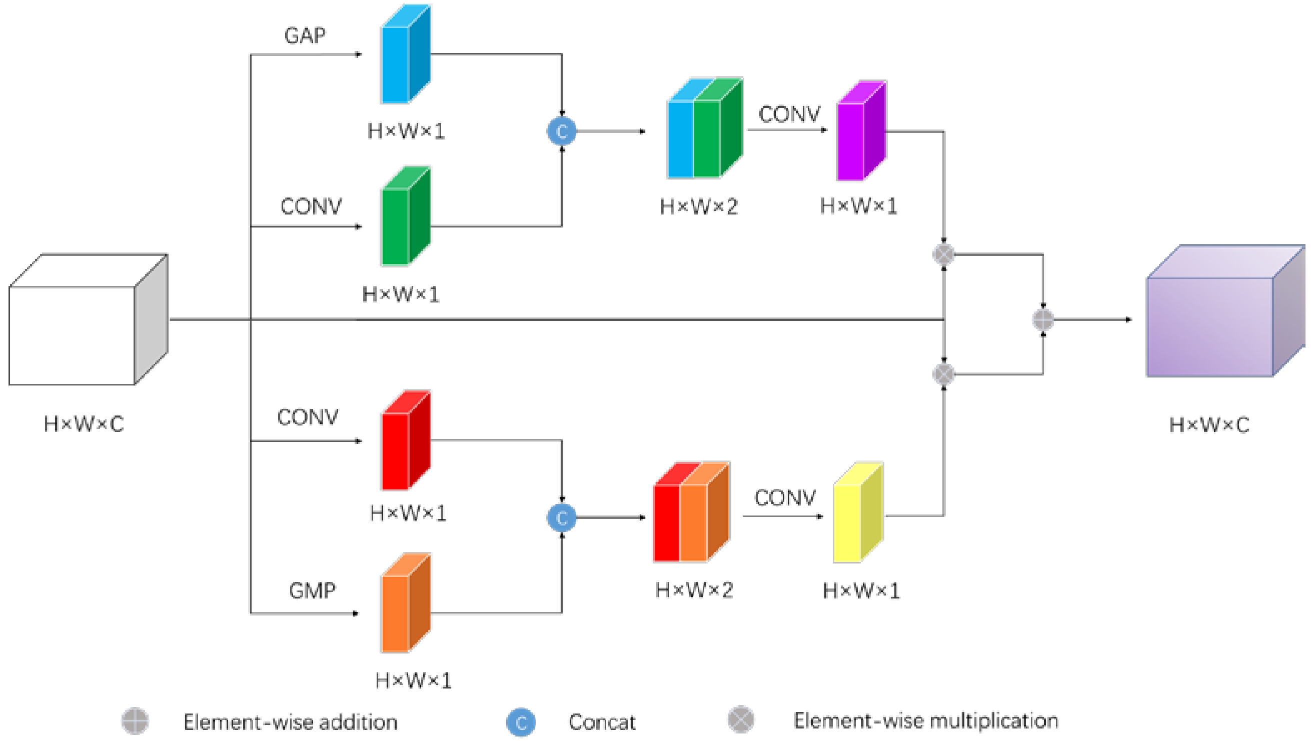
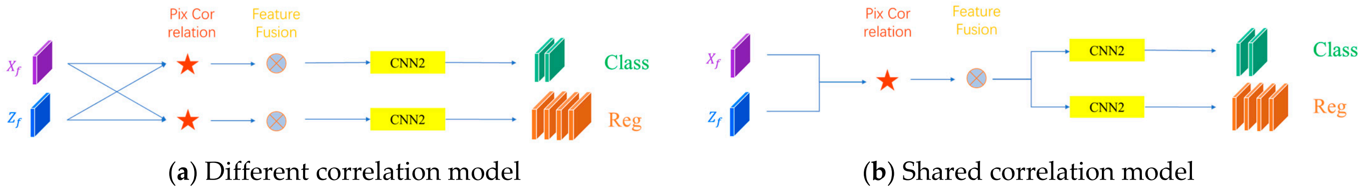


Figure 7. The global to local information fused spatial attention block (GSA).



(**a**) Different correlation model

(**b**) Shared correlation model

Figure 8. Different connections between the correlation module and prediction sub-network: (**a**) separate correlation module connected to the classification and regression sub-networks; (**b**) use of a shared correlation module.

4. Experiments

4.1. Implementation Details

The initial model of the backbone was derived from ResNet50 [4] trained on the COCO [29] dataset, a migration learning approach that is commonly used for network training today. We used the Lasot [30], Got10k [31], ImageNet VID [32], and YouTube Bounding Boxes [33] datasets as training sets. The search region was cropped to 255 × 255, and the template region was cropped to 127 × 27 for training. The initial learning rate was 0.001, and 20 training epochs were performed using stochastic gradient descent (SGD). In the first 5 epochs, the learning rate increased from 0.001 to 0.005, and in the last 15 epochs, it gradually decreased from 0.005 to 0.0005. Meanwhile, the parameters of the backbone network were frozen in the first 10 epochs, where only the neck and output parts were trained, and in the last 10 epochs, the parameters of the backbone network were unfrozen, and the network was trained as a whole. Finally, the model was tested and evaluated on the UAV123 [34] and OTB100 [35] datasets.

4.2. Ablation Study

In order to explore the effect of the multi-layer feature fusion, ablation comparison experiments were conducted. Table 1 shows that the use of multi-layer feature fusion is better than just using a single feature, and the effect is better when using the three-layer feature fusion of CONV3, CONV4, and CONV5 than when using the two-layer feature fusion of CONV4 and CONV5, indicating that the features extracted from the different stages of the network are not the same, and fusing multi-layer features is beneficial to improving the tracking accuracy. The correlation method based on pixel matching of the template features also shows an improvement compared to the channel-by-channel correlation method, with an improvement of 0.8% on the UAV123 dataset. The addition of the attention module to the network further improves the effect of the network, and the use of both spatial and channel attention models enables the network to achieve the best effect, with a final accuracy of 65.5% on the UAV123 dataset.

Table 1. Ablation study of the proposed tracker on UAV123. L3, L4, and L5 represent conv3, conv4, and conv5, respectively. DW/Pix stands for depth-wise correlation and pixel to global correlation.

L3	L4	L5	Correlation	MCA Block	GSA Block	AUC
		√	DW			0.616
	√	√	DW			0.620
√	√	√	DW			0.628
√	√	√	Pix			0.636
√	√	√	Pix	√		0.647
√	√	√	Pix	√	√	0.655

In order to analyze the effect of fusing multi-layer features, we tested the model on three datasets. As shown in Table 2, the use of three feature maps from different convolution layers leads to the best results on all three datasets, which shows that the use of multi-layer feature fusion is beneficial to improving the accuracy.

Table 2. Ablation study of the use of feature maps from different layers.

Conv Layers Used	UAV123		OTB100		GOT10K	
	AUC	P	AUC	P	AO	$SR_{0.5}$
Conv5	0.616	0.814	0.690	0.905	0.585	0.680
Conv4, 5	0.620	0.822	0.693	0.907	0.591	0.689
Conv3, 4, 5	0.628	0.827	0.695	0.908	0.594	0.693

Another ablation experiment was conducted to explore the attention mechanism and pixel-level correlation. As shown in Table 3, the baseline uses three convolution layers with pixel-level correlation, while MCA and GSA are the multi-scale feature aggregated channel attention block and the global-to-local-information-fused spatial attention block. Every addition improves the accuracy. In the end, all modules are used, achieving the best performance with an AUC of 65.5% and a precision rate of 85.2%.

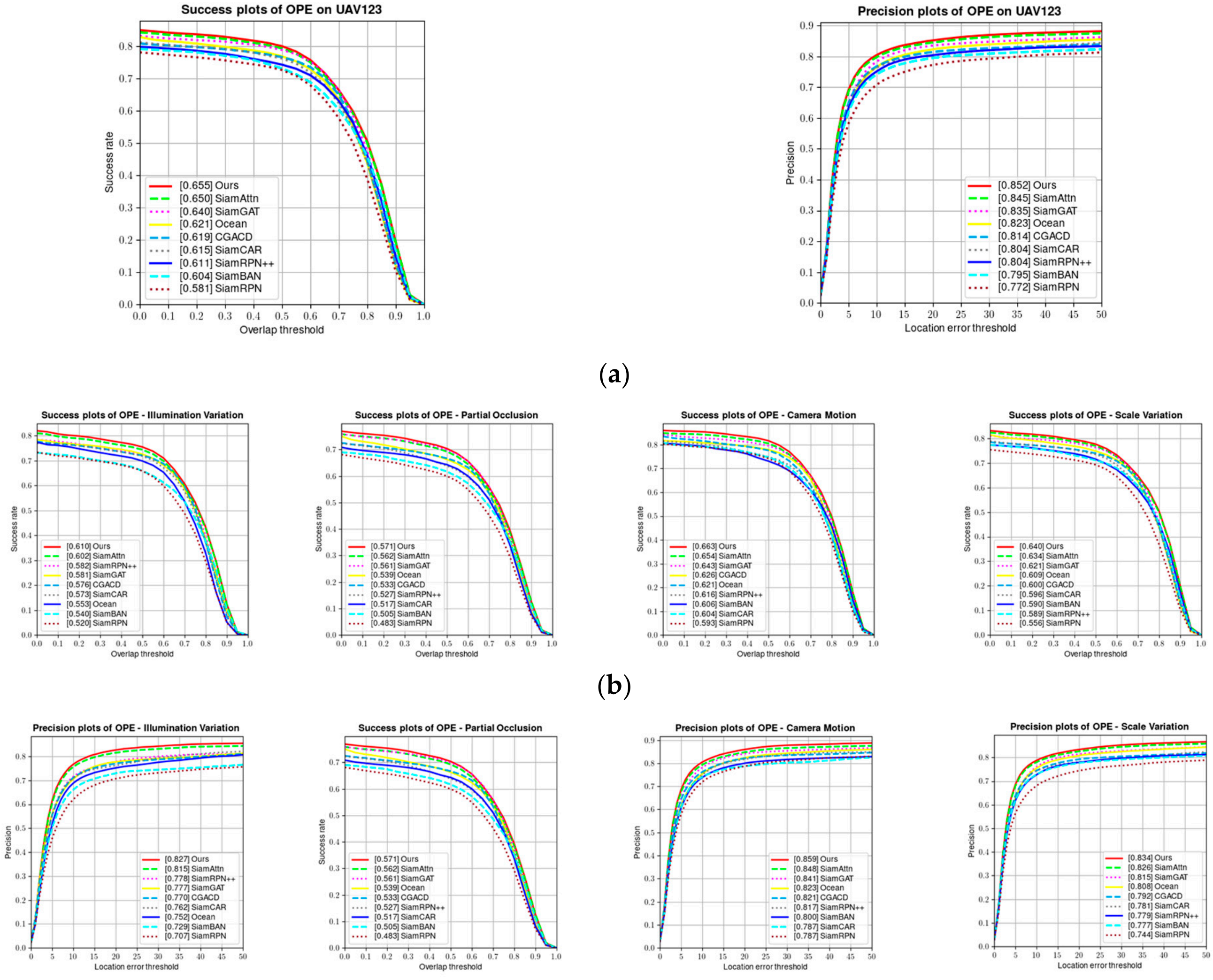
Table 3. Ablation study of the attention model and correlation method.

Method	AUC	P_{Norm}	P
Baseline (3layers + pix)	0.636	0.857	0.830
+MCA	0.647	0.869	0.844
+MCA +GSA	0.655	0.876	0.852

4.3. Results on UAV123

UAV123 [34] is a collection of 123 high-definition videos captured using UAVs during aerial photography, containing a variety of targets such as pedestrians, ships, planes, and cars; a variety of scenes including fields, roads, and water, with many activity styles; and occlusions, scale changes, lighting changes, and camera movements in order to increase the tracking challenge. The evaluation metrics include success, precision, and norm precision. Precision is the center position error, using the average center position error of all frames in a sequence to evaluate the performance of the trackers. Success is the proportion of area overlapped between the detection and the real area; generally, the area under the curve is used as its value.

We compared our work with other state-of-the-art trackers, including SiamRPN++ [3], Ocean [36], SiamBAN [37], and SiamGAT [38]. As shown in Figure 9, compared with SiamCAR, our tracker shows a 4.0% improvement in success and a 4.8% improvement in precision.



(**a**)

(**b**)

(**c**)

Figure 9. (**a**) Overall success and precision plots of our tracker on UAV123 compared with other trackers. (**b**) Success plot for visual attributes. (**c**) Precision plot for visual attributes.

We also compared the trackers in terms of visual attributes, including illumination changes, occlusion, scale changes, and background clutter, as shown in Figure 10. Our tracker ranks first, which shows that our tracker has the ability to cope with illumination changes, occlusion, and scale changes.

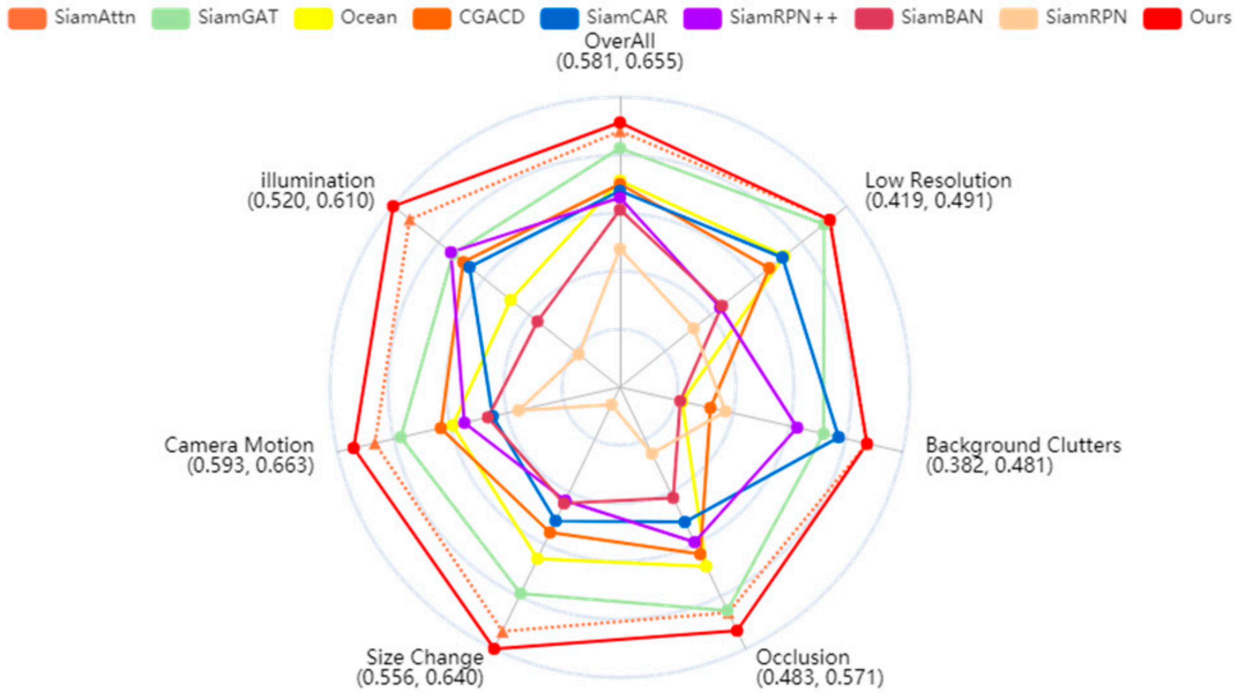
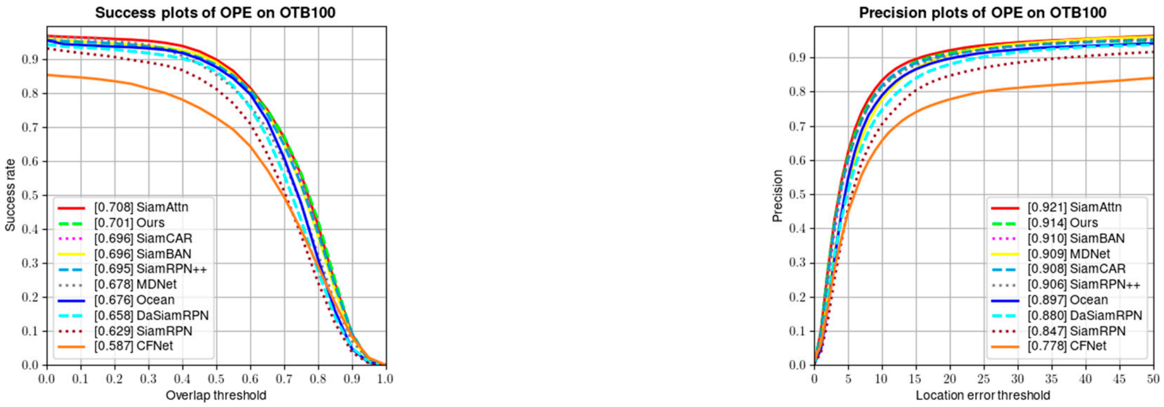


Figure 10. Comparison of success in terms of visual attributes.

4.4. Results on OTB100

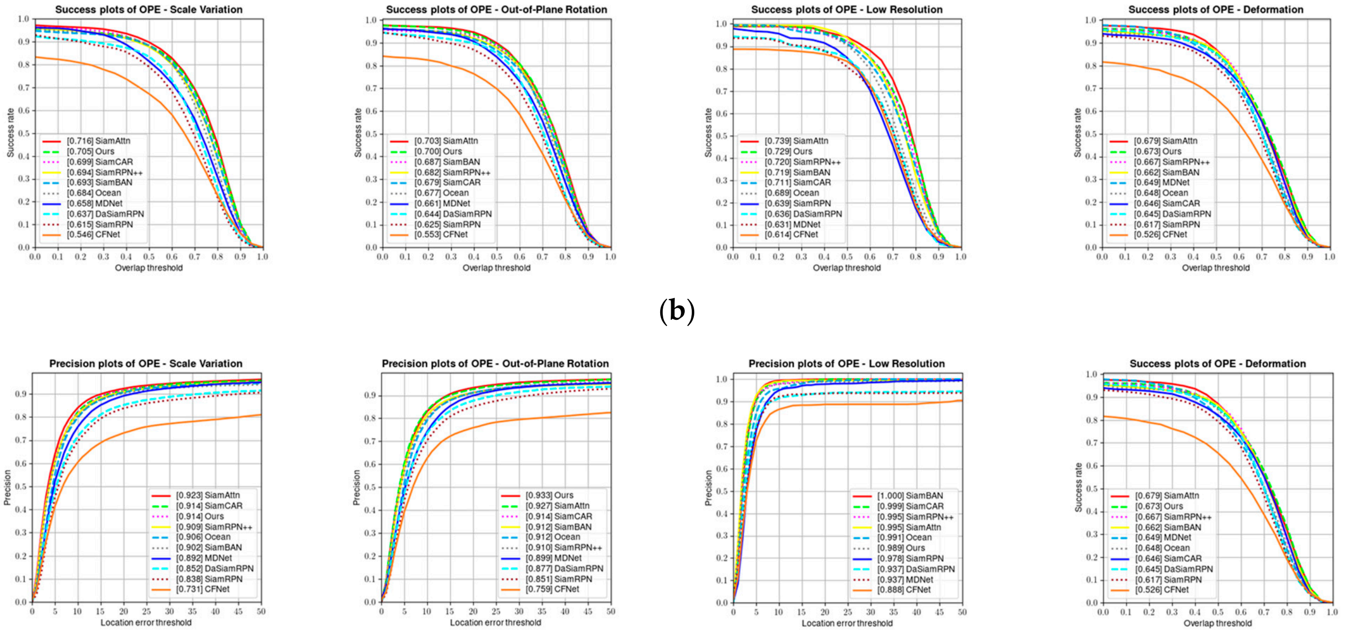
OTB100 is a widely used object-tracking dataset. It contains 100 video sequences with attributes such as fast motion, motion blur, and low resolution. We compared our tracker with other state-of-the-art trackers including SiamCAR [39], SiamRPN++ [3], SiamBAN [37], and CFNet [40].

Figure 11 illustrates the success and precision plots of the compared trackers. Our track-er achieves better results than SiamCAR [39] and SiamBAN [37], with a faster speed in terms of scale variation, out-of-plane rotation, low resolution, etc. Our tracker obtains a success rate of 0.701 and a precision rate of 0.914. The integration of the attention and pix-el-level correlation methods enables the tracker to work well in scenarios with low resolution, scale variation, etc.



(**a**)

Figure 11. *Cont.*



(**b**)

(**c**)

Figure 11. (**a**) Overall success and precision plots of our tracker on OTB100 compared with other trackers. (**b**) Success plot for visual attributes. (**c**) Precision plot for visual attributes.

4.5. Results on GOT10K and LaSOT

As a large tracking dataset, GOT10K contains more than 10,000 videos, and it is populated with more than 560 categories of moving objects and 87 motion patterns—more than other datasets. We tested our model on the test set. As shown in Table 4, compared with SiamCAR [39], SiamFC++ [6], and Ocean [36], our tracker achieves an AO of 60.7%, which is 1.2% better than that of SiameseFC++ and generally better than that of the other trackers.

Table 4. Comparison with other trackers on the GOT10k test set.

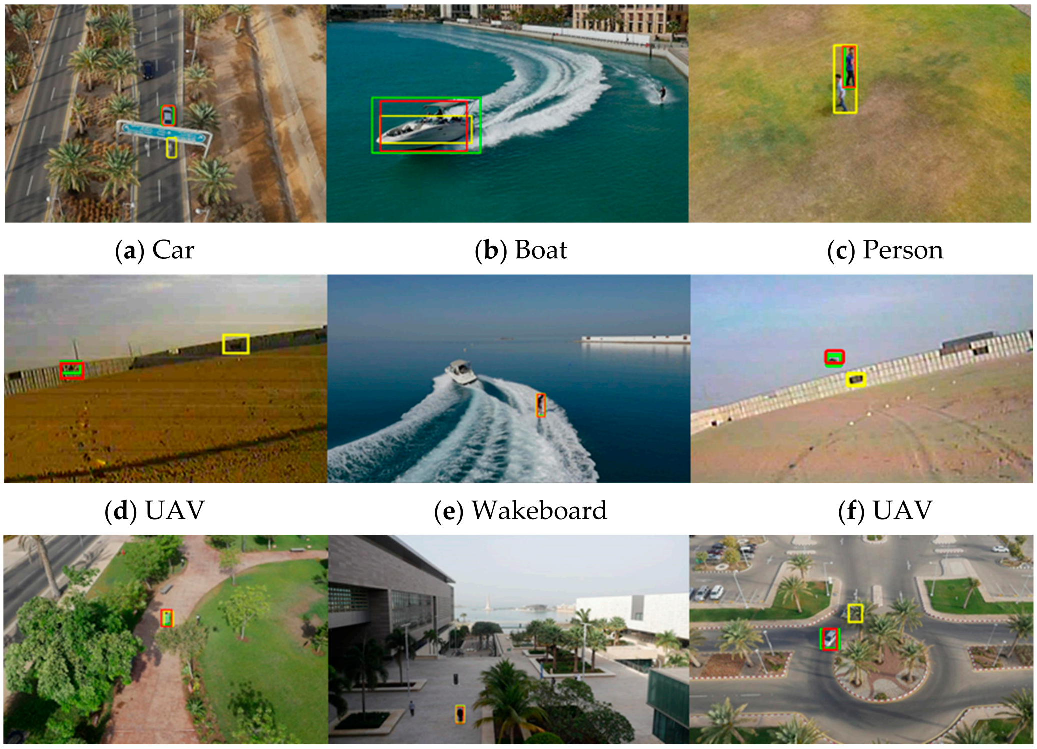
	SiamFC	**SiamRPN**	**SiamRPN++**	**SiamCAR**	**SiamFC++**	**Ocean**	**Ours**
AO	0.374	0.483	0.517	0.569	0.595	0.611	0.607
$SR_{0.5}$	0.404	0.581	0.616	0.670	0.695	0.721	0.713

LaSOT contains 70 object categories and provides an equal number of sequences for each category to mitigate potential category bias, resulting in a collection of 1400 sequences with an average video length of 2512 frames, constituting a high-quality tracking dataset. We tested our tracker on this test set. As shown in Table 5, our tracker outperforms Ocean by 1.2% and has a better performance than the other trackers, which shows its effectiveness and generalizability.

Table 5. Comparison with other trackers on the UAV123, OTB100, and LaSOT datasets in terms of the AUC.

	SiamRPN++	**SiamCAR**	**SiamBAN**	**CGCAD**	**PGNet**	**Ocean**	**Ours**
UAV123	0.611	0.604	0.615	0.623	0.619	0.621	0.655
OTB100	0.695	0.696	0.696	0.691	0.703	0.676	0.701
LaSOT	0.469	0.507	0.514	0.518	0.531	0.560	0.572

Figure 12 shows that our model can track successfully in the face of size variation, occlusion, and low resolution, improving the success and precision rates. The inaccuracy of the boat tracking is due to the fixed viewpoint, and as the boat is traveling from far to near, its size changes rapidly, so the tracker does not work well. Our model aggregates multi-layer features with different receptive fields, which reduces the problem of accuracy degradation due to the change in the size of the object. The person tracking inaccuracy is due to the close distance and high similarity of the two people, resulting in the bounding box containing both. Pixel-level correlation is a more refined correlation method that can refine the bounding box and diminish tracking exceptions caused by background interference. Due to the small size and fast movement of UAVs, tracking errors often occur. The attention module can enhance the feature extraction ability of the network, allowing the network to focus on important features and track successfully. Therefore, our tracker provides a better accuracy than the other algorithms in different situations. Meanwhile, compared to SiamCAR's inference speed of 31FPS, our model runs at 40FPS, representing an improvement of 9FPS, which is an improvement in both speed and success.


(**a**) Car (**b**) Boat (**c**) Person

(**d**) UAV (**e**) Wakeboard (**f**) UAV

(**g**) Person (**h**) Person (**i**) Car

Figure 12. Comparisons of tracking results from different trackers. Targets including person, car, boat, UAV, and images present challenging attributes such as low resolution, occlusion, fast motion, and size variation. Green boxes denote ground truth, yellow boxes are results from SiamCAR, and red boxes are our model results.

5. Conclusions

In this work, we propose a Siamese framework with a group convolution and pixel-level correlation for visual object tracking, with training from end to end, using multi-layer feature fusion and attention mechanisms to improve the feature extraction capability of the network, which works well under fast motion, occlusion, etc. We designed two attention modules: a multi-scale channel attention block (MCA) and a global-to-local spatial attention block (GSA), which enable the network to extract more meaningful features in the classification and regression sub-network. During tracking, pixel-level correlation reduces background interference and provides more refined target boundaries, and it decomposes the template features from the channel and spatial dimensions and uses every pixel feature to correlate the template and search regions. Furthermore, in order to improve the inference speed, our tracker uses a group convolution, which reduces the number of parameters in the network, as well as the use of activation functions and normalization in the backbone. The final inference speed reaches 40FPS, nearly 9FPS faster than that of SiamCAR. Our model achieved a 65.5% success rate and an 85.2% precision rate on the UAV123 dataset, outperforming SianCAR by 4%; a 70.1% success rate and a 91.4% precision rate on the OTB100 dataset; and a 57.2% success rate on LaSOT, outperforming Ocean by 1.2%. Accordingly, our tracker performs better than other trackers and effectively improves the results under lighting changes and occlusion, showing its effectiveness and generalizability.

Author Contributions: Conceptualization, L.L. and B.H.; methodology, L.L.; software, L.L.; validation, L.L., Y.L. and T.N.; formal analysis, G.L.; data curation, L.L.; writing—original draft preparation, L.L.; writing—review and editing, B.H., G.L. and C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science Foundation of China, grant number 62105328.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the Computer Vision–ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–10 and 15–16 October 2016; pp. 850–865.
2. 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; pp. 8971–8980.
3. 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/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4282–4291.
4. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
5. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
6. Xu, Y.; Wang, Z.; Li, Z.; Yuan, Y.; Yu, G. Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 12549–12556.
7. 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, Nashville, TN, USA, 20–25 June 2021; pp. 8126–8135.
8. Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; He, K. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1492–1500.
9. Bolme, D.S.; Beveridge, J.R.; Draper, B.A.; Lui, Y.M. Visual object tracking using adaptive correlation filters. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 2544–2550.
10. Henriques, J.F.; Caseiro, R.; Martins, P.; Batista, J. High-speed tracking with kernelized correlation filters. *IEEE Trans. Pattern Anal. Mach. Intell.* **2014**, *37*, 583–596. [CrossRef] [PubMed]
11. Danelljan, M.; Häger, G.; Khan, F.S.; Felsberg, M. Discriminative scale space tracking. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1561–1575. [CrossRef] [PubMed]
12. Tao, R.; Gavves, E.; Smeulders, A.W. Siamese instance search for tracking. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1420–1429.
13. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
14. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
15. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
16. Zhang, Z.; Peng, H.; Fu, J.; Li, B.; Hu, W. Ocean: Object-aware anchor-free tracking. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 771–787.

17. Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 11976–11986.
18. Wu, Y.; He, K. Group normalization. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
19. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
20. Lin, M.; Chen, Q.; Yan, S. Network in network. *arXiv* **2013**, arXiv:1312.4400.
21. Liao, B.; Wang, C.; Wang, Y.; Wang, Y.; Yin, J. Pg-net: Pixel to global matching network for visual tracking. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 429–444.
22. 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 Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
23. 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, Long Beach, CA, USA, 15–20 June 2019; pp. 5374–5383.
24. 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]
25. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
26. 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; pp. 3074–3082.
27. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for uav tracking. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; pp. 445–461.
28. 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; pp. 2411–2418.
29. 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, Seattle, WA, USA, 13–19 June 2020; pp. 6668–6677.
30. Guo, D.; Shao, Y.; Cui, Y.; Wang, Z.; Zhang, L.; Shen, C. Graph attention tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 9543–9552.
31. Guo, D.; Wang, J.; Cui, Y.; Wang, Z.; Chen, S. SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6269–6277.
32. Real, E.; Shlens, J.; Mazzocchi, S.; Pan, X.; Vanhoucke, V. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5296–5305.
33. 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.
34. Yu, F.; Koltun, V.; Funkhouser, T. Dilated residual networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 472–480.
35. Zhao, H.; Wang, D.; Lu, H. Representation Learning for Visual Object Tracking by Masked Appearance Transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 18696–18705.
36. Gao, S.; Zhou, C.; Zhang, J. Generalized Relation Modeling for Transformer Tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 18686–18695.
37. 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; pp. 4303–4311.
38. Yang, Y.; Li, G.; Qi, Y.; Huang, Q. Release the power of online-training for robust visual tracking. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 12645–12652.
39. Qi, Y.; Qin, L.; Zhang, S.; Huang, Q.; Yao, H. Robust visual tracking via scale-and-state-awareness. *Neurocomputing* **2019**, *329*, 75–85. [CrossRef]
40. Ren, H.; Han, S.; Ding, H.; Zhang, Z.; Wang, H.; Wang, F. Focus on Details: Online Multi-object Tracking with Diverse Fine-grained Representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 11289–11298.

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