Computers, Materials & Continua
DOI: 10.32604/cmc.2022.026784
Article

Tech Science Press

Anchor-free Siamese Network Based on Visual Tracking

Shaozhe Guo[1], Yong Li[1,*], Xuyang Chen[2] and Youshan Zhang[1]

[1]Engineering University of People's Armed Police, Xi'an, 710000, China
[2]Dresden University of Technology, Dresden, 01069, Germany
*Corresponding Author: Yong Li. Email: liyong@nudt.edu.cn
Received: 15 January 2022; Accepted: 06 May 2022

Abstract: The Visual tracking problem can usually be solved in two parts. The first part is to extract the feature of the target and get the candidate region. The second part is to realize the classification of the target and the regression of the bounding box. In recent years, Siameses network in visual tracking problem has always been a frontier research hotspot. In this work, it applies two branches namely search area and tracking template area for similar learning to track. Some related researches prove the feasibility of this network structure. According to the characteristics of two branch shared networks in Siamese network, we also propos a new fully convolutional Siamese network to solve the visual tracking problem. Based on the Siamese network structure, the network we designed adopt a new fusion module, which realizes the fusion of multiple feature layers at different depths. We also devise a better target state estimation criterion. The overall structure is simple, efficient and has wide applicability. We extensive experiments on challenging benchmarks including generic object tracking-10k (GOT-10K), online object tracking benckmark2015 (OTB2015) and unmanned air vehicle123 (UAV123), and comparisons with state-of-the-art trackers and the fusion module commonly used in the past, Finally, our network performed better under the same backbone, and achieved good tracking effect, which proved the effectiveness and universality of our designed network and feature fusion method.

Keywords: Classification; regression; anchor-free; fusion module

1 Introduction

Visual object tracking has been paid more and more attention and research in the past few years, and it has been widely used in the fields of video surveillance, human-computer interaction and unmanned driving [1–3]. However, because there are still many constraints in real conditions, such as changes in illumination, scale changes, and object occlusion, this is still regarded as a challenging task.

The architecture of Siamese network is a mainstream system in today's Visual object tracking, and there are already a series of mature visual tracking methods [4–6]. This network architecture regards the visual object tracking problem as a matching problem between targets, and learns the similarity of matching through the cross-correlation between target template features and search area features.

A common strategy is to improve tracking performance by adding a region proposal network (RPN) [7], such as Siamese-RPN [5]. These methods are usually divided into two stages. The first stage is to extract the candidate regions on the relevant feature map, and the second stage is to encode the target appearance information on the template branch to the RPN feature. Although this strategy is a great breakthrough compared with previous detection methods, there are still some problems. Such Siamese trackers rely on pre-defined anchor boxes, which will lead to too many hyperparameters, such as the numbers, sizes and aspect ratios of anchor boxes, which need to be adjusted in different situations to achieve good performance. Moreover, this strategy is prone to produce too many negative samples, making the matching result wrong and causing tracking failure. Recently, another new type of siamwse network system has appeared. Its representative is SiamFC++ [6], which uses an anchorless detector and a new quality evaluation strategy, and the effect has been greatly improved.

Most Siamese tracks use shallow networks such as AlexNet or deeper networks such as residual neural network18 (ResNet18) as the backbone network. This approach does not make use of shallow features. A small number of Siamese tracks have also tried to use shallow and deep layers for simple fusion or splicing as feature maps, but the effect is not very obvious. Such matching of language information in different dimensions is of great significance to tracking performance.

Feature fusion has also been tried in recent years, but the current feature fusion does not take into account the particularity of the two branches of the Siamese network, and it is easier to find feature targets when searching for areas at the same latitude. Based on this principle, we have designed a new way of fusing feature layers in several different dimensions in the Siamses network.

The main contributions of this article are as follows:

- Propose a target tracking framework based on Siamese classification and regression, and adjust it on the original AlexNet to provide richer semantic information.
- A brand-new fusion module is designed, which uses the interrelationship between features of different layers to improve the ability to recognize the foreground and the background.
- Improved on the anchor-free proposal network of the original SiamFC++, and proposed a better target state estimation criterion.
- Our algorithm achieves state-of-the-art performance on OTB2015 tracking datasets and runs in realtime.

2 Related Work

Visual tracking has been a hot topic in recent years. With the development of big data and the deepening of neural network research, many tracking methods have achieved amazing results [8–10]. The following will mainly introduce the family of the trackers based on Siamese network and anchor-free in dection.

2.1 Siamese Network Based Tracking

The tracking effect of a tracker often depends on features extraction [11], template update [12] and bounding box regression [13]. The idea of convolution theorem was first applied to tracking by Bolme et al. [14]. This idea has greatly improved trackers based on correlation filters. With the development of deep learning, models based on Convolutional Neural Networks (CNN) have been widely used. Many trackers with deep feature representation have achieved the best results in popular tracking benchmarks, and the family of Siamese occupies a lot of them.

As an important system in target tracking, Fully-Convolutional Siamese (SiamFC) [4] first proposed a fully convolutional Siamese network structure, which uses a new matching strategy to maintain real-time monitoring while maintaining high tracking accuracy. Since then, Siamese network has gradually been researched and innovated. Dynamic siamese (Dsiame) [15] constructed a dynamic Siamese network, and proposed a fast learning model and a multi-layer fusion method of unitization. SAsiam [16] proposed a two-layer Siamese network to achieve better learning of semantic information and appearance features. SiamRPN integrates the Regional Proposal Network on the original basis, and proposes a new idea for many researchers to make people pay more attention to the design of the tracking frame.

However, most of them still use AlexNet as the backbone. Although its structure is simple and the number of parameters is small, the extracted features are limited, and it is impossible to better learn shallow languages. SiamRPN++ [16] replaced AlexNet with ResNet, used a deep backbone to improve performance, and achieved a good tracking effect. SiamDW shows an unfilled residual unit, which is a deeper network structure. These experiments have shown that Siamese network can adopt appropriate model structure and training strategy on backbone to improve the tracking effect of trackers.

2.2 Anchor-free in Dection

Visual tracking and object detection have many similarities, and some methods can learn from each other. The anchor can also be divided into anchor-free and anchor-based in target detection. In the deep learning era, object detection usually adopts the strategy of classification and regression of candidate regions for detection. Most of the candidate regions are anchors generated by methods such as sliding window methods. For example, in the Anchor-based method, the RPN structure proposed by Ren et al. [7] has achieved good results when introduced into Siamese. However, because the anchor-based method often requires a large number of hyperparameters to set the anchor, it has a greater impact on the tracking effect of trackers [17].

In recent years, the anchor-free method has gradually been widely studied and used. The early You Only Look Once (YOLO) [18] directly predicted the bounding box and classification score from the entire picture, reaching an amazing detection speed. CornerNet [19] used a pair of corner coordinates to detect the target, which greatly improves the detection accuracy. Since then, Anchor-free models have emerged in endlessly. The fully convolutional one-stage (FCOS) model used multi-level prediction, which achieves that the effect can be comparable to mainstream anchor-based target detection algorithms without anchors. In Siamese network, SiamFC++ adopts the idea of anchor-free, The FCOS structure is used, which saves a lot of parameters and ranks in the forefront on multiple benchmarks. But the shallow semantic learning is not enough, and the detection target frame for small objects is not accurate enough.

3 Proposed Method

This part will introduce our Siamese network, which is an anchor-free tracking method. It uses the Siamese network to extract the features of the target module X and the search area Z, and uses the improved FCOS module for classification. Compared with SiamFC++, our proposed fusion module and target state estimation criteria can better combine shallow features and track small targets, and this idea can be applied to other Siamese networks. The network architecture is shown in Fig. 1.

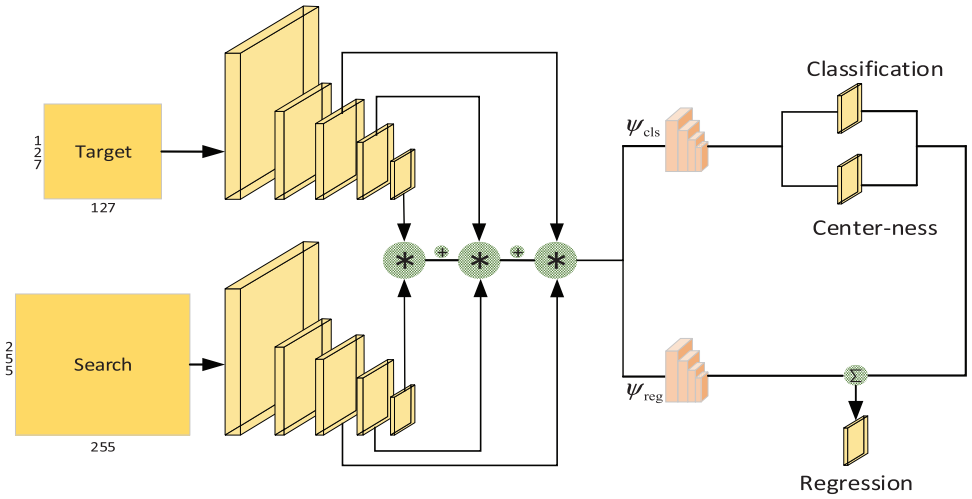


Figure 1: Network architecture diagram. boxes denote feature maps. backbone uses the Alexnet structure, and for the sake of clarity, we omit its details. the output enters our feature fusion structure, improves the feature information, and finally outputs the score results

3.1 Siamese-based Feature Extraction and Matching

Here we use a fully convolutional network to construct a Siamese network. Take AlexNet as an example. The network can be seen as a backbone of two branches. One branch is used to learn the feature representation of the target, and the other branch is used to search for the area. The two branches share the same CNN architecture as its backbone model, and the input is a tracking template of $127 * 127$ and a search template of $255 * 255$. The subsequent prediction part of the network obtains the response map, and decodes it to obtain the location and scale information of the target. In order to better extract the shallow feature information, we perform feature fusion on the output of the backbone. Because there is a certain gap between the shallow information and the deep information, if simple fusion is directly carried out, the cross-correlation results of the deep and shallow features are difficult to achieve the desired effect. Inspired by SiamRPN++ [16] and SiamFC++ [6], this paper adopts a new feature extraction method, extracting feature maps from the 3rd, 4th, and 5th layers of the network respectively, and performing cross-correlation operations on feature maps of the same depth. The cross-correlation operations are as follows: Formula (1).

$$f(z,x) = \varphi(x) * \varphi(z) \tag{1}$$

The $*$ here represents the channel-by-channel convolution operation between the tracking template and the search template. The generated $f(z,x)$ feature map has a large amount of feature information for subsequent classification and regression.

The semantic information contained in the deep features can better classify the target, and the large amount of visual information such as color and shape contained in the shallow features can better locate the target. In previous studies, there have been many experiments combining shallow and deep features of networks, and they have been improved to a certain extent [19]. This article considers that although there is a certain connection between the shallow visual information and the deep semantic information, if the relevant calculations are directly performed, they cannot be fully integrated, and certain characteristic information will be lost. We extract the last 3 convolutional blocks of AlexNet

separately, and use the extracted feature maps as tracking templates and search templates for related operations, and finally cascade to obtain feature maps. Formula (2) is as follows

$$f_i(z,x) = \varphi_i(x_3) * \varphi_i(z_3) + \varphi_i(x_4) * \varphi_i(z_4) + \varphi_i(x_5) * \varphi_i(z_5)\ i \in cls, reg \tag{2}$$

In formula (2), i represents the task type ("cls" represents classification, "reg" represents regression). In order to improve the training and tracking speed, we use 1∗1 kernel convolution to reduce the dimensionality of the extracted feature maps to achieve The amount of parameters is greatly reduced, and finally the $f(z,x)$ feature map enters the subsequent network for classification and regression operations.

3.2 Target State Estimation and Loss Function Criterion

On the basis of SiamFC, combined with the ideas of FCOS and SiamFC++, we made the following designs for classification tasks and regression tasks.

After the backbone and feature fusion, the output is divided into two categories. The classification part passes through Ψ_{cls} to generate a feature map $A^{cls}_{w\times h\times 2}$, and each point contains $(i,j,;)$, which represents the 2D vector of each point, divides the scores of the foreground and the background, and achieves the function of dividing the positive and negative samples in the image; the regression part passes through Ψ_{reg} to generate a feature map $A^{reg}_{w\times h\times 4}$ Generate offset regression, each point contains $(i,j,;)$, which represents the 4D vector $t^* = (l^*, t^*, r^*, b^*,)$ to improve the position of the output bounding box, w represents the width of the feature map, and h represents the height of the feature map.

In this article, set the backbone stride to 8(s = 8), assuming that the original point on the feature map $f(z,x)$ is (x,y), then the corresponding prediction in the classification part The position of the bounding box is $\left(\left|\frac{s}{2}\right| + xs, \left|\frac{s}{2}\right| + ys\right)$, and the predicted point is in the predicted bounding box The middle is regarded as a positive sample, and vice versa is a negative sample. In the regression part, through the 4D vector containing the position information, the position of the predicted point can be expressed as formula (3), where (x_0, y_0) and (x_1, y_1) represent the upper left corner of the true bounding box and Bottom right corner.

$$\begin{aligned} l^* &= \left(\left|\frac{s}{2}\right| + xs\right) - x_0, t^* = \left(\left|\frac{s}{2}\right| + ys\right) - y_0 \\ r^* &= x_1 - \left(\left|\frac{s}{2}\right| + xs\right),\ b^* = y_1 - \left(\left|\frac{s}{2}\right| + ys\right) \end{aligned} \tag{3}$$

Then using the predicted bounding box information, combined with formula (3), the regression loss of the regression part is calculated as follows.

$$L_{reg} = \frac{1}{\sum^{\delta_{(i,j)}}} \sum_{ij} \delta_{(i,j)} L_{IOU}\left(A^{reg}(i,j,;), t^*\right) \tag{4}$$

Among them, $\delta(i,j)$ means dividing positive and negative samples, the positive sample is 1, and the negative sample is 0, as shown in formula (5).

$$\delta_{(i,j)} = \begin{cases} 1 & i < \left|\frac{s}{2}\right| + xs \text{ and } j < \left|\frac{s}{2}\right| + ys \\ 0 & otherwise \end{cases} \tag{5}$$

Refer to the loss function design of the classification part in the article [6], and introduce the design of centrality. The target center position score is 1, and the farther the target center position is, the smaller the score, and the predicted bounding box and outside the box are 0. Such a design can

perform a good quality assessment and improve the tracking accuracy. Calculation formula (6) is as follows

$$C(i,j) = \delta_{(i,j)} * \sqrt{\frac{min(l^*, r^*)}{max(l^*, r^*)} \times \frac{min(t^*, b^*)}{max(t^*, b^*)}} \tag{6}$$

Because the size, scale, and shape of different target tracking objects have certain differences, and the appearance and shape of the target will also change when there is occlusion, intersection over union (IOU) design cannot accurately reflect the overlap between the real frame and the predicted frame. For scale changes It is not sensitive and does not contain much directional information. This article combines the IOU design of article [20], the design of L_{IOU} is as formula (7):

$$L_{IOU} = \frac{Intersection(B, B^*)}{Union(B, B^*)} - \frac{\left|\frac{C}{Union(B,B^*)}\right|}{|C|} \tag{7}$$

where B is the real frame area, B^* is the predicted frame area, and C is the smallest rectangular area that can contain B and B^*. Finally, our classification part loss function is as follows:

$$L_{cen} = C(i,j) \times logA^{reg}_{w\times h\times 1}(i,j) + (1 - C(i,j)) \times logA^{reg}_{w\times h\times 1}(i,j) \tag{8}$$

The total loss function of the Siamese network we designed is as follows:

$$L = L_{cls} + \lambda_1 \delta_{(i,j)} L_{cen} + \lambda_2 \delta_{(i,j)} L_{reg} \tag{9}$$

4 Experiments

4.1 Implementation Datails

The Siamese network we proposed is implemented using Python with PyTorch and trained on 2 GPUs for NVIDIA GTX 2080. For the sake of fairness, the backbone network chooses the most common and simple AlexNet network. Considering the amount of computation and computational burden, a modified AlexNet pretrained from ImageNet is used. The input size of the target template and the search template are the same as the size of [6] 127 respectively. $\times$ 127 and 255 $\times$ 255.

4.2 Training Datails

In order to make the tracking results more realistic, we chose two different data sets as training and testing. Use GOT-10k [21] for training. In order to compare with [6], I adopted the same training parameter strategy, freezing the parameters from Conv1 to Conv3, and fine-tuning Conv4 and Conv5. For the convolutional layer without pre-training, I used the zero center with a standard deviation of 0.01 The Gaussian distribution is initialized, and the learning rate adopts a dynamic change strategy, linearly increasing from the initial 10^{-7} to 2×10^{-3}, and finally the optimizer selects stochastic gradient descent (SGD) and sets the parameter to 0.9.

4.3 Test Phase

The commonly used OTB2015 is selected as the test set. The OTB2015 data set has 100 videos, and the tracking results are used in the form of reports to ensure fair comparison. The test phase uses an offline tracking strategy, and only uses the objects in the first frame of the sequence as the template patch. Select their corresponding confidence scores as the metric, and the box with the highest score as the output, to update the target status.

There are two main evaluation indicators: The expected average overlap considering both bounding box overlap (success) and Location error (precision).

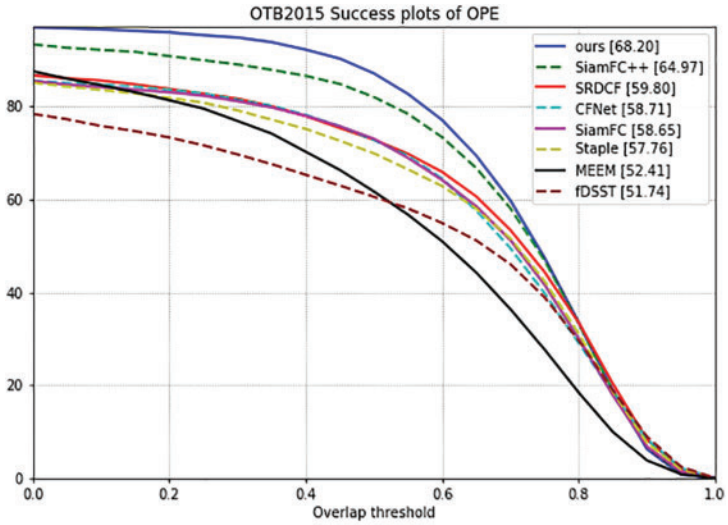
It is then compared with other advanced trackers in UAV123 dataset, which contains 123 video sequences and more than 110K frames. The target in the dataset has challenging features such as rapid motion, light change, and masking, which makes the dataset of wide comparative value in tracking projects.

4.4 Comparison Results

We evaluated ours method on OTB2015 and used the most advanced trackers SiamFC++ [6], SiamFC [4], Efficient Convolution Operators for Tracking (ECO) [22], Correlation Filter based tracking (CFNET) [23] and some other baselines or state-of-the art approaches.

As shown in the Fig. 2, in the overlap ratio (success) and Location error (precision) of 8 algorithms on the OTB2015 data set. It can be seen that the performance of the algorithm in this paper is superior. In terms of success plot, the algorithm of this paper is 0.68, which is 6.2% higher than the 0.64 of the same type of SiamFC++ as the same backbone. In terms of precision plot, the algorithm of this paper is 0.90, which is an improvement of 0.86 compared to SiamFC. That's 4.6%.

In order to more accurately show the performance of the algorithm and the accuracy of the improved positioning frame, we selected 4 representative video sequences on OTB2015 as the display, from top to bottom, they are Basketball, Biker, CarDark, Football1, these The video sequence includes multiple challenging content such as occlusion, shadow, rotation, and characters. Each extracts the same 5 frames, and compares the labels of the positioning boxes of each algorithm. As shown in the Fig. 3, the white is the tracking result of the algorithm in this article, and the blue is the tracking result. It is the SiamFC tracking result, the green is the SiamFC++ tracking result, and the purple is the SiamRPN tracking result. It can be seen from the image that the algorithm in this paper has a good tracking effect for target movement, drift, vehicle movement, and people. Compared with other algorithms, the size of the tracking frame is more suitable.



(a)

Figure 2: (Continued)

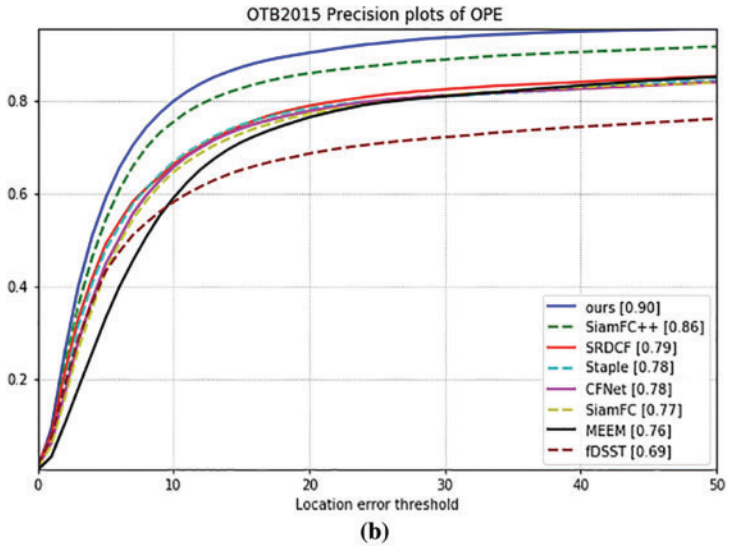


Figure 2: Success and precision plots of one-pass evaluation (OPE)

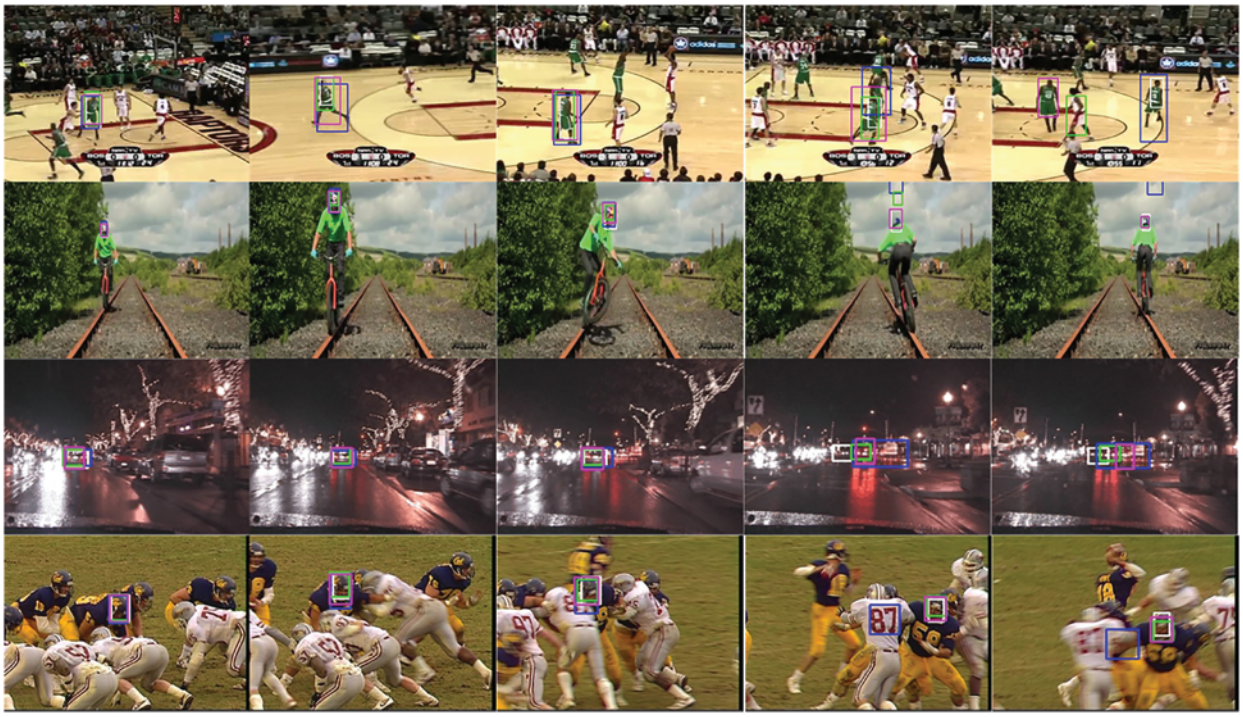


Figure 3: Algorithm tracking results display, the results of the oustracker output are more accurate in the event of a major change in the target

Finally, we tested it on UAV123 and compared it to the trackers of the Siamese architecture, and our model also reached the leading position. Results on UAV123 are as shown in Tab. 1.

Table 1: Results on UAV123, compared with the two indicators Area Under Curve (AUC) and distance precision (DP)

Trackers	AUC	DP
Siamfc	0.504	0.702
DaSiamRPN	0.604	0.801
SiamRPN++	0.578	0.769
SiamFC++	0.611	0.781
Ours	0.621	0.809

4.5 Feature Fusion Assessment

The main purpose of our research is mainly to discuss the fusion of shallow features and deep features. We have adopted the same deep feature maps for cross-correlation, and then performed cascading operations. Considering the particularity of the Siamese network, it has two network branches, and the branches use the same network structure and parameters, which makes the feature layers of two different branches in the same dimension more meaningful for fusion features. The feature layer of the same dimension is more conducive to feature extraction and search, and reduces the use of model parameters compared with the original feature fusion. As shown in the Figs. 4 and 5, taking a picture in OTB2015 as an example, although the information expressed by different depth feature maps are similar, they still have certain differences.

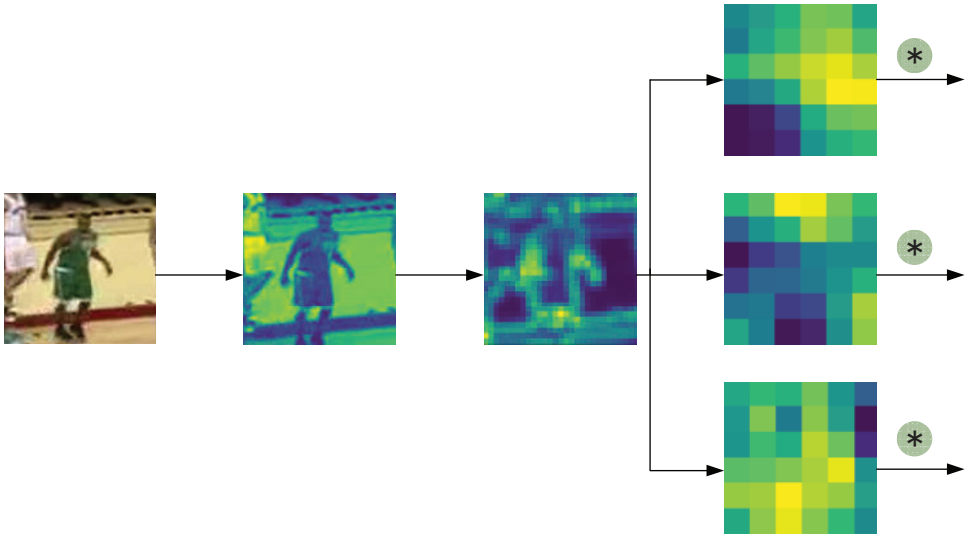


Figure 4: Visualization of our fusion features

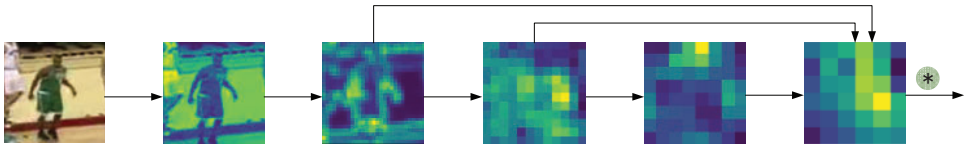


Figure 5: Visualization of old fusion features

In order to verify the accuracy of our ideas, we conducted a comparative experiment. The content of the experiment is to cascade first and then perform cross-correlation. The other structure remains unchanged. The feature map of the comparison model is as follows. The test set selects OTB2015, and the tracking result is not performed. There is little difference in tracking performance. This experiment verifies that the fusion module we designed is more effective than the previous direct fusion, and the tracking results are more accurate. Specific tracking results are as in Tab. 2.

Table 2: Results on OTB2015 and fusion comparison

Trackers	Success	Precision
SiamFC++	64.97	0.86
Cascading SiamFC++	64.42	0.84
Ours	68.20	0.90

5 Conclusions

In this paper, we proposed a new Siamese network and explored a new fusion module, which is very different from the previous model and proved its effectiveness, and more adapted to the structural characteristics of the two branches of the Siamese network. Finally, a better target state estimation criterion is proposed. Training and testing were performed on different challenging benchmarks to verify the generalization ability of the architecture. Because the modules we designed are simple and effective, there is still a lot of room for improvement.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] A. Bochkovskiy, C. -Y. Wang and H. -Y. M. J. Liao, "YOLOv4: Optimal speed and accuracy of object detection," ArXiv Preprint ArXiv:2004.10934, 2020.

[2] X. R. Zhang, X. Sun, W. Sun, T. Xu and P. P. Wang, "Deformation expression of soft tissue based on BP neural network," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 1041–1053, 2022.

[3] X. R. Zhang, J. Zhou, W. Sun and S. K. Jha, "A lightweight CNN based on transfer learning for COVID-19 diagnosis," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 1123–1137, 2022.

[4] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision*, The Netherlands, Amsterdam, vol. 13, pp. 850–865, 2016.

[5] B. Li, J. Yan, W. Wu, Z. Zhu and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 8971–8980, 2018.
[6] Y. Xu, Z. Wang, Z. Li, Y. Yuan and G. Yu, "SiamFC++: Towards robust and accurate visual tracking with target estimation guidelines," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 7, pp. 12549–12556, 2020.
[7] S. Ren, K. He, R. Girshick and J. J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, vol. 28, pp. 91–99, 2015.
[8] W. Sun, L. Dai, X. R. Zhang, P. S. Chang and X. Z. He, "RSOD: Real-time small object detection algorithm in UAV-based traffic monitoring," *Applied Intelligence*, vol. 42, pp. 1–16, 2021.
[9] S. Rajendar and V. K. Kaliappan, "Sensor data based anomaly detection in autonomous vehicles using modified convolutional neural network," *Intelligent Automation & Soft Computing*, vol. 32, no. 2, pp. 859–875, 2022.
[10] W. Sun, G. Z. Dai, X. R. Zhang, X. Z. He and X. Chen, "TBE-Net: A three-branch embedding network with part-aware ability and feature complementary learning for vehicle re-identification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 1–13, 2021.
[11] X. R. Zhang, X. Chen, W. Sun and X. Z. He, "Vehicle Re-identification model based on optimized densenet121 with joint loss," *Computers, Materials & Continua*, vol. 67, no. 3, pp. 3933–3948, 2021.
[12] W. Sun, X. Chen, X. R. Zhang, G. Z. Dai, P. S. Chang *et al.* "A Multi-feature learning model with enhanced local attention for vehicle re-identification," *Computers, Materials & Continua*, vol. 69, no. 3, pp. 3549–3561, 2021.
[13] X. R. Zhang, W. F. Zhang, W. Sun, X. M. Sun and S. K. Jha, "A robust 3-D medical watermarking based on wavelet transform for data protection," *Computer Systems Science & Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.
[14] D. S. Bolme, J. R. Beveridge, B. A. Draper and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, IEEE, pp. 2544–2550, 2010.
[15] Q. Guo, W. Feng, C. Zhou, R. Huang, L. Wan *et al.* "Learning dynamic siamese network for visual object tracking," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Venice, Italy, pp. 1763–1771, 2017.
[16] A. He, C. Luo, X. Tian and W. Zeng, "A twofold siamese network for real-time object tracking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 4834–4843, 2018.
[17] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proc. of the European Conf. on Computer Vision (ECCV)*, Munich, Germany, pp. 734–750, 2018.
[18] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 779–788, 2016.
[19] G. Shaozhe, L. Yong, Z. Youshan, L. Qiming, Y. Kaikai *et al.* "A asymmetric attention siamese network for visual object tracking," in *2021 2nd Int. Conf. on Big Data and Informatization Education (ICBDIE)*, Hangzhou, China, IEEE, pp. 163–168, 2021.
[20] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid *et al.* "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 658–666, 2019.
[21] L. Huang, X. Zhao, K. Huang and M. Intelligence, "Got-10k: A large high-diversity benchmark for generic object tracking in the wild," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 1562–1577, 2019.

[22] M. Danelljan, G. Bhat, F. Shahbaz Khan and M. Felsberg, "Eco: Efficient convolution operators for tracking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 6638–6646, 2017.

[23] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi and P. H. Torr, "End-to-end representation learning for correlation filter based tracking," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 2805–2813, 2017.