WEP Warwick Evans Publishing

International Journal of Computer Science and Information Technology

ISSN: 3005-9682 (Print), ISSN: 3005-7140 (Online) | Volume 2, Number 2, Year 2024
DOI: https://doi.org/10.62051/ijcsit.v2n2.14
Journal homepage: https://wepub.org/index.php/IJCSIT/index

A Siamfc Target-Tracking Algorithm Based on an Improved Spatiotemporal Attention Mechanism

Xu Zhang*, Jun Lu, Lin Shi, Yuan Cao

School of Software Engineering, Chengdu University of Information Technology, Chengdu 610000, China

*Corresponding Author: 1915162478@qq.com

ABSTRACT

The motion target tracking algorithm has developed rapidly. In this paper, the SiamFC (Fully Convolutional Siamese Networks) algorithm mainly relies on the first frame of the video as a template, and lacks an effective update mechanism. Based on the SiamFC algorithm, This paper introduced an improved spatiotemporal attention mechanism, and the model pays more attention to key historical frames and target regions in the video sequence by introducing an improved spatiotemporal attention mechanism in the backbone network. Moreover, the pixels in the response map are divided between background and foreground by a pixel-by-pixel classification regression method. By combining the centrality branch to limit the generation of lower quality prediction box, increase the accuracy of target prediction and reduce the complexity of prediction, the algorithm improves the accuracy and success rate, effectively realizing the target tracking in complex scenarios, while maintaining the accuracy and stability of tracking.

KEYWORDS

Target Tracking; Siamese Network; Space-Time Attention

1. INTRODUCTION

Although object detection[1] and object tracking[2] are closely related, they focus on different aspects when executing tasks. Object detection aims to identify objects of specific categories within an image and accurately determine their locations and confidence levels. In contrast, object tracking focuses on continuously tracking the position of one or more specific objects in a video sequence, requiring the integration of information such as the object's features, size, depth, and motion trajectory to maintain consistent tracking.

Recently, due to the efficient and fast performance of the target tracking algorithm[3]. With the introduction of various features such as directional gradient histogram[4], multi-channel calculation[5], deep convolution feature [6], color attribute[7], as well as the application of auxiliary strategies such as spatial information constraint[8], the development of tracking algorithm is greatly promoted and the tracking performance[9]is significantly improved. In recent years, with the continuous development of deep learning theory and the continuous improvement of visual tracking training datasets, deep learning-based visual tracking has also made significant progress.This paper is based on the deep learning related target tracking algorithm for improvement, in order to achieve better results.

2. TARGET TRACKING ALGORITHM

This study focuses on deep feature tracking of a single target within image sequences, that is, tracking a unique object across video frames. The range of targets is extensive, covering all kinds of objects. Therefore, the characteristics that target tracking research should possess include adaptability to various target categories and the ability to focus on tracking a single target. This paper posits that target tracking tasks should have the following features:

(1) Initial Frame Annotation: The target is usually annotated in the first frame of the target sequence to establish a model for identifying and tracking the target in subsequent frames. Annotation in the initial frame is essential to ensure that the model maintains continuity in target recognition throughout the tracking process.

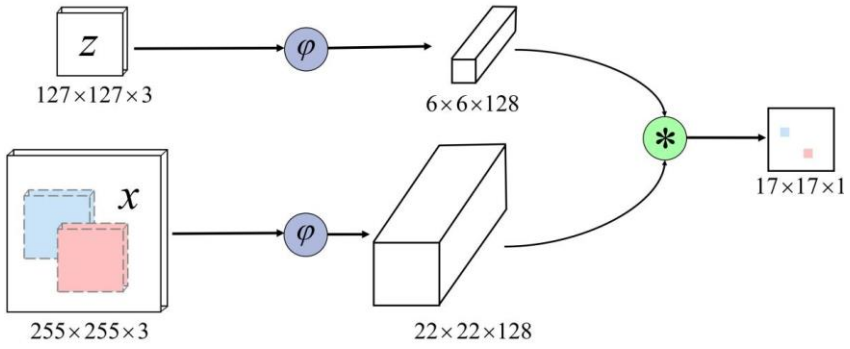
(2) Arbitrariness of Targets: The scope of objects for target tracking is very broad, including any type of object or organism. This requires the tracking model to be highly adaptable and generalizable in order to handle a variety of different tracking subjects.

(3) Specificity of a Single Target: During the target tracking process, the model focuses on tracking a single target without involving simultaneous tracking of multiple targets. Therefore, the model needs to have sufficient discriminative power to differentiate between the target and the background, ensuring the accuracy and effectiveness of tracking.

2.1. The Flow of the Target Tracking

The key to single-target tracking research lies in identifying and tracking specific targets from video sequences. It is divided into five critical steps: target initialization, motion model, feature extraction, observation model, and final prediction. First, target initialization involves calibrating the tracking target in the initial frame of the video, laying the groundwork for subsequent steps. Next, the motion model analyzes the target's motion trajectory and pattern across consecutive frames. Then, feature extraction involves extracting key image features from the tracking target to improve tracking accuracy. Following this, the observation model further enhances tracking precision by analyzing the interaction between the target and its environment. Finally, the final prediction integrates information from the previous steps to determine the optimal target location.

2.2. Siam FC Algorithm



1**Figure 1.** SiamFC network structure

Introduced in 2016, SiamFC, standing for Fully Convolutional Siamese Networks, utilizes a fully convolutional siamese network for similarity learning to achieve tracking of arbitrary targets. This structure receives two inputs, namely the reference template and the candidate samples. In the process of motion target tracking, the tracking target is treated as the reference template, typically chosen from the target in the first frame of the video sequence, and then candidate samples are searched for in the images of each subsequent frame. The core of the SiamFC training process is to train a metric

function to assess similarity. The task of this metric function is to compare the template images with the candidate images. When the template image and the candidate image show a high degree of similarity in describing the same target, the metric function assigns a higher score. Conversely, if the similarity in target description between the two is lower, the function awards a lower score. In this way, the SiamFC algorithm can effectively discriminate and track targets by determining the accuracy of tracking based on the similarity between images. The network structure of SiamFC is illustrated in Figure 1.

The structure initially sets z as the input template image, representing the target frame of the first frame of the video, with the dimensions of the target frame being (127×127×3). x denotes the search image inputted into the network, which has dimensions of (255×255×3). Both the target frame and search area image undergo a transformation process φ, with AlexNet serving as the underlying architecture for feature extraction. This step produces feature maps with dimensions of (6×6×128) for the template and (22×22×128) for the search image. Upon completing feature extraction, mutual correlation techniques (i.e., convolution operations) are employed to analyze the extracted features. The specific formula for the operation is expressed as Equation 1. Here, '$*$' represents the convolution operation, and $b\Pi$ is the value corresponding to each position. The convolution operation extracts the portion of the search image x that is most similar to the template image z Finally, a response map is generated.

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

3. SIAMFC ALGORITHMIC IMPROVEMENT

While the SiamFC algorithm has achieved notable results in some visual tracking challenges and engineering applications, it primarily relies on the first frame of the video as a template during the tracking process and lacks an effective update mechanism. To address these issues with the SiamFC algorithm, this section introduces two different modules on top of the SiamFC framework. The first module is an improved spatiotemporal attention module, and the second is a target estimation module. The first module uses a non-local operation neural network with the aim of accurately modeling the spatiotemporal information of the target during the tracking process, deeply analyzing the association between the current position of the target and historical frames, thereby enhancing the algorithm's ability to recognize targets. The second module classifies each pixel in the response map into foreground and background categories through a pixel-by-pixel classification method, calculating the distance from the prediction box to each pixel and utilizing a centrality branch to reduce the number of low-quality bounding boxes generated. This not only improves the precision of target positioning but also enhances accuracy.

3.1. Improved Spatiotemporal Attention Module

The attention mechanism[10] processes images in a way similar to how humans process visual information. When faced with a vast amount of visual data, people focus on certain special areas to obtain more details and ignore other irrelevant regions. By integrating an improved spatiotemporal attention module into the SiamFC algorithm, this section aims to more effectively capture key information and regions in video frames and their sequences, reducing the influence of background on dynamic target tracking. This optimization module is divided into two sub-modules: dynamic temporal attention and spatial attention, each responsible for capturing the correlations of temporal dimensions in the video sequence and spatial positions in single-frame images. The improved spatiotemporal attention module is designed to align the input-output feature dimensions, allowing it to be flexibly embedded in any convolutional layer within the SiamFC algorithm.

3.1.1. Dynamic Temporal Attention Module

Traditional Siamese network tracking algorithms often adopt a fixed attention allocation strategy for time-series data. This paper proposes a dynamic temporal attention mechanism that calculates a dynamic weight adjustment factor through an additional network branch, enabling real-time adjustment of attention weights.

The role of the dynamic temporal attention module in the algorithm is primarily to enhance the model's ability to understand time-series data. It assigns different weights to different time points, allowing the model to focus more on moments that are more important to the current task. This dynamic adjustment mechanism makes attention more flexible and adaptable to changes in the target's movement and appearance, leading to two main advantages: improved tracking accuracy for fast-moving or changing targets and increased model robustness against occlusion or similar interfering objects. These advantages are key factors in enhancing the overall performance of the tracking algorithm.

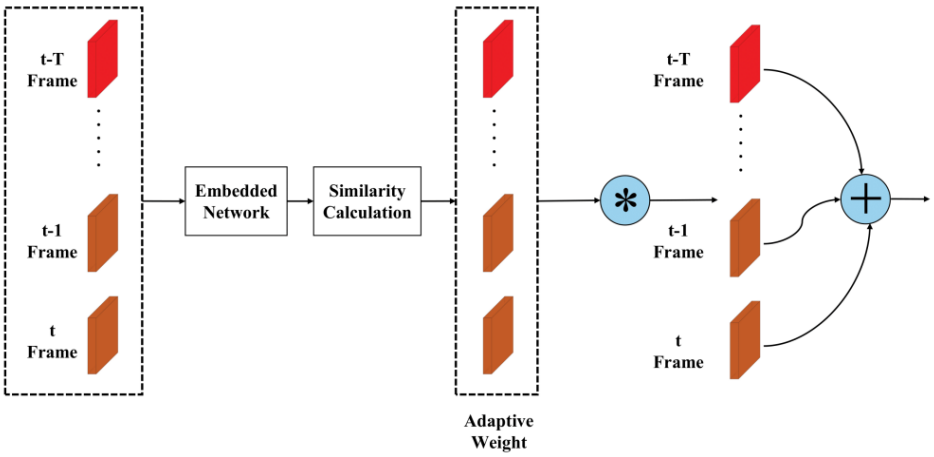


Figure 2. Time attention module diagram

As shown in Figure 2, the feature extraction network processes the front T frame of the target to determine its similarity to the first frame template. These features were then integrated into the embedded network to derive the temporal attention features $A(z)$. It is calculated by the Equation 2:(2)1

$$A(z) = \sum_{i=t-T}^{t-1} \alpha_i W_i \tag{2}1$$

In this model, α_i represents the attention weight of the time point i, and W_i is the feature vector corresponding to time point i. These features are scored by assessing their similarity to the initial frame template; if the similarity is high, the temporal attention module assigns greater weight. These weights are then normalized to ensure that their sum equals 1, with the calculation formula defined as Equation 3:(3)2

$$\alpha_i = \text{softmax}\left(\frac{W_i \bullet W_0}{|W_i| \bullet |W_0|}\right) \tag{3}2$$

This paper introduces a dynamic weight adjustment factor δ_i ,calculated from a small neural network $f(\cdot)$ .The input of the small network is the target state vector $\mathbf{S}_i$ and the historical state information, and the output is the weight adjustment factor for each time point.The network is designed to capture the dynamic changes of the target in the sequence, such as changes in size, appearance. The dynamic weight adjustment factor is calculated as shown in Equation 4:(4)3

$$\alpha_i^{'} = \delta_i \cdot \alpha_i \tag{4}3$$

In the formula $\delta_t = f\left(\mathbf{S}_i, \mathbf{S}_{i-1}, \ldots, \mathbf{S}_0; \Theta_f\right)$,and Θ_f represents the parameters representing the small network.ReLU is used as the activation function for the hidden layers, while the output layer employs a sigmoid function to ensure that δ_i is within a reasonable range. Additionally, to prevent the over-concentration of attention, a regularization term $R(\alpha)$ is introduced to smooth out the distribution of the weights.Expression, such as Equation 5:(5)4

$$\alpha_i^{''} = \frac{\alpha_i^{'} - R(\alpha)}{\sum_k \left(\alpha_k^{'} - R(\alpha)\right)} \tag{5}4$$

Here, $R(\alpha)$ is composed of a combination of L1 regularization, L2 regularization, and regularization for the smoothness of the time series, with the expression given as Equation 6:(6)5

$$R(\alpha) = \lambda_1 \| \alpha \|_1 + \lambda_2 \| \alpha \|_2^2 + \lambda_3 \sum \left(\alpha_i - \alpha_{i-1}\right)^2 \tag{6}5$$

λ_1、λ_2、λ_3 is a hyperparameter selected for its optimal value through cross-validation.

3.1.2. Spatial attention module

Traditional convolutional neural networks, due to the limitations of their convolutional kernels, can usually only process images through local sliding windows. Therefore, Wang et al.[11] proposed a non-local neural network method to capture global information more comprehensively. Building on this research, this section introduces a spatial module based on the non-local network concept to enhance the network model's acquisition of key target information in features. As shown in Figure 3, the spatial attention module utilizes the non-local neural network method to broaden the convolutional receptive field. This approach allows for the capture of interactions of object positions over a larger range, thus forming a feature map of spatial attention.

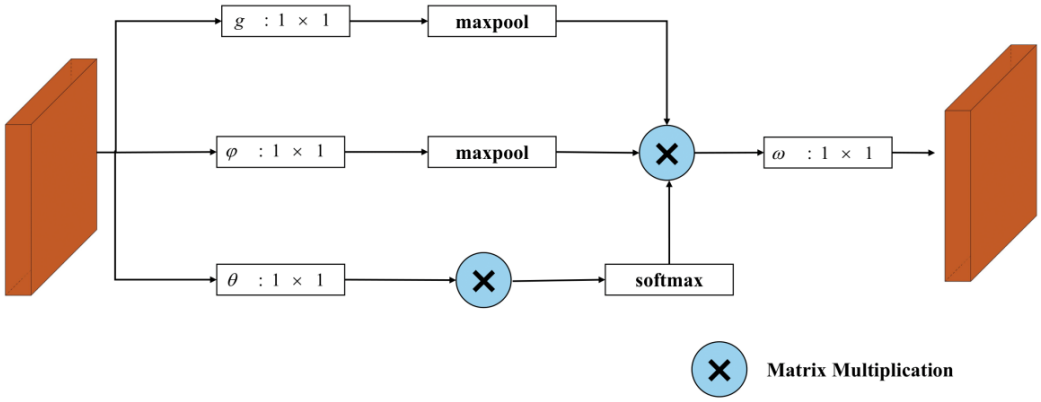


Figure 3. Spatial Attention Module Diagram

The spatial attention module input features are $z_{i=1,2,\ldots,T}^t$ and $z^s = \left[z_1^s, z_2^s, \ldots, z_T^s\right] \in R^{W \times H \times C}$, where W、H、C represent the length, width, and channel.The calculation formula for the spatial attention module is defined as Equation 7:

$$y_i^s = \frac{1}{C\left(Z^s\right)} \sum_{\forall j} f\left(z_i^s, z_j^s\right) g\left(z_j^s\right) \tag{7}6$$

Where, z_i^s and z_j^s represent the features of positions i and j, respectively, and the normalization factor

$C\left(Z^s\right)=\sum_{\forall j} f\left(z_i^s, z_j^s\right)$. In the function, $f(\cdot)$ indicates that the spatial attention module first adopts Gaussian functions, such as Equation 8:(8)7

$$f\left(z_i^s, z_j^s\right)=e^{\theta\left(z_i^s\right)^T \varphi\left(z_j^s\right)} \tag{8}$$

The equation is used to calculate the association between two different positions of the input feature and to perform a weighting processing for each position, where $\theta\left(z_i^s\right)=W_\theta^s z_i^s$, $\varphi\left(z_j^s\right)=W_\varphi^s z_j^s$,In a function, $g(\cdot)$ is a unary function, defined as Equation 9::

$$\mathrm{g}\left(z_j^s\right)=W_g^s z_j^s \tag{9}$$

In the formula, W_g^s is a learnable parameter. A maximum pooling layer of 2×2 is introduced after the functions $\varphi(\cdot)$ and $g(\cdot)$ to reduce the computation.

The function of the function ω is to ensure that the dimension of the weighted output is the same as the input, and the output dimension is $W \times H \times C$, which represents the spatial attention information o_i^s , and its calculation formula is Equation 10:(10)9

$$o_i^s=W_o^s y_i^s \tag{10}$$

3.2. Target Estimation Module

In the field of object detection, algorithms typically adopt the anchor box method to predict the bounding boxes of targets. This method has a clear issue: during the process of tracking moving objects, the algorithm generates a large number of candidate boxes, most of which are not utilized, leading to a waste of computational resources. To address this problem, this section proposes a new approach. A target estimation module has been constructed that classifies each pixel in the image to determine whether it belongs to the foreground (i.e., the target) or the background, aiming to increase the accuracy of predictions and reduce the rate of false positives. Subsequently, these pixel-level classification results are used to predict the bounding boxes of targets pixel by pixel, moving away from the traditional reliance on the anchor box method, thus employing an anchor-free mechanism. The anchor-free mechanism significantly reduces the number of unnecessary predicted bounding boxes, thereby improving the algorithm's computational efficiency. Furthermore, to address the issue of low-quality predicted bounding boxes, a centrality strategy has been introduced to reduce the low-quality bounding boxes generated by edge pixels.

错误!未找到引用源。The target estimation section designed in this section includes three main branches: regression, classification, and centrality. The Siamese network generates a response map, and after the response map is processed for dimensionality increase, feature $R^{a \times H_s \times W_s}$ is obtained. The regression branch produces a regression map $A_{reg} \in \mathbb{R}^{H_x \times W_x \times 4}$ by measuring the distance between the bounding box and each pixel, the classification branch predicts and classifies each pixel to produce a classification map $A_{cls} \in \mathbb{R}^{H_x \times W_x \times 2}$, and the centrality branch evaluates the distance between the target's center and the pixel to derive the centrality map $A_{cen} \in \mathbb{R}^{H_x \times W_x \times 1}$. Each pixel point (i, j) on the response map corresponds to the search area (x, y) relative to the input branch.

3.2.1. Regression Branch

For each point $(i, j,:)$ of the regression graph A_{reg} contains a 4D vector, as shown in Figures 4, the upper left vertex coordinate of the defined target true region is $\left(x_0, y_0\right)$ and the lower right vertex

coordinate is (x_1, y_1), $m^*_{(i,j)} = (l^*, t^*, r^*, b^*)$ represents four distances from the boundary of the real area of the target, defined as follows:

$$m^{*0}_{(i,j)} = l^* = x - x_0, m^{*1}_{(i,j)} = t^* = y - y_0$$
$$m^{*2}_{(i,j)} = r^* = x_1 - x, m^{*3}_{(i,j)} = b^* = y_1 - y \quad (11)10$$

Similarly, $m_{(i,j)} = (l\ , t\ , r\ , b\ )$ represents the four distances of the point (x, y) to the boundary of the predicted region. The regression loss function L_{reg} uses the IOU(Intersection over Union) loss function, which is defined as Equation 12:(12)11

$$L_{reg} = \frac{1}{\sum T\left(m^*_{(i,j)}\right)} \sum_{(i,j)} T\left(m^*_{(i,j)}\right) L_{IOU}\left(m(i,j), m^*(i,j)\right) \quad (12)11$$

Where the indicator function is defined such as Equation 13:(13)12

$$\Gamma\left(m^*_{(i,j)}\right) = \begin{cases} 1 & \text{if } m^{*k}_{(i,j)} > 0, k = 0,1,2,3 \\ 0 & \text{otherwise} \end{cases} \quad (13)12$$

This indicator function is used to check whether the m^{*k} value of each index (i, j) position is greater than 0, corresponding to k is an index of the four parameters of the bounding box.If m^{*k} is greater than 0 for any k value, the value of $\Gamma\left(m^*_{(i,j)}\right)$ is 1, indicating that $m^{*k}_{(i,j)}$ is positive, meaning that the difference between the position parameter of the predicted bounding box and the true value is positive, which is meaningful in the bounding box regression.

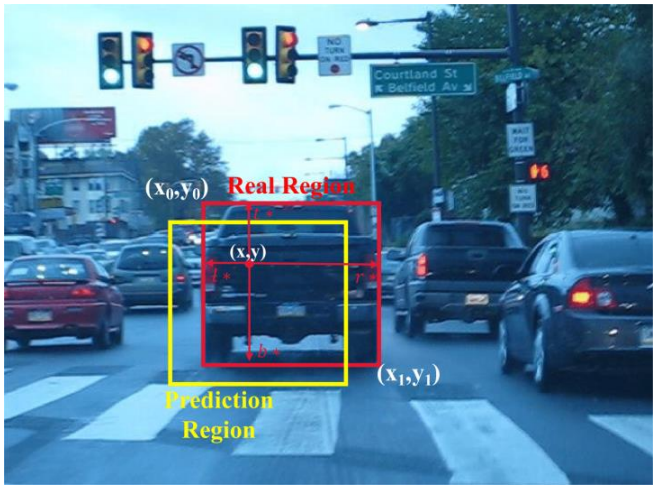


Figure 4. Regression branch prediction results

3.2.2. Classification Branch

Each pixel $(i, j,:)$ in the classification map A_{cls} contains a two-dimensional vector that represents the confidence levels of the foreground and background in the search area. Experiments have shown that within the target area, sampling points that are farther from the target center tend to have poorer prediction outcomes, while those closer to the center yield better results. To improve the accuracy of the algorithm, it is necessary to constrain the classification scores of the edge points. Therefore, this design incorporates centrality within the classification branch to reinforce the importance of central points and mitigate the impact of edge points, thereby more effectively filtering out anomalous data.After removing the abnormal data, the centrality branch feature map A_{cen} is generated, and each point $(i, j,:)$ on the feature map A_{cen} corresponds to a one-dimensional vector $C(i, j)$, representing

the distance between the target center and the corresponding pixel point in the search area. It is as defined as Equation 14:(14)13

$$C(i,j)=\Gamma\left(m^{*}_{(i,j)}\right)\sqrt{\frac{\min\left(l^{*},r^{*}\right)}{\max\left(l^{*},r^{*}\right)}\times\frac{\min\left(t^{*},b^{*}\right)}{\max\left(t^{*},b^{*}\right)}} \tag{14}13$$

Where, $\min(\cdot)$ represents the minimum value, and $\max(\cdot)$ represents the maximum value. If point (x,y) falls in the background area, the value of $C(i,j)$ is 0; if the point is far from the target center, the centering score is low; if point (x,y) is close to the target center, the centrality score corresponding to the point is high.

This section employs a joint loss function for the training of the algorithm model. It encompasses three loss functions: regression loss, classification loss, and centrality loss. The regression loss function L_{reg} uses the IOU loss function, the classification loss function L_{cls} utilizes the cross-entropy loss function, and the centrality loss function is defined as follows:

$$L_{cen}=\frac{-1}{\sum\Gamma\left(m_{(i,j)}\right)}\sum\Gamma\left(m_{(i,j)}\right)=1^{C(i,j)*\log\left(A^{cen}_{w\times h\times 1}(i,j)\right)+\left(1-C(i,j)*\log\left(1-A^{cen}_{w\times h\times 1}(i,j)\right)\right)} \tag{15}14$$

The algorithm loss function in this section is defined as Equation 16:

$$L=L_{cls}+\lambda_1 L_{reg}+\lambda_2 L_{cen} \tag{16}15$$

Where λ_1 and λ_2 are the weight coefficients used to balance the different loss terms.

3.3. The Overall Framework of the Algorithm

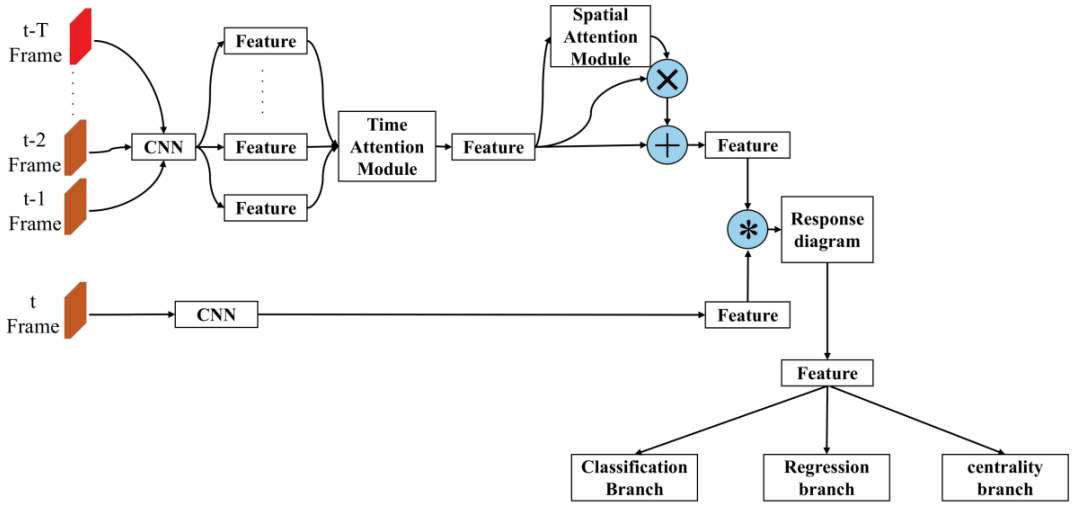


Figure 5. Algorithm model network frame diagram

Figure 5 illustrates the overall structure of the motion target tracking algorithm proposed in this section, utilizing SiamFC as the underlying framework and adopting AlexNet as the backbone network. This Siamese network is composed of two branches: the search branch and the template branch, which share parameters during training. The improved spatiotemporal attention module is designed to maintain consistent input-output feature dimensions, enabling it to be flexibly embedded into any convolutional layer of the SiamFC algorithm. In this section, the improved spatiotemporal attention module is embedded in the middle layer of the AlexNet network to capture the dynamic changes of the target across temporal and spatial dimensions, enhancing the network's ability to integrate time and space features. Target estimation is conducted by classifying each pixel in the

response map into foreground and background; each pixel in the foreground is treated as a relative bounding box, and bounding box regression processing is performed. By combining with the centrality branch, the generation of low-quality or invalid bounding boxes is reduced, decreasing the computational power required by the algorithm. This process improves the accuracy of positioning, thereby enhancing the tracking performance.

3.4. Experiment and Analysis

The algorithm experiments in this chapter were conducted under the Windows 10 operating system, utilizing an Intel Core i5 3.4GHz CPU and 16GB RAM. The graphics card used was an NVIDIA GeForce GTX 1060Ti. The experiments were written in Python and employed the Pytorch deep learning framework.

The training of the Siamese network target tracking algorithm with an improved spatiotemporal attention mechanism (referred to as MySiam in this paper) utilized Stochastic Gradient Descent (SGD) with a momentum of 0.01 as the optimization strategy. The weight decay was set to 0.01, the initial learning rate was set to 0.01, the learning rate decay factor was 0.1, the number of epochs was set to 50, and each batch size was 12.

Table 1. OTB100 Experiment results

Algorithm name	Precision	Success
MySiam	0.826	0.615
SiamFC	0.772	0.587

The evaluation of the algorithm's improvement also included tests on the OTB100 dataset, testing both MySiam and SiamFC. The OTB100 dataset testing used the One-Pass Evaluation (OPE) method, employing success rate and precision metrics defined in the literature for measuring algorithm performance. Corresponding performance curves were also displayed. In these curves, the threshold for center location error was set between 0 to 50, while the threshold for target overlap ratio was set between 0 to 1.

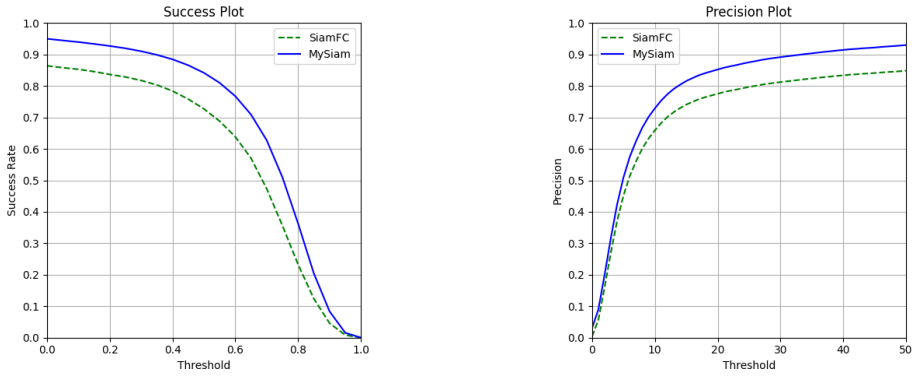


Figure 6. Algorithm performance graph

As shown in Table 1, this section conducts a performance evaluation using the OTB100 dataset for four algorithms, utilizing Precision and Success as metrics to assess the algorithms' performance. Precision is used to evaluate the accuracy of the algorithms, while Success is used to assess their robustness. In calculating the Precision metric, it is necessary to calculate the error between the center of the algorithm's predicted target bounding box and the center of the manually annotated target bounding box in each frame, which is the Euclidean distance between the center of the annotated target box and the center of the predicted target box. When calculating the Success metric of the algorithm, it is required to calculate the overlap ratio between the manually annotated target bounding box and the algorithm's predicted target bounding box in each frame.

Figure 6 shows the performance graph of the improved algorithm MySiam and the benchmark algorithm SiamFC tested on the OTB100 dataset. As can be seen from the data in the figure, the improved Siamese network algorithm in this chapter outperforms the benchmark algorithm, which can be seen that the algorithm in this chapter has good tracking performance.

In general, the score of MySiam on Precision index is 0.826, and that of Success index is 0.615. Compared with SiamFC, the score of Precision is improved after the integration of improved spatiotemporal attention module, from 0.772 to 0.826, indicating an improvement in the accuracy of the algorithm. In terms of Success score of the algorithm, MySiam also improved significantly with SiamFC, from 0.587 to 0.615.

4. CONCLUSION

This paper makes improvements to SiamFC. For the SiamFC Siamese network algorithm, the algorithm is enhanced by using an improved spatiotemporal attention module and a target estimation module. Through the dynamic temporal attention module, the model learns and adaptively adjusts the similarity weights of multiple historical frames and template frames, thereby constructing a matching template with temporal series associations to utilize the information of historical frames. The role of the spatial attention mechanism is to increase the algorithm's perception of global information, aiming to improve the precision of motion target localization. In the target estimation module, each pixel in the response map is subjected to both regression and classification, and a centrality branch is introduced into the target estimation module. The introduction of this module can reduce the difficulty of predicting the position of moving targets.

REFERENCES

[1] Zhao Z Q, Zheng P, Xu S, et al. Object detection with deep learning: A review[J]. IEEE transactions on neural networks and learning systems, 2019, 30(11): 3212-3232.

[2] Yilmaz A, Javed O, Shah M. Object tracking: A survey[J]. Acm computing surveys (CSUR), 2006, 38(4): 13-es.

[3] Han S, Huang P, Wang H, et al. Mat: Motion-aware multi-object tracking[J]. Neurocomputing, 2022, 476: 75-86.

[4] Petsiuk A, Pearce J M. Towards smart monitored AM: Open source in-situ layer-wise 3D printing image anomaly detection using histograms of oriented gradients and a physics-based rendering engine[J]. Additive Manufacturing, 2022, 52: 102690.

[5] Alvarado-Robles G, Osornio-Rios R A, Solis-Munoz F J, et al. An approach for shadow detection in aerial images based on multi-channel statistics[J]. IEEE Access, 2021, 9: 34240-34250.

[6] Ali R, Chuah J H, Talip M S A, et al. Structural crack detection using deep convolutional neural networks[J]. Automation in Construction, 2022, 133: 103989.

[7] Vijayan T, Sangeetha M, Kumaravel A, et al. Feature selection for simple color histogram filter based on retinal fundus images for diabetic retinopathy recognition[J]. IETE Journal of Research, 2023, 69(2): 987-994.

[8] Cooper L A. Spatial information processing: Strategies for research[M]//Aptitude, learning, and instruction. Routledge, 2021: 149-176.

[9] Szeliski R. Computer vision: algorithms and applications[M]. Springer Nature, 2022.

[10] Niu Z, Zhong G, Yu H. A review on the attention mechanism of deep learning[J]. Neurocomputing, 2021, 452: 48-62.

[11] Wang X, Girshick R, Gupta A, et al. Non-local neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7794-7803.