Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH

Open Access

A cloud-oriented siamese network object tracking algorithm with attention network and adaptive loss function

Jinping Sun[1*] and Dan Li[1*]

Abstract

Aiming at solving the problems of low success rate and weak robustness of object tracking algorithms based on siamese network in complex scenes with occlusion, deformation, and rotation, a siamese network object tracking algorithm with attention network and adaptive loss function (SiamANAL) is proposed. Firstly, the multi-layer feature fusion module for template branch (MFFMT) and the multi-layer feature fusion module for search branch (MFFMS) are designed. The modified convolutional neural networks (CNN) are used for feature extraction through the fusion module to solve the problem of features loss caused by too deep network. Secondly, an attention network is introduced into the SiamANAL algorithm to calculate the attention of template map features and search map features, which enhances the features of object region, reduces the interference of background region, and improves the accuracy of the algorithm. Finally, an adaptive loss function combined with pairwise Gaussian loss function and cross entropy loss function is designed to increase inter-class separation and intra-class compactness of classification branches and improve the accuracy rate of classification and the success rate of regression. The effectiveness of the proposed algorithm is verified by comparing it with other popular algorithms on two popular benchmarks, the visual object tracking 2018 (VOT2018) and the object tracking benchmark 100 (OTB100). Extensive experiments demonstrate that the proposed tracker achieves competitive performance against state-of-the-art trackers. The success rate and precision rate of the proposed algorithm SiamANAL on OTB100 are 0.709 and 0.883, respectively. With the help of cloud computing services and data storage, the processing performance of the proposed algorithm can be further improved.

Keywords Attention network, Adaptive loss function, Siamese network, Object tracking, CNN

Introduction

The task of object tracking [1, 2] is to stably locate the object to be tracked from subsequent frames when the size and location information of the object in the first frame of the video sequence is given. To make up the limited computing resources and storage resources of a single computer, video sequences can be deployed in the cloud, and cloud computing technology can be used to further improve the tracking performance. Object tracking is currently applied to various fields of artificial intelligence such as intelligent monitoring based on edge-cloud computing [3], human-computer interaction based on vision [4, 5], intelligent transportation, and autonomous driving [6].

Object tracking algorithms are mainly divided into two types, namely, generative model and discriminative model. Generative models, such as optical flow methods [7, 8] and mean shift algorithm [9–11], are difficult to resist scale changes, deformation, and similar interference. The mainstream discriminative object tracking

*Correspondence:
Jinping Sun
sjp@xzit.edu.cn
Dan Li
lidanonline@xzit.edu.cn
[1] School of Information Engineering (School of Big Data), Xuzhou University of Technology, Xuzhou 221018, China

algorithms are mainly divided into the correlation filter algorithm and the depth learning algorithm. The correlation filter algorithm aims to learn a filter with high response to the object center and low response to the surrounding background through mathematical modeling. Among the object tracking algorithms based on the correlation filter, MOSSE (Minimun output sum of square error, MOSSE) [12], CSK (Circulant structure of tracking-by-detection with kernels, CSK) [13, 14], KCF (Kernel correlation filter, KCF) [15], and DSST (Discriminative scale space tracker, DSST) [16] are the most representative algorithms. KCF introduces Gaussian kernel function based on CSK, uses ridge regression method to train filter template, and simplifies calculation in the form of circular matrix, which significantly improves the operation speed.

By using off-line training, the tracking algorithm [17–19] based on convolutional neural networks (CNN) can learn the common feature model which represents the robustness of the object, and dynamically update the coefficient of the classifier through online learning to improve the tracking performance. However, it will involve the adjustment and update of huge network parameters in the tracking process, which will consume large amount of calculation time and cannot fully meet industrial standards in the term of real-time performance.

In recent years, object tracking methods based on the siamese network have received significant attention at home and abroad for their strong accuracy and excellent processing speed. The object tracking algorithms based on the correlation filter perform well in real-time performance, but their accuracy is difficult to improve due to the extracted single feature attribute. The existing object tracking algorithms based on the siamese neural network achieve high accuracy, but they have network complexity, limited operation speed and poor real-time performance. Aiming at solving the above mentioned problems, a siamese network object tracking algorithm with attention network and adaptive loss function (SiamANAL) is proposed in this paper. The main contributions of this paper are as follows:

- A multi-layer feature fusion module using modified ResNet50 network is proposed, which fuses the hierarchical features in the last three layers of the ResNet50 network to avoid missing important features in the process of feature extraction.
- Aiming at solving the limited accuracy of tracking algorithm, an attention network is introduced to encode self-attention and cross-attention of feature maps. The features of the elements with rich semantic information of the object are enlarged, while those of irrelevant elements are reduced, and the generalization ability of search map features is improved.
- In order to improve the accuracy of object classification, the cross entropy loss function and pairwise Gaussian loss function are proposed in the classification branch to increase inter-class separation and intra-class compactness.
- By comparing the proposed SiamANAL algorithm with other trackers on the existing mainstream object tracking datasets, it is verified that the accuracy and robustness of the proposed algorithm have been significantly improved.

The remainder of this study is organized as follows. Related work section summarizes and discusses existing methods of object tracking based on siamese network. The proposed SiamANAL algorithm section describes the overall framework of the proposed algorithm, and constructs feature extraction network, self-attention network and cross-attention network, as well as classification-regression subnetwork. Result analysis and discussion section verifies the tracking effect of the proposed algorithm in different datasets, and carries out quantitative and qualitative analysis and discussion with comparative algorithms. Finally, conclusion section summarizes the conclusions.

Related work

A typical siamese network consists of two branches with shared parameters, namely, a template branch representing the object features and a search branch representing the current search area. The template is usually obtained from the label box of the first frame in the video sequence, marked as Z, and the search area of each subsequent frame is marked as X. The siamese network takes two branches Z and X as the inputs, and uses an off-line trained backbone network φ with shared weights to take the characteristics of the two branches. The parameter of the backbone network is θ. By convolving the features of the template branch and the search branch, the tracking response map of the current frame can be obtained. The value on the response map represents the score of the object at each position. The response map is calculated as follows:

$$f_\theta(Z, X) = \varphi_\theta(Z) * \varphi_\theta(X) + b, \tag{1}$$

where b represents the deviation term of simulated similarity deviation. In Eq. (1), the template Z performs exhaustive search on the image X to obtain the similarity score of each position.

Generally speaking, the siamese network trains (Z, X) and the corresponding real label y offline by collecting many images from the training video. The backbone

network parameter θ is continuously optimized during the training process. To match the maximum value in response map $f_\theta(Z, X)$ with the object position, the loss ↕ is usually minimized in the training set, that is:

$$\operatorname{argmin}\left(\mathrm{y}, f_\theta(Z, X)\right). \tag{2}$$

Based on the above mentioned theories, the tracking algorithms based on the siamese network have been modified to improve the tracking performance. The SiamFC (Full coherent siamese networks for object tracking, SiamFC) algorithm [20] firstly proposes the concept of siamese structure, which has two inputs: one is the benchmark template for manually labeling the object in the first frame, and the other is the search candidate area for all the other frames in the tracking process. The purpose of the siamese structure is to find the area that is most similar to the reference template of the first frame in each frame. The design and optimization of the loss function play a key role in the tracking effect. The SiamRPN (High performance visual tracking with Siamese region proposal network, SiamRPN) algorithm [21] introduces the region proposal network (RPN) based on SiamFC. The RPN sends the features extracted by the siamese neural network into the classification branch and regression branch, and uses the predefined anchor boxes as the reference for the regression value of the boundary box. The speed and accuracy of tracking algorithm are significantly improved. Guo et al. [22] propose a fast universal transformation learning model, which can effectively learn changes in the appearance of the object and suppress the background, but online learning has lost the real-time ability of the model. Wang et al. [23] explore the effects of different types of attention mechanisms on template map features in the SiamFC method, including general attention, residual attention, and channel attention. However, this algorithm does not explore the attention network on search map features. He et al. [24] propose double feature branches, namely, semantic branch and appearance branch, which effectively improve the generalization of the algorithm. However, these two branches are trained separately and only combined during reasoning, thus are lack of the coupling. The SiamRPN++ (SiamRPN++: Evolution of Siamese Visual Tracking with Very Deep Networks, SiamRPN++) algorithm [25] based on SiamRPN uses modified ResNet50 [19] network for feature extraction, and also achieves good results. Other classical algorithms based on siamese networks include the Siam R-CNN (Siam R-CNN: visual tracking by re-detection, Siam R-CNN) algorithm [26] and the SiamCAR (SiamCAR: siamese fully convolutional classification and regression for visual tracking, SiamCAR) algorithm [27], and they both achieve significant tracking effect. Chen et al. [28] propose a siamese network tracking algorithm based on SiamRPN++ algorithm for online object classification to enhance the context information of the object and improve the robustness of the algorithm. Tan et al. [29] design a full convolution siamese tracker without anchor frame, which can directly classify and predict on pixels to improve the robustness of the tracker.

Although the tracking model based on the siamese network improves the tracking performance while ensuring the real-time tracking, the model based on offline training is difficult to effectively distinguish the tracking object from the background information under dim ambient light. How to reduce tracking drift or tracking failure and improve the tracking success rate and robustness is still the key research content when the object is occluded, deforms and faces other interference.

Proposed SiamANAL algorithm

The overall framework of the proposed SiamANAL algorithm is shown in Fig. 1, which is divided into four parts: the feature extraction of siamese network, the self-attention network, the cross-attention network, and the classification-regression subnetwork. The main processing flow is divided into the following four parts:

1. Feature extraction of siamese network (see 'Proposed siamese network feature extraction module' section for details). The template map Z and the search map X are the input of the feature extraction module, which are injected into the modified ResNet50 network through weight sharing. The hierarchical features in the last three layers of the template map Z are fused through the multi-layer feature fusion module of the template branch (MFFMT), and the template map features are output in the form of $f(Z)$. The hierarchical features of the last three layers of the search map X are fused through the multi-layer feature fusion module of the search branch (MFFMS), and the search map features are output in the form of $f(X)$.
2. Self-attention network (see 'Self-attention calculation' section for details). The self-attention network includes the template map attention network and the search map attention network. $f(Z)$ and $f(X)$ are used as the input matrix of the self-attention network to calculate their self-attention features, and their self-attention outputs $f^*(Z)$ and $f^*(X)$ are obtained respectively.
3. Cross-attention network (see 'Cross-attention calculation' section for details). The search map feature $f^*(X)$ is used as the input matrix and the template

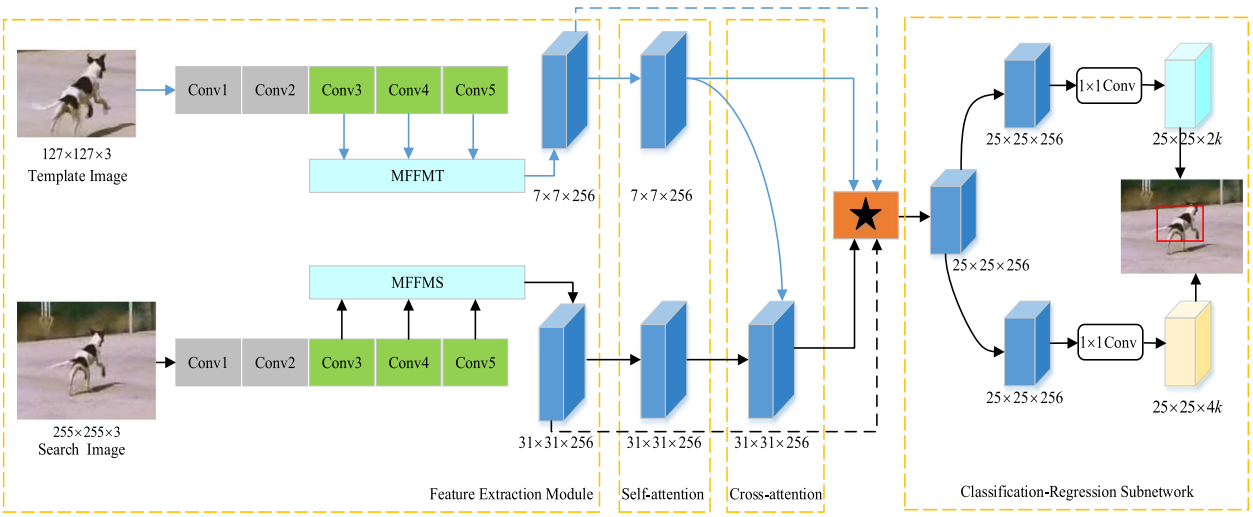


Fig. 1 Algorithm framework

map feature $f^*(Z)$ is used as the coding matrix, which are input into the cross-attention network to obtain the cross-attention output $f^{**}(X)$.

4. Classification-regression subnetwork (see 'Classification and regression subnetwork' section for details). As the inputs of the classification-regression subnetwork, $f^*(Z)$ and $f^{**}(X)$ are performed the deep cross-correlation operation. The pairwise Gaussian loss function and the cross entropy loss function are designed to achieve classification and regression results of boundary boxes.

Proposed feature extraction module

Low-level features of CNN, such as edge, color and shape, provide rich position information, and can deal with the tracking problems in scenes such as illumination changes, but they are not robust to appearance deformation. High-level features are better to represent rich semantic features and have strong robustness to significant changes in the appearance of the object. However, the spatial resolution is too low to achieve accurate object localization. The object tracking effect can be improved by making full use of the different resolution of low-level features and high-level features. Many methods take advantage of fusing both low-level features and high-level features to improve the tracking accuracy. Considering the above mentioned factors, the last three layers of convolutional network with both location features and semantic features are selected to represent the object.

The input of the template branch is the template map Z withe the size $27 \times 127 \times 3$ and the input of the search branch is the search map X with size $255 \times 255 \times 3$. The template map feature $f(Z)$ and the search map feature $f(X)$ are output by the modified weight sharing ResNet50 network respectively. As the deep learning network becomes deeper and deeper, the extracted features become more and more abstract. To avoid the loss of some useful features due to the deep network, a multi-layer feature fusion module is proposed, including MFFMT and MFFMS for the template branch and the search branch respectively.

MFFMT

As shown in Fig. 2, MFFMT represents multi-feature extraction of the template branch.

Step 1: Compress the hierarchical features in the last three layers of the template map Z to keep the number of channels consistent.

Step 2: Compress the hierarchical features in the last three layers by using a convolution kernel of size 1×1 to keep the number of channels consistent that is 256.

Step 3: To reduce the amount of calculation in the template branch, the hierarchical features in the last three layers are clipped in the center so that the size of the feature map is kept as $7 \times 7 \times 256$.

Step 4: Concat these three feature maps together to obtain a feature map with 3×256 channel number and 7×7 size.

Step 5: The ConTranspose2d operation is used to obtain the feature map $f(Z)$ with the size of $7 \times 7 \times 256$, which contains all useful information of the last three layers of the ResNet50 network structure.

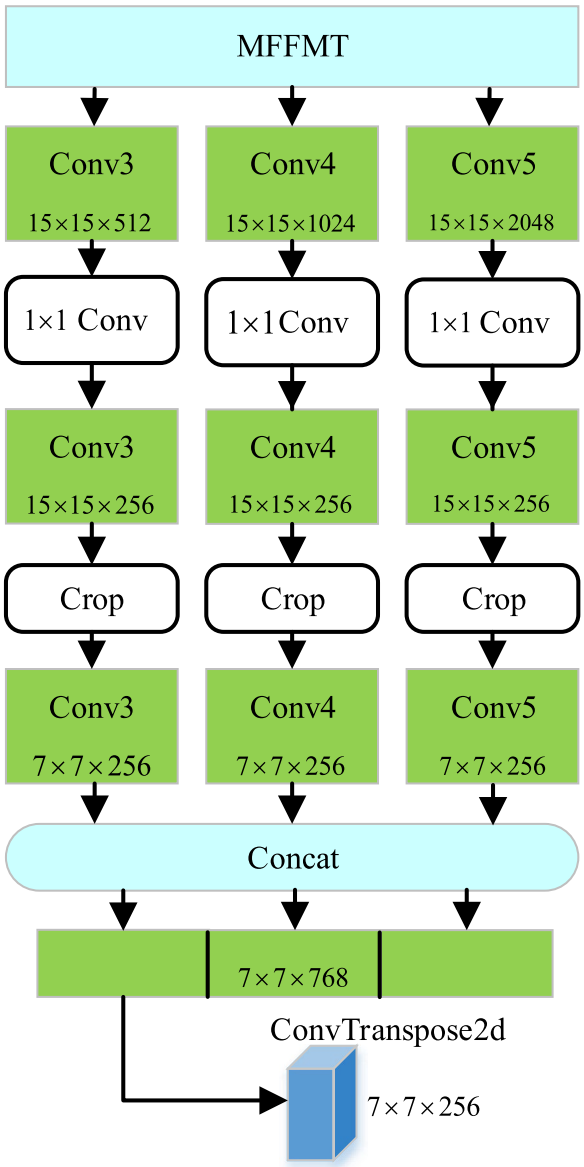


Fig. 2 Feature extraction of MFFMT module

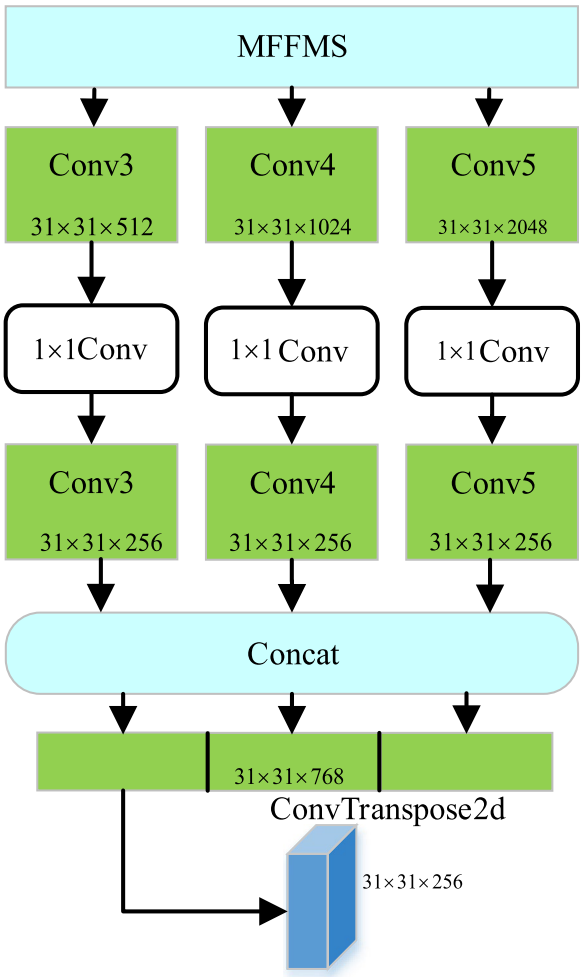


Fig. 3 Feature extraction of MFFMS module

MFFMS

As shown in Fig. 3, MFFMS represents multi-feature extraction of the search branch.

Step 1: Compress the hierarchical features in the last three layers of the search map X to keep the number of channels consistent.

Step 2: Compress the hierarchical features in the last three layers by using a convolution kernel of size 1×1 to keep the number of channels consistent that is 256.

Step 3: Concat these three feature maps together to obtain a feature map with 3×256 channel number and 31×31 size.

Step 4: The ConTranspose2d operation is used to obtain the feature map $f(X)$ with the size of $31 \times 31 \times 256$, which contains all useful information of the last three layers of the ResNet50 network structure.

Proposed attention network model

The most effective way to improve the accuracy of the algorithm is to improve the expression ability of the feature matrix, and attention network can further improve the expression ability of backbone features. The model scheme of attention network in this paper mainly includes self-attention and cross-attention. Self-attention can encode the correlation between the feature elements and the channel, which can help better highlight the

feature elements that are useful for tracking in the object tracking task. Cross-attention can encode the element correlation between two different features, which acts on both search map features and template map features in object tracking task. It is more beneficial to improve the accuracy of cross correlation results to let the feature elements of the search map execute a weight allocation in advance according to the influence of the feature elements of the template map.

Considering the impact on real-time performance, the lightweight attention network introduced in this paper is Non-local [30], which will have minimal impact on the number of parameters and floating point arithmetic, and can effectively improve the expression of backbone features. Non-local is a kind of non-local network operation, which is the opposite of local operations such as convolution and cyclic operation. The long-distance dependency of each element in the input features is captured, which is an extremely informative dependency. The structure diagram is shown in Fig. 4.

The inputs of the Non-local network are two matrices $A \in H_1 \times W_1 \times D$ and $B \in H_2 \times W_2 \times D$ respectively. H_i and W_i represent the height and width of the matrix respectively, and D represents the number of channels of the matrix. After matrix A is input, the residual matrix is calculated with matrix B. The inputs of residual matrix operation are *query*, *key*, and *value*, where matrix *query* is assigned by matrix A, and matrices *key* and *value* are assigned by matrix B. We Perform the convolution kernel operation of $1 \times 1 \times D$ on the input matrix, and then perform the matrix dimension transformation. After two matrix multiplication operations and the final $1 \times 1 \times D$ convolution operation, the output is the residual matrix A^*. The final output $\widehat{A}$ is obtained by adding the residual matrix A^* and the original matrix A. To simplify the expression, the convolution kernel encoding of $1 \times 1 \times D$ is represented by function *Conv*. The expressions of each operation step are as follows:

$$A_{query} = Conv(A)_M, \tag{3}$$

$$B_{key} = Conv(B)_M^T, \tag{4}$$

$$B_{value} = Conv(B)_M, \tag{5}$$

$$A_{query+key} = soft\max\left(A_{query} \bullet B_{key}\right), \tag{6}$$

$$A_{query+key+value} = A_{query+key} \bullet B_{value}, \tag{7}$$

$$A^* = Conv\left(A_{query+key+value}\right), \tag{8}$$

$$\widehat{A} = A \oplus A^*. \tag{9}$$

"$\cdot$" represents the matrix multiplication operation, "$\oplus$" represents the addition of the matrix element by element, "T" represents the transpose operation of the matrix, and "$(\cdot)_M$" represents the first and second two-dimensional combinations of the matrix.

The input dimensions of matrix multiplication in Eq. (6) are $A_{query} \in (H_1 \times W_1) \times D$ and $B_{key} \in (H_2 \times W_2) \times D$, and the output dimension is $A_{query+key} \in (H_1 \times W_1) \times (H_2 \times W_2)$, which means that the elements in the space dimension of matrix A and the elements in the space dimension of matrix B carry out attention correlation operation one by one. The input dimensions of matrix multiplication in Eq. (7) are $A_{query+key} \in (H_1 \times W_1) \times (H_2 \times W_2)$ and $B_{value} \in (H_2 \times W_2) \times D$, and the output dimension is $A_{query+key+value} \in (H_1 \times W_1) \times D$. The residual matrix A^* is output through the 1×1 Conv of Eq. (8), which injects the attention influence coefficient of matrix B on matrix A in the dimensions H_1, W_1 and D. By adding the attention influence coefficient matrix A^* and A one by one through Eq. (9), the final output result of the Non-local attention network can be obtained.

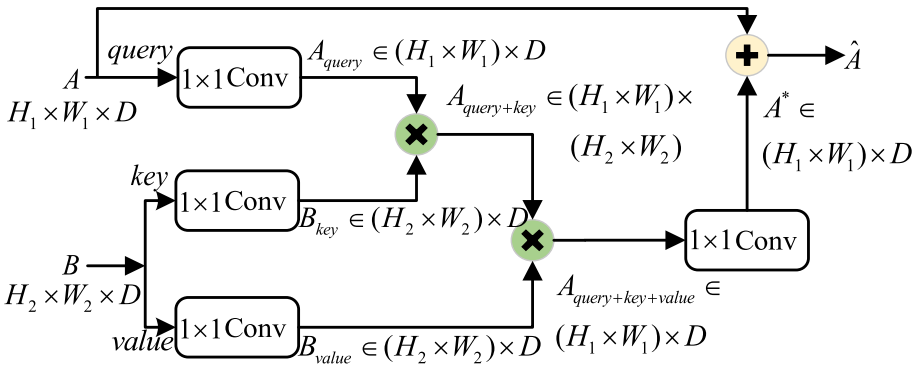


Fig. 4 Non-local network structure diagram

Self-attention calculation

The self-attention-non-local (SANL) network designed in this paper takes the feature matrix itself as attention correlation, that is, uses search map features and template map features as the input of *query*, *key* and *value* matrices. For the template map attention network, the template map feature $f(Z)$ is input into the Non-local attention network as the input matrix, that is, $A = f(Z)$ and $B = f(Z)$, and they are substituted into the Non-local attention network model to obtain the self-attention output as follows:

$$f^*(Z) = f(Z) \oplus \left(soft\max\left(Conv\left(f(Z)\right)_M\right.\right. \cdot Conv\left(f(Z)\right)_M^T \cdot \left.\left. Conv\left(f(Z)\right)_M\right)\right). \tag{10}$$

$f^*(Z)$ is the template map feature using SANL attention coding.

For the search map attention network, the search map feature $f(X)$ is input into the Non-local attention network as the input matrix, that is, $A = f(X)$ and $B = f(X)$, and they are substituted into the Non-local attention network model to obtain the self-attention output as follows:

$$f^*(X) = f(X) \oplus \left(soft\max\left(Conv\left(f(X)\right)_M\right.\right. \cdot Conv\left(f(X)\right)_M^T \cdot \left.\left. Conv\left(f(X)\right)_M\right)\right). \tag{11}$$

$f^*(X)$ is the search map feature using SANL attention coding.

After the feature matrices $f(Z)$ and $f(X)$ are encoded by SANL network, the correlation between each feature element of the matrix and the other elements is calculated to obtain $f^*(Z)$ and $f^*(X)$. Compared with the feature matrix without coding, the elements with tracking semantic information in $f^*(Z)$ and $f^*(X)$ are enhanced, so as to obtain better scores in the classification branch. The background elements in $f^*(Z)$ and $f^*(X)$ are weakened, which will cause less interference to the score results of the classification branch. The feature values of the final object elements with rich semantic information are enlarged, while those of irrelevant elements are reduced.

Cross-attention calculation

The cross-attention-non-local (CANL) network designed in this paper takes the features of the search map as the input of *query* matrix, and the features of the template map as the input of *key* and *value* matrices, that is, $A = f^*(X)$ and $B = f^*(Z)$. CANL network structure diagram is shown in Fig. 5.

The template map feature $f^*(Z)$ is used to encode the search map feature $f^*(X)$, which is exerted the attention influence. The relevant elements in the template map will enhance the features of the core semantic elements in the search map, and the cross-attention output is as follows:

$$f^{**}(X) = f^*(X) \oplus \left(soft\max\left(Conv\left(f^*(X)\right)_M\right.\right. \cdot Conv\left(f^*(Z)\right)_M^T \cdot \left.\left. Conv\left(f^*(Z)\right)_M\right)\right), \tag{12}$$

Where $f^{**}(X)$ is the search map feature using CANL attention coding, $f^*(X)$ is the search map feature using SANL attention coding, and $f^*(Z)$ is the template map feature using SANL attention coding.

After the feature matrix $f^*(X)$ is encoded by CANL network, each element of its own is calculated by correlation with each element of $f^*(Z)$, and the output $f^{**}(X)$ is obtained. Compared with $f^*(X)$, the feature elements with semantic information of $f^{**}(X)$ are enhanced by the influence of $f^*(Z)$, while irrelevant background elements are weakened. Before the cross correlation of the classification and regression network, the search map features perceive the attributes of the template map features in advance, improving the generalization ability of the search map features. Thereafter, $f^*(Z)$ and $f^{**}(X)$ will be sent to the classification branch and the regression branch respectively.

Classification and regression subnetwork

The algorithm designed in this paper uses RPN to achieve classification and regression of the object, in which the

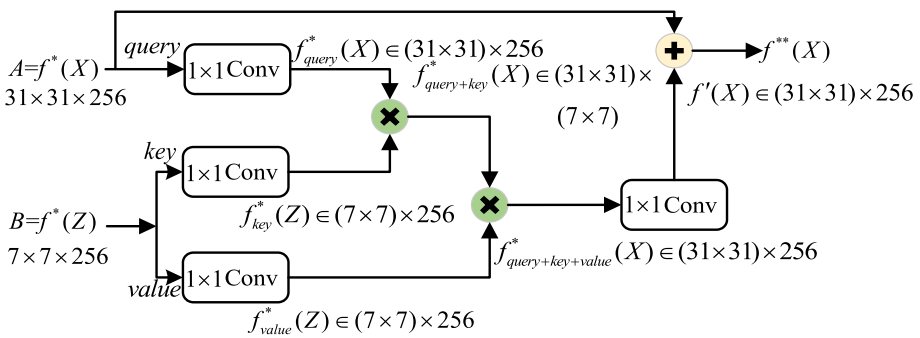


Fig. 5 CANL network structure diagram

function of the classification branch is to distinguish the foreground from the background, the foreground refers to the location of the object, and the background refers to the location of non-object. The function of the regression branch is to determine the size of the object. If k anchor boxes with different scales are added to the object, the classification features will have $2k$ channels and the regression features will have $4k$ channels. The template features $f(Z)$ and the search features $f(X)$ are depth cross correlated to obtain the depth features $Y_1 \in \mathbb{R}^{H\times W\times D}$, and the template features $f^*(Z)$ after SANL network coding and the search features $f^{**}(X)$ after CANL network coding are depth cross correlated to obtain the depth features $Y_2 \in \mathbb{R}^{H\times W\times D}$. Matrix Y_1 and matrix Y_2 are added together to get the final output $Y \in \mathbb{R}^{H\times W\times D}$.

$$Y_1 = f(Z) * f(X) \tag{13}$$

$$Y_2 = f^*(Z) * f^{**}(X) \tag{14}$$

where $*$ denotes the channel-by-channel correlation operation.

$$Y = Y_1 \oplus Y_2 \tag{15}$$

$Y \in \mathbb{R}^{H\times W\times D}$ is divided into two branches used for classification and regression. In the classification branch, the channels are compressed to $2k$ by using the convolution whose convolution kernel size is 1×1 to obtain $Y_{cls} \in \mathbb{R}^{H\times W\times 2k}$. In the regression branch, the channels are compressed to $4k$ by using the convolution whose convolution kernel size is 1×1 to obtain $Y_{reg} \in \mathbb{R}^{H\times W\times 4k}$.

Design of loss function for classification branch

It is assumed that there are N classification tasks and K samples, the feature vector of input samples is represented by x_i, and the category label of input samples is $y_i \in [1, N]$. Then it is easy to obtain the cross entropy loss of this sample:

$$\begin{aligned}\ell_{\text{Softmax}}(x_i) &= -\log P_{i,y_i} = \frac{1}{K}\sum_i -\log\frac{e^{W_{y_i}^T x_i}}{\sum_j^N e^{W_j^T x_i}} \\ &= \frac{1}{K}\sum_i -\log\frac{e^{\|W_{y_i}^T\|\|x_i\|\cos(\beta_{y_i})}}{\sum_j^N e^{\|W_j^T\|\|x_i\|\cos(\beta_j)}}\end{aligned}. \tag{16}$$

P_{i,y_i} represents the probability that the sample belongs to class y_i, $W = [W_1, W_2, \cdots, W_N]$ represents the parameter of the last fully connected layer of the network, and $\beta_j\left(j \in [1, N]\right)$ represents the angle between W_j and x_i. Suppose $N = 2$, the cross entropy loss of x_i can be expressed as:

$$\begin{aligned}\ell(x_i) &= -\log\frac{e^{\|W_1^T\|\|x_i\|\cos(\beta_{y_i})}}{e^{\|W_1^T\|\|x_i\|\cos(\beta_{y1})} + e^{\|W_2^T\|\|x_i\|\cos(\beta_{y2})}} \\ &= -\log\frac{1}{1+e^{\|x_i\|\left(\|W_2^T\|\cos(\beta_{y2}) - \|W_1^T\|\|x_i\|\cos(\beta_{y1})\right)}}\end{aligned}. \tag{17}$$

It can be seen that when the cross entropy loss $\ell(x_i)$ is minimized in the process of network training, x_i will gradually approach vector W_1 while moving away from W_2. Similarly, if the class label of x_i is y_2, it will be closer to W_2 and farther away from W_1. Therefore, the cross entropy loss function ignores the intra-class compactness while maximizing the inter-class separability. In the object tracking task, even the same object will be diversified due to different angles of view and illumination. Therefore, the pairwise Gaussian loss function [31] (PGL) is proposed to calculate the classification loss to improve the intra-class compactness of the object. The Eq. for calculating PGL is as follows:

$$\ell_{PGL} = \frac{4}{K^2}\sum_{i=1}^{K}\sum_{j=i+1}^{K}\left[\eta d_{ij}^2 + (y_{ij}-1)\log\left(e^{\eta d_{ij}^2} - 1\right)\right], \tag{18}$$

where η represents the simplified proportional parameter of the Gaussian function and d_{ij} represents the Euclidean distance between two features. y_{ij} indicates whether two features have the same class label. If two features are the same, y_{ij} is 1; otherwise,y_{ij}. is 0. For two features that belong to the same object, y_{ij}. is 1. Then, PGL can be expressed as:

$$\ell_{PGL} = \frac{4}{K^2}\sum_{i=1}^{K}\sum_{j=i+1}^{K}\eta d_{ij}^2. \tag{19}$$

It can be seen that if the Euclidean distance d_{ij} is bigger, the loss ℓ_{PGL} is greater, the penalty imposed is greater, and the intra-class compactness is higher. Therefore, the cross entropy loss and pairwise Gaussian loss are combined as the total loss function of the classification model. The intra-class compactness is improved through PGL, and the inter-class separability is improved through the cross entropy loss function.

The loss function ℓ_{cls} of the classification branch can be expressed as the followings:

$$\ell_{cls} = \ell_{PGL} + \ell_{soft\max}. \tag{20}$$

Design of loss function for regression branch

The smooth L_1 loss function is used to train regression. We let (x, y) and (w, h) represent the coordinate of the center point and the size of the anchor box, and (x_0, y_0) and (w_0, h_0) represent the coordinate of the center point and the size of the real-time frame. After normalization of the regression distance, Eq. (21) is obtained.

$$\begin{aligned}\Re[0] &= \tfrac{x_0 - x}{w},\ \Re[1] = \tfrac{y_0 - y}{h},\\ \Re[2] &= 1\mathrm{n}\tfrac{w_0}{w},\ \Re[3] = 1\mathrm{n}\tfrac{h_0}{h}\end{aligned} \tag{21}$$

Then, the regression is calculated through the smooth L_1 loss, as shown in Eq. (22).

$$\mathrm{smooth}_{L_1}(x,\delta) = \begin{cases} 0.5\delta^2 x^2 & |x| < \frac{1}{\delta^2} \\ |x| - \frac{1}{2\delta^2} & |x| \geq \frac{1}{\delta^2} \end{cases}, \tag{22}$$

where δ is a hyper-parameter of the Huber Loss. The loss of the regression branch is calculated as follows:

$$\ell_{reg} = \sum_{j=0}^{3} \mathrm{smooth}_{L_1}\left(\Re[j], \delta\right). \tag{23}$$

The final loss of the classification and regression network is calculated as follows:

$$\ell = \ell_{cls} + \lambda \ell_{reg} = \ell_{soft\max} + \lambda_1 \ell_{PGL} + \lambda_2 \ell_{reg}, \tag{24}$$

where the constants λ_1 and λ_2 are hyper-parameters that balance the classification loss and the regression loss. During the model training, λ_1 and λ_2 are set at 1.8 and 2.5 respectively.

Algorithm flow

The classification and regression subnetwork is used to locate the object, in which the classification branch distinguishes foreground and background, and the regression branch determines the size of the object. The main working steps of the proposed algorithm are shown as follows:

Input: Template map Z and search map X
Output: Estimated object position
1: Build MFFMT and MFFMS, extract the hierarchical features in the last three layers of the template map Z and the search map X to obtain $f(Z)$ and $f(X)$.
2: Calculate self-attention features $f^*(Z)$ and $f^*(X)$ using the self-attention network.
3: Obtain the cross-attention output $f^{**}(X)$ using the cross-attention network.
4: Perform the deep cross-correlation operation on $f^*(Z)$ and $f^{**}(X)$ to obtain the depth features $Y_1 \in \mathbb{R}^{H\times W\times D}$ and $Y_2 \in \mathbb{R}^{H\times W\times D}$.
5: Calculate the final output $Y \in \mathbb{R}^{H\times W\times D}$ which is divided into two branches for classification and regression.
6: Design the loss functions ℓ_{cls} and ℓ_{reg} to achieve classification and regression results of boundary boxes.

Result analysis and discussion

Implementation details

In this paper, the algorithm is built based on the pytorch deep learning framework. The GPU is NVIDIA GeForce GTX 1080 and the processor is Intel Core i7-8550U at 2.0GHZ CPU. ResNet50 is initialized with the weights pre-trained by ImageNet [32], leaving the parameters of the first two layers unchanged. During the training phase, the stochastic gradient descent (SGD) is used to calculate loss functions of different layer features, and then calculate gradients to optimize the network parameters. The training data sets are ImageNetDET [32], COCO [33] and LaSOT [34]. Data sets used for testing include the visual object tracking 2018 (VOT2018) [35] and the object tracking benchmark 100 (OTB100) [36]. The learning rate decreases from 0.01 to 0.0005, the batch size is 64, and the training epoch is 30. In the first 15 epochs, the learning rate decreases from 0.01 to 0.005. In the last 15 epochs, the learning rate decreases from 0.005 to 0.0005. The number of anchor boxes used for the classification and regression subnetwork is set to $k = 5$.

Evaluation indicator

Evaluation indicator for VOT

The evaluation indicators used in VOT dataset include accuracy, robustness, and expected average overlap (EAO). The accuracy rate is used to evaluate the accuracy of the tracker. As the accuracy increases, the success rate increases. In each frame, the tracking accuracy is represented by the intersection ratio (IoU), which is defined as:

$$IoU = \frac{B_G \cap B_T}{B_G \cup B_T}, \tag{25}$$

where B_G represents the boundary box marked manually and B_T represents the predicted boundary box.

Robustness is used to evaluate the stability of the tracker. The more times the tracker restarts, the greater the robustness value, indicating that the tracker is more unstable. EAO is an indicator derived from the comprehensive evaluation of the intersection ratio, restart interval, and restart times, which can reflect the comprehensive performance of the tracker.

Evaluation indicator for OTB

The evaluation indicators of OTB dataset are success rate and precision rate respectively. The success rate is the rate of tracking success across all video frames. A threshold is set and the cross merge ratio is used to determine whether it is successful. The precision rate pays attention to whether the object center position predicted by the algorithm is close to the marked center position. The precision rate represents the percentage of the center

location errors between predicted position and ground-truth with different thresholds.

Ablation experiment

To verify the performance of the proposed SiamANAL algorithm, ablation experiment and analysis of each component are performed and verified on OTB100. The baseline algorithm used for comparison is SiamFC, and the independent role of each component of the algorithm is experimentally tested. The benchmark algorithm is represented by Baseline, the tracking result without using fusion module is represented by BaseLine_UN_M, the multi-feature fusion module is represented by BaseLine_M, the self-attention network is represented by BaseLine_M_SANL, the cross-attention network is represented by BaseLine_M_CANL, the self-attention and cross-attention network is represented by BaseLine_M_SANL_CANL, and the fusion of all components is represented by BaseLine_SUM. The experimental results are shown in Table 1. It can be seen from the Table 1 that BaseLine_SUM adopted by the proposed SiamANAL algorithm achieves the best tracking result by using the multi-layer feature fusion module, the self-attention network, and the cross-attention network on OTB100. The across-attention network BaseLine_CANL has obvious advantages in improving the success rate and precision rate, and plays a greater role than the multi-feature fusion module BaseLine_M. By selecting the attention of the feature map, the background interference of the object region can be filtered out to enhance the expression ability of the object region and effectively improve the tracking performance.

Table 1 Comparison of different optimized components of SiamANAL algorithm on OTB100

Different optimized components	Precision	Success rate
BaseLine	0.771	0.582
BaseLine_UN_M	0.776	0.590
BaseLine_M	0.789	0.601
BaseLine_M_SANL	0.812	0.639
BaseLine_M_CANL	0.827	0.668
BaseLine_M_SANL_CANL	0.865	0.689
BaseLine_SUM	**0.883**	**0.709**

Table 2 Accuracy, robustness and EAO of various algorithms on VOT2018

Algorithm	Accuracy	Robustness	EAO
KCF	0.447	0.773	0.135
Staple	0.530	0.688	0.169
SiamFC	0.503	0.585	0.187
SiamRPN	0.526	0.376	0.383
SiamRPN++	0.531	0.318	0.286
Siam R-CNN	0.585	0.311	0.381
SiamANAL	0.592	0.304	0.403

Quantitative experiment and analysis

The proposed SiamANAL algorithm has achieved excellent results with mass testing on VOT2018 and OTB100 comparing with other competing tracking algorithms.

Result analysis on VOT2018

The VOT dataset is a classic object tracking test dataset, which is proposed by the VOT challenge in 2013, and its data content is updated every year. The VOT2018 dataset contains a total of 60 video sequences, all of which are marked by the following visual attributes: occlusion, illumination variation, motion variation, size variation, and camera motion. As shown in Table 2, the proposed SiamANAL algorithm is compared with KCF [15], Staple [37], SiamFC [20], SiamRPN [21], SiamRPN++ [25], and Siam R-CNN [26] tracking algorithms. SiamANAL algorithm obtains high accuracy and EAO values. In conclusion, the SiamANAL algorithm shows good tracking performance on VOT2018. Compared with the SiamRPN algorithm, the accuracy rate is improved by 0.066, the robustness is improved by 0.072, and the EAO is improved by 0.020. The attention network structure enhanced the expression of core semantic elements in the template map features and search map features, thus improving the accuracy of tracking frame extraction in the tracking process.

To compare testing results more intuitively, the comparison results are displayed in the form of a histogram as shown in Fig. 6.

In scenes with background interference, there may be complex background and similar objects. The attention network can weaken background feature elements, thus reducing the influence of background on tracking effect.

Result analysis on OTB100

OTB100 benchmark divides object tracking scenes into 11 types of visual challenge attributes and labels the challenge attributes for each video sequence. Each video sequence has more than one attribute tag corresponding to it. In this way, the tracking ability of the algorithm in different challenge attributes can be analyzed. 11 visual challenge attributes are: scale variation (SV), illumination variation (IV), motion blur (MB), deformation (DEF), occlusion (OCC), out-of-plane rotation (OPR), fast motion (FM), background clutter (BC), out-of-view (OV), in-plane rotation

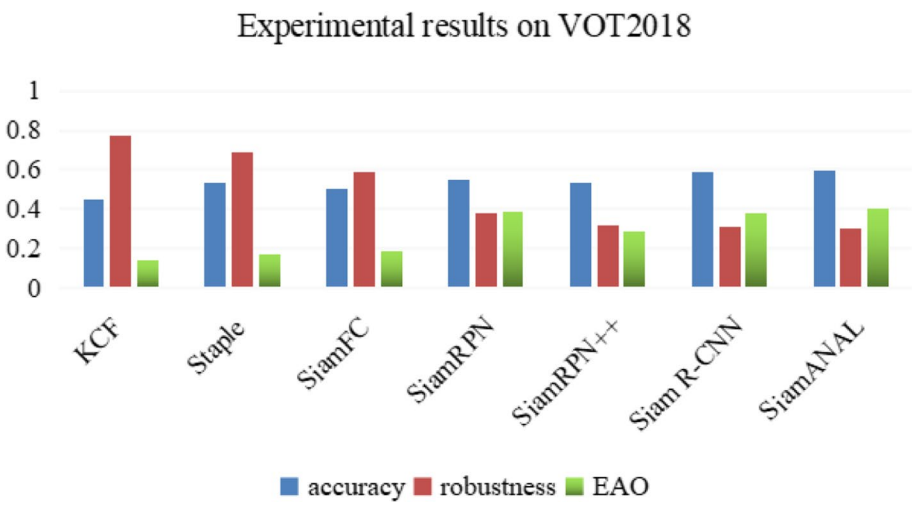
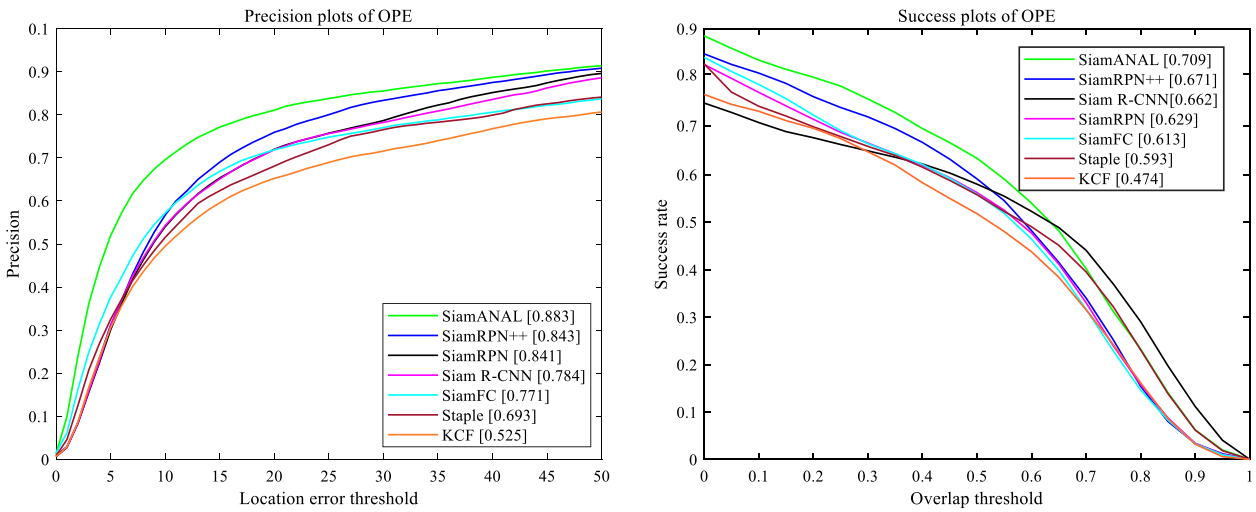


Fig. 6 Experiment results on VOT2018

(IPR), and low resolution (LR). The algorithm starts tracking from the real position of the object in the first frame, and obtains tracking accuracy and success rate using one pass evaluation (OPE). As shown in Fig. 7, the proposed SiamANAL algorithm is compared with KCF [15], Staple [37], SiamFC [20], SiamRPN [21], SiamRPN++ [25], and Siam R-CNN [26] tracking algorithms. The success rate and precision rate of SiamANAL algorithm rank first. Compared with SiamRPN, the success rate and precision are improved by 4.7% and 10.6%, respectively. It is proved that the extracted features have strong discrimination ability, and the design of loss function in classification and regression subnetwork is effective.

For video sequences with challenging attributes, the tracking results are compared as shown in Fig. 8. In the video sequences with BC attribute, the designed attention network can effectively filter out the background information and enhance the features of the object position, achieving high precision and success rate. This shows that the proposed algorithm performs well in the BC scenes. However, in the video sequence with OPR attribute, the precision score of the proposed SiamANAL algorithm is the second, and the object position located by the regression branch has a certain deviation.

Table 3 further shows the precision indicator and center location error (CLE) indicator of each comparison algorithm on OTB100. Specifically, the SiamANAL algorithm exceeds the comparison algorithms in terms of precision and CLE on the OTB100 dataset. The CLE value obtained by SiamANAL is 9.6, which is significantly



(a) Precision of different algorithms (b) Success rate of different algorithms

Fig. 7 Comparison of success rate and precision on OTB100. **a** Precision of different algorithms. **b** Success rate of different algorithms

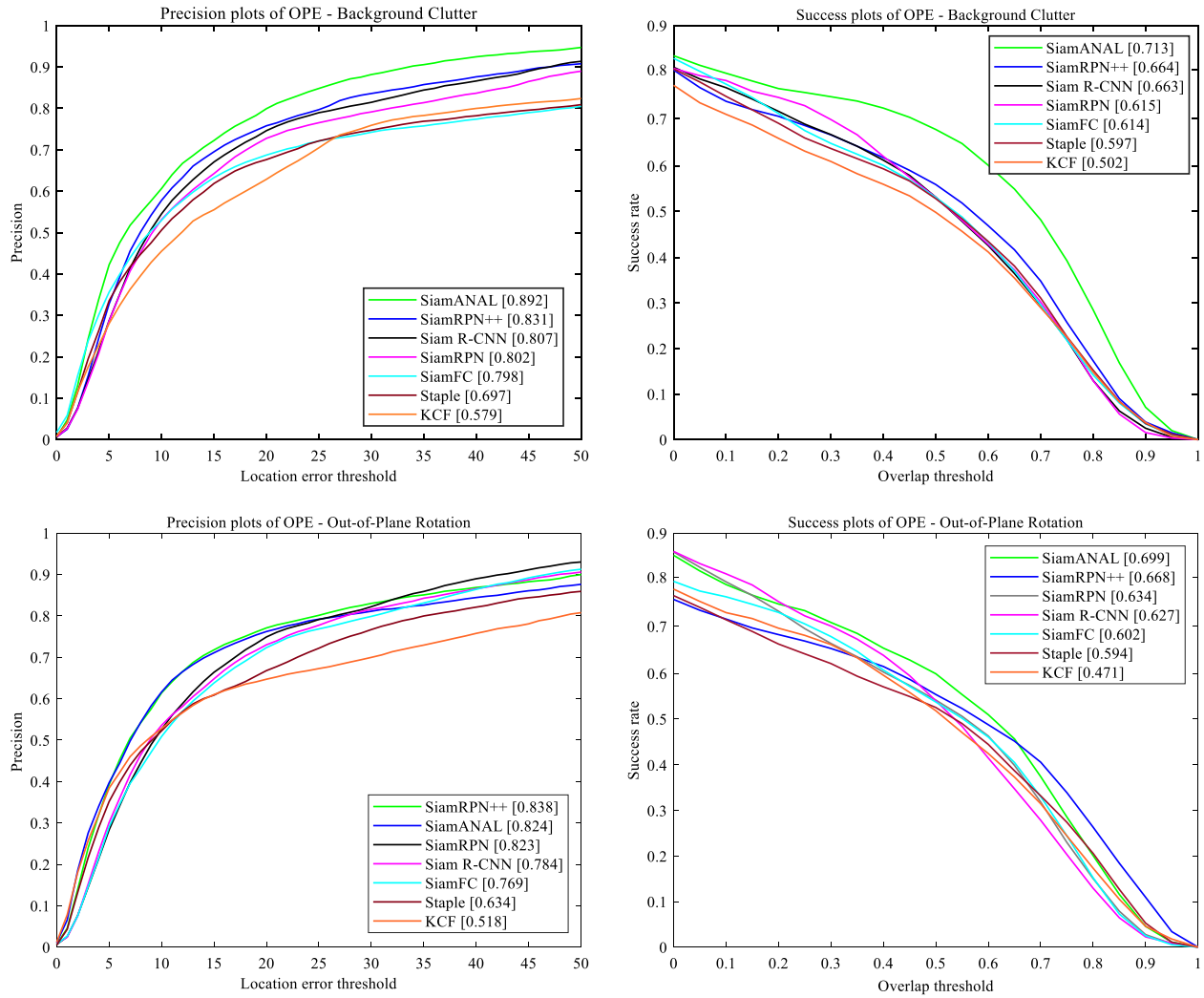


Fig. 8 Comparison of different algorithms with different video attributes

higher than that of SiamRPN++ (14.3FPS), which ranks second in the tracking effect. It is verified that the Non-local attention network performs the self-attention and cross-attention calculation on features, which enhances the expression ability of deep features, and reduces the parameter amount and calculation amount of CNN.

Performance and speed analysis

To verify the real-time tracking performance of the SiamANAL algorithm, the SiamANAL algorithm is compared with other comparison algorithms on OTB100 in terms of success rate and speed. Some high-performance algorithms are usually designed to achieve high tracking accuracy, but this will affect real-time tracking. Similarly, some simple algorithms have good real-time performance, but the tracking accuracy is difficult to meet. It can be seen from Fig. 9 that the SiamANAL algorithm has achieved a high success rate with a speed of 49 FPS (Frame Per Second), which is not the fastest, but can meet the basic real-time tracking requirements with

Table 3 Comparison of precision and CLE on OTB100

Benchmark	Evaluating indicator	KCF	Staple	SiamFC	SiamRPN	SiamRPN++	Siam R-CNN	SiamANAL
OTB100	precision	0.525	0.693	0.771	0.841	0.843	0.784	0.883
	CLE	35.2	33.1	23.1	15.5	14.3	17.2	9.6
Speed (FPS)		89	23.8	50	21	15	3.7	49

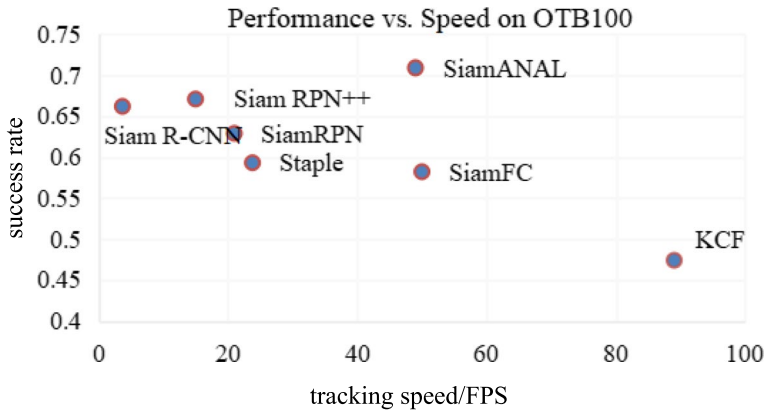


Fig. 9 Performance vs. speed on OTB100

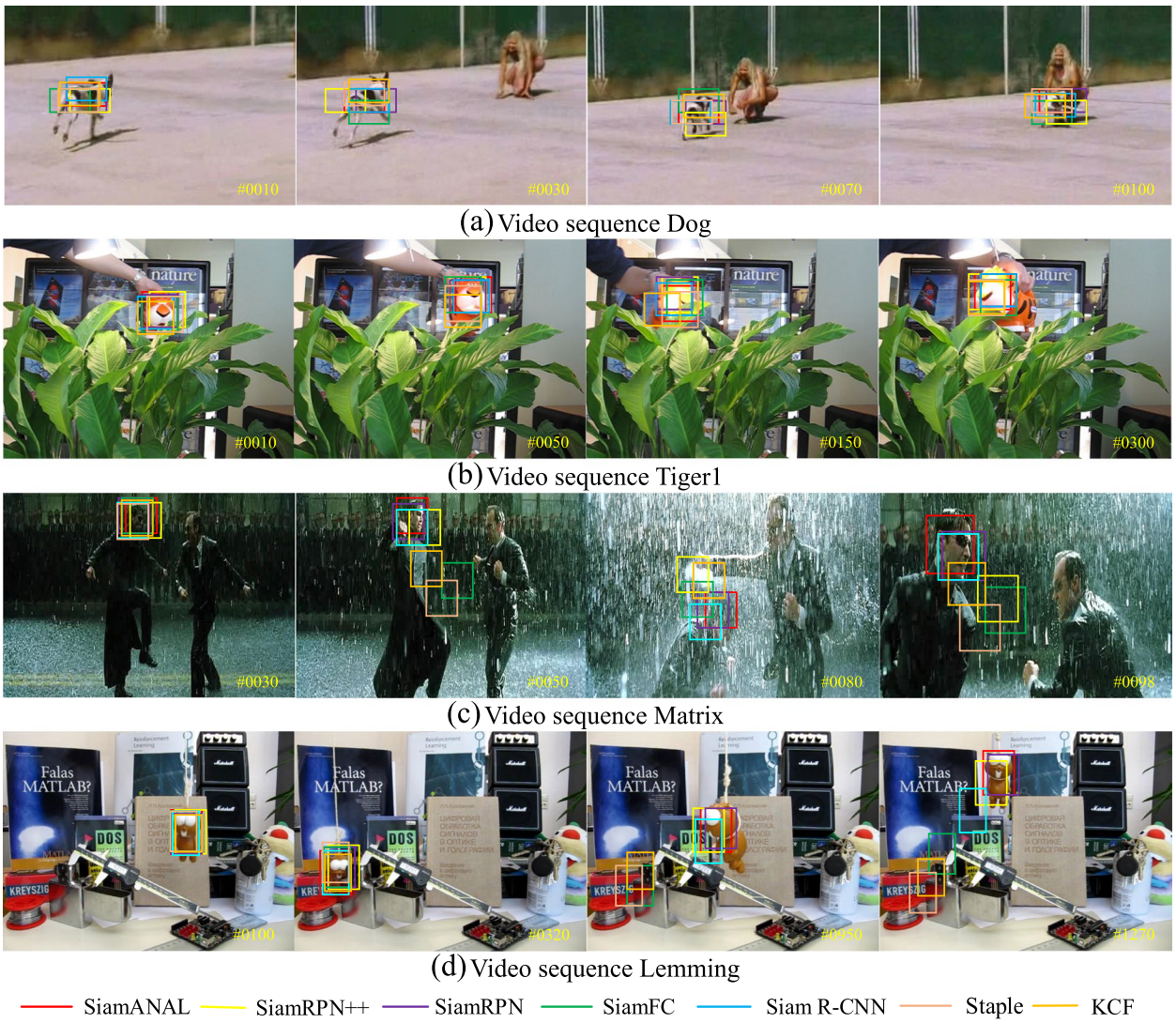


Fig. 10 Qualitative comparison in typical video sequences of the OTB100 benchmark. **a** Video sequence Dog. **b** Video sequence Tiger1. **c** Video sequence Matrix. **d** Video sequence Lemming

25 FPS. The speed achieved by SiamANAL algorithm is slightly lower than that of the SiamFC, but the tracking success rate achieved by SiamANAL algorithm is much higher than that of the SiamFC. This is because the hierarchical features in the last three layers of the ResNet50 network in this paper are optimized by the fusion module, which makes the extracted features more discriminative. The floating-point computation of SiamANAL algorithm is 5.821 GFLOPS (Giga Floating-point Operations Per Second), in which the introduced attention network requires 0.434 GFLOPS. Further more, the lightweight attention network takes up a very low amount of computation in the real-time object tracking task, so the tracking algorithm has a good real-time performance. The large distance displacement between frames caused by high-speed motion will rarely occur, which is also conducive to the tracking algorithm to track the object more accurately.

Qualitative experiment and analysis

To intuitively illustrate the accuracy of different algorithms, tracking results in the tracking sequence will be compared and analyzed. Figure 10 shows the results of visual comparison with six popular algorithms (KCF [15], Staple [37], SiamFC [20], SiamRPN [21], SiamRPN++ [25], and Siam R-CNN [26]) in four typical video sequences Dog, Tiger1, Matrix, and Lemming.

In the video sequence with LR and DEF attributes (Fig. 10a Dog), the object has lower pixels and insufficient detail features, but the attention network can enhance the semantic expression of object feature elements, so as to identify the tracking object more accurately.

In the video sequence with the IV attribute (Fig. 10b Tiger1), the proposed algorithm can effectively overcome the influence brought by illumination variation and achieve robust tracking effect. The tracking results show that the multi-feature fusion model enhances the features of the object region and reduces the background interference.

In the video sequence with the FM and BC attributes (Fig. 10c Matrix), the details of the object are weakened due to the motion blur of the object. The attention network can also enhance the semantic expression ability of the object, and the tracking object can also be accurately obtained under the fuzzy state.

In the video sequence with the OCC attribute (Fig. 10d Lemming), the proposed SiamANAL algorithm can accurately locate the object after the occluded object reappeared through the classification and regression subnetwork according to the template map.

Other comparison algorithms achieve good tracking effect in video sequences Dog and Tiger1, and reduce the influence of low resolution and illumination variation on the object location. However, in the scene where the object is occluded, the comparison algorithms appear different degrees of tracking drift and cannot relocate the object after the object reappears.

Conclusion

The object tracking algorithm SiamANAL based on the siamese network is designed by introducing attention network and adaptive loss function. The following conclusions can be drawn:

(1) The multi-feature fusion module integrates hierarchical features in the last three layers of the ResNet50 network, which can solve the problem of partial feature loss caused by too deep network.
(2) The self-attention and cross-attention modules in the attention network calculate the attention of the template feature map and the search feature map, so that the calculated features highlight the object area, making the tracking process pay more attention to the object.
(3) Two loss functions, cross entropy loss and pairwise Gaussian loss, are designed to maximize intra-class compactness and inter-class separability, and improve the accuracy of object classification.
(4) Through quantitative and qualitative analysis of the tracking results on VOT2018 and OTB100, the proposed SiamANAL algorithm performs well in performance and various challenging video sequences.

In this paper, the tracking algorithm uses a fixed template map, which is not updated during the tracking process, resulting in the tracking results concussion when the algorithm deals with the problem of long-time occlusion of the object. In the future study, an effective object tracking method based on dual template fusion will be designed and the algorithm will be deployed in the cloud to further improve the robustness of the tracking algorithm.

Acknowledgements
Not applicable.

Authors' contributions
Jinping Sun carried out the design of multi-feature fusion algorithm, the loss function, and attention network, performed all experimental tests, and drafted the manuscript. Dan Li mainly proofread the manuscript. All authors reviewed and agreed to the published version of the manuscript.

Funding
This research was funded by Basic Science Major Foundation (Natural Science) of the Jiangsu Higher Education Institutions of China (Grant: 22KJA520012), Natural Science Foundation of Shandong Province (Grant: ZR2021MD082), Jiangsu Province Industry-University-Research Cooperation Project (Grant: BY2022744), Xuzhou Science and Technology Plan Project (Grant: KC21303, KC22305), and the sixth "333 project" of Jiangsu Province.

Availability of data and materials
The datasets supporting the conclusions of this article are available publicly in https://www.votchallenge.net/vot2018/dataset.html and http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Both authors provide consent for publication.

Competing interests
The authors declare no competing interests.

Received: 9 January 2023 Accepted: 23 March 2023
Published online: 04 April 2023

References

1. Li D, Bei LL, Bao JN, Yuan SZ, Huang K (2021) Image contour detection based on improved level set in complex environment. Wirel Netw 27(7):4389–4402
2. Sun JP, Ding EJ, Sun B, Chen L, Kerns MK (2020) Image salient object detection algorithm based on adaptive multi-feature template. Dyna-bilbao 95(6):646–653
3. Chen X, Han YF, Yan YF, Qi DL, Shen JX (2020) A unified algorithm for object tracking and segmentation and its application on intelligent video surveillance for transformer substation. Proc CSEE 40(23):7578–7586
4. Bao WZ (2021) Artificial intelligence techniques to computational proteomics, genomics, and biological sequence analysis. Curr Protein Pept SC 21(11):1042–1043
5. Bao WZ, Yang B, Chen BT (2021) 2-hydr_Ensemble: Lysine 2-hydroxyisobutyrylation identification with ensemble method. Chemom Intell Lab Syst. https://doi.org/10.1016/j.chemolab.2021.104351
6. Zhang XY, Gao HB, Guo M, Li GP, Liu YC, Liu YC, Li DY (2016) A study on key technologies of unmanned driving. CAAI T Intell Techno 1(1):4–13
7. Zhang XL, Zhang LX, Xiao MS, Zuo GC (2020) Target tracking by deep fusion of fast multi-domain convolutional neural network and optical flow method. Computer Engineering & Science 42(12):2217–2222
8. Liu DQ, Liu WJ, Fei BW, Qu HC (2018) A new method of anti-interference matching under foreground constraint for target tracking. ACTA Automatica Sinica 44(6):1138–1152
9. Sun JP, Ding EJ, Li D, Zhang KL, Wang XM (2020) Continuously adaptive mean-shift tracking algorithm based on improved gaussian model. Journal of Engineering Science and Technology Review 13(5):50–57
10. Akhtar J, Bulent B (2021) The delineation of tea gardens from high resolution digital orthoimages using mean-shift and supervised machine learning methods. Geocarto Int 36(7):758–772
11. Pareek A, Arora N (2020) Re-projected SURF features based mean-shift algorithm for visual tracking. Procedia Comput Sci 167:1553–1560
12 Bolme DS, Beveridge JR, Draper BA, Lui YM (2010) Visual object tracking using adaptive correlation filters. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, San Francisco, pp 2544–2550
13. Henriques JF, Caseiror R, Martins P, Batista J (2012) Exploiting the circulant structure of tracking- by-detection with kernels. 12th European Conference on Computer Vision (ECCV). Springer, Florence, pp 702–715
14. Henriques JF, Carreira J, Rui C, Batista J (2013) Beyond hard negative mining: efficient detector learning via block-circulant decomposition. IEEE International Conference on Computer Vision (ICCV). IEEE, Sydney, pp 2760–2767
15. Henriques JF, Caseiro R, Martins P, Batista J (2015) High-speed tracking with kernelized correlation filters. IEEE T Pattern ANAL 37(3):583–596
16. Danelljan M, Häger G, Khan FS, Felsberg M (2014) Accurate scale estimation for robust visual tracking. In: Proceedings of the British Machine Vision Conference (BMVA). Nottingham, British Machine Vision Association.
17. Leibe B, Matas J, Sebe N, Welling M (2016) Beyond correlation filters: Learning continuous convolution operators for visual tracking. European Conference on Computer Vision (ECCV). Springer, Amsterdam, pp 472–488
18 Danelljan M, Bhat G, Khan FS, Felsberg M (2017) Eco: Efficient convolution operators for tracking. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Hawaii, IEEE, pp 6638–6646
19. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Las Vegas, pp 770–778
20. Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PHS (2016) Fully-convolutional siamese networks for object tracking. European Conference on Computer Vision (ECCV). Springer, Amsterdam, pp 850–865
21 Li B, Yan JJ, Wu W, Zhu Z, Hu XL (2018) High performance visual tracking with siamese region proposal network. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Salt Lake City, pp 8971–8980
22 Guo Q, Wei F, Zhou C, Rui H, Song W (2017) Learning dynamic siamese network for visual object tracking. Proceedings of the IEEE International Conference on Computer Vision (ICCV). Venice, IEEE, pp 1763–1771
23 Wang Q, Teng Z, Xing J, Gao J, Maybank S (2018) Learning attentions: residual attentional siamese network for high performance online visual tracking. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Salt Lake City, pp 4854–4863
24 He A, Luo C, Tian X, Zeng W (2018) A twofold siamese network for real-time object tracking. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Salt Lake City, pp 4834–4843
25 Li B, Wu W, Wang Q, Zhang FY, Xing JL, Yan JJ (2019) SiamRPN++: evolution of siamese visual tracking with very deep networks. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Long Beach, pp 4277–4286
26 Voigtlaender P, Luiten J, Torr PHS (2020) Siam R-CNN: visual tracking by re-detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Washington, pp 6577–6587
27. Guo DY, Wang J, Cui Y, Wang ZH, Chen SY (2020) SiamCAR: siamese fully convolutional classification and regression for visual tracking. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Washington, pp 6269–6277
28. Chen ZW, Zhang ZX, Song J (2021) Tracking algorithm of Siamese network based on online target classification and adaptive template update. Journal on Communications 42(8):151–163
29. Tan JH, Zheng YS, Wang YN, Ma XP (2021) AFST: Anchor-free fully convolutional siamese tracker with searching center point. ACTA Automatica Sinica 47(4):801–812
30 Wang XL, Girshick R, Gupta A, He KM (2018) Non-local neural networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Salt Lake City, pp 7794–7803
31. Qin Y, Yan C, Liu G, Li Z, Jiang C (2020) Pairwise gaussian loss for convolutional neural networks. IEEE T Ind Inform 16(10):6324–6333
32. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M (2015) Imagenet large scale visual recognition challenge. Int J Comput Vision 115(3):211–252
33 Lin TY, Maire M, Belongie S, Hays J, Zitnick CL (2014) Microsoft coco: common objects in context. European Conference on Computer Vision (ECCV). Springer, Zurich, pp 740–755
34 Fan H, Lin L, Fan Y, Peng C, Ling H (2019) LaSOT: A high-quality benchmark for large-scale single object tracking. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Long Beach, pp 5374–5383
35. Kristan M, Leonardis A, Matas J, Felsberg M, He ZQ (2018) The sixth visual object tracking VOT2018 challenge results. Proceedings of the European Conference on Computer Vision (ECCV). Springer, Munich, pp 3–53
36. Wu Y, Lim J, Yang MH (2015) Object tracking benchmark. IEEE T Pattern ANAL 37(9):1834–1848
37. Bertinetto L, Valmadre J, Golodetz S, Miksik O, Torr PHS (2016) Staple: complementary learners for real-time tracking. IEEE Conference on Computer Vision and Pattern Recognition(CVPR). IEEE, Las Vegas, pp 1401–1409

Publisher's Note