International Journal of Computational Intelligence Systems (2024) 17:107
https://doi.org/10.1007/s44196-024-00500-0

RESEARCH ARTICLE

TFITrack: Transformer Feature Integration Network for Object Tracking

Xiuhua Hu[1] · Huan Liu[1] · Shuang Li[1] · Jing Zhao[1] · Yan Hui[1]

Received: 17 January 2024 / Accepted: 4 April 2024

Abstract
Due to the ignoring of rich spatio-temporal and global contextual information with convolutional neural networks in features extraction, the traditional method is prone to tracking drift or even failure in complex scenario, especially for the tiny targets in aerial photography scenario. In this work, it proposes a transformer feature integration network (TFITrack) to obtain diverse and comprehensive target feature for the robust object tracking. Based on the typical transformer architecture, it optimizes encoder and decoder structure for aggregating discriminative spatio-temporal information and global context-awareness feature. Furthermore, the encoder introduces the similarity calculation layer and dual-attention module; the aim is to deepen the similarity between features and make corrections for channel and spatial dimensions, and feature representation is improved. Finally, with the introduction of the temporal context filtering layer, unimportant feature information is ignored adaptively, obtaining a balance between the parameters number reduction and stable performance. Experimental results show that the proposed tracking algorithm exhibits excellent tracking performance on seven benchmark datasets, especially on the aerial dataset UAV123, UAV20L, and UAV123@10fps, which presents the advantages of the novel method in dealing with fast motion and external interference.

Keywords Object tracking · Transformer · Feature integration · Spatio-temporal · Contextual feature

1 Introduction

Object tracking is an important aspect of computer vision and is widely used in various domains including drones, robotics and security surveillance. The main purpose of object tracking is to recognize and track the target based on the initial frame of the video, achieving continuous tracking of the subsequent frames. Most current tracking algorithms use convolutional neural network for feature extraction, calculating the similarity between the template feature and the search feature to estimate the state of the target during the tracking process. However, methods based on convolutional neural network mainly emphasize local modeling, often ignoring global contextual and spatio-temporal information. As a result, these algorithms may suffer from tracking drift or failure when confronted with complex scenarios such as fast motion or tracking aerial object. Considering the widespread use of current trackers, it is essential to develop a tracker that ensures stability and robust performance while adapting effectively to diverse scenarios. This design will address the limitations of existing methods and improve tracking accuracy and reliability.

Most current trackers are based on the powerful matching capabilities of Siamese networks. The first application of Siamese networks in tracking appeared in SiamFC [1]. It used a full convolutional Siamese network structure to match the similarity between the search image and the template image, thus successful tracking the target. Researchers introduced RPNs from the domain of object detection into object tracking to enhance tracking accuracy further. This led to the development of the SiamRPN [2] tracking architecture, which utilized RPN to predict accurate anchor frames and enhance tracker precision. However, these trackers encounter difficulties when disturbed by similar semantic information or when the size of the tracked target changes significantly. In such situations, the tracking effect tends to be poor.

CNN networks with more convolutional layers can effectively capture features with semantic information and fully represent the target features. Therefore, SiamRPN++ [3] and SiamDW [4] enabled the tracking

✉ Xiuhua Hu
huxhxatu@163.com

[1] School of Computer Science and Engineering, Xi'an Technological University, Xi'an 710021, Shaanxi, People's Republic of China

Published online: 29 April 2024

algorithms to capture deep semantic information more deeply and comprehensively, and enhanced the robustness of the tracking algorithm. C-RPN [5] connected multiple RPNs. Each RPN is trained using the output of the previous RPN and performs better in distinguishing challenging backgrounds. However, it is worth noting that these trackers increase the number of network hyper-parameters and get a higher computational burden due to the RPN construction anchor frame strategy.

In order to reduce the computationally intensive problems associated with the use of anchor frames. Researchers have introduced attention mechanism in tracking algorithms. NT-DPTC [6] developed a spatio-temporal dimension-preserving tensor complementary model that fully extracts intrinsic features. An effective strategy was designed to remove non-negative constraints during training. Meanwhile, temporal constraints and the AdamW method were employed to achieve high accuracy and fast convergence. SiamAttn [7] proposed deformable Siamese attention networks to improve the feature learning capability of Siamese network trackers. SiamAPN++[8] utilized the attention mechanism to perform attention aggregation networks through Self-AAN and Cross-AAN. Eventually, features representation was improved. HiFT [9] proposed a lightweight tracker utilizing transformer and multilayer features to obtain spatial interaction fusion in shallow convolutional layers and semantic information in deep convolutional layers. To balance model accuracy and computational efficiency, -DARTS [10] designed a lightweight Differentiable Architecture Search model. The proposed channel fusion compensation module eliminates inter-channel semantic information discrepancies and mitigates potential accuracy degradation. In addition, the augmented regularization technique with margins improves the system's architectural stability. Although the above tracking algorithms have made breakthroughs in utilizing attention mechanisms, most of them ignored spatio-temporal or global contextual information.

To enhance feature characterization and improve tracking robustness for tiny targets in the aviation domain, in this paper, the deep and shallow feature information is maximally utilized based on the transformer structure. Meanwhile, rich spatio-temporal and global contextual information is obtained by optimizing the encoder and decoder, which achieves the purpose of enhancing feature relevance and filtering unimportant feature information.

In general, the main contributions of this work can be summarized as follows.

(1) The transformer feature integration network frame aims to enhance feature representation. The frame focuses on combining local and global features using encoder and decoder to establish dependencies between features and within features.
(2) The new method reconstructs the similarity calculation layer and temporal context filtering layer to highlight salient features. The former is used to improve the similarity between features, while the latter adaptively filters unimportant information, helping to find a balance between reducing the number of parameters and stabilizing performance.
(3) The dual-attention mechanism is introduced to excavate the spatial structure information ignored by CNN networks.
(4) The proposed tracking algorithm achieves state-of-the-art tracking performance on several challenging benchmarks.

The remaining contents of this paper are organized as follows. In Sect. 2, a brief review of related work in the field of object tracking is concluded. In Sect. 3, the implementation process of the object-tracking algorithm is described in detail. In Sect. 4, the tracking method is evaluated and analyzed on several challenging benchmarks. In Sect. 5, the whole paper is concluded briefly.

2 Related Work

2.1 Transformer Network

Transformer [11], originally from Attention is all your need, abandons the traditional CNN and RNN approach of using convolutional layers to build networks. The whole network structure consists of an attention mechanism and a FFN. Transformer has superb long time series modeling ability and capturing global information perception ability. It has been widely used in the field of object tracking in the last 2 years based on this advantage.

TransT [12] proposed an attention mechanism for a feature fusion network, which can efficiently fuse search features and template features using attention. The issue of local linear matching in correlation operations losing semantic information and falling into local optimization has been resolved. DAPAT [13] introduced a novel training network model, which combines the anchor-free concept with Transformer theory. Stack [14] introduced Transformer to single target-tracking task and provided a transformer-based intermediate module, which improved feature representation. HiFT [9] proposed a hierarchical feature transformer module for fusing similarity images from multiple layers. The module not only captures global dependencies, but also efficiently learns the dependencies between multilevel features. TCTrack [15] better achieved a balance of speed and precision by continuously integrating temporal information in both the feature dimension and the similarity map dimension. E.T. Track [16] proposed a real-time visual

object-tracking network based on an Exemplar Transformer, which achieved up to 47 FPS on the CPU. LightVIT [17] proposed a convolution-free lightweight network, and used efficient strategies on self-attention and FFN, which led to significant progress in target detection and semantic segmentation. The mentioned studies propose valuable research ideas for key feature extraction and lightweight model construction for object tracking.

It can be seen that the introduction of a transformer in the tracking domain has achieved good results. However, some tracking algorithms ignore two drawbacks in FFN (feed forward network) when using the transformer structure. First, the channel dimension is drastically limited in order to reduce the computational cost, resulting in a lack of model representation. Second, the conventional FFN structure ignores the dependency modeling on the feature spatial level and fails to consider the spatio-temporal information. In addition, the multi-headed attention layer in transformer contains a large number of parameters, which can affect the performance of the tracker.

Therefore, this paper constructs a transformer feature integration network containing an encoder and a decoder inspired by the prevalent transformer structure [9, 15, 17]. In the encoder, a similarity calculation layer and a temporal context filtering layer are introduced, the self-attention mechanism to learn interdependencies between feature layers and the spatial information and integrate the global context and spatio-temporal information. In the decoder, Integrating template branch information with search branch information to effectively propagate temporal context information. Meanwhile, the main role of Transformer is to construct the dependencies between the features and fewer corrections for the channel and spatial features, thus a dual-attention mechanism is introduced in the FFN in the transformer network to further improve feature representation.

2.2 Attention Mechanism

Attention is a lightweight network that allows feature enhancement in different dimensions. It improves the accuracy of extracted features while having a small impact on the real-time performance of the tracker. Therefore, attention mechanism has been widely used in target detection, person re-identification, object tracking, etc.

RASNet [18] combined three attention mechanisms, channel, residual and general, to weight the spatial and channel of SiamFC features, this approach effectively decompose the coupling of feature extraction and discriminative analysis, which is used to improve the discriminative ability. SASiam [19] is a dual Siamese network consisting of semantic and appearance branches. The semantic branch with a channel attention mechanism filters out the background. Meanwhile, the appearance branch focuses on generalize indicator changes and enhances the judgment capabilities of the semantic Siamese network. SiamAttn [7] proposed a deformable Siamese attention network as a way to enhance the feature learning capability of Siamese network tracker. In addition, the attention mechanism provides a way for the tracker to update the features of the template feature adaptively. SiamAPN++ [8] contained two parts, Self-AAN and Cross-AAN. Self-AAN aggregates and models the self-semantic interdependencies of individual feature maps through spatial and channel dimensions. Cross-AAN aims to aggregate the interdependencies between different semantic features including anchor location information. This ultimately improves the feature representation. TDKD-Net [20] proposed the resource-saving exact network with a dual-attention mechanism that enables the model to focus on salient features from small-scale targets, improving detection accuracy. CDKD [21] designed a spatial and channel-oriented structural discriminative module to establish consistency and dependency of features between the teacher and student models. The module extracts discriminative spatial locations and channels while eliminating noise effects.

Some of the tracking method mentioned above disregard spatio-temporal and contextual information, and their performance is mostly tested on regular target datasets, which may not be representative of aerial datasets. Inspired by the above tracker, the influence of channel, spatial information and global contextual information on tracking robustness are considered. In this paper, we simultaneously use the multi-head self-attention mechanism, channel attention and spatial attention in transformer to aggregate global context and spatio-temporal information to enhance feature representation.

3 Transformer Feature Integration Network for Object Tracking

In order to obtain richer spatio-temporal and global context information and supplement the limitations of local operations in convolutional neural networks, the tracker can accurately track targets in complex scenario or aerial scenario, this paper proposes a target-tracking algorithm based on the transformer feature integration network. Specifically, the feature extraction network uses AlexNet, and the features of the last three layers are fused by convolution to obtain the new three feature vectors F_3, F_4, and F_5, respectively. F_3 and F_4 are dimensionally reshaped and are used as the inputs to the transformer encoder in order to construct global dependencies between the feature layers.

The encoder introduces a similarity calculation layer and the temporal context filtering layer. They are used to improve the similarity between features and adaptively filter unimportant information, respectively. The FFN introduces a dual-attention mechanism to enhance critical channel and

spatial information to further improve the tracker performance. The output of the encoder interacts with the dimensionally reshaped F_5 features in the transformer decoder to propagate temporal contextual information and improve feature representation to obtain the final feature vector for classification and regression networks. The network block diagram of the tracking algorithm proposed in this paper is shown in Fig. 1.

The proposed tracking algorithm consists of three main parts, the feature extraction network, the transformer feature integration network, and the classification regression network. Section 3.1 introduces the specific implementation of the feature extraction network, Sect. 3.2 introduces the design and implementation details of the transformer feature integration network, and Sect. 3.3 introduces the implementation of the whole algorithm. The classification regression networks use a traditional multilayer perceptron. The classification network is used to obtain the foreground score of the target in the image, and the regression network is used to obtain the predicted bounding box of the target. The specific location of the target in the video frame is determined by the foreground score of the classification network and the target bounding box is predicted by the regression network. In the subsequent subsections, the classification network and the regression network will not be presented again.

3.1 Feature Extraction Network

The Siamese network has the same upper and lower branch sub-networks and has two different inputs. Therefore, the architecture is widely used for the tracking task of computing the similarity between these inputs and estimating the position of the target in subsequent frames based on the initial frame features. In this paper, we use Siamese networks as the feature extraction network architecture, the upper branch serves as the template branch, while the lower branch serves as the search branch. Importantly, both branches share the same weights.

This paper utilizes the AlexNet network as a feature extraction network, the template image (z) and the search image (x) as inputs and are sent to the Siamese network architecture. The shallow features capture important appearance and color information that is crucial for accurate localization. The deep features capture semantic information that better describes the target. By utilizing shallow and deep features, the tracker can improve tracking accuracy and accurately localize the target even in complex scenarios. In this paper, a feature extraction network is used to extract the feature vectors of Layer3, Layer4, and Layer5. The template features and search features from these three layers are individually convolved to complement the fused feature information. This process produces three fused feature vectors, which are used as inputs for the proposed transformer feature fusion network. The multilayer feature extraction fusion network is shown in Fig. 2.

The feature vectors of the template and search image for Layer3, Layer4, and Layer5 layers are shown in Eqs. (1) and (2):

$$F_{zl3}, F_{zl4}, F_{zl5} = \text{AlexNet}(z) \tag{1}$$

$$F_{xl3}, F_{xl4}, F_{xl5} = \text{AlexNet}(x) \tag{2}$$

where z is the template image, x is the search image, $F_{zl3}, F_{zl4}, F_{zl5}$ are the feature vectors of Layer3, Layer4 and Layer5 layers of the template branch, respectively,

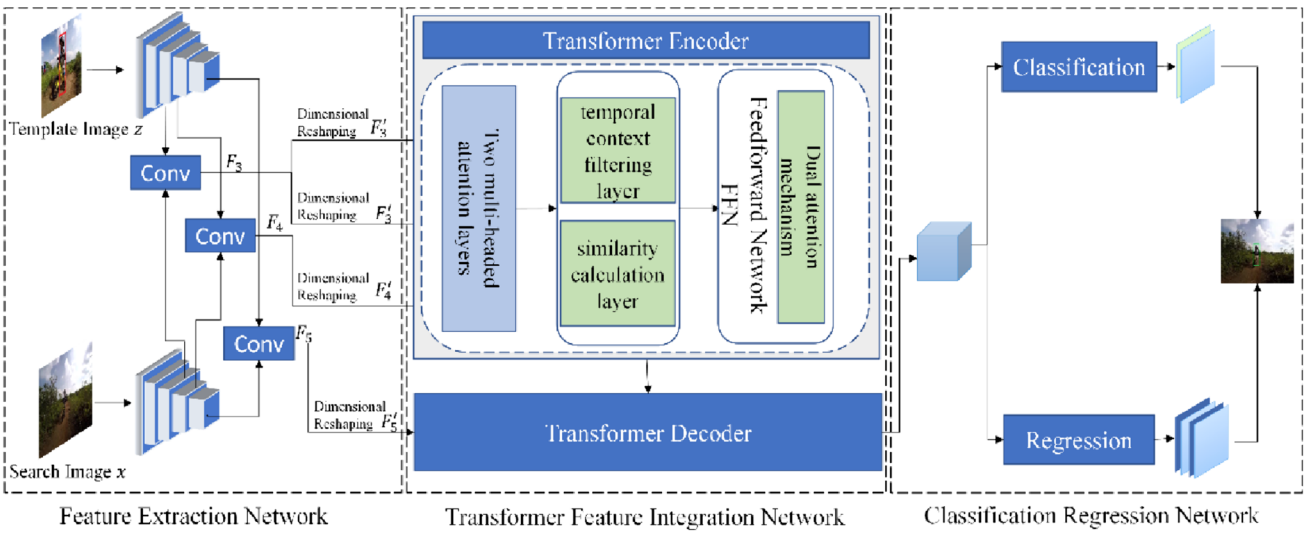


Fig. 1 Transformer feature integration network for object-tracking network block diagram

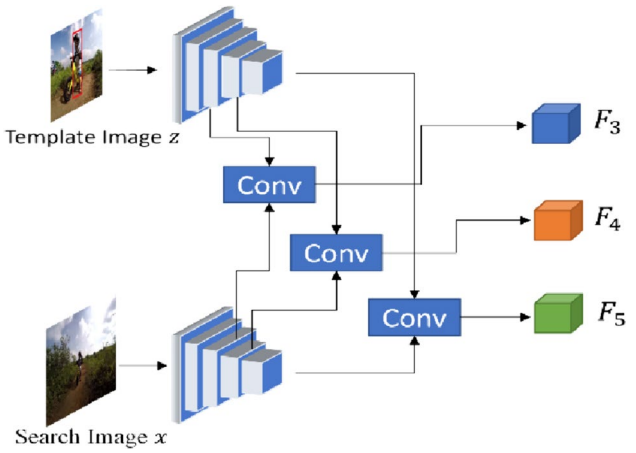


Fig. 2 Multilayer feature extraction fusion network

$F_{xl3}, F_{xl4}, F_{xl5}$ are feature vectors of Layer3, Layer4 and Layer5 layers of the search branch, respectively. The feature vectors from the three layers are fused using convolutional layers individually. This fusion process aims to effectively combine shallow and deep feature information to enhance feature representation. The feature fusion process is shown in Eqs. (3)–(5):

$$F_3 = \text{Conv}(F_{zl3}, F_{xl3}) \tag{3}$$

$$F_4 = \text{Conv}(F_{zl4}, F_{xl4}) \tag{4}$$

$$F_5 = \text{Conv}(F_{zl5}, F_{xl5}) \tag{5}$$

where Conv(.,.) is the convolutional layer, F_3, F_4, F_5 are the fused features of Layer3, Layer4, and Layer5 in the template branch and the search branch, respectively.

3.2 Transformer Feature Integration Network

The traditional transformer network consists of an encoder and a decoder, which are used by stacking the encoder and decoder to deepen the depth of the network. The encoder contains a multi-headed self-attention layer and a FFN layer. The multi-headed self-attention layer allows features to conduct self-attention learning, which effectively capture the internal relevance of target features. The decoder contains the same two layers as the encoder, and an interactive attention layer. The interactive attention layer enables the model to learn feature correlations between template features and search features, thus improving the overall performance of the network.

Self-attention is an important part of the transformer. The three feature matrices of Query (Q), Key (K) and Value (V) in self-attention are obtained from the embedded feature vectors, and the similarity between features is calculated using Q and K. Meanwhile, in order to prevent the value from being too large, it is first divided by dimensional constant and finally normalized using softmax and multiplied by V to get the self-attention feature vector. The formula for calculating the self-attention is shown in Eq. (6):

$$\text{Self\_Att}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

where Q, K and V are the linear transformations from the input features and d_k is the number of columns of the Q, K matrix, i.e., the dimensions of the vectors.

The multi-headed self-attention layer in the transformer is mapped to Q, K and V by nth linear transformations. Finally, the different attention results are concatenated. The multi-headed self-attention layer is computed in Eqs. (7) and (8):

$$\text{Multi\_Head}(Q, K, V) = \text{Concat}\left(h_1, \ldots, h_n\right) \tag{7}$$

$$h_i = \text{Self\_Att}(QW_i^Q, KW_i^K, VW_i^V) \tag{8}$$

where W_i^Q, W_i^K, W_i^V denote the weight matrix vectors of Q, K, V, respectively, and the Q, K, V are used in the self-attention.

In order to make the transformer structure suitable for the tracking task, a transformer feature integration network is designed. This network includes an encoder and a decoder. The encoder consists of two multi-headed attention layers, a similarity calculation layer, a temporal context filtering layer, and a FFN. In order to reduce the computational effort, a temporal context filtering layer is introduced at the output of the encoder's second multi-headed self-attention layer to adaptively ignore the unimportant feature information. Meanwhile, the FFN introduces a dual-attention mechanism in the encoder and decoder. This mechanism consists of two branches: channel attention and spatial attention. The channel attention branch captures global information, while the spatial attention branch focuses on local information. The outputs of the two branches are concatenated together to further improve the robustness of the tracking model. The decoder in the transformer feature integration network follows the structure of the conventional transformer network, but introduces a dual-attention mechanism in the FFN. The principle structure of the transformer feature integration network designed in this paper is shown in Fig. 3.

In the transformer feature integration network, the encoder can efficiently obtain global context and spatio-temporal information between the features. The template and search images are obtained by AlexNet in the upper and lower branches of the feature extraction network to get the features of Layer3 and Layer4, and these feature images are fused using a convolution

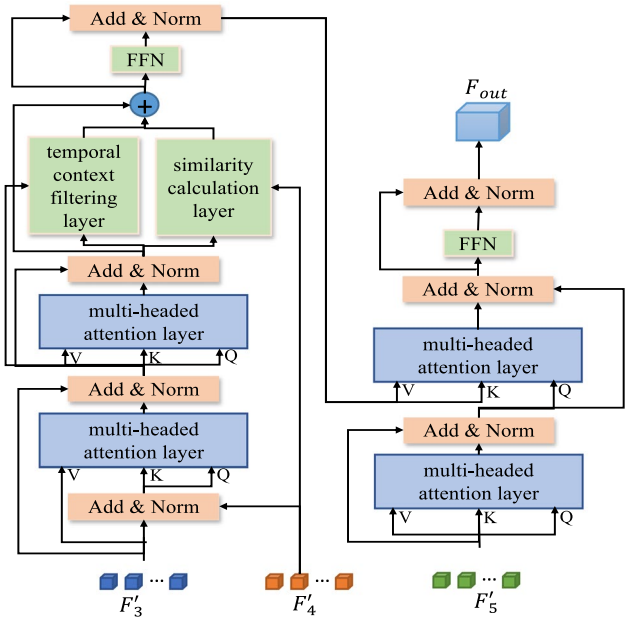


Fig. 3 Transformer feature integration network

operation to obtain the fused features F_3 and F_4. F_3 and F_4 are reshaped to obtain encoder inputs F'_3 and F'_4.

3.2.1 Encoder with Temporal Context Filtering Layer and Similarity Calculation Layer

In order to model the feature global context information more effectively, F'_3 and F'_4 are first normalized and summed; it is possible to fully and effectively utilize the global context information after two multi-headed self-attention layers. The normalization of F'_3 and F'_4 is shown in Eq. (9):

$$F_E^1 = \text{Norm}(F_3^{'} + F_4^{'}) \tag{9}$$

where F_E^1 is the output of feature vectors F'_3 and F'_4 after summation and normalization, and F_E^1 serves as input to the Q and K in the first layer of the multi-headed self-attention mechanism. F'_3 serves as the V, and Norm() is the normalization operation. The input and output through the first multi-headed self-attention mechanism layer is shown in Eq. (10):

$$F_E^2 = \text{Norm}(F_3^{'} + \text{Multi\_Head}(F_E^1, F_E^1, F_3^{'})) \tag{10}$$

where F_E^2 is the output of the first multi-headed self-attention mechanism in the first encoder. This output is used as the input of the second multi-headed self-attention mechanism to further enrich the global contextual information of the features. The input and output through the second multi-headed self-attention mechanism layer is shown in Eq. (11):

$$F_E^3 = \text{Norm}(F_E^2 + \text{Multi\_Head}(F_E^2, F_E^2, F_E^2)) \tag{11}$$

where F_E^3 is the output result of using the second multi-headed self-attention mechanism. In order to highlight the correlation between the input features F'_3 and F'_4 and focus on the part with higher correlation between them, this paper introduces a similarity calculation layer, which includes a global average pooling and FFN, and the calculation process of the similarity calculation layer is shown in Eqs. (12)-(14):

$$\text{Sign}\left(F_4^{'}\right) = \text{FFN}(\text{GAP}(F_4^{'})) \tag{12}$$

$$F_E^{Cs} = \text{Conv}(\text{Cat}(F_E^3, F_4^{'})) \tag{13}$$

$$F_E^S = F_E^{Cs} \times \text{Sign}\left(F_4^{'}\right) \tag{14}$$

where $\text{Sign}\left(F'_4\right)$ is the significant features obtained from the feature vector F'_4 after global average pooling and FFN. F_E^{Cs} is the output of the combination of the feature vector F'_4 and the output F_E^3 after two layers of self-attention mechanism. F_E^S is the output that highlights the significance of the correlation between F_E^{Cs} and $\text{Sign}\left(F'_4\right)$.

In the tracking task, there is not a direct temporal contextual link between all the features. Only the links between the tracking target features need to be highlighted in the tracking task, and the temporal contextual links between background and background do not need to be computed, this paper proposes a temporal context filtering layer to prevent unnecessary parameter computation. This layer automatically filters and removes unimportant temporal context information. The computation process of the temporal context filtering layer is shown in Eqs. (15)–(17):

$$\text{Filter}\left(F_E^2\right) = \text{FFN}(\text{GAP}(\text{Conv}(F_E^2))) \tag{15}$$

$$F_E^{Cf} = \text{Conv}(\text{Cat}(F_E^2, F_E^3)) \tag{16}$$

$$F_E^F = F_E^{Cf} \times \text{Filter}\left(F_E^2\right) \tag{17}$$

where $\text{Filter}\left(F_E^2\right)$ is the output of feature F_E^2 after convolution, global average pooling, and FFN. F_E^{Cf} is the output obtained from the interaction of feature information between feature F_E^2 and feature F_E^3.F_E^F is the output of temporal context filtering layer, which can effectively filter out the feature associations that do not need to be computed. The features F_E^S and F_E^F have gone through the similarity calculation layer and the temporal context filtering layer, which are fused with the output F_E^3 of the second multi-headed self-attention mechanism to obtain the final enhanced feature F_E^4. The computation process is shown in Eq. (18):

$$F_E^4 = F_E^3 + F_E^F + F_E^S \tag{18}$$

The transformer focuses on spatio-temporal and global context information. In order to make significant enhancement of both channel and spatial dimensions, this paper introduces a dual-attention module to enhance the features in both spatial and channel dimensions. The feature vector F_E^4 inputs to two branches of the dual-attention module after the FFN. One branch is used to compute the channel attention, which highlights the features by increasing the weights of the important features on the channel. Another branch is used to compute spatial attention, which serves the same way as channel attention. The features with global context information are further augmented on the channel and spatial to obtain the final output of the integrated feature F_E. The schematic diagram of the dual-attention module in FFN is shown in Fig. 4.

For the channel attention branch, a global average pooling operation is first used on the features to make the height (H) and width (W) become 1, and then the weights of different channels are obtained by two fully connected layers and sigmoid activation function. For the spatial attention branch, the fully connected layers are utilized to reduce the number of channels, and then in the middle of spatial attention, the local information is interacted with the global information using the cat function. Finally, the information from two branches is used to obtain the same features as the input feature layer using a broadcast-like mechanism. The dual-attention mechanism is realized in the formulas (19)–(22):

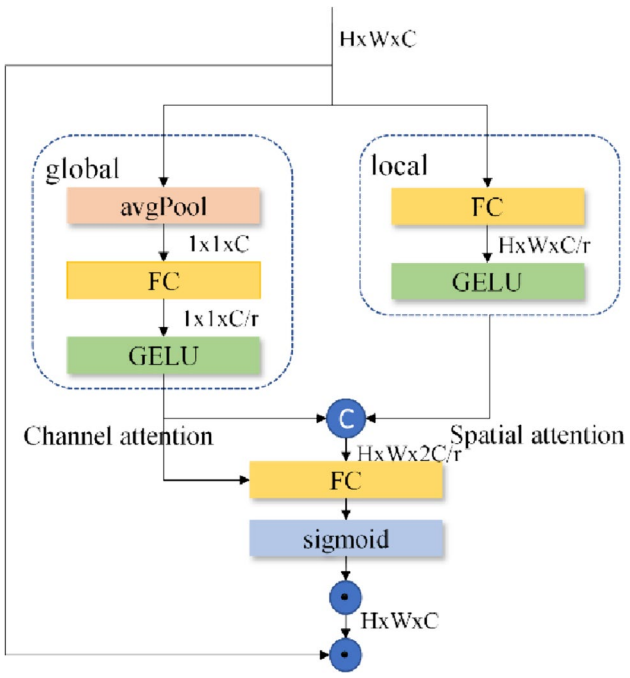


Fig. 4 Module diagram of the dual-attention module

$$F_E^4(c) = \mathrm{GELU}(\mathrm{Conv}(\mathrm{AVG}(F_E^4))) \tag{19}$$

$$F_E^4(s) = \mathrm{GELU}(\mathrm{Conv}(F_E^4)) \tag{20}$$

$$F_E^{4'} = \mathrm{Sigmoid}(\mathrm{Conv}(F_E^4(c)) \cdot \mathrm{Sigmoid}(\mathrm{Conv}(\mathrm{Cat}(F_E^4(c), F_E^4(s))))) \tag{21}$$

$$F_E = Norm(F_E^4 + F_E^4 \cdot F_E^{4'}) \tag{22}$$

where $F_E^4(c)$ is the output of global component, $F_E^4(s)$ the output of local component, $F_E^{4'}$ is the output of the dual-attention module, and F_E the output of the integrated features of the encoder.

It improves the representation ability of FFN and reduces the computational effort. GELU is an activation function that adjusts the output through a gating mechanism, applying the concept of stochastic regularity in nonlinear activation [22]. The GELU in both branches of Fig. 4 is adapt to the nonlinearity of network model, which can effectively avoid the gradient disappearing problem.

3.2.2 Decoder for Transformer Feature Integration Network

In the transformer feature integration network, a conventional decoder is used, and the initial input is F_5' obtained from the deep feature F_5 of the AlexNet network by dimensional reshaping. The F_5' is passed through a multi-headed self-attention layer and then interacts with the output F_E of the encoder to propagate useful temporal information and enhance feature representation. The input and output process of the first layer in the decoder is shown in Eq. (23):

$$F_D^1 = \mathrm{Norm}(F_5' + \mathrm{Multi\_Head}(F_5', F_5', F_5')) \tag{23}$$

where F_D^1 is the output of the first multi-headed self-attention layer, F_D^1 is used as the Q of the second multi-headed self-attention layer and the output F_E of the encoder as the K and V for the information interaction between the features, which is used to enrich the key information of the search features. The input and output process of the second layer in the decoder is shown in Eq. (24):

$$F_D^2 = \mathrm{Norm}(F_D^1 + \mathrm{Multi\_Head}(F_D^1, F_E, F_E)) \tag{24}$$

where F_D^2 is the output of the second multi-headed self-attention layer, which is passed through the FFN to obtain the final feature vector F_{out} for the classification and regression network, the computation process is shown in Eq. (25):

$$F_{\mathrm{out}} = \mathrm{Norm}(F_D^2 + \mathrm{FFN}(F_D^2)) \tag{25}$$

where F_{out} is the final output of the decoder.

3.3 The Overall Algorithm Implementation

Throughout the implementation of the tracking algorithm, the final tracking algorithm model is obtained by offline training using the publicly available datasets VID [23], LaSOT [24], GOT-10k [25], and COCO [26]. The target state of the tracked target is first determined in the initial template frame, and then the trained model is used to predict the state of the target in the subsequent frames. The specific process is to pass the template image and the search image through the feature extraction network to obtain the input of the transformer feature integration network encoder and decoder. In the encoder, the temporal context filtering layer adaptively filters out unimportant feature information, while the similarity calculation layer highlights relevant feature. The FFN employs a dual-attention mechanism to enhance features in the channel and spatial dimensions. This produces feature vectors that integrate spatio-temporal and global contextual information. The final output features are obtained by interacting the template information with the information of the search branch through a decoder and propagating the temporal context information. The output features are sent to the classification regression network to complete the target localization and tracking. During the target-tracking process, the video frames are read sequentially. After completing the image normalization operation, the feature extraction process is carried out, and then the tracking is realized.

The specific implementation process is as follows.

Input: Selecting an image pair in the video sequence, the template image is preprocessed to size 127×127, and the search image size is 287×287.

Output: the target state of the tracking target in the subsequent frames.

Training network models:

(1) The designed object-tracking network is trained offline on four publicly available datasets, VID [23], LaSOT [24], GOT-10k [25], and COCO [26].
(2) The AlexNet network is used as a feature extraction network for extracting the feature vectors of the template images and search images.
(3) Classification loss using cross-entropy loss and binary cross-entropy loss and regression loss using IOU loss.
(4) The SGD optimizer is used to optimize the network model, and the initial learning rate is set to 5×10^{-4} and the weight decay is set to 10^{-4}.

Object tracking:

(1) Image pairs are extracted from the test dataset as template image z and search image x, respectively.
(2) Feature extraction of template images and search images using a feature extraction network to obtain features F_3, F_4, F_5 by Eqs. (1)–(5).
(3) F_3, F_4 are used as the input to the encoder in transformer feature integration network, and pass through two layers of multi-headed self-attention layer, temporal context filtering layer, similarity calculation layer, and FFN with dual-attention mechanism, respectively, then the output F_E of the encoder is obtained by Eqs. (6)–(22).
(4) The F_5 passes dimensional reshaping and interacts with F_E for information interaction in the decoder of transformer feature integration network, and finally the feature vector F_{out} that integrates global context and spatio-temporal information.
(5) F_{out} is fed to the classification and regression network to get the target state of the tracking target in the subsequent frames.

4 Experimental Results and Analysis

4.1 Experimental Details and Evaluation Criteria

In order to verify the effectiveness of the tracking algorithm proposed in this paper, image pairs were extracted from four publicly available large datasets, VID [23], LaSOT [24], GOT-10K [25], and COCO [26], and the template image was set to 127×127 and the search image was set to 287×287, and the whole tracking algorithm was trained for 70 epochs. During the training process, the last three layers of AlexNet were fine-tuned in the last 60 epochs, while the first two layers were frozen. In addition, the learning rate was initialized to 5×10^{-4}, the batch size was set to 220, the weight decay and momentum were set to 10^{-4} and 0.9, respectively, and optimized using the gradient descent method SGD optimizer.

In order to objectively evaluate the performance of the proposed tracking algorithm, this paper uses a One-Pass Evaluation (OPE) to evaluate tracking performance, including precision and success rate. Specifically, precision is calculated by comparing the distance (in pixels) between the tracking results and the real bounding box, and different trackers are ranked according to a threshold (20 pixels) The obtained precision is normalized to reduce the sensitivity of the precision metric to the target size and image resolution. The tracking algorithms are ranked between 0 and 0.5 using the area under the curve (AUC) under the normalized precision metric. The success rate was calculated using the Intersection over Union (IOU).

4.2 Comparison with Typical Trackers

In order to comprehensively analyze the overall performance of the proposed tracker under the interference of complex background and other factors, the tracking algorithm are evaluated on three public authoritative benchmark datasets OTB100 [27], LaSOT [24], GOT-10k [25], four well-known aerial tracking benchmark datasets DTB70 [28], UAV123 [29], UAV@10fps [29], and UAV20L [29] are evaluated on typical tracking algorithms, and the performance of the tracking algorithms was analyzed from multiple perspectives. A comprehensive evaluation and comparison are conducted in 55 existing typical tracking algorithms and results of the compared algorithms are obtained from the top session paper.

4.2.1 Comparative Analysis with Typical Tracking Algorithms on the OTB100 Dataset

The OTB100 dataset has a total of 98 videos and 100 test scenarios and contains a total of 11 tracking challenge factors, a fair and accurate evaluation of various tracking algorithms can be compared. On the OTB100 dataset, this paper compares and analyzes the performance of typical trackers using two evaluation metrics, precision and success rate.

The trackers compared on the OTB100 dataset include SRDCF [30], Staple [31], MEEM [32], CFNet [33], MUSTER [34], SiamFC [1], DSST [35], and Struck [36], totaling 8 types. Figure 5 shows the precision and success rate plots of the 9 typical trackers on the OTB100 dataset.

As can be seen from Fig. 5, the proposed tracking algorithm significantly outperforms other typical tracking algorithms such as SiamFC in terms of precision and success rate. This is due to the fact that SiamFC uses a simple similarity calculation to estimate the position of the target, making it difficult to accurately track the target in environments such as fast moving.

CFNet uses correlation filters as an update module and integrates it with CNN, and the performance is still lacking compared to typical tracking algorithms, because correlation filtering is difficult to deal with boundary effects. Thanks to the newly designed transformer feature integration network, the proposed tracking algorithm improves 7.7% in precision and 5.8% in success rate compared to the typical Siamese tracking algorithm SiamFC.

4.2.2 Comparative Analysis with Typical Tracking Algorithms on LaSOT Dataset

The LaSOT dataset contains 1400 video sequences totaling over 3.5 M frames. It provides not only visual bounding box annotations, but also rich natural language specifications, LaSOT dataset is more challenging than previous datasets and all training is long-term. It is a large-scale, high-quality dedicated benchmark dataset for object-tracking training and evaluation of tracking algorithms. Performance is evaluated using two evaluation metrics, the normalized precision plot and the success plot.

The tracker compared on the LaSOT dataset include SiamMask [37], C-RPN [5], SiamDW [4], VITAL [38], SPLT [39], MDNet [40], and D3S [41]. Figure 6 shows the result of normalized precision plot and success plot with other nine typical trackers on the LaSOT dataset.

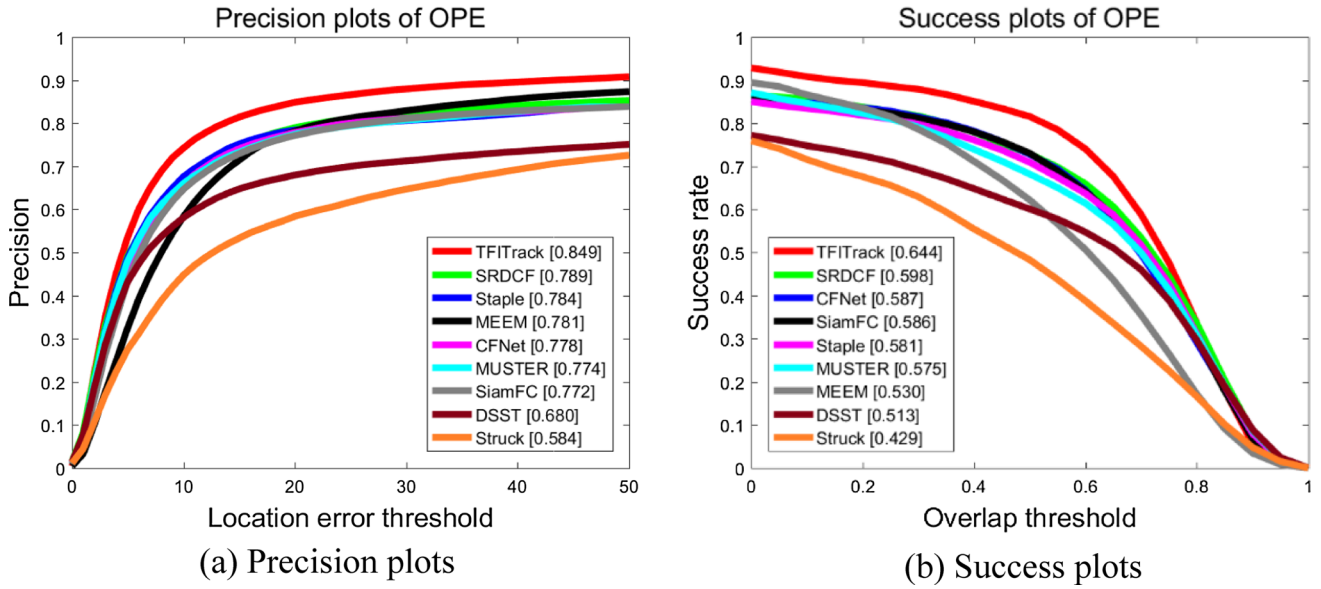
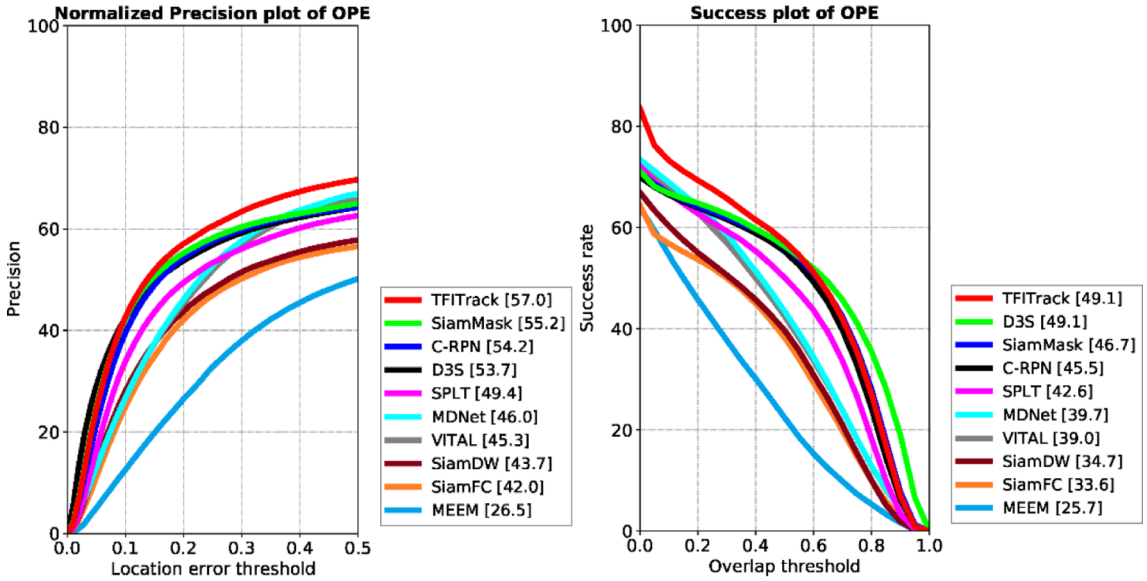


Fig. 5 Precision and success plots of the tracking algorithm on the OTB100 dataset



(a) Normalized Precision plots (b) Success plots

Fig. 6 Normalized precision and success plots of the tracking algorithm on the LaSOT dataset

As can be seen in Fig. 6, the proposed tracking algorithm achieves better tracking performance on the LaSOT dataset with the AUC score (49.1%) and the precision score (57.0%), which surpass the best results of the typical tracking algorithms.

The SiamMask method only considers the appearance features of the current frame and can hardly use global context and spatio-temporal information. This makes it difficult to distinguish similar interference. Meanwhile, the SiamMask tracking algorithm sacrifices precision for speed. Therefore, the tracking precision needs to be improved. C-RPN networks are prone to model drift because the link of key information between features is not considered in the tracking process.

The tracking algorithm proposed in this paper benefits from the designed transformer feature integration network, which can effectively construct long-term dependencies between features and global contextual information ignored by CNN. Therefore, the tracking algorithm proposed in this paper can obtain better robustness on the long-time dataset LaSOT.

4.2.3 Comparative Analysis with Typical Tracking Algorithms on the GOT-10k Dataset

The GOT-10k dataset provides more than 10,000 video and 1.5 million manually labeled bounding boxes, which has the characteristics of rich scenario and difficult algorithmic challenges, and realizes the uniform training and stable evaluation of deep trackers. Average overlap (AO) and success rate (SR) are two evaluation methods for the GOT-10k dataset. Success rate is the precision of successful tracking at a certain AO threshold, and take two thresholds of 0.5 and 0.7. The tracking results obtained from the test are uploaded to the official website to get the performance results of the test, which reflects the fairness and effectiveness of the tracking algorithm in this paper.

The trackers compared on the GOT-10k dataset include SPM [42], GOTURN [43], CCOT [44], ECO [45], and others. Figure 7 shows the success rate plots of 9 typical trackers on the GOT-10k dataset.

As can be seen in Fig. 7, the proposed tracking algorithm significantly outperforms other tracking algorithms on the GOT-10k dataset. To better reflect the performance of the proposed tracking algorithm on the GOT-10k dataset, Table 1 illustrates the average overlap rate and success rate thresholds of 0.5 and 0.75 for the tracking algorithm in detail.

Table 1 shows the performance analysis of the tracking algorithm proposed in this paper compared with typical tracking algorithms on the GOT-10K test dataset, including specific data for AO, SR0.5, and SR0.75. The comparative tracking algorithms include: SiamRPN + + and SiamRPN tracking algorithms, which introduce RPN networks in the detection domain, SPM tracking algorithm combines coarse matching and fine matching, THOR and ECO tracking algorithms are with update mechanism. GOTURN tracking algorithm is for offline learning neural networks, and MDNet tracking algorithm is for multi-domain learning model.

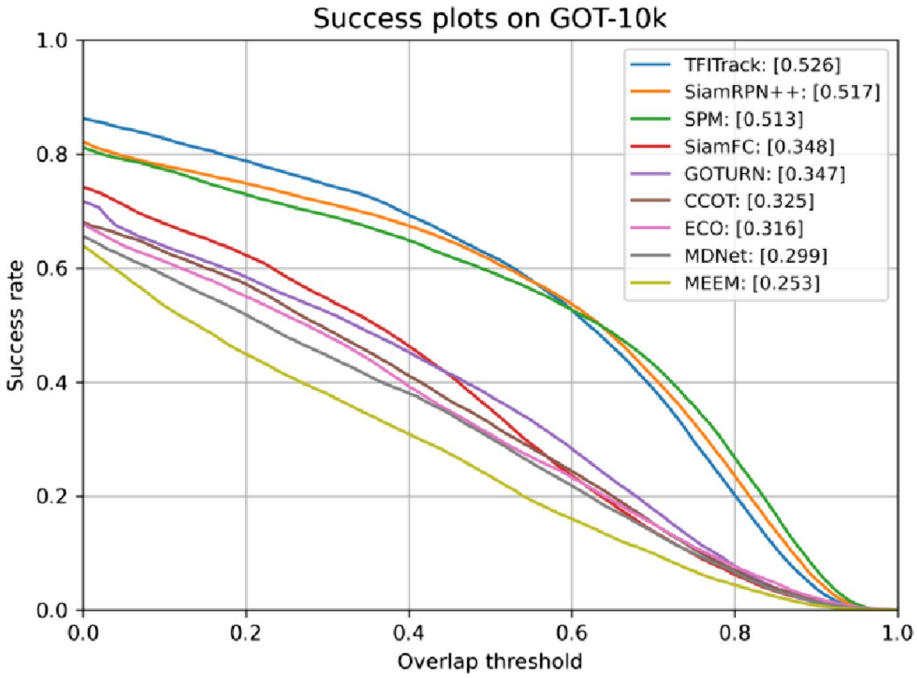


Fig. 7 Comparison results of GOT-10k dataset

Table 1 Performance comparison of typical trackers on the GOT-10k dataset

Methods	AO	SR0.5	SR0.75
TFITrack	0.526	0.622	0.304
SiamRPN + + [3]	0.517	0.616	0.325
SPM [42]	0.513	0.593	0.359
SiamRPN [2]	0.463	0.549	0.253
THOR [46]	0.447	0.538	0.204
SiamFCv2 [1]	0.374	0.404	0.144
SiamFC [1]	0.348	0.353	0.098
GOTURN [43]	0.347	0.375	0.124
ECO [45]	0.316	0.309	0.111
MDNet [40]	0.299	0.303	0.099

As can be seen from Table 1, the tracking algorithm proposed in this paper achieves better tracking performance on the AO, SR (0.5), and SR (0.75) evaluation metrics on the GOT-10k dataset. Compared with SiamFC, AO, SR (0.5), and SR (0.75) are improved by 17.8%, 29.6%, and 20.6%, respectively. Compared with SiamRPN, AO, SR (0.5), and SR (0.75) improved by 6.3%, 7.3%, and 5.1%, respectively.

SiamFC considers tracking as a matching problem, where similarity is computed on the outputs of the upper and lower branches of the Siamese network and the location of the target is estimated. This method is simple and straightforward, it is difficult to accurately localize the tracking target due to insufficient representation of the extracted features, and the similarity calculation is simple. Therefore, the performance needs to be improved compared with other tracking. SiamRPN and SiamRPN + + use the traditional anchor generation mechanism, which has predefined anchor frames and cannot adapt to the fast motion and motion blur tracking scenario. Tracking drift is likely to occur and even tracking target failure when tracking target with fast motion.

Due to the design of transformer feature integration network, the tracking algorithm proposed in this paper considers both the global contextual information and spatio-temporal information and the long-term dependencies between the features using encoder. Therefore, the tracking algorithm proposed in this paper has better performance on the GOT-10k dataset compared with other typical tracking algorithms.

4.2.4 Comparative Analysis with Typical Tracking Algorithms on the DTB70 Dataset

The DTB70 dataset is a high-diversity benchmark video dataset containing a total of 70 video sequences of short-term and long-term aerial targets in a variety of challenging scenarios, which can be used to study motion modeling in the field of visual tracking. The robustness of the tracking algorithm proposed in this paper can effectively evaluate when dealing with motion.

The trackers compared on the DTB70 dataset include 27 typical trackers such as HiFT, SiamRPN + +, SiamAPN + +, SiamAPN, CCOT, DeepSTRCF, MCCT, DaSiamRPN, and others. Table 2 shows the precision and success rate of these typical trackers on the DTB70 dataset.

The precision and success rate of the proposed tracking algorithm on the DTB70 dataset are improved by 1.1% compared to the HiFT tracking algorithm. Compared with

Table 2 Precision and success rates of typical trackers on the DTB70 dataset

Methods	Precision	Success	Methods	Precision	Success
TFITrack	0.813	0.605	IBCCF[55]	0.669	0.460
HiFT [9]	0.802	0.594	MCPF [56]	0.664	0.433
SiamRPN++[3]	0.795	0.589	UDT+[57]	0.658	0.462
SiamAPN++[8]	0.789	0.594	STRCF [58]	0.649	0.437
SiamAPN [8]	0.784	0.586	ECO-HC [45]	0.643	0.453
CCOT [44]	0.769	0.517	CF2 [49]	0.616	0.415
DeepSTRCF [58]	0.734	0.506	UDT [57]	0.602	0.422
MCCT [50]	0.725	0.484	CoKCF [59]	0.599	0.378
ECO [45]	0.722	0.502	BACF [47]	0.590	0.402
SiamFC [1]	0.719	0.483	fDSST [48]	0.534	0.357
AutoTrack [51]	0.716	0.478	SRDCF [30]	0.512	0.363
ARCF [52]	0.694	0.472	DSiam [60]	0.495	0.337
DaSiamRPN [53]	0.694	0.472	DSST [35]	0.463	0.276
TADT [54]	0.693	0.464	Staple [31]	0.365	0.265

SiamAPN++and SiamRPN++, the precision is improved by 2.4% and 1.8%, and the success rate is improved by 1.9% and 1.6%, respectively.

HiFT and SiamAPN++are tracking algorithms proposed for the UAV domain. HiFT combines shallow and deep semantic information by extracting features from a benchmark network to make full use of the valuable feature information. At the same time, the transformer network is used to enhance the feature representation. SiamAPN++uses similar self-attention and cross-attention in transformer network to enhance feature representation.

Different from these two approaches, this paper uses two multi-headed self-attention layers in transformer encoder to enhance the key information dependencies between features. Meanwhile, a dual-attention mechanism is introduced in FFN for enhancing the channel and spatial information ignored by FFN, and it further obtains feature vectors with higher representation.

SiamRPN++, CCOT, and DASiamRPN are conventional target trackers, and it can be seen from Table 2 that the proposed method shows superior tracking performance both to conventional trackers and trackers designed for UAV. This also proves that the tracker proposed in this paper can be used not only for conventional object tracking, but also for UAV object tracking in the aerial domain.

4.2.5 Comparative Analysis with Typical Tracking Algorithms on UAV Dataset

UAV123 contains 123 fully labeled high definition video datasets and benchmarks captured from low-altitude aerial views, it consists 91 UAV videos with several longer video sequences that are split into three or four shorter videos, so there are 123 ground truth. UAV20L is a long-term tracking benchmark in the UAV123 dataset, which contains 20 long-time video sequences with an average length of more than 2900 frames, and has high authority in the performance evaluation dataset.

The trackers compared on the UAV dataset include typical trackers such as HiFT [9], SiamRPN [2], and DaSiamRPN [53]. Figures 8 and 9 show the results of the precision and success rate of the tracking algorithm proposed in this paper compared with 8 and 9 typical trackers on the UAV123 and UAV20L datasets, respectively.

As can be seen from Fig. 8, the tracking algorithm in this paper has the same precision as HiFT on the UAV123 dataset, but the success rate is improved by 0.5% compared to HiFT, which reflects that the tracking algorithm in this paper frames the target more accurately. Compared with DASiamRPN and SiamRPN of multi-scale action, the precision is improved by 0.6% and 1.5%, and the success rate is improved by 2.5% and 1.2%, respectively.

As can be seen from Fig. 9, the tracking algorithm in this paper is compared to the tracking algorithms HiFT and SiamAPN++designed for UAV on the UAV20L dataset, the precision is improved by 4.3% and 7%, and the success rate is improved by 4.6% and 5.2%, respectively. UAV20L is a long-time tracking dataset, the results from Fig. 9 show the performance of the proposed tracking algorithm on the long-time dataset. The proposed tracking algorithm still maintains a good tracking performance on the long-time dataset.

Figure 10a–l shows the precision and success rate results of this tracker compared to a typical tracker on 12 challenge attributes for the UAV123 dataset. 12 attributes include scale variation (SV), aspect ratio change (ARC), low resolution (LR), fast motion (FM), full occlusion (FOC), partial occlusion (POC), out-of-view (OV), background clutter (BC), low resolution (LR), illumination variation (IV), viewpoint change (VC), camera motion (CM), and similar object (SOB).

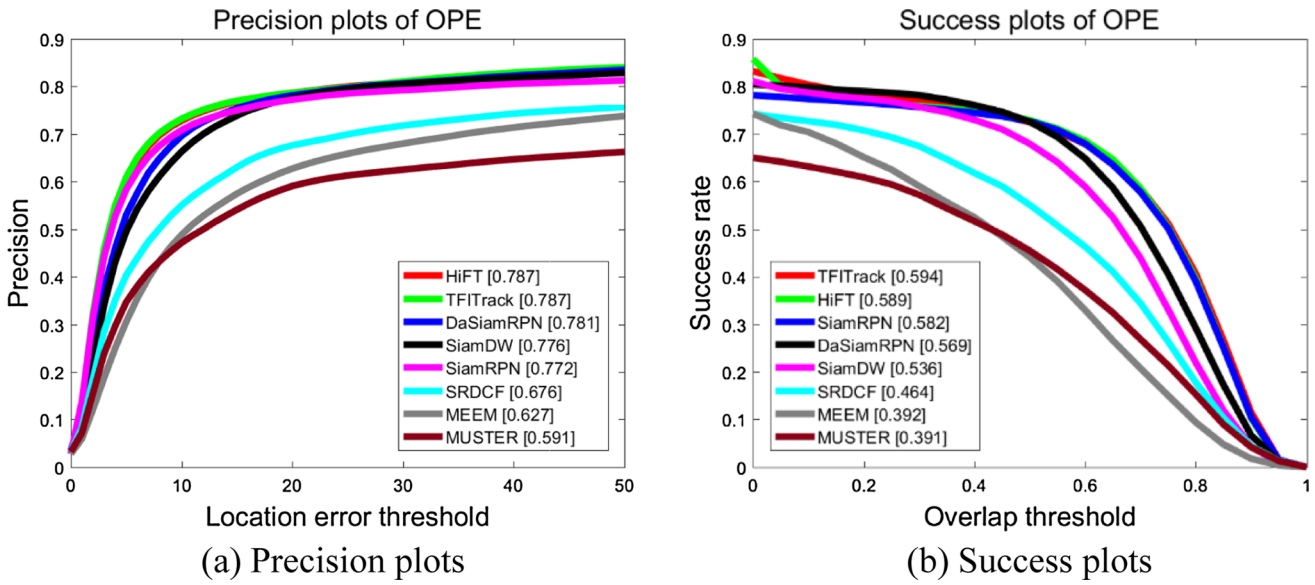


Fig. 8 Precision and success plots of the tracking algorithm on the UAV123 dataset

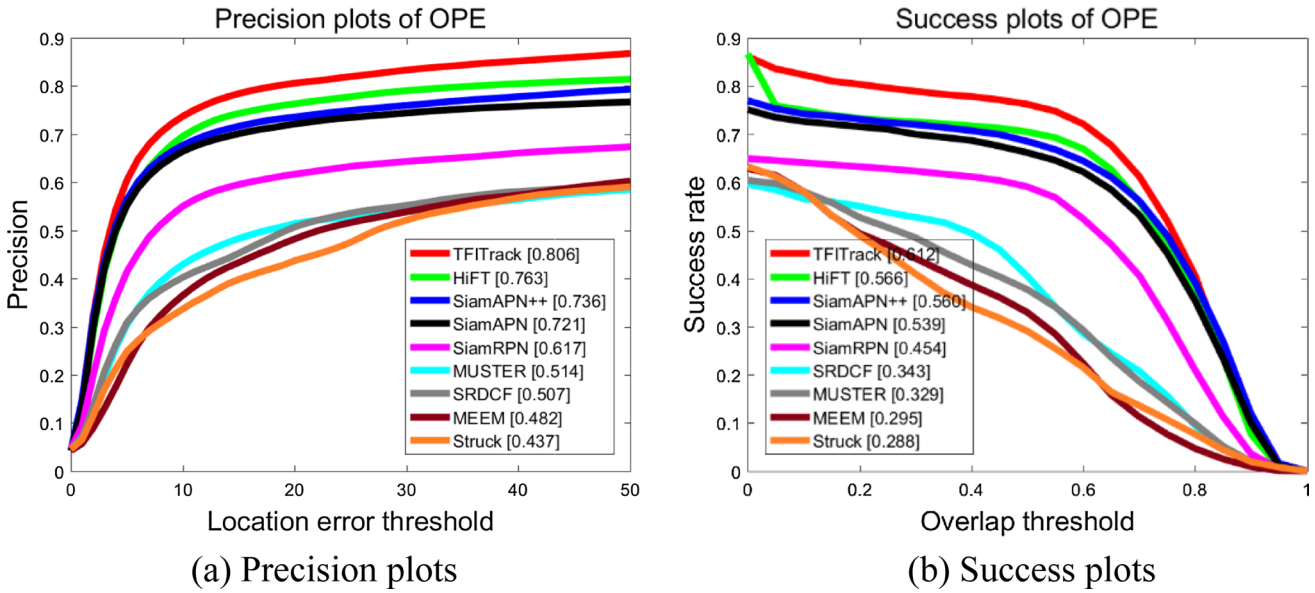
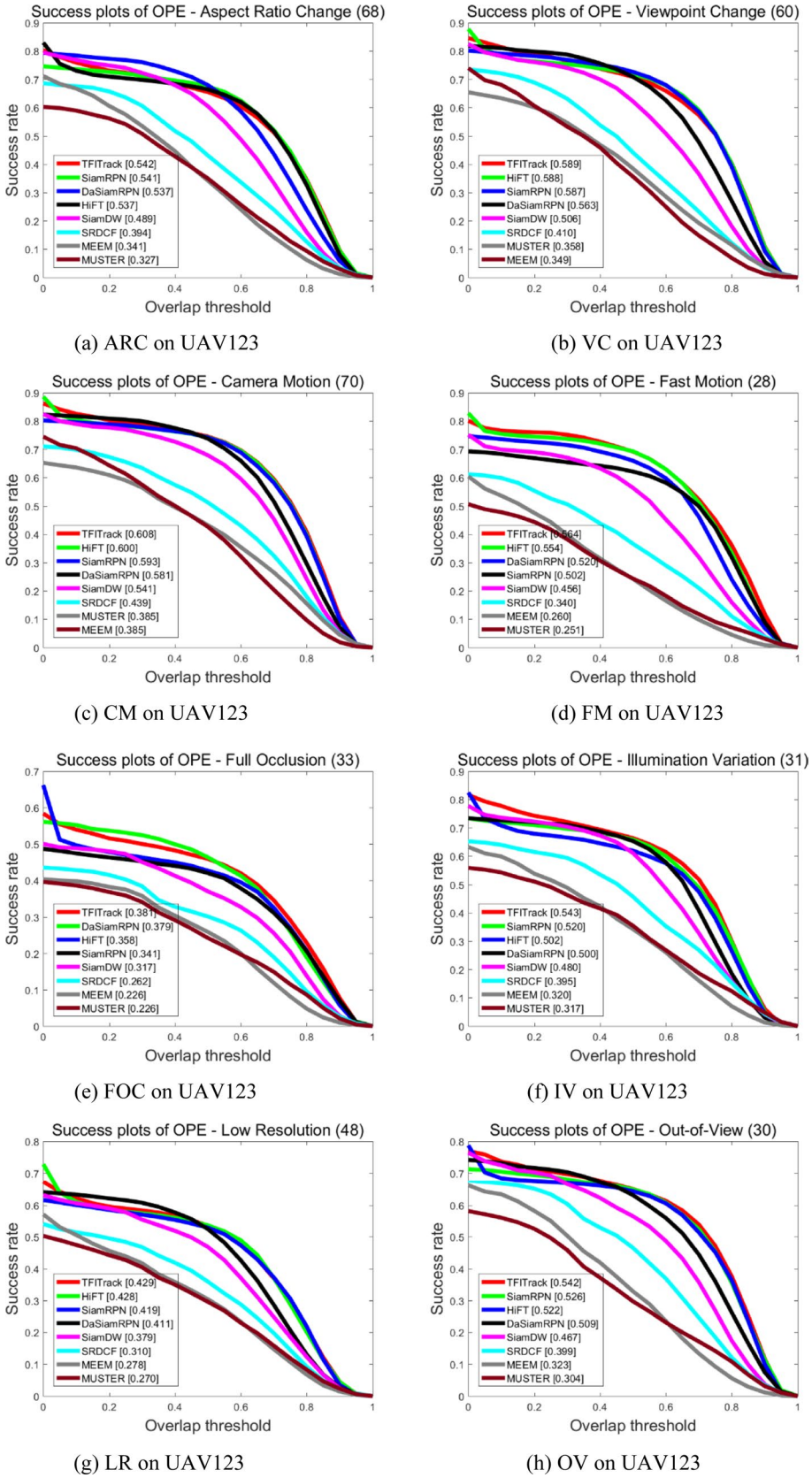


Fig. 9 Precision and success plots of the tracking algorithm on the UAV20L dataset

As can be seen from Fig. 10a–j, the tracking algorithm proposed in this paper improves the localization precision of features due to make full use of both shallow and deep information. Meanwhile, similarity calculation layer and dual-attention mechanism are introduced in the encoder of the transformer feature integration network. It enables the tracking algorithm to achieve better performance on the UAV123 dataset with scale variation, aspect ratio change, low resolution, fast motion, full occlusion, partial occlusion, out-of-view, illumination variation, viewpoint change, and camera motion.

As can be seen from Fig. 10k and l, DaSiamRPN constructs semantically negative samples that enrich the difficult negative sample data and allow the network to learn discriminative abilities. Therefore, it has better tracking performance under the challenging factors of similarity and background clutter. However, the performance is slightly lower than the tracking algorithm proposed in this paper under full occlusion, partial occlusion, and fast motion due to ignore the dependencies between features.

UAV123@10fps was created by downsampling from the original 30FPS recording. Therefore, the strong motion



(a) ARC on UAV123 (b) VC on UAV123

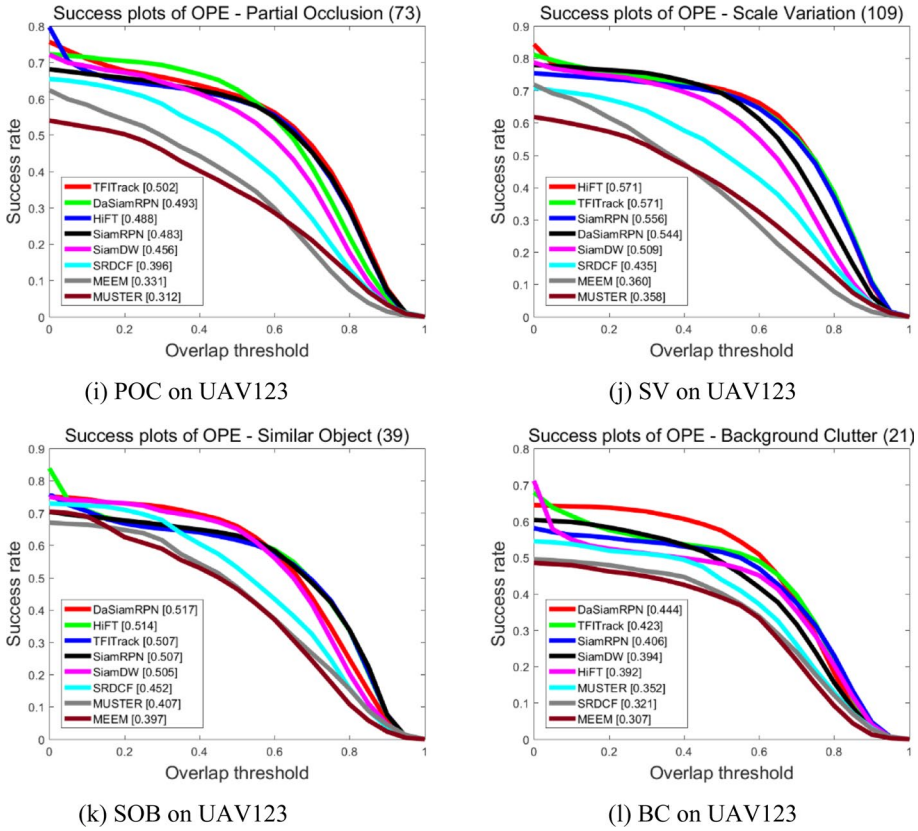
(c) CM on UAV123 (d) FM on UAV123

(e) FOC on UAV123 (f) IV on UAV123

(g) LR on UAV123 (h) OV on UAV123

Fig. 10 Performance evaluation results for different attributes of the UAV123 dataset

Fig. 10 (continued)

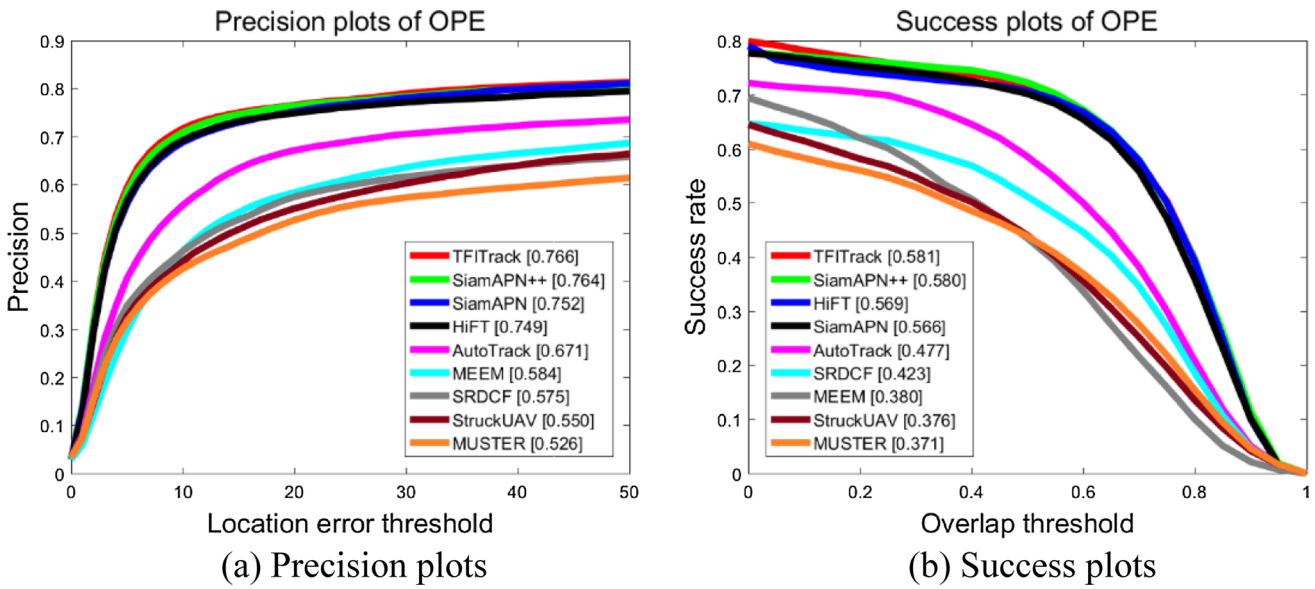


(i) POC on UAV123 (j) SV on UAV123

(k) SOB on UAV123 (l) BC on UAV123

problem is more severe in UAV123@10fps compared to UAV123, which greatly improves the difficulty of tracking. It is clear that the proposed tracking algorithm maintains superior robustness from the comparison with other typical trackers. Figure 11 shows the results of the precision and success rate of the proposed tracking algorithm compared with 18 typical trackers on the UAV123@10fps dataset.

UAV123@10fps is a dataset for the strong motion problem, testing on this dataset and improving the tracking difficulty. As can be seen from Fig. 11, the precision and success



(a) Precision plots (b) Success plots

Fig. 11 Precision and success plots of the tracking algorithm on the UAV123@10fps dataset

rate of the tracking algorithm proposed in this paper reach 76.6 and 58.1 on the UAV123@10fps dataset. Compared with HiFT and SiamAPN++, the precision has improved by 1.7% and 0.2%, respectively, while the success rate has improved by 1.2% and 0.1%, respectively. It shows the high robustness of the tracker when it encounters strong motion scenario.

When combined with the DTB70 data results, it is further demonstrated that the tracking algorithm proposed in this paper improves the localization precision of features due to make full use of shallow and deep information. Meanwhile, in the encoder of transformer feature integration network, the correlation between features is further highlighted using the similarity calculation layer, and the dual-attention mechanism is introduced in the FFN to correct the channel and spatial features, which enriches the channel and spatial feature information ignored by transformer. The newly proposed tracking method can adapt conventional object tracking and domain-specific object tracking.

4.3 Qualitative Analysis of the Tracking Algorithm

To further validate the performance of the newly proposed tracking algorithm, it displays the qualitative evaluation results with 7 typical tracking method including HiFT [9], DaSiamRPN [53], SiamDW [4], SiamRPN [2], MEEM [32], MUSTER [34], and SRDCF [30].The comparison results of 3 video sequences are selected for analysis on the UAV123 dataset, namely bike1_1, boat9_1, and group1_2_1. The visualization tracking results can be seen from Figs. 12, 13, 14.

Figure 12 shows that the trackers for all comparisons frame the target accurately at frame 11. However, the SiamRPN and SRDCF tracker undergoes tracking drift and gradually moves away from the target in frame 268. At frames 858, the tracking algorithms SiamDW and MUSTER failed to track. On the contrary, thanks to the constructed transformer feature integration network, the newly designed tracking method exhibits much better tracking results.

The videos in Fig. 13 have less information about the tracked target, and the whole image is background information. It is challenging for the tracking algorithm, which can easily lead to tracking failure if the foreground and background are not correctly determined. As the target continues to move into the distance, the HiFT tracker frames out too much background information at frame 163 and definitely fails to track at frame 720. The tracking scheme designed in the paper is due to the combination of Transformer's rich spatio-temporal context, which is used to construct long-time dependencies between the features. It enables the algorithm in this paper to still find the location of the target and contain little contextual information.

As can be seen from Fig. 14, at the beginning, the trackers compared can track the target more accurately. At frame 1759 and 2407, the SiamDW and SRDCF tracker tracks

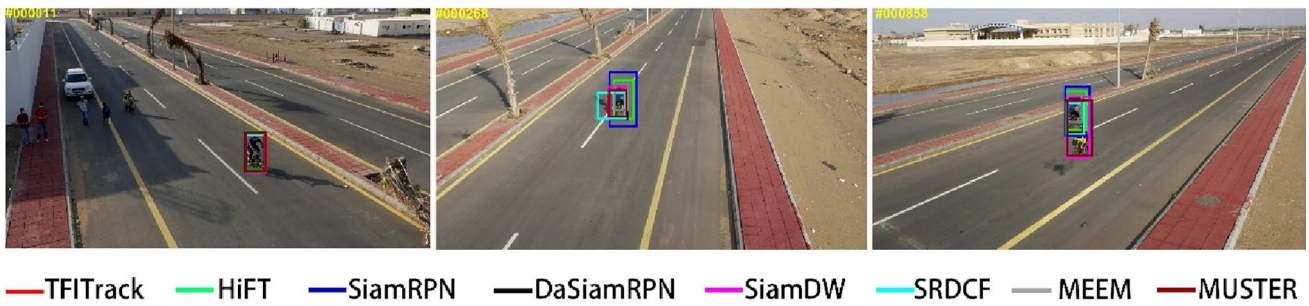


Fig. 12 Visualization of bike1_1 tracking results in UAV123 dataset

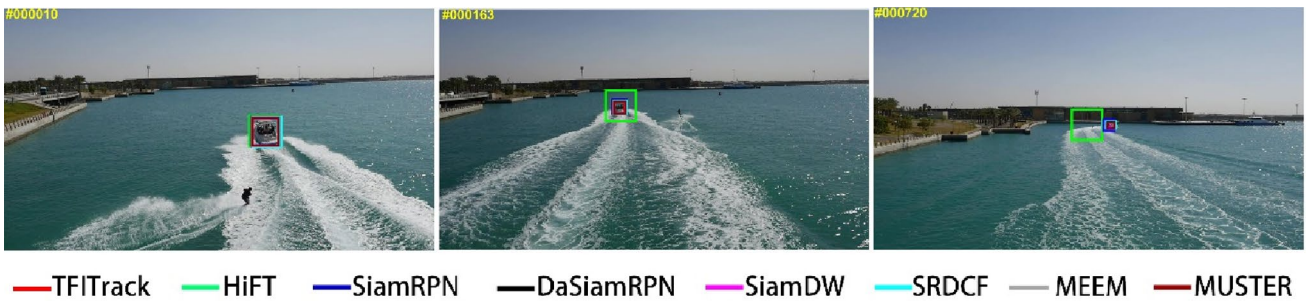


Fig. 13 Visualization of boat9_1 tracking results in UAV123 dataset

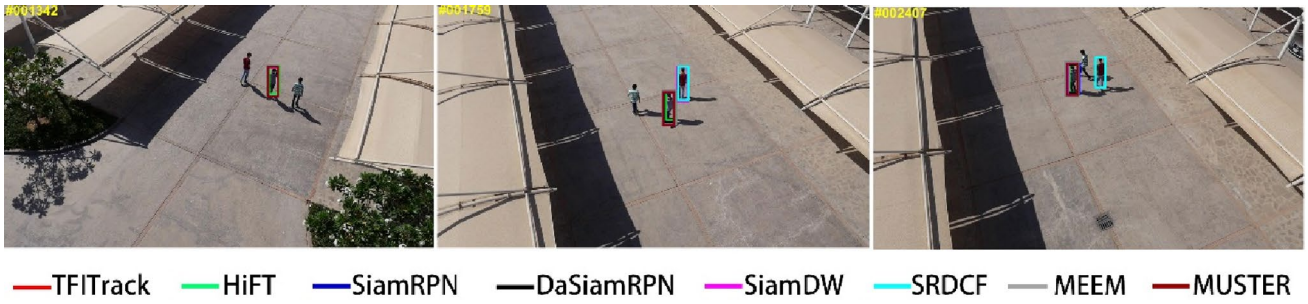


Fig. 14 Visualization of group1_2_1 tracking results in UAV123 dataset

other objects due to the interference of similar objects. The proposed new method can show better robustness when the tracker encounters similar object interference due to the benefit of spatio-temporal and contextual information.

5 Conclusion

In this paper, a tracking algorithm based on transformer feature integration network is proposed. First, AlexNet is used as the feature extraction network to pre-extract features from the template image and the search image to get the input of transformer feature integration network. Transformer feature integration network contains an encoder and a decoder, this paper introduces temporal context filtering layer, similarity calculation layer and dual-attention module in the encoder. The temporal context filtering layer is used for adaptively filtering the unimportant feature information, reducing the amount of parameter calculations and improving the efficiency of the tracker. The similarity calculation layer is used to enhance the correlation between different layers and enhance the feature representation. The dual-attention module can enhance the channel and spatial information ignored in FFN to obtain feature vectors with stronger feature representation ability. It is combined with transformer for constructing dependencies within and between features, which enhances the global attention of features, enriches the global context and spatio-temporal information, and obtains integrated feature vectors with stronger representational ability, it can be used for subsequent localization tracking with classification regression networks. The proposed tracking algorithm shows a significant improvement in precision and success rate compared to the state-of-the-art trackers on seven authoritative test datasets, which demonstrates the feasibility of the algorithm in conventional tracking targets and aerial scenarios. Although current research methods have produced positive evaluation results for testing typical public datasets, there are still numerous challenges to overcome in the future. Given the practical complexities of the application requirements, and the limitations of existing methods due to the lack of labeled data and model generalization ability, future research will focus on model training dataset labeling augmentation and model lightweight design.

Author Contributions Conceptualization, X.H.; methodology, X.H.; supervision, X.H.; writing—review and editing, X.H. and H.L.; software, H.L.; writing—original draft preparation, H.L.; resources, S.L.; formal analysis, J.Z.; project administration, Y.H. All the authors have read and agreed to the published version of the manuscript.

Funding This work was supported by the Natural Science Basic Research Project of Shaanxi Provincial Department of Science and Technology under Grant 2022JQ-677.

Availability of Data and Materials The data sets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of Interest The authors declare that they have no competing interests.

Ethical Approval and Consent to Participate Not applicable.

Consent for Publication Not applicable.

References

1. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: Proceedings of the 2016 European Conference on Computer

Vision, ECCV 2016, Amsterdam, The Netherlands, October 8–16, 2016, pp. 850–865 (2016).https://doi.org/10.48550/arXiv.1606.09549
2. Li, B., Yan, J., Wu, W., Zhu, Z., Hu, X.: High performance visual tracking with Siamese region proposal network. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt, Lake, City, UT, USA, June18–22, 2018, pp. 8971–8980 (2018). https://doi.org/10.1109/CVPR.2018.00935
3. Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., Yan, J.: Siamrpn++: Evolution of Siamese visual tracking with very deep networks. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June16–20, 2019, pp. 4282–4291 (2019). https://doi.org/10.48550/arXiv.1812.11703
4. Zhang, Z., Peng, H.: Deeper and wider Siamese networks for real-time visual tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 15–20, 2019, pp. 4591–4600 (2019). https://doi.org/10.1109/CVPR.2019.00472
5. Fan, H., Ling, H.: Siamese cascaded region proposal networks for real-time visual tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018, Long Beach, CA, USA, June 15–20, 2018, pp. 7952–7961 (2019). https://doi.org/10.1109/CVPR.2019.00814
6. Chen, H., Lin, M., Liu, J., Yang, H., Zhang, C., Xu, Z.: NT-DPTC: a non-negative temporal dimension preserved tensor completion model for missing traffic data imputation. Inf. Sci. **653**, 119797 (2024). https://doi.org/10.1016/j.ins.2023.119797
7. Yu, Y., Xiong, Y., Huang, W., Scott, M.R.: Deformable Siamese attention networks for visual object tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June13–19, 2020, pp. 6728–6737 (2020). https://doi.org/10.1109/CVPR42600.2020.00676
8. Cao, Z., Fu, C., Ye, J., Li, B., Li, Y.: SiamAPN++: Siamese attentional aggregation network for real-time UAV tracking. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Prague, Czech Republic, September 27–October 01, 2021, pp. 3086–3092 (2021). https://doi.org/10.1109/IROS51168.2021.9636309
9. Cao, Z., Fu, C., Ye, J., Li, B., Li, Y.: HiFT: Hierarchical Feature Transformer for Aerial Tracking. In: IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10–17, 2021, pp. 15437–15446 (2021). https://doi.org/10.1109/ICCV48922.2021.01517
10. Hu, L., Wang, Z., Li, H., Wu, P., Mao, J., Zeng, N.: ℓ-DARTS: Light-weight differentiable architecture search with robustness enhancement strategy. Knowl. Based. Syst. **288**, 111466 (2024). https://doi.org/10.1016/j.knosys.2024.111466
11. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, December 4–9, 2017, pp. 5998–6008 (2017). https://doi.org/10.48550/arXiv.1706.03762
12. Chen, X., Yan, B., Zhu, J., Wang, D., Yang, X., Lu. H.: Transformer tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2021, Nashville, TN, USA, June20–25, 2021, pp. 8122–8131 (2021). https://doi.org/10.1109/CVPR46437.2021.00803
13. Fan, L., Kim, P.: Dual Siamese anchor points adaptive tracker with transformer for RGBT tracking. Int J. Comput. Intell. Syst. **16**, 1–18 (2023). https://doi.org/10.1007/s44196-023-00360-0
14. Yan, B., Peng, H., Fu, J., Wang, D., Lu, H.: Learning spatio-temporal transformer for visual tracking. In: IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10–17, 2021, pp. 10428–10437 (2021). https://doi.org/10.48550/arXiv.2103.17154
15. Cao, Z., Huang, Z., Pan, L., Zhang, S., Liu, Z., Fu, C.: TCTrack: Temporal Contexts for Aerial Tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June18–24, 2022, pp. 14778–14788 (2022). https://doi.org/10.1109/CVPR52688.2022.01438
16. Blatter, P., Kanakis, M., Danelljan, M., Gool, L.V.: Efficient Visual Tracking with Exemplar Transformers (2021). https://doi.org/10.48550/arXiv.2112.09686
17. Huang, T., Huang, L., You, S., Wang, F., Qian, C., Xu, C.: LightViT: Towards Light-Weight Convolution-Free Vision Transformers (2022). https://doi.org/10.48550/arXiv.2207.05557
18. Wang, Q., Teng, Z., Xing, J., Gao, J., Hu, W., Maybank, S.: Learning attentions: residual attentional Siamese network for high performance online visual tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–23, 2018, pp. 4854–4863 (2018). https://doi.org/10.1109/CVPR.2018.00510
19. He, A., Luo, C., Tian, X., Zeng, W.: A twofold Siamese network for real-time object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2008, Anchorage, AK, USA, June23–28, 2008, pp.4834–4843 (2008). https://doi.org/10.1109/CVPR.2018.00508
20. Zeng, N., Li, X., Wu, P., Li, H., Luo, X.: A novel tensor decomposition-based efficient detector for low-altitude aerial objects with knowledge distillation scheme. IEEE-CAA J. Automatica Sin. **11**(2), 487–501 (2024). https://doi.org/10.1109/JAS.2023.124029
21. Chen, Y., Lin, M., He, Z., Polat, K., Alhudhaif, A., Alenezi, F.: Consistency-and dependence-guided knowledge distillation for object detection in remote sensing images. Expert Syst. Appl. **229**, 120519 (2023). https://doi.org/10.1016/j.eswa.2023.12051
22. Hendrycks, D., Gimpel, K.: Gaussian Error Linear Units (GELUs). arXiv preprint arXiv:1606.08415 (2016).https://doi.org/10.48550/arXiv.1606.08415
23. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**, 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
24. Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., Bai, H., Xu, Y., Liao, C., Ling, H.: LaSOT: A high-quality benchmark for large-scale single object tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018, Long Beach, CA, USA, June15–20, pp. 5374–5383 (2018). https://doi.org/10.1109/CVPR.2019.00552
25. Huang, L., Zhao, X., Huang, K.: GOT-10k: a Large high-diversity benchmark for generic object tracking in the wild. IEEE Trans. Pattern Anal. Mach. Intell. **43**, 1562–1577 (2022). https://doi.org/10.1109/TPAMI.2019.2957464
26. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In :Proceedings of the European Conference on Computer Vision, ECCV 2014, Zurich, Switzerland, September 6–12,2014, pp.740–755 (2014). https://doi.org/10.1007/978-3-319-10602-1_48
27. Wu, Y., Lim, J., Yang, M.H.: Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1834–1848 (2015). https://doi.org/10.1109/TPAMI.2014.2388226
28. Li, S., Yeung, D.Y.: Visual object tracking for unmanned aerial vehicles: a benchmark and new motion models. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1–7 (2017). https://doi.org/10.1609/aaai.v31i1.11205
29. Mueller, M., Smith, N., Ghanem, B.: A Benchmark and Simulator for UAV Tracking. In: Proceedings of the European Conference on Computer Vision, pp. 445–461 (2016). https://doi.org/10.1007/978-3-319-46448-0_27

30. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Learning spatially regularized correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 11–18, pp. 4310–4318 (2015).https://doi.org/10.1109/ICCV.2015.490
31. Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., Torr, P.H.: Staple: Complementary learners for real-time tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 26–July 1, 2016, pp. 1401–1409 (2016). https://doi.org/10.1109/CVPR.2016.156
32. Zhang, J., Ma, S., Sclaroff, S.: MEEM: Robust tracking via multiple experts using entropy minimization. In: European Conference on Computer Vision, ECCV 2014, Zurich, Switzerland, September 6–12, 2014, pp. 188–203 (2014). https://doi.org/10.1007/978-3-319-10599-4_13
33. Valmadre, J., Bertinetto, L., Henriques, J., Vedaldi, A., Torr, P.H.: End-to-end representation learning for correlation filter based tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017, pp. 2805–2813 (2017). https://doi.org/10.48550/arXiv.1704.06036
34. Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D.V., Tao, D.: MUlti-Store Tracker (MUSTer): A cognitive psychology inspired approach to object tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 07–12, 2015, pp. 749–758 (2015). https://doi.org/10.1109/CVPR.2015.7298675
35. Danelljan, M., Hager, G., Khan, F.: Accurate scale estimation for robust visual tracking. In: British Machine Vision Conference, Nottingham, September 1–5, pp. 1–11 (2015). https://www.bmva.org/bmvc/2014/papers/paper038/index.html
36. Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M., Hicks, S.L., Torr, P.H.: Struck: structured output tracking with kernels. IEEE Trans. Pattern Anal. Mach. Intell. **38**, 2096–2109 (2016). https://doi.org/10.1109/TPAMI.2015.2509974
37. Wang, Q., Zhang, L., Bertinetto, L., Hu, W., Torr, P.H.: Fast online object tracking and segmentation: a unifying approach. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June16–20, 2019, pp. 1328–1338 (2019). https://doi.org/10.1109/CVPR.2019.00142
38. Song, Y., Ma, C., Wu, X., Gong, L., Bao, L., Zuo, W., Shen, C.: Vital: Visual tracking via adversarial learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR 2018, Salt Lake City, UT, USA, June18–23, 2018, pp. 8990–8999 (2018). https://doi.org/10.1109/CVPR.2018.00937
39. Yan, B., Zhao, H., Wang, D., Lu, H., Yang, X.: 'Skimming-Perusal' tracking: a framework for real-time and robust long-term tracking. In: IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27–November 02, 2019, pp. 2385–2393 (2019). https://doi.org/10.1109/ICCV.2019.00247
40. Nam, H., Han, B.: Learning multi-domain convolutional neural networks for visual tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June27–30, pp. 4293–4302 (2016). https://doi.org/10.1109/CVPR.2016.465
41. Lukežič, A., Matas, J., Kristan, M.: D3S-A discriminative single shot segmentation tracker. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2019, Seattle, WA, USA, June 13–19, 2019, pp. 7131–7140 (2019). https://doi.org/10.1109/CVPR42600.2020.00716
42. Wang, G., Luo, C., Xiong, Z., Zeng, W.: Spm-tracker: Series-parallel matching for real-time visual object tracking. In :Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2016,Long Beach, CA, USA, June 16–20, 2016, pp. 3643–3652 (2016). https://doi.org/10.1109/CVPR.2019.00376
43. Held, D., Thrun, S., Savarese, S.: Learning to track at 100 fps with deep regression networks. In: European Conference Computer Vision, ECCV 2016, Amsterdam, The Netherlands, October 11–14, pp. 749–765 (2016). https://doi.org/10.1007/978-3-319-46448-0_45
44. Danelljan, M., Robinson, A., Khan, F.S. Felsberg, M.: Beyond correlation filters: Learning continuous convolution operators for visual tracking. In: Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, pp. 472–488 (2016). https://doi.org/10.1007/978-3-319-46454-1_29
45. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: Eco: Efficient convolution operators for tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,CVPR2017, Honolulu, HI, USA, July21–26, 2017, pp. 6638–6646 (2017). https://doi.org/10.1109/CVPR.2017.733
46. Sauer, A., Aljalbout, E., Haddadin, S.: Tracking holistic object representations. In: Proceedings of the 30th British Machine Vision Conference (BMVC), Cardiff, UK, September 9–12, 2019. arXiv preprint arXiv:1907.12920 (2019). https://doi.org/10.48550/arXiv.1907.12920
47. Galoogahi, H. K., Fagg, A., Lucey, S.: Learning background-aware correlation filters for visual tracking. In: Proceedings of the IEEE International Conference on Computer Vision, ICCV 2017, Santiago, Chile, December 07–13, 2017, pp. 1144–1152 (2017). https://doi.org/10.1109/ICCV.2017.129
48. Danelljan, M., Häger, G., Khan, F.S., Felsberg, M.: Discriminative scale space tracking. IEEE Trans. Pattern Anal. Mach. Intell. **39**(8), 1561–1575 (2017). https://doi.org/10.1109/TPAMI.2016.2609928
49. Ma, C., Huang, J.B., Yang, X., Yang, M.H.: Hierarchical convolutional features for visual tracking. In: IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 07–13, 2015, pp. 3074–3082 (2015). https://doi.org/10.1109/ICCV.2015.352
50. Wang, N., Zhou, W., Tian, Q., Hong, R., Meng, W., Li, H.: Multi-cue correlation filters for robust visual tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–23, 2018, pp. 4844–4853 (2018). https://doi.org/10.1109/CVPR.2018.00509
51. Li, Y., Fu, C., Ding, F., Huang, Z., Lu, G.: AutoTrack: towards high-performance visual tracking for UAV with automatic spatio-temporal regularization. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, 2020, June 13–19, pp. 11920–11929 (2020). https://doi.org/10.1109/CVPR42600.2020.01194
52. Huang, Z., Fu, C., Li, Y., Lin, F., Lu, P.: Learning aberrance repressed correlation filters for real-time UAV tracking. In: IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27–November 02, pp. 2891–2900 (2019). https://doi.org/10.1109/ICCV.2019.00298
53. Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., Hu, W.: Distractor-aware Siamese networks for visual object tracking. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds) Computer Vision – ECCV 2018. ECCV 2018. Lecture Notes in Computer Science, vol. 11213. Springer, Cham. https://doi.org/10.1007/978-3-030-01240-3_7
54. Li, X., Ma, C., Wu, B., He, Z., Yang, M.: Target-aware deep tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 15–20, 2019, pp. 1369–1378 (2019). https://doi.org/10.1109/CVPR.2019.00146

55. Li, F., Yao, Y., Li, P., Zhang, D., Zuo, W., Yang, M.H.: Integrating boundary and center correlation filters for visual tracking with aspect ratio variation. In: IEEE International Conference on Computer Vision Workshops, ICCVW 2017, Venice, Italy, October 22–29, 2017, pp. 2001–2009 (2017). https://doi.org/10.1109/ICCVW.2017.234
56. Zhang, T., Xu, C., Yang, M.H.: Multi-task correlation particle filter for robust object tracking. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21–26, 2017, pp. 4335–4343 (2017). https://doi.org/10.1109/CVPR.2017.512
57. Wang, N., Song, Y., Ma, C., Zhou, W., Liu, W., Li, H.: Unsupervised deep tracking. In IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 15–20, 2019, pp. 1308–1317 (2019). https://doi.org/10.1109/CVPR.2019.00140
58. Li, F., Tian, C., Zuo, W., Zhang, L., Yang, M.H.: Learning spatial-temporal regularized correlation filters for visual tracking. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18–23, 2018, pp. 4904–4913 (2018). https://doi.org/10.1109/CVPR.2018.00515
59. Zhang, L., Suganthan, P.N.: Robust visual tracking via co-trained kernelized correlation filters. Pat. Rec. **69**, 82–93 (2017). https://doi.org/10.1016/j.patcog.2017.04.004
60. Guo, Q., Feng, W., Zhou, C., Huang, R., Wan, L., Wang, S.: Learning dynamic siamese network for visual object tracking. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22–29, 2017, pp. 1781–1789 (2017). https://doi.org/10.1109/ICCV.2017.196