Zhang *et al.*
EURASIP Journal on Advances in Signal Processing (2022) 2022:124
https://doi.org/10.1186/s13634-022-00954-4

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Long-term tracking with transformer and template update

Hongying Zhang*, Xiaowen Peng and Xuyong Wang

*Correspondence: carole_zhang0716@163.com

Civil Aviation University of China, Jinbei Road, Tianjin 300300, China

Abstract

Aiming at the tracking failure due to the disappearance of the target in the long-term target tracking process, this paper proposes a long-term target tracking network based on the visual transformer and template update. First of all, we construct a feature extraction network based on the transformer and adopt a knowledge distillation strategy to improve the effectiveness of the network for global feature extraction. Secondly, in the modeling transformer, the target features are fully fused with the search area features by using encoder, and the position information in the target query is learned by the decoder. Then, target predictions are performed on the information from the encoder–decoder to obtain tracking results. Meanwhile, we design a score head model to judge the validity of the dynamic template of the current frame before tracking in the next frame. We select the appropriate dynamic template for the tracking of the next frame according to the score result. In this paper, we performed extensive experiments on LaSOT, VOT2021-LT, TrackingNet, TLP, and UAV123 datasets, and the experimental results prove the effectiveness of our method. In particular, it exceeds STARK by 0.8 % (F score) on VOT2021-LT, 1.0 % (S score) on LaSOT, and TrackingNet exceed STARK by 1.1 % (NP score), which also demonstrates the superiority of the method in this paper.

Keywords: Transformer, Long-term tracking, Template update

1 Introduction

Although long-term visual tracking has received significant attention from researchers as a research hotspot in visual target tracking, studying a robust long-term tracking framework remains a daunting task as it is conducted in more realistic scenarios with many unresolved difficulties, particularly in the case of target disappearance & reappearance [1].

Currently, most approaches use a combination of CNN and Transformer [2] for long-term visual tracking and have also achieved good results [3, 4]. Typically, researchers extract generic features of the input image through a CNN-based backbone network. However, CNN only focuses on the feature relationships between local neighborhoods when processing image content and features, ignoring the impact of global information on feature extraction. In order to extract global information better, inspired by the success of the transformer and its recent application in computer vision [5], this paper

uses transformer-based DeiT [6] to replace convolutional neural network. Specifically, a distillation mechanism is used in the feature extraction process and combined with a teacher model for bootstrap optimization. The advantage of this is that the local sensory field and parameter sharing in CNN can be combined into the transformer to improve the processing capability of the transformer model for features.

In addition, in using the transformer model to solve the process of target disappearance and reappearance, most methods pay attention to the importance of temporal information for long-time tracking in addition to global information. Template update is the typical approach to introduce temporal information, which typically specifies a target template in the first frame and uses this same template in subsequent tracking, ignoring the changes of the target across frames. The long tracking time of long-term tracking tasks can lead to problems of target disappearance and deformation. However, the existing method of fixing the template cannot effectively update the target template, which eventually leads to a situation of tracking failure. Therefore, we propose a score prediction head to judge the effectiveness of dynamic target templates using temporal information and the target state of the current frame, so as to flexibly select high-quality dynamic update templates to improve tracking accuracy.

To conclude, this work improves the long-term tracking model in two ways. Firstly, we use transformer-based DeiT to replace the original ResNet [7] as the feature extraction backbone, DeiT takes advantage of the ability of the transformer to extract features with global dependency, reducing the accumulation of errors in the subsequent trace process, thereby improving long-term tracking performance. Secondly, a score prediction head is designed to be applied to the dynamic template update branch, and the cross-attention operation of the score token is performed in the search area and the initial template, to calculate the effectiveness of the dynamic template of the current frame of the score judgment, and provide a reliable dynamic template for the tracking of the next frame.

In summary, this work has three contributions.

1. A transformer-based long-term tracking model is proposed to capture globally dependent target features in video sequences, allowing extensive communication between the target and the search region.
2. An efficient score prediction head is designed to select high-quality dynamic templates through which additional temporal information is introduced to achieve an efficient transformer-based long-term tracker.
3. Our model demonstrates strong performance on five challenging benchmarks and achieved near real-time operation at 25 FPS on dual RTX 2080Ti GPUs.

2 Related works

The related work in this paper incorporates both long-term tracking and tracking paradigm.

2.1 Long-term tracking

Since 2018, long-term trackers have been developed with the release of long-term tracking datasets. Equipping short-term trackers with re-detectors to improve the ability of

long-term tracking to deal with frequent target disappearance and reappear is a mainstream approach at present. For example, The MBMD [8] exploits a SiamPRN-based network to regress the target in a local search region or every sliding window when re-detection. Valmadre et al. [9] propose a long-term tracker. This tracker adds a simple re-detector to SiamFC [10], and its performance is much better than the original SiamFC. In [9, 11–13], trackers are equipped with the re-detection scheme for long-term tracking, but they merely track the targets in a local search region to expect that the lost targets will reappear around the previous location. This approach carries a high level of risk because the output of the short-term tracker is not as reliable. To avoid this risk, GlobalTrack [14] performs a global instance search of the target for each frame, but this method not only requires a lot of computational costs but also has unsatisfactory results. In the same year, the template updated strategy pushed the comprehensive performance of long-term tracking to a new commanding height. Updatenet [15] applied template matching to SiamFC and DaSiamRPN [16] to predict target locations. LTMU [17] utilizes SiamRPN [18] as a re-detector and Metaupdater as an online updater to predict whether the current state is reliable enough to be used for the update in long-term tracking.

2.2 Tracking paradigm

At present, the popular tracking methods [3, 4, 19, 20] are mostly combined with CNN and transformer, using CNN as a backbone to extract the general features of the targets, and the transformer with its powerful modeling ability is usually used for the fusion work between the target and the template, and finally through the simple head network to generate the target state. This method shows powerful performance in many works, for example, Transtrack [21] takes the features extracted in CNN as the query and key, learns to query the target location from the key by the detecting branch, and queries the location of the current frame by tracking the object feature of the previous frame in the key by the tracking branch. Based on the DETR [22], Trackformer [23] queries the target's embedding through a series of learnable object queries, which successfully predict the output embedding through subsequent tasks, such as border regression or category prediction, to pass in the tracking of the next frame. STARK extracts the common features through the ResNet, and then passes in the transformer to model the global spatio-temporal feature dependencies between the target and the search area, and learns query embeddings to predict the target location. This work believes that although the use of CNN to extract common features can adapt to most of the tasks, in the long-term tracking process, this general method is not very suitable. To avoid CNN's shortcomings in handling long-term dependency and understanding the global structure of objects, we propose a full transformer tracker, solely containing encoders and decoders and two simple heads, leading to a more accurate tracker with neat and compact architecture.

3 Methods

In this section, we describe our approach in detail. In Sect. 3.1, we introduce the motivation. In Sect. 3.2, we introduce the overall tracking framework of the approach. In Sect. 3.3, we introduce the transformer-based feature extraction network. In Sect. 3.4,

we introduce the transformer network for the modeling part. In Sect. 3.5, we introduce two simple heads. Section 3.6 describes the training loss.

3.1 Motivation

In recent years, the more popular long-term tracking networks have used convolutional neural networks to extract target features. However, the disadvantage of convolutional neural networks is that the convolutional kernel focuses on the information in local regions and ignores the global information of the target and the frame-to-frame dependencies, therefor if we can improve the tracking network's ability to extract global information with long-term dependencies, the overall tracking performance of the long-term tracker will be improved. At the same time, this paper argues that temporal information is important for the performance of the trackers. If all the morphological and positional information in a video frame is assumed to be known when tracking a target in a frame, then there should be some performance improvement for problems such as target disappearance and deformation.

3.2 Long-term tracking framework

In this section, we propose the transformer network for long-term visual tracking. The network architecture is demonstrated in Fig. 1, which is mainly composed of four parts. It is divided into feature extraction backbone, transformer structure for building a feature dependency model, a head network to track the target position, and a head network to control the update of dynamic templates. In Fig. 1, X represents a search region of the current frame, T represents a template image of the initial target object, and Z represents dynamically updated template sampled from intermediate frames.

3.3 Visual transformer feature extraction network

We use the transformer-based DeiT as the backbone, which introduces a teacher-student strategy for transformer. It reduces dependence on large amounts of data by optimizing data augmentation and regularization strategies. And it improves the running speed to a certain extent. The core of the DeiT core is the introduction of distillation method into the training of VIT [24], and the proposal of token-based distillation. An important component of DeiT is the distillation training, which combines with the teacher model to guide DeiT to learn the target's feature extraction better. The distillation process is rough as shown in Fig. 2, this process is mainly to use a distillation

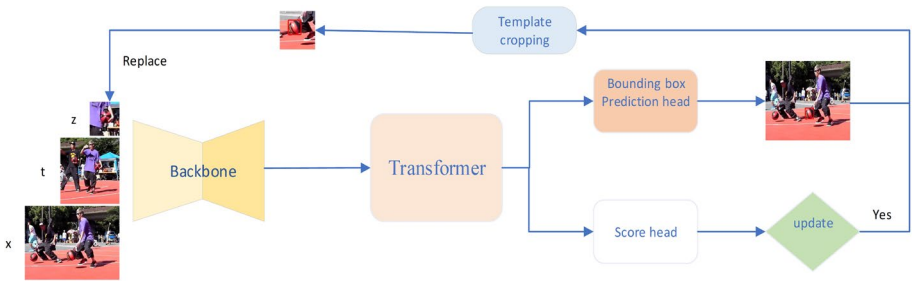

Fig. 1 The proposed tracking architecture

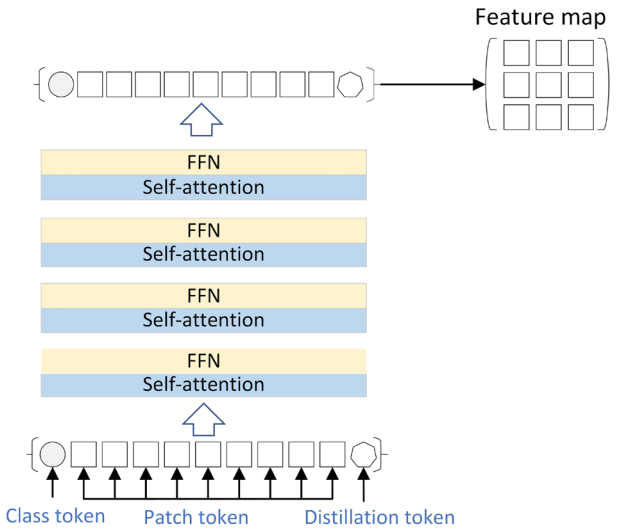

Fig. 2 The distillation procedure of the DeiT

tokens to interact with class tokens and patch tokens at the self-attention level, and the distillation token entered into the transformer is learned through backpropagation. This training strategy uses convolutional networks as a teacher network for distillation, which achieves better results with fewer data and fewer computing resources than a network using the transformer architecture as a teacher network.

The input of the DeiT backbone is a triplet: a template image of the initial target object $X \in R^{3*H_x*W_x}$, a search area for the current frame$T \in R^{3*H_t*W_t}$, and a dynamically updated template image $Z \in R^{3*H_z*W_z}$ sampled from the intermediate frame. We split the input image group into patches, and then linearly projected each patch to obtain a sequence of patch tokens. At the same time, we spliced a class token for classification before the patch tokens, and a distillation token for distillation training after the patch tokens. In the process of training DeiT using distillation strategies, lots of error messages are learned from the teacher network. And the distillation token is designed to solve these error messages, and is specifically designed to receive the label generated by the teacher network and participate in the overall information interaction process. In order to preserve the spatial location information between patches, we added position embedding to encode the location information of the token. We input the class token, patch token, and distillation token with position embedding added to the transformer encoder for processing. The outputs are the initial template feature$F_x \in R^{\frac{H_x}{s}*\frac{W_x}{s}*C}$, the search area feature $F_t \in R^{\frac{H_t}{s}*\frac{W_t}{s}*C}$, and the dynamic template feature$F_z \in R^{\frac{H_z}{s}*\frac{W_z}{s}*C}$respectively. This work only uses the feature extraction part of the DeiT, that is, removing the MLP layer and the subsequent parts, and the rest has not changed.

Transformer layers in DeiT contain only encoders, which are mainly stacked by combining self-attention and feed-forward network (FFN). The specific structure in the encoder is shown in Fig. 3, and it consists of a multi-head self-attention (MSA), a norm layer (LN), and a multi-layer perceptron (MLP) through residual connections. The specific process can be represented as:

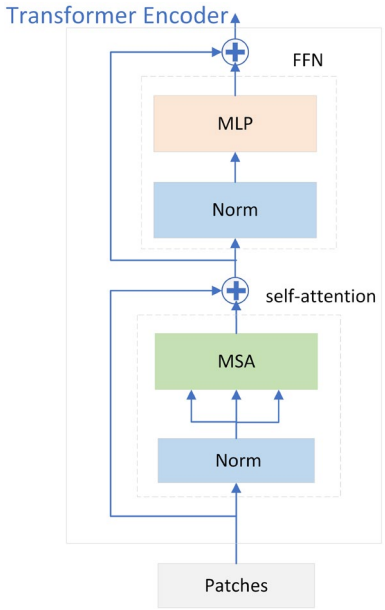

Fig. 3 Structure of the Encoder

$$X_n = MSA\ (LN(X_{n-1})) + X_{n-1} \tag{1}$$

$$\hat{X}_n = MLP\ (LN(X_n)) + X_n \tag{2}$$

where X_n represents the output of the multi-head self-attention and $\hat{X}_n$ represents the output of the MLP output, and use a norm layer before each block.

3.4 Modeling transformer

The transformer in the modeling phase consists of an encoder and decoder, and there are 6 layers in both the encoder layer and the decoder layer. The encoder captures dependencies between all elements in the sequence and reinforces the original features with global contextual information. And it allows the model to learn the discriminating features for target localization. The decoder allows the target query to focus on all location features and search area features on the template, learning a robust representation that is ultimately associated with the heads.

The feature groups output from the DeiT are stitched and then passes through a 1*1 convolutional layer to reduce the number of channels from C to d, which is consistent with the hidden layer dimension in the subsequent transformer encoder–decoder structure. The flatten and concatenated operations to obtain a total feature $\hat{F} = \frac{H_x}{s}\frac{W_x}{s} + \frac{H_t}{s}\frac{W_t}{s} + \frac{H_z}{s}\frac{W_z}{s}$, and input it to transformer encoder.

3.4.1 Encoder

Similar to the encoder in DeiT, this also consists of continuous encoder layers, each of which includes a multi-head self-attention and feed-forward network, where the feed-forward network contains two-layer perceptron and GEIU activation.

3.4.2 Decoder

Similar to the encoder, the decoder also includes a self-attention, encoder–decoder attention, and a feed-forward network. The input of this part is the enhanced feature sequence from the encoder and the preset target query, the target query and the enhanced feature sequence have interacted in the decoder layer, from which the tracked target information is extracted, and a more robust representation is learned for subsequent bounding box prediction and dynamic template update judgment.

3.5 Head

3.5.1 Bounding box prediction head

In order to predict the information of the bounding box more accurately, a more stable prediction box is generated. Like the STARK Corner Prediction Head, we use a fully convoluted corner point locator head to directly estimate the bounding box of the tracked object, solely with several Conv-BN-ReLU layers to predict the coordinates of top-left and bottom-right corners, respectively. Finally, the bounding box is obtained by calculating the expectation of the angular point probability distribution.

3.5.2 Score head

Dynamic template plays a key role in capturing time information and changes in the appearance of the target. If the target is completely obscured or out of view or due to the deformation of the target and causes the model to drift, the crop of the dynamic template is not trustworthy. To solve these problems, we designed a score head, which is composed of a scoring prediction head, dynamic template update judgment, and simple crop operation. This head controls how the dynamic template is updated by predicting whether the confidence level of the current frame is correct.

The structure of the scoring prediction head is shown in Fig. 4, which mainly consists of a depth-wise cross-correlation, an attention block, and an MLP. First, a learnable score token acts as a query to interact with the decoder's output in-depth, allowing the score token to encode the extracted enhanced target information. At the same time, the score token focuses on the position of the target token in the dynamic template to compare with the target state in the next frame. Finally, the score is calculated through the MLP layer and sigmoid activation. We use the score to judge the timing of dynamic template updating, to prevent the generation of inferior quality dynamic

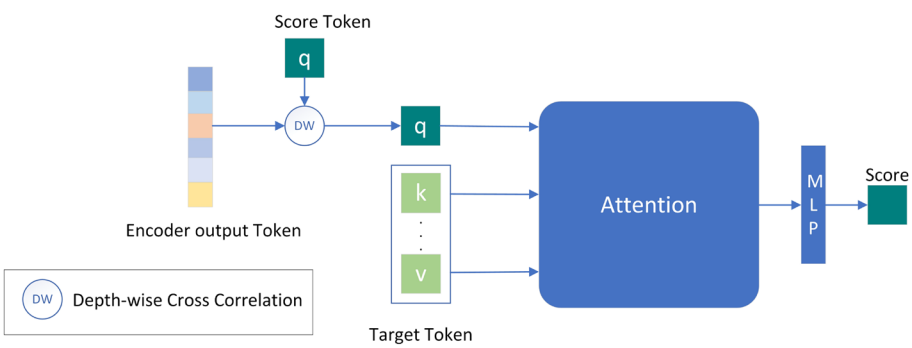

Fig. 4 Structure of the score prediction head

templates with fuzzy targets or severely deformed targets. Therefore, we set a threshold τ to compare with the score, if the score is higher than the threshold τ, the current state is considered reliable, and the dynamic template of this frame is cropped, if the score is below the threshold τ, the current state is considered unreliable, and the dynamic template of the previous frame is maintained. Here τ we set it to 0.5.

To focus on more target local spatial information, the attention block in the score prediction head uses the asymmetric mixed attention proposed by MixFormer [25], It performs separable depth-wise convolutional projection on each feature map (i.e., query, key, and value), then flattens each feature map and process it through a linear projection to generate a query, key, and value for attention operations. This mixed attention is defined as follows:

$$Attention = Softmax\left(\frac{q_t k_t^T}{\sqrt{D}}\right) v_t \tag{3}$$

where D represents the dimension of the key, q_t , k_t and v_t represent target, Attention is the attention maps of the target.

3.6 Training loss

This work uses the learning method of joint learning positioning and classification, so our training is divided into two stages: localization and classification. In the first stage, the whole network, except for the scoring head, is trained end-to-end only with the L1 loss and GIoU [26] loss to supervise the bounding box prediction results, and the calculation formula of the entire framework loss function L is as follows:

$$L_{GIoU} = 1 - GIoU(B, \hat{B}) \tag{4}$$

$$L_1 = B - \hat{B} \tag{5}$$

$$L = \lambda_1 L_1(B, \hat{B}) + \lambda_2 L_{GIoU}(B, \hat{B}) \tag{6}$$

In the second stage, only the score head is optimized with binary cross-entropy loss defined as

$$\hat{L} = \hat{B} log(P_i) + (1 - \hat{B}) log(1 - P_i) \tag{7}$$

where $\hat{B}$ is the bounding box groundtruth, B is the prediction result, λ_1 and λ_2 are the loss weight coefficient, this work sets 5 and 2, P_i is the predicted confidence.

During inference, three templates and corresponding features are initialized in the first frame, and they are fed into the network to generate a bounding box and a confidence score. The dynamic template is updated only when the update interval is reached and the confidence level is greater than the threshold τ. To improve efficiency, we set the update interval to 200 frames. The new template is cropped from the original image and then imported as a new dynamic template image for feature extraction.

4 Experimental results and discussion

In this section, we evaluate the proposed method on five benchmarks and compare it with other advanced tracking networks. Section 4.1 introduces the relevant details of the experiments. Section 4.2 presents the results of the quantitative evaluation of the 5 benchmarks, including the experimental datasets and the evaluation criteria of the experiments. Section 4.3 presents the ablation experiments, analyzing the results of the qualitative evaluation, and Sect. 4.4 describes the visualization results, providing visualizations of the LaSOT datasets to demonstrate the superiority of our model.

4.1 Implementation details

Our trackers are implemented using Python 3.6 and PyTorch 1.7.0. The experiments are conducted on a server with GeForce RTX 2080 Ti/PCIe/SSE2. Especially, this is a neat tracker without post-processing and multi-layer feature aggregation strategy.

4.1.1 Model

The backbone is initialized with the parameters pre-trained with 300 epochs on ImageNet with a distillation strategy. The transformer of the backbone has only an encoder, and no decoder, and the heads and layers are both 12. The transformer structure of the modeling section consists of 1 encoder layer and 1 decoder layer, for a total of 6 transformers, including a multi-head attention layer (MSA) and forward network (FFN). The MSA has 8 heads with a width of 256, while the FFN has hidden units of 2048.

4.1.2 Training

The training data consists of the train-splits of LaSOT [27], GOT-10K [28], COCO2017 [29], and TrackingNet [30]. The sizes of search images and templates are 320 $\times$ 320 pixels and 128 $\times$ 128 pixels, respectively, corresponding to 52 and 22 times the target bounding box area. The minimal training data unit for our model is a triplet, consisting of two templates and one search image. The entire training process consists of two stages, the first stage requires 500 epochs and the second stage requires 50 epochs, each with 6000 samples. The network is optimized using AdamW optimizer and weight decay 10^{-4}, The initial learning rates for the backbone and the rest are 10^{-5} and 10^{-4}, respectively. To stabilize the training to ensure convergence, the gradient cropping and learning rate attenuation strategies were adopted, and the learning rate decreased by 10 times after the 400th epoch in the first stage and 10 times after the 40th epoch in the second stage.

4.2 Comparison with the state-of-the-art models

We verify the performance of our model on five benchmarks, including VOT2021-LT [31], LaSOT [27], TrackingNet [30], UAV123 [32], and TLP [33]. The main algorithms involved are SiamRPN++ [34] and SiamRCNN [11] based on twin networks; TREG [19] and STARK [4] based on transformer. LTMU [17], LT-DSE [35], and UpdateNet [15] based on template update, where STARK-ST [4] also uses the template update strategy; TLD [36],SPLT [37], RLT-DiMP [38] and Globletrack [14] based on

Local-global Switching strategy; and other classical algorithms, such as DiMP [39], PrDiMP [40], ATOM [41], DMTrack [42], etc.

4.2.1 LaSOT

LaSOT is a large-scale long-term tracking benchmark, which contains 1400 videos with an average length of 2512 frames. It has 70 target categories and 20 videos in each category, covering a variety of challenges in the field. Divided into 20% as test sets, it includes a total of 280 videos. Fig. 5 shows that our model surpasses all other trackers by a large margin. Compared to STARK-ST101, SiamRCNN, and LTMU, our model achieves 1.0% ,2.8%, and 7.6% gains on the LaSOT test set, respectively.

To verify the effectiveness of our method for different attributes, detailed results of the success rate of eight typical difficulties on the LaSOT dataset are provided in Figs. 6 and 7, including fast motion (FM), background clutter (BC), motion blur (MB), deformation (DEF), illumination change (IV), partial occlusion (POC), out-of-field (OV), and scale change (SV).

As shown by the results in Figs. 6 and 7, the adaptability of the tracking network becomes necessary in the case of deformation and out-of-field. The method proposed in this paper can effectively establish the long-term dependence of features, and accurately utilize the historical feature information in the case of target deformation or target disappearance. At the same time, our method also has some performance improvement in occlusion and scale change, thanks to the transformer feature extraction network's ability to obtain the most representative feature vectors. In the case of motion blur, the tracking networks need to be able to perform accurate target tracking in low-resolution video frames. Our method compensates for the effect of motion blur on the tracker to some extent and improves the performance of the tracker. However, the method in this paper seems to be ineffective in the case of background clutter and illumination changes, probably because too much background information is introduced during feature extraction and dynamic template update to affect the judgment of the tracker.

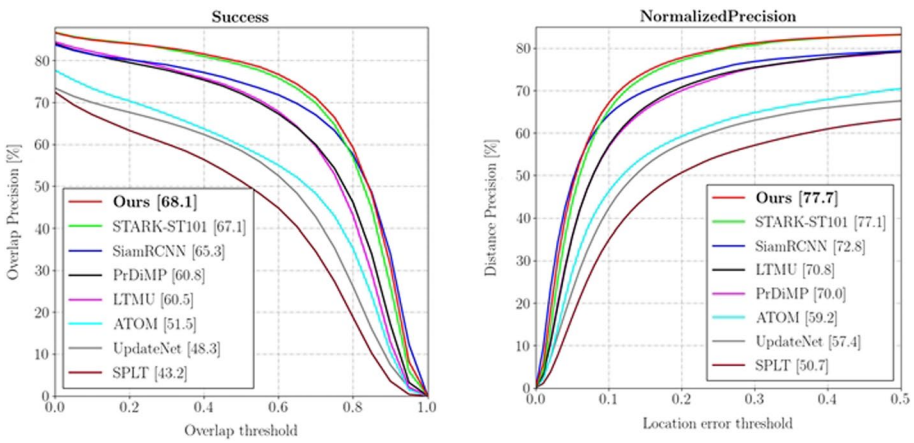

Fig. 5 Comparisons on LaSOT test set

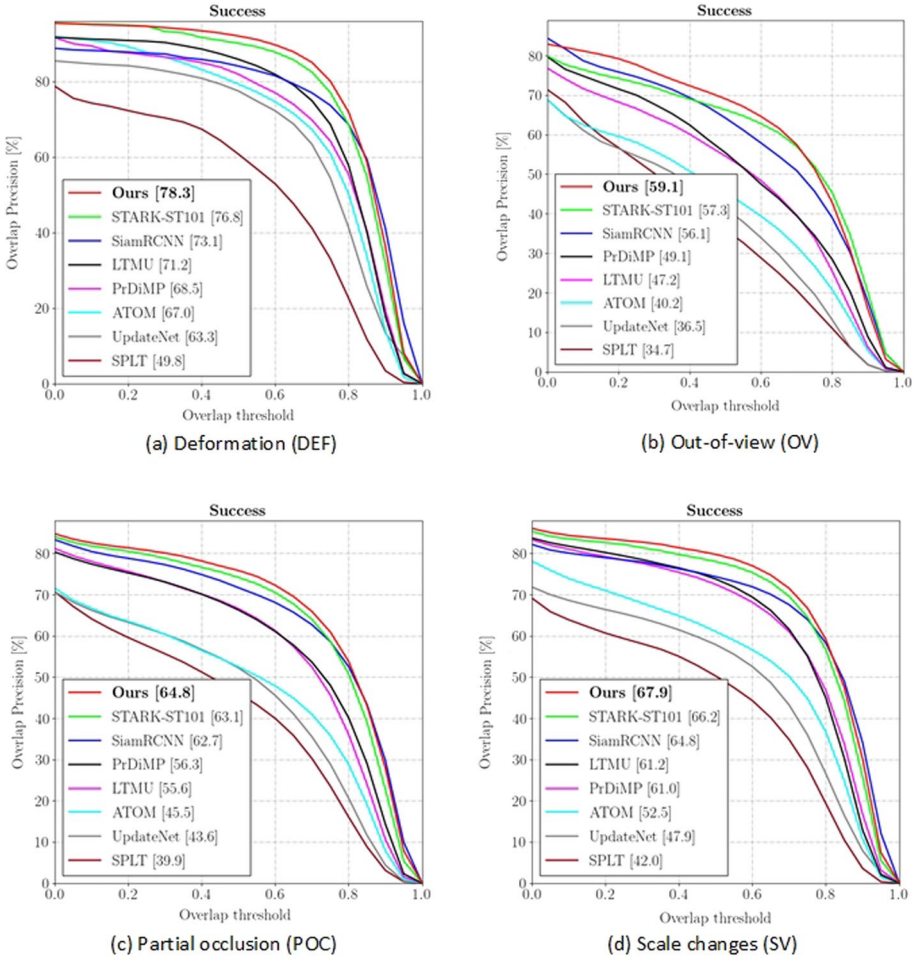

Fig. 6 The success rate of different attributes on the LaSOT

4.2.2 TrackingNet

TrackingNet is a carefully selected video dataset specifically for target tracking from large-scale object detection datasets. A total of 30,643 videos, with an average duration of 16.6s, included 511 videos and 70 target categories in the test set. Table 1 shows that our model surpasses all other models with a large margin. Specifically, our model achieves the top-ranked performance on NP of 88.0%, surpassing STARK by 1.1%.

4.2.3 UAV123

UAV123 includes 123 videos captured by the low-altitude drone platform, with a clean background and a wide variation in viewing angle, averaging 915 frames. The dataset has the problems of the invisible target, complete occlusion, and small target scale, which requires our tracker to have faster learning ability and the ability to extract global information. Table 1 shows our results on the UAV123 dataset. Our model outperforms all other models.

TrackingNet and UAV123 datasets adopt the one-pass evaluation (OPE) strategy, and the evaluation indicators are Precision (P), Normalized Precision (NP), and Success (S).

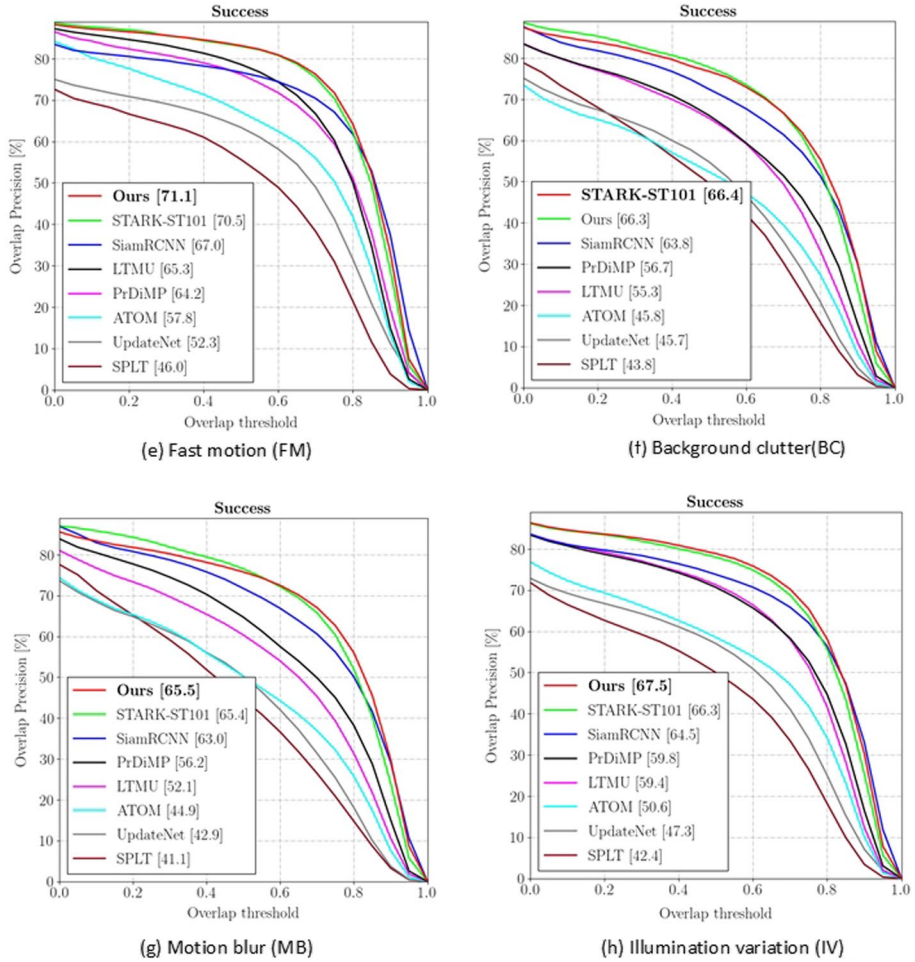

Fig. 7 The success rate of different attributes on the LaSOT

Table 1 AUC on TrackingNet and UAV123 dataset

	TrackingNet			UAV123	
	AUC	**NP**	**P**	**AUC**	**P**
ATOM	70.3	77.1	64.8	64.3	–
DiMP	74.0	80.1	68.7	65.4	–
SiamRPN++	73.3	80.0	69.4	61.0	80.3
SiamRCNN	81.2	85.4	80.0	64.9	83.4
Globletrack	65.6	75.4	70.4	–	–
TREG	78.5	83.8	78.5	66.9	88.4
UpdateNet	67.6	75.2	62.5	–	–
STARK-ST101	82.0	86.9	–	68.3	89.3
Ours	82.9	88.0	83.1	69.0	90.2

The precision is calculated by comparing the distance between the predicted result and the ground truth, and the success rate is calculated by measuring the Intersection over Union (IoU) between the two. Since the precision is very sensitive to the target size and

Table 2 AUC and P on the TLD dataset

	TLD	SPLT	Globaltrack	DMTrack	LTMU	STARK-ST101	Ours
AUC(%)	15.4	41.6	50.1	54.1	55.8	56.9	57.6
P(%)	16.7	40.3	70.1	59.1	60.8	61.3	61.1

Table 3 Comparisons on VOT-LT2021 benchmark. The best results are marked in the red font

	SPLT	RLT-DiMP	LTMU	LT-DSE	STARK-ST101	Ours
F-score(%)	56.5	67.0	69.1	69.5	69.5	70.3
Pr(%)	58.7	65.7	70.1	71.5	69.1	73.9
Re(%)	54.4	63.5	68.1	67.7	69.8	67.1

image resolution, the normalized accuracy is calculated by normalizing the accuracy according to the size of the ground truth. The final result is usually compared based on the Area Under Curve (AUC) of the success rate graph.

4.2.4 TLP

This is a long-term single-target tracking dataset that includes 50 long HD videos from real scenes with an average sequence length of over 13,000 frames and a total video duration of over 400 minutes. Table 2 reports the AUC and precision scores on the TLP dataset. Our method has some gains compared to other trackers. For example, our method achieves 0.7% and 1.8% AUC gain compared to STARK-ST101 and LTMU, two methods that use template updates, respectively.

4.2.5 VOT2021-LT

The VOT2021-LT dataset contains 50 videos with 215294 frames in total, in which target objects disappear and reappear frequently. The accuracy evaluation of the dataset mainly includes tracking precision (Pr), tracking recall (Re), and tracking F-score. Precision and recall are computed under a series of confidence thresholds. F-score, defined as $F = \frac{2P_r R_e}{P_r + R_e}$, is used to rank different trackers. Different trackers are ranked according to the tracking F-score. We compare our model to the currently popular tracker and report the evaluation results in Table 3 It can be seen that the F-score of our model is 70.3, which is better than all previous methods.

4.3 Ablation study

In this section, we use the LaSOT dataset to perform ablation analysis of our model. Through different experimental settings, we did four experiments with different trackers, namely "Baseline," "Model 1," "Model 2, " and "Ours." The meaning of these concepts is explained below. (1) "Baseline" denotes the STARK-ST101 model. (2) "Model1" denotes a model that uses DeiT as a feature extraction network. (3) "Model2" denotes a model of adding the score prediction head based on STARK-ST101. (4) "Ours" denotes a long-term tracking network that uses both the DeiT feature extraction network and the score prediction head.

Table 4 Ablation analysis of the proposed tracker on the LaSOT dataset

	P	NP	S
Baseline	71.88	76.7	66.8
Model1	72.35	77.3	67.3
Model2	72.03	77.0	67.03
Ours	74.13	77.5	68.1

Table 5 Comparison about the speed, FLOPs, and Params

	Speed(fps)	FLOPs(G)	Params(M)
Baseline	30	20.5	47.2
Model1	25	39.7	105.1
Model2	30	20.4	47.1
Ours	24	40.5	105.2

The results of the different variants on the LaSOT dataset in Table 4, and the speed, FLOPs, and Params of the different variants in Table 5, from which we can obtain the following conclusions. (1) 'model1' achieves an S of 67.3, which is also competitive compared to other state-of-the-art methods (shown in Fig. 5). The applicability of the transformer to long-term tracking tasks is also verified. But it also brings a corresponding flaw, the large number of parameters of the transformer model drags down the running speed. (2) By comparing "model2" and "baseline," we can conclude that the score prediction head proposed in this work can improve the long-term performance to some extent, which also means that reliable dynamic templates can bring better gains to the tracker. (3) Comparing 'ours' with 'model1' and 'model2' yields that both the transformer backbone and the score prediction head control template update are indispensable throughout the tracing process.

4.4 Visualization

To show the actual tracking effect of different networks in complex situations such as target disappearance, deformation, and scale change. We select video sequences (fox, motorcycle, skate, racing, and volleyball) from LaSOT, a large real-world scene tracking dataset, to visualize the tracking results, as shown in Fig. 8.

As shown in Fig. 8, the top-down sequences are the "fox" sequence, "motorcycle" sequence, "skate" sequence, "racing" sequence, and "volleyball" sequence, respectively. In the "motorcycle" and "racing" sequences, the targets appear to disappear and reappear frequently, and the proposed algorithm could still obtain accurate target state estimation and high-quality tracking results due to the combined effect of the transformer and template update mechanisms. When the target disappears, STARK-ST101, SiamRCNN, and LTMU all drift to other similar targets easily, and the target does not track the correct target in time when it reappears. In the "fox" and "state" sequences, the targets have severe scale changes, occlusion, and deformation phenomenon, STARK-ST101 and other trackers cannot get accurate target state estimation and have tracking failure, while our tracker could track the target accurately, and handle these situations

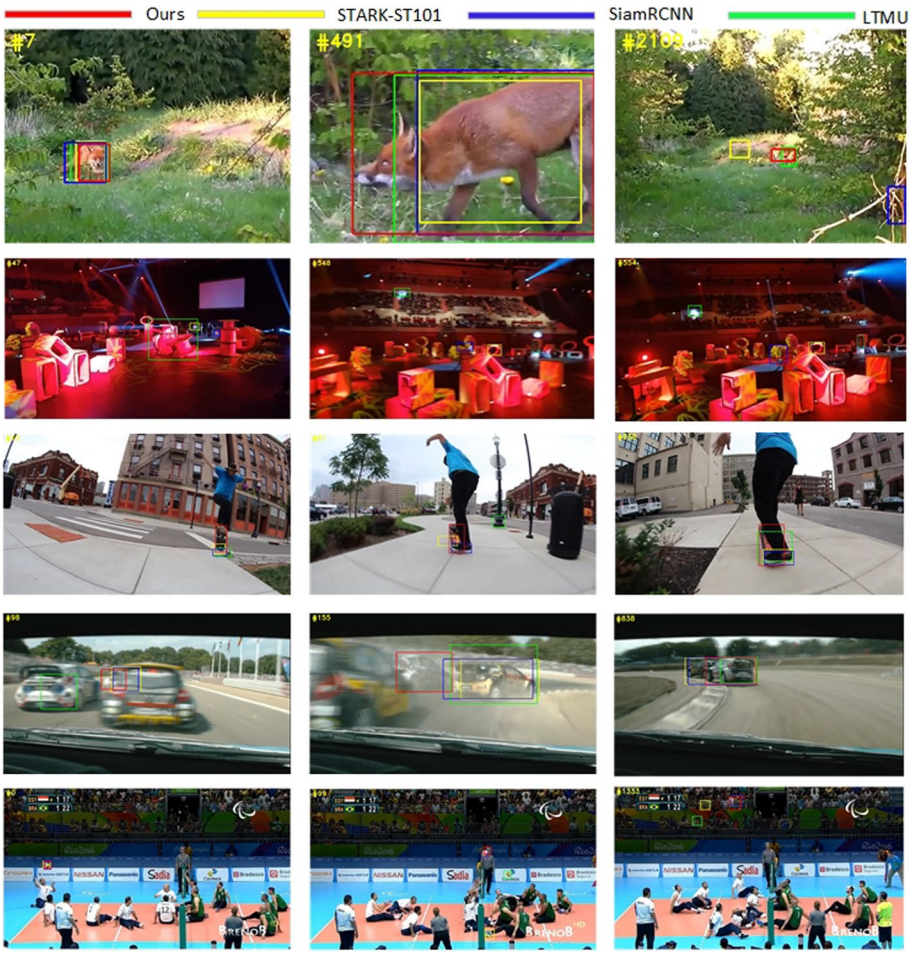

Fig. 8 Qualitative comparison of ours with other state-of-the-art trackers on 5 video sequences from the LaSOT dataset

better. It is worth noting that there are difficulties in the 'volleyball' sequence with fast motion, background clutter, and small targets, our tracker has experienced tracking errors. Specifically, when attributes such as fast motion, background clutter, and small targets appear in the same video at the same time, the dynamic template strategy in our approach introduces too much background information to affect the tracker's judgment, which leads to a tracking failure situation. In this case, SiamRCNN is better than the method in this paper because it does not use a complex template update strategy.

5 Conclusion

This paper proposes a new transformer-based long-term tracking framework. We improve the existing transformer-based tracking network and enhance the feature extraction ability by introducing the visual transformer based on the attention mechanism as the backbone network. A score prediction head is designed to control dynamic template updating using asymmetric mixed attention and score token interactive learning. Experimental results show that our model designed in this work is better than the current mainstream tracking networks on five long-term tracking benchmarks. As there

are more parameters in the network, the running speed does not reach real-time, so next, we intend to study in the direction of a low number of parameters and high running speed.

Abbreviations

CNN:	Convolutional neural network
FFN:	Feed-forward network
MSA:	Multi-head self-attention
LN:	Norm layer
MLP:	Multi-layer perceptron
OPE:	One-pass evaluation
GELU:	Gaussian error linear units, a high-performance neural network activation function

Acknowledgements
Thanks to the anonymous reviewers and editors for their hard work.

Author contributions
HZ and XP proposed the original idea of the full text. XP and XW designed and performed the experiment. XP wrote the manuscript under the guidance of HZ. XP and HZ revised the manuscript. All authors read and approved this submission

Funding
This work was supported in part by the National Key Research and Development Program of China (2018YFB1601200); Graduate Research Innovation Grant Program of Civil Aviation University of China (2021YJS026)

Availability of data and materials
The datasets used during the current study are the LaSOT [27] , the VOT2021-LT [31], the TrankingNet [30], and the UAV123 [32] datasets, which are available online or from the corresponding author upon reasonable request.

Declarations

Competing interests
The authors declare that they have no competing interests

Received: 11 August 2022 Accepted: 27 November 2022
Published online: 28 December 2022

References

1. S.M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, S. Kasaei, Deep learning for visual tracking: a comprehensive survey. IEEE Trans. Intell. Transp. Syst. **23**(5), 3943–3968 (2021)
2. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need. Adv. Neural Inform. Process. Syst. **30** (2017)
3. T. Bian, Y. Hua, T. Song, Z. Xue, R. Ma, N. Robertson, H. Guan, Vtt: long-term visual tracking with transformers. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 9585–9592 (2021). IEEE
4. B. Yan, H. Peng, J. Fu, et al. Learning spatio-temporal transformer for visual tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10448-10457 (2021)
5. N. Parmar, A. Vaswani, J. Uszkoreit, et al. Image transformer. In: International conference on machine learning. PMLR, pp. 4055–4064 (2018)
6. H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jégou, Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning, pp. 10347–10357 (2021). PMLR
7. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
8. Y. Zhang, D. Wang, L. Wang, J. Qi, H. Lu, Learning regression and verification networks for long-term visual tracking. arXiv preprint arXiv:1809.04320 (2018)
9. J. Valmadre, L. Bertinetto, J.F. Henriques, R. Tao, A. Vedaldi, A.W. Smeulders, P.H. Torr, E. Gavves, Long-term tracking in the wild: a benchmark. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 670–685 (2018)
10. L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, P.H. Torr, Fully-convolutional siamese networks for object tracking. In: European Conference on Computer Vision, pp. 850–865 (2016). Springer
11. P. Voigtlaender, J., Luiten, P.H. Torr, B. Leibe, Siam r-cnn: Visual tracking by re-detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6578–6588 (2020)
12. T. Li, S. Zhao, Q. Meng, Y. Chen, J. Shen, A stable long-term object tracking method with re-detection strategy. Pattern Recogn. Lett. **127**, 119–127 (2019)
13. N. Wang, W. Zhou, H. Li, Reliable re-detection for long-term tracking. IEEE Trans. Circuits Syst. Video Technol. **29**(3), 730–743 (2018)
14. L. Huang, X. Zhao, K. Huang, Globaltrack: A simple and strong baseline for long-term tracking. Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11037–11044 (2020)

15. L. Zhang, A. Gonzalez-Garcia, J.V.d. Weijer, M. Danelljan, F.S. Khan, Learning the model update for siamese trackers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4010–4019 (2019)
16. Z. Zhu, Q. Wang, B. Li, W. Wu, J. Yan, W. Hu, Distractor-aware siamese networks for visual object tracking. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 101–117 (2018)
17. K. Dai, Y. Zhang, D. Wang, J. Li, H. Lu, X. Yang, High-performance long-term tracking with meta-updater. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6298–6307 (2020)
18. B. Li, J. Yan, W. Wu, Z. Zhu, X. Hu, High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8971–8980 (2018)
19. Y. Cui, C. Jiang, L. Wang, G. Wu, Target transformed regression for accurate tracking. arXiv preprint arXiv:2104.00403 (2021)
20. Z. Fu, Q. Liu, Z. Fu, Y. Wang, Stmtrack: template-free visual tracking with space-time memory networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13774–13783 (2021)
21. P. Sun, J. Cao, Y. Jiang, R. Zhang, E. Xie, Z. Yuan, C. Wang, P. Luo, Transtrack: Multiple object tracking with transformer (2020). arXiv preprint arXiv:2012.15460
22. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers. In: European Conference on Computer Vision, pp. 213–229 (2020). Springer
23. T. Meinhardt, A. Kirillov, L. Leal-Taixe, C. Feichtenhofer, Trackformer: multi-object tracking with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8844–8854 (2022)
24. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, C. Heigold, S. Gelly, et al.: An image is worth 16x16 words: transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
25. Y. Cui, C. Jiang, L. Wang, G. Wu, Mixformer: End-to-end tracking with iterative mixed attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13608–13618 (2022)
26. H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 658–666 (2019)
27. H. Fan, L, Lin, F. Yang, P. Chu, G. Deng, S. Yu, H. Bai, Y. Xu, C. Liao, H. Ling, Lasot: a high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5374–5383 (2019)
28. L. Huang, X. Zhao, K. Huang, Got-10k: a large high-diversity benchmark for generic object tracking in the wild. IEEE Trans. Pattern Anal. Mach. Intell. **43**(5), 1562–1577 (2019)
29. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: common objects in context. In: European Conference on Computer Vision, pp. 740–755 (2014). Springer
30. M. Muller, A. Bibi, S. Giancola, S. Alsubaihi, B. Ghanem, Trackingnet: a large-scale dataset and benchmark for object tracking in the wild. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 300–317 (2018)
31. M. Kristan, J. Matas, A. Leonardis, M. Felsberg, R. Pflugfelder, J.-K. Kämäräinen, H.J. Chang, M. Danelljan, L. Cehovin, A. Lukežič, et al: The ninth visual object tracking vot2021 challenge results. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2711–2738 (2021)
32. B. Leibe, J. Matas, N. Sebe, M. Welling, [lecture notes in computer science] computer vision – eccv 2016 volume 9907 || fast guided global interpolation for depth and motion https://doi.org/10.1007/978-3-319-46487-9(Chapter 44), 717–733 (2016)
33. A. Moudgil, V. Gandhi, Long-term visual object tracking benchmark. In: Asian Conference on Computer Vision, pp. 629–645 (2018). Springer
34. B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, J. Yan, Siamrpn++: evolution of siamese visual tracking with very deep networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4282–4291 (2019)
35. M. Kristan, A. Leonardis, J. Matas, M. Felsberg, R. Pflugfelder, J.-K. Kämäräinen, M. Danelljan, L.Č., Zajc, A. Lukežič, O. Drbohlav, et al., The eighth visual object tracking vot2020 challenge results. In: European Conference on Computer Vision, pp. 547–601 (2020). Springer
36. Z. Kalal, K. Mikolajczyk, J. Matas, Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. **34**(7), 1409–1422 (2011)
37. B. Yan, H. Zhao, D. Wang, H. Lu, X. Yang, 'Skimming-perusal' tracking: a framework for real-time and robust long-term tracking. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
38. S. Choi, J. Lee, Y. Lee, A. Hauptmann, Robust long-term object tracking via improved discriminative model prediction. In: European Conference on Computer Vision, pp. 602–617 (2020). Springer
39. G. Bhat, M. Danelljan, L.V. Gool, R. Timofte, Learning discriminative model prediction for tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6182–6191 (2019)
40. M. Danelljan, L.V. Gool, R. Timofte, Probabilistic regression for visual tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7183–7192 (2020)
41. M. Danelljan, G. Bhat, F.S. Khan, M. Felsberg, Atom: accurate tracking by overlap maximization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4660–4669 (2019)
42. Z. Zhang, B. Zhong, S. Zhang, Z. Tang, X. Liu, Z. Zhang, Distractor-aware fast tracking via dynamic convolutions and mot philosophy. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1024–1033 (2021)

Publisher's Note