frontiers | Frontiers in Physics

TYPE Original Research
PUBLISHED 01 September 2023
DOI 10.3389/fphy.2023.1266927

OPEN ACCESS

EDITED BY
Ben-Xin Wang,
Jiangnan University, China

REVIEWED BY
Fupeng Wang,
Ocean University of China, China
Giuseppe Brunetti,
Politecnico di Bari, Italy

*CORRESPONDENCE
Qiang Fu,
cust_fuqiang@163.com
Hua Cai,
caihua@cust.edu.cn

RECEIVED 25 July 2023
ACCEPTED 09 August 2023
PUBLISHED 01 September 2023

CITATION
Zhu R, Leng J, Fu Q, Wang X, Cai H, Wen G, Zhang T, Shi H, Li Y and Jiang H (2023), Transformer-based target tracking algorithm for space-based optoelectronic detection.
Front. Phys. 11:1266927.
doi: 10.3389/fphy.2023.1266927

Transformer-based target tracking algorithm for space-based optoelectronic detection

Rui Zhu[1], Jinsong Leng[2], Qiang Fu[1,3]*, Xiaoyi Wang[1,4], Hua Cai[2]*, Guanyu Wen[5], Tao Zhang[6], Haodong Shi[1,3], Yingchao Li[1,3] and Huilin Jiang[1]

[1]College of Opto-Electronic Engineering, Changchun University of Science and Technology, Changchun, China, [2]College of Electronics and Information Engineering, Changchun University of Science and Technology, Changchun, China, [3]National and Local Joint Engineering Research Center of Space Optoelectronics Technology, Changchun University of Science and Technology, Changchun, China, [4]Changchun Institute of Optics, Fine Mechanics, and Physics, Chinese Academy of Sciences, Changchun, China, [5]Changchun Observatory National Astronomical Observatories Chinese Academy of Sciences, Changchun, China, [6]China Academy of Space Technology, Beijing, China

The target tracking by space-based surveillance systems is difficult due to the long distances, weak energies, fast speeds, high false alarm rates, and low algorithmic efficiencies involved in the process. To mitigate the impact of these difficulties, this article proposes a target tracking algorithm based on image processing and Transformer, which employs a two-dimensional Gaussian soft-thresholding method to reduce the image noise, and combines a Laplace operator-weighted fusion method to augment the image, so as to improve the overall quality of the image and increase the accuracy of target tracking. Based on the SiamCAR framework, the Transformer model in the field of natural language processing is introduced, which can be used to enhance the image features extracted from the backbone network by mining the rich temporal information between the initial and dynamic templates. In order to capture the information of the target's appearance change in the temporal sequence, a template update branch is introduced at the input of the algorithm, which realizes the dynamic update of the templates by constructing a template memory pool, and selecting the best templates for the candidate templates in the memory pool using the cosine similarity-based selection, thus ensuring the robustness of the tracking algorithm. The experimental results that compared with the SiamCAR algorithm and the mainstream algorithms, the TrD-Siam algorithm proposed in this article effectively improves the tracking success rate and accuracy, addressing poor target tracking performance under space-based conditions, and has a good value of application in the field of optoelectronic detection.

KEYWORDS

optoelectronic detection, image processing, target tracking, transformer, dynamic template updates

1 Introduction

Optoelectronic detection technology possesses the benefits of high-resolution images, large detection distances, compact system sizes, and low costs; these favourable properties facilitate the detection of many objects in space and meet the requirements of space-based target detection [1–5]. Target tracking is an important research element in the field of optoelectronic detection. Moreover, target tracking is the foundation for computer vision tasks such as pose estimation, behavior recognition, behavioral analysis, and video analysis.

Currently, it is difficult to monitor targets with high precision, specifically in four areas: 1) Radical variations in target appearance throughout the tracking task, including target rotation, illumination changes, scale changes, etc., 2) frequent occlusion of targets during tracking; 3) drifting tracking frame caused by interactive motion between targets. 4) poor image quality with unclear targets in complex backgrounds [6].

Image preprocessing is defined as the processing of images prior to the detection and tracking of spatial targets in the image [7]. For space-based target tracking, the target can be very distant, and it is imaged on the detector's image plane as a dot or short strip, occupying only a few image elements. This results in an extremely low signal-to-noise ratio and causes the uneven background noise to obscure the target. To solve these problems, researchers have examined how the characteristics of space-based targets differ from those of background stars and noise via methods such as multiframe time series projection [8], trajectory identification [9], matching correlation [10], and hypothesis testing [11]. However, the background noise composition of the space-based environment is intricate, and the distribution of noise within the images is nonuniform because of the effect of external stray light and the detector itself. Furthermore, the forms and greyscale values of the spatial targets in the images resemble those of the noise. These algorithms are frequently unsuccessful at denoising space-based images of stars, resulting in the loss of target information or the production of spurious targets.

With the advent of deep learning techniques, monitoring researchers began experimenting with the application of deep neural networks. In the beginning, more emphasis was placed on the use of pre-trained neural networks; however, from 2017 onwards, researchers have paid more attention to Siamese network trackers, whose algorithms exhibit ultra-fast tracking speed while ensuring greater tracking accuracy. The classical twin-based tracking algorithm determines the tracking model through offline training and only employs the tracking model learned based on the template of the initial frame during the tracking process, which makes it difficult for the algorithm to adapt to changes in the target's appearance and reduces the algorithm's robustness. updateNet [12] automatically learns appearance samples of the target during the tracking process, thereby mitigating the issue that a single template cannot account for changes in the target's appearance during motion. However, the update module proposed by UpdateNet is distinct from the embedded tracking algorithm, does not profit from end-to-end training, and updates the template at a fixed frequency, which adds superfluous computational effort when the target's appearance does not change significantly.

SiamFC [13] utilized an image pyramid approach for the prediction of target bounding boxes, which is not only inefficient in inference but also incapable of adapting to scale changes in the target's appearance. In target detection, algorithms such as SiamRPN [14] and SiamRPN++ [15] borrowed from anchor point-based region recommendation networks, which are more adaptable to the target than the multi-scale search approach. However, the pre-setting of anchor points is dependent on the configuration of hyperparameters, which increases the complexity of model training.

In order to resolve the aforementioned problems, this article proposes a target tracking algorithm based on image preprocessing and transformer [16]. First, the original image is pre-processed using a two-dimensional Gaussian soft thresholding method based on the denoising factor [17] to eliminate background noise, and the image is enhanced using a Laplace operator weighted fusion method after noise reduction [18, 19]. Secondly, SiamCAR [20] is used as the overall framework of the target tracking algorithm, given that SiamCAR employs an anchorless bounding box based regression strategy for target state estimation go. Transformer is then incorporated to improve feature representation. Transformer is widely used in the field of computer vision, and the DETR [21] algorithm in target detection uses this model to expand the features of the image and process them into sequence form, so that each feature node in the sequence can calculate the correlation between each other and have the capability of global modelling, and the global modelling capability using the correlation between each feature node in the sequence to calculate the correlation between each feature node. The transformer's global modeling capability can be used to derive information on the temporal variations in the

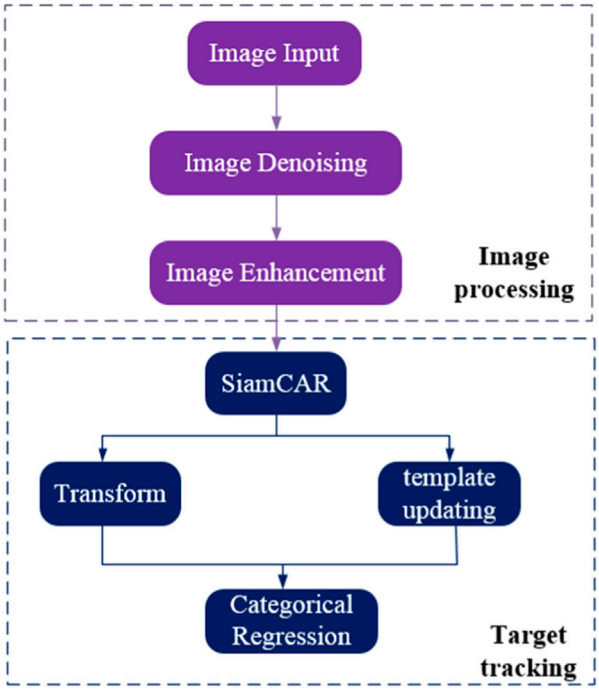

FIGURE 1
Algorithm flow chart.

target's motion, thereby enhancing the performance of the target tracking algorithm. Finally, a dynamic template update is designed to capture changes in the target's appearance during motion in order to increase the tracking algorithm's robustness to appearance changes.

The rest of this article is organized as follows: in Section 2, the implementation process of the target tracking algorithm based on image preprocessing and Transformer is proposed. In Section 3 experimental validations are made and the results are proved, demonstrating the superior performance of the proposed TrD-Siam. Finally, the conclusions are drawn in Section 4.

2 Algorithms in this article

Figure 1 depicts the algorithm's flow chart, which consists of image denoising, image enhancement, the SiamCAR backbone network, the Transformer, the template update branch, and the classification and regression networks. Using ResNet-50 to extract template features, the template pool selects the dynamic templates, the Transformer encoder enhances the initial template features and dynamic template features, and the Transformer decoder aggregates the information of the initial and dynamic templates in the search area to accomplish deep mining of temporal information in the image blocks of the search area. After modeling by Trasformer, the template features are cross-correlated with the search features to generate a high-quality feature response map, which is then input into an anchorless-based classification regression network to decode the predicted target bounding box.

2.1 Image processing

Image denoising: A two-dimensional Gaussian soft threshold method is used to pre-process the image to derive the processed $R(x,y)$, with the following equation:

$$R(x,y) = \frac{\partial g(x,y)}{\partial t} = a \cdot \left(\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \right) \tag{1}$$

Where $f(x,y)$ denotes a original image; x denotes the spatial horizontal coordinate position of a pixel and y denotes the spatial vertical coordinate position of a pixel; t denotes the image denoising processing time, a denotes the denoising factor and $g(x,y)$ denotes a two-dimensional Gaussian function. $R(x,y)$ denotes the output image.

The two-dimensional Gaussian functions are

$$g(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(i^2+j^2)t}{2\sigma^2} \tag{2}$$

Where i denotes the distance of the pixel from the origin on the x-axis, j denotes the distance of the pixel from the origin on the y-axis and σ is the standard deviation of the Gaussian distribution.

The denoising factor a is.

$$a = \begin{cases} [g(x,y) + \nabla f(x,y)]^n & g(x,y) \leq -\nabla f(x,y) \\ [g(x,y) - \nabla f(x,y)]^n & g(x,y) \geq \nabla f(x,y) \end{cases} \tag{3}$$

Where n denotes the number of iterations and $\nabla f(x,y)$ denotes the original image two-dimensional gradient value. The denoising factor a is not a fixed value; the final value of a is determined by $\nabla f(x,y)$; and as the number of iterations n increases, image denoising is becoming more and more apparent.

Using the Laplace operator weighted fusion method, the enhancement of the pre-processed image is conducted, and the enhanced image $B(x,y)$ is obtained.

$$B(x,y) = f(x,y) + \beta \left(\frac{\partial^2 I(x,y)}{\partial^2 x} + \frac{\partial^2 I(x,y)}{\partial^2 y} \right) \tag{4}$$

Where $I(x,y)$ denotes the Laplace transform operator and β denotes the weighting factor. The value of the weighting factor β depends on the image sharpness and contrast, with the dark target weighting being large and the light target weighting being small. The weighting factor β is

$$\beta = \left[|i-j|^2 P(i,j) + p \right] \cdot \left[\left| q - \sum_{k=0}^{L-1} g_k \log_2 g_k \right| \right] \tag{5}$$

Where $P(i,j)$ denotes the probability of the pixel distribution of the grey level difference between pixels, L denotes the image grey level value, g_k denotes the kth image histogram, and p, q are constant terms. Taking the star atlas as an example, the results are shown in Figure 2.

2.2 Backbone network

Figure 3 depicts the algorithm structure block diagram with SiamCAR as the backbone network and a modified ResNet-50 as the backbone sub-network for feature extraction. ResNet-50's perceptual field was expanded to make it suitable for dense prediction tasks by reducing the spatial step size to retain more target features and implementing dilation convolution. To increase the perceptual field, the network was designed by setting the step size to 1 in the Conv4 and Conv5 blocks and the dilation rate to 2 in the Conv4 block and 4 in the Conv5 block. The shallow features can effectively represent visual attributes and thus aid in target localisation, while the deep semantic features are more conducive to classification; combining shallow and deep features improves tracking accuracy [15]. To improve the classification of the regression prediction bounding box, the algorithm described in this article cascades the features extracted from the last three residual blocks of the ResNet-50 backbone network:

$$\varphi(X) = Cat(F_3(X), F_4(X), F_5(X)) \tag{6}$$

Where $F_3(X)$, $F_4(X)$ and $F_5(X)$ are represented as features of ResNet-50 backbone network layers conv3_4, conv4_6 and conv5_3 respectively, and their channel numbers are all adjusted to 256 by applying 1×1 convolution. Cat stands for channel cascade operation and $\varphi(X)$ is the fused feature after channel cascade with 3×256 channels.

The Siamese network for target feature extraction consists of two backbone subnetworks with shared weights: the template branch, which receives template patch Z as input and returns template feature $\varphi(Z)$; and the search branch, which receives search region X

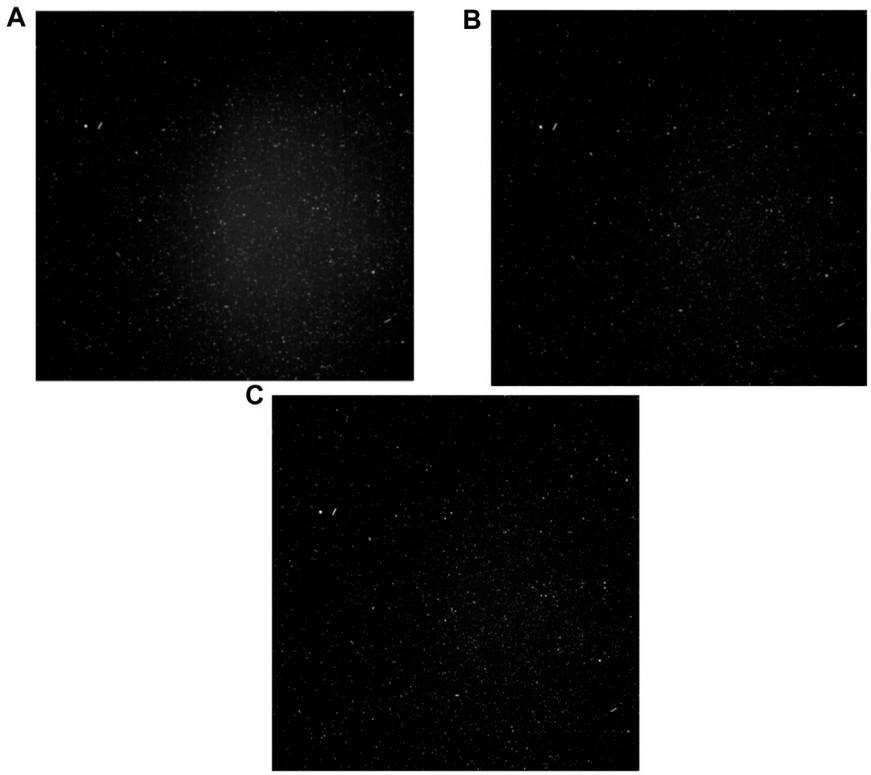

FIGURE 2
Image pre-processing results. **(A)**Original image **(B)**Image denoising **(C)**Image enhancement.

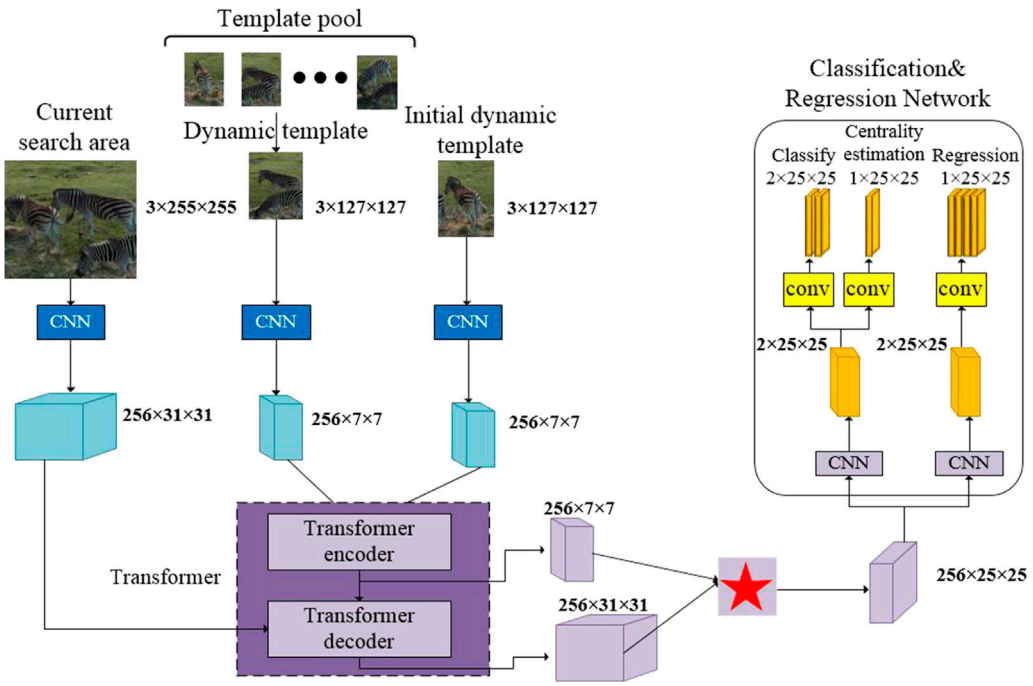

FIGURE 3
Block diagram of the tracking algorithm structure. Adapted with permission from, "Zebras Grazing by Taryn Elliot, https://www.pexels.com/zh-cn/video/5146558/.

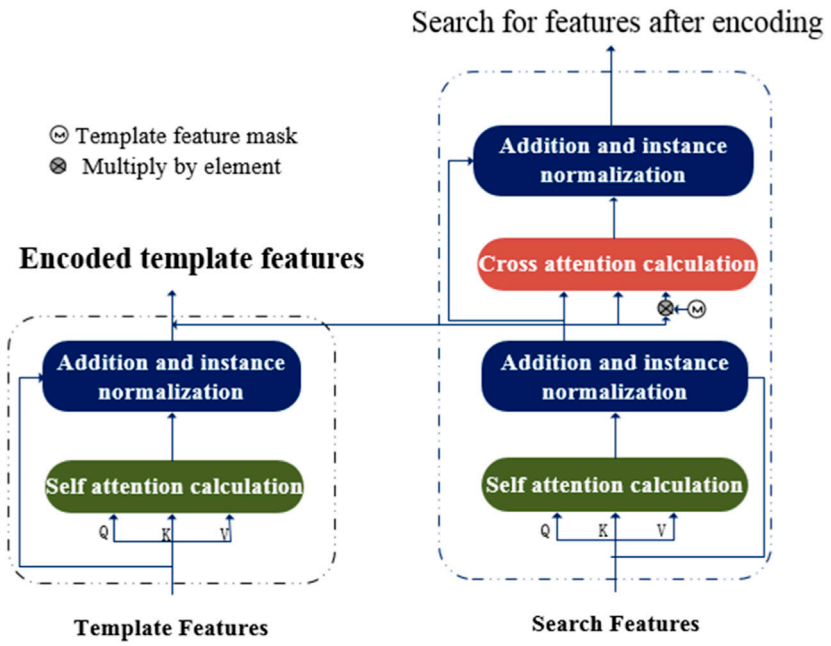

FIGURE 4
Transformer structure.

and returns search feature $\varphi(X)$. By sharing convolutional neural network parameters, both branches guarantee that the same transformation is applied to the input. Both branches of the convolutional neural network share the same parameters to ensure that the same transformation is applied to the input. In the subsequent prediction subnetwork, the algorithm is executed with template feature $\varphi(Z)$ as the kernel and the intercorrelation operation on search feature $\varphi(X)$ to obtain the feature response map R.

$$R = \varphi(Z)^{*}\varphi(X) \tag{7}$$

In order to adapt to space-based optoelectronic detection systems, the number of channels of $\boldsymbol{\phi}(X)$ is decreased from 3 × 256 to 256 by performing a dimensionality reduction operation on the 1 × 1 convolution kernel in order to satisfy real-time requirements and reduce the algorithm's complexity, thereby enhancing the tracking performance.

2.3 Transformer for target tracking

Figure 4 depicts the structure of the Transformer for target tracking. The structure of the Transformer is derived from the traditional Transformer in natural language processing. The left half is the Transformer encoder and the right half is the Transformer decoder, with the self-attention and cross-attention modules serving as the respective fundamental construction elements.

2.3.1 Transformer encoder

A feature mapping of the initial and dynamic templates, designated $T_1 \in \mathbb{R}^{C\times H\times W}$ and $T_d \in \mathbb{R}^{C\times H\times W}$, is the input to the Transformer encoder. Cascade it to $T = Concat(T_1, T_d) \in \mathbb{R}^{2\times C\times H\times W}$. A self-attentive calculation weights the integrated initial frame with cascaded feature T of the dynamic template so that cascade T can benefit from the features of both frames, resulting in a higher quality representation of the template features. The primary method of calculation is as follows: First, the features of the template feature T are adjusted to $T' \in \mathbb{R}^{N_T\times C}$, i.e., serially processed, to acquire a one-dimensional feature vector containing $N_T = 2 \times H \times W$.The encoder module computes the self-attentiveness matrix of the template features, as shown in Eq. 8:

$$A_{TT} = Attention(\varphi(\mathrm{T}'), \varphi(\mathrm{T}')) \in \mathbb{R}^{N_T\times N_T} \tag{8}$$

In Eqs. 1, 8 × 1 is processed from $\varphi(T)$ to $\varphi(T')$ under the linear transformation operation $\varphi(\bullet)$, which is intended to adjust the dimensionality of the one-dimensional sequence features from C to $C/4$. $\varphi(T')$ is then fed into the self-attention module of the encoder to do the calculation of the self-attention matrix. Based on the self-attentive matrix A_{TT}, the transformed template feature $A_{TT} \times T' \in \mathbb{R}^{N_T\times C}$ can be obtained, and this term is used as the residual term and added to the original feature T'. As in Eq. 9:

$$T_f = Ins.Norm(\mathrm{A}_{TT}\mathrm{T}' + \mathrm{T}') \tag{9}$$

Where $T_f \in \mathbb{R}^{N_T\times C}$ is the final encoded feature vector from the encoder. Ultimately, this one-dimensional feature vector is recovered as a two-dimensional feature map $T_{encode} \in \mathbb{R}^{2\times C\times H\times W}$.

2.3.2 Transformer decoder

The input to the transformer decoder is a search for feature $S \in \mathbb{R}^{C\times H\times W}$ of the image block, which first resizes the feature to $S' \in \mathbb{R}^{N_s\times C}$, where $N_s = H \times W$. S' is then fed into the self-attention

module to obtain the self-attention matrix A_{SS} for the search region, as in Eq. 10:

$$A_{SS} = Attention(\varphi(S'), \varphi(S')) \in \mathbb{R}^{N_S \times N_S} \tag{10}$$

The final output of the self-attentive module of the decoder is given in Eq. 11:

$$S_f = Ins.Norm(A_{SS}S' + S') \tag{11}$$

From the search feature S_f and the output T_f of the encoder, the cross-attention matrix between them can be calculated as

$$A_{TS} = Attention(\varphi(S_f), \varphi(T_f)) \in \mathbb{R}^{N_S \times N_T} \tag{12}$$

In addition to constructing the propagation of temporal information, Gaussian labels for the initial and dynamic template features were constructed to utilize the spatial information and make the tracking algorithm more focused on areas where the target could be present, as shown in Eq. 13:

$$m(y) = \exp\left(-\frac{\|y - c\|^2}{2\sigma^2}\right) \tag{13}$$

Where c represents the true position of the target, for the initial template mask m_1 and the dynamic template mask m_d, which is stitched together as $M = Concat(m_1, m_d) \in \mathbb{R}^{2\times H\times W}$. And further adjusted to M for $M' \in \mathbb{R}^{N_T \times 1}$.

In order to transfer the information between the template features and the search features, the template mask M' is first multiplied element by element with the template feature T_f to obtain $T_f \otimes M'$, which is used to suppress the background region. The transformed feature $A_{TS}(T_f \otimes M')$ is then obtained based on the cross-attention matrix A_{TS}, and this feature is added as a residual term to the search feature S_f. Instance normalisation is then performed to obtain the output of the Transformer decoder, the process being Eq. 14:

$$S_{feat} = Ins.Norm(A_{TS}(T_f \otimes M') + S_f) \tag{14}$$

Converts the final output feature vector $S_{feat} \in \mathbb{R}^{N_s \times C}$ of the Transformer decoder into a two-dimensional feature map $S_{decode} \in \mathbb{R}^{C\times H\times W}$.

Finally, by 1×1 convolution, $T_{encode} \in \mathbb{R}^{2\times C\times H\times W}$ is downscaled and its channel is adjusted to C. Finally, the adjusted T_{encode} is intercorrelated with S_{decode} to obtain the response map R^*. This response map is fed into the classification regression network for classification and regression of the target.

2.4 Dynamic template updates

For dynamic template branching, an N-sized template memory pool with the feature encoding $\{T_i | T_i \in \mathbb{R}^{C\times H\times W}, i = 1{:}\ N\}$ is first constructed. The feature encoding in the memory pool is then aggregated and converted into a vector to produce the encoded feature vector in the memory pool: $\{e_i | e_i \in \mathbb{R}^{1\times m}, i = 1{:}\ N\}$.The same operation is performed on the feature encoding in the search region to obtain $e_{decode} \in \mathbb{R}^{1\times m}$. The cosine similarity between the individual feature vectors in the memory pool and the feature vectors in the search region is then calculated according to the following Formula (15):

$$\cos(e_i, e_{decode}) = \frac{e_i \cdot e_{decode}}{\|e_i\| \times \|e_{decode}\|} \tag{15}$$

After calculating the similarity between the feature vectors in the memory pool and the feature vectors in the search region, the frame with the highest similarity is selected, cropped, and used as a dynamic template for subsequent tracking, as shown in Eq. 16:

$$T_d = crop(\text{argmax}(\cos(e_i, e_{decode}))) \tag{16}$$

Where $crop(\cdot)$ is the cropping operation, the dynamic template selection and update process is shown in Figure 5.

Define a template noise degree to determine whether to update the template. Due to the limited computational capability of the space-based target tracking system, continuous updating of the template not only increases the computational burden of the system, but also introduces noise, so when the template noise level increases sharply, the choice is not to update the template. The template noise degree calculation formula is as follows:

$$N_t = \lambda_1 \frac{|F_{\max} - mean(F_{\max})|}{mean(F_{\max})} + \lambda_2 \frac{|apce - mean(apce)|}{mean(apce)} \tag{17}$$

Where λ_1 and λ_2 are constants, usually set to 1 and 2. $F_{\max}$ is the maximum response value in the feature response map R obtained by inter-correlating the target template features with the search frame features; $apce$ is the average peak correlation energy; $mean(F_{\max})$ and $mean(apce)$ represent the mean of the historical frame $F_{\max}$ and $apce$ values. $apce$ is calculated as follows

$$apce = \frac{|F_{\max} - F_{\min}|^2}{mean\left(\sum_{w,h}(F_{w,h} - F_{\min})^2\right)} \tag{18}$$

Where $F_{\max}$, $F_{\min}$ and $F_{w,h}$ represent the maximum response value, the minimum response value, and the response value of the element in row w and column h, respectively, in the feature response map R.

3 Experimental results and discussion

3.1 Experimental setup

The training and testing environment for the algorithm in this article is Ubuntu 18.04 with Python 3.7 and PyTorch 1.2. ResNet-50 with the same parameters as the baseline algorithm SiamCAR served as the backbone network. The training set consisted of ImageNet DET [22], COCO2017 [23], YouTube - BB [24], and LaSOT training set [25]. Randomly chosen images from the training set were used as static template frames and search image frames, respectively, with a 50-pixel distance between them. For the purpose of acquiring a dynamic template, a random frame between the initial template frame and the search image frame was chosen as the prototype dynamic template image frame. The network optimiser was trained for 50 iterations utilizing the ADAM optimiser, with an initial learning rate of 0.01 scaled down to 0.2 times the original every 10 iterations. A and B in the loss function were respectively set to 1 and 3.

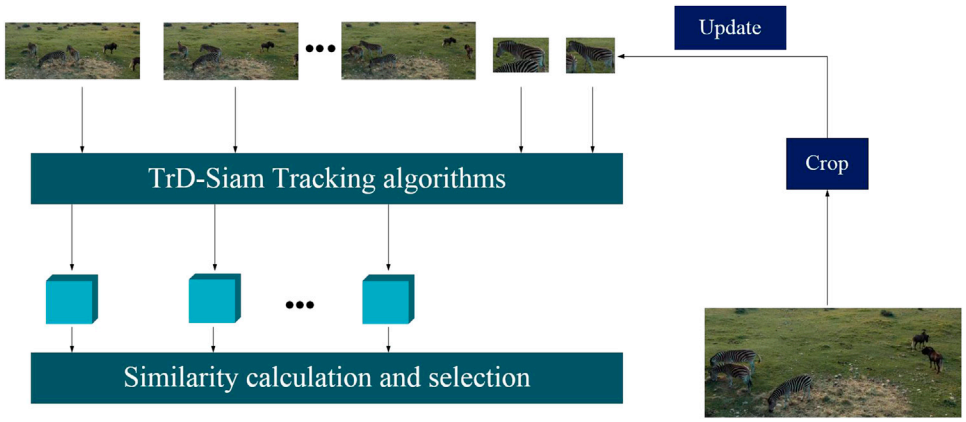

FIGURE 5
Dynamic template update process. Adapted with permission from, "Zebras Grazing by Taryn Elliot, https://www.pexels.com/zh-cn/video/5146558/.

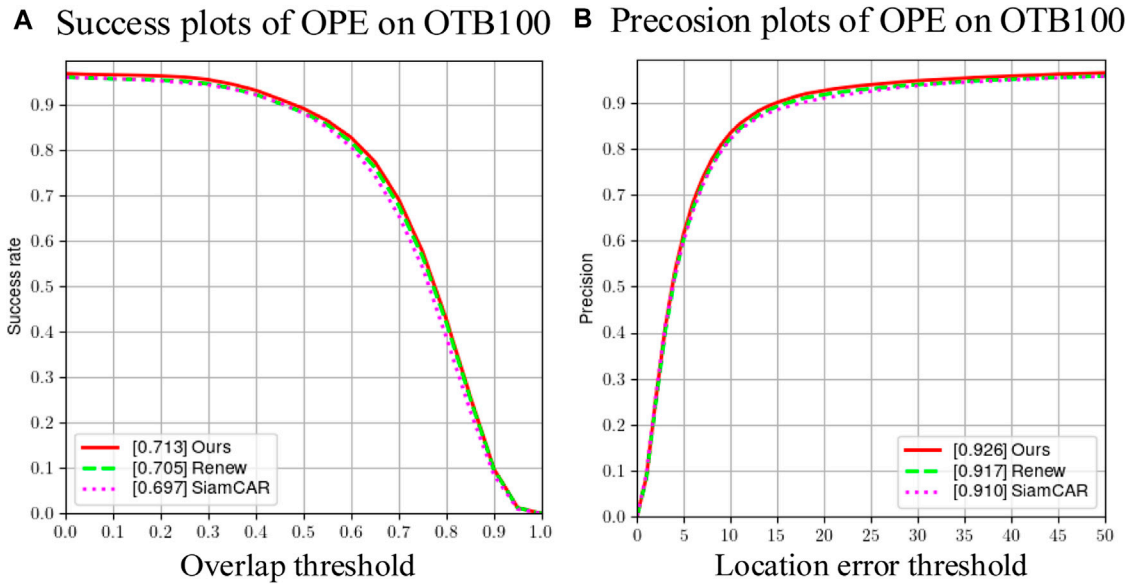

FIGURE 6
Results of ablation experiments. **(A)**Success plots of OPE omn OTB100 **(B)**Precision plots of OPE on OTB100.

3.2 Ablation experiments

In order to assess the efficacy of the Transformer module and the dynamic template update module, three sets of control algorithms were established in this section: SimCAR algorithm, Renew algorithm and Ours algorithm. The results of the ablation analysis using the OTB100 dataset for the three categories of algorithms previously mentioned in Figure 6.

Figure 6 demonstrates that TrD-Siam (Ours) obtains the highest success rate and accuracy, with 71.3% and 92.6%, respectively. In terms of success rate, Renew improves the baseline algorithm SiamCAR by 0.012, while Ours improves it by 0.008 relative to Renew; in terms of accuracy, Renew improves the SiamCAR algorithm by 0.007, while Ours improves it by 0.009 relative to Renew. The structure and template update modules enhance the efficacy of the SiamCAR base algorithm.

3.3 Quantitative experiments

3.3.1 Experimental results for the OTB100 dataset

For the OTB100 dataset, TrD-Siam was compared to several dominant and representative tracking algorithms, such as SiamCAR, SiamFC, SiamRPN, MDNet [26], SiamDW [27], SiamFC++ [28], Ocean [29], SiamBAN [30], and SiamAttn [31].

Figure 7 depicts the success and accuracy profiles of the algorithms derived from the OTB100 dataset using the OPE evaluation strategy. TrD-Siam obtains success and accuracy

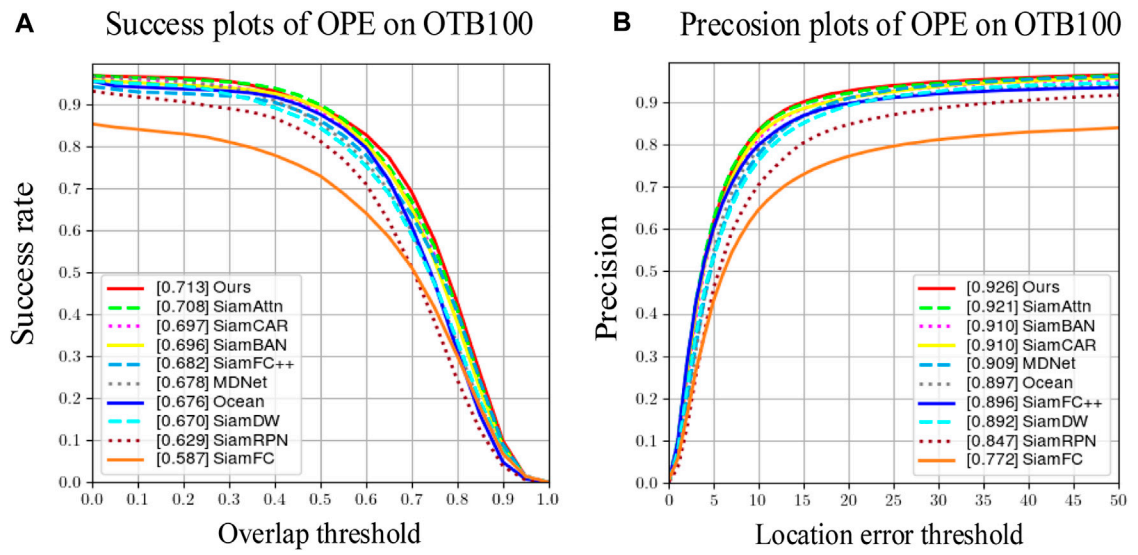

FIGURE 7
Quantitative comparison results on the OTB100 dataset. **(A)** Success plots of OPE omn OTB100 **(B)** Precision plots of OPE on OTB100.

TABLE 1 Performance comparison of the algorithms on the VOT2018 dataset.

	A↑	R↓	EAO ↑	Lost number
SiamFC	0.503	0.585	0.187	125.0
SiamRPN	0.586	0.276	0.383	59.8
SiamRPN++	0.601	0.234	0.415	50.0
SiamBAN	0.590	0.178	0.447	38.0
SiamCAR	0.578	0.197	0.427	42.0
TrD-Siam	0.599	0.169	0.467	36.0

rates of 71.3% and 92.6%, respectively, substantially outperforming the other nine algorithms compared. TrD-Siam's tracking success rate and precision are both enhanced by 1.6% compared to SiamCAR. The aforementioned results demonstrate that the dynamic template and Transformer structure's effectiveness in this chapter's algorithm TrD-Siam mitigate the degradation of tracking performance when the target's appearance drastically changes.

3.3.2 Experimental results for the VOT2018 dataset

Five sets of tracking algorithms, SiamCAR, SiamFC, SiamRPN, SiamRPN++, and SiamBAN, were introduced and evaluated using three evaluation metrics from the VOT2018 benchmark: A (Accuracy), R (Robustness), and EAO (Expected Average Overlap). Additionally, the Lost Number was utilized as a secondary metric. A greater value of A indicates that the algorithm is more precise, a lesser value of R indicates that the tracking algorithm is more robust, and a greater value of EAO indicates that the tracking algorithm is more exhaustive. Table 1 provides the results.

As shown in Table 1, TrD-Siam demonstrated excellent tracking performance, achieving the second maximum accuracy (0.599), robustness (0.169), and EAO (0.44). Compared to the baseline algorithm SiamCAR, TrD-Siam demonstrated an enhancement of 1.2%, 1.8%, and 4% in accuracy, robustness, and expected mean expected overlap, respectively, demonstrating its efficacy.

FIGURE 8
Ground-based large aperture telescope.

3.3.3 Ground-based large-aperture telescope experiment

To visualize the tracking results, optoelectronic detection equipment is used to track long-distance trailing targets, and Figure 8 shows the ground-based large-aperture optoelectronic detection equipment, and the results of the visualisation are presented in Figure 8. For qualitative analysis, TrD-Siam was compared to SiamRPN, SiamFC.

As illustrated in Figure 9. Red, ground truth; bule, the proposed TrD-Siam; green, SiamRPN; yellow, SiamFC. The data collected by the

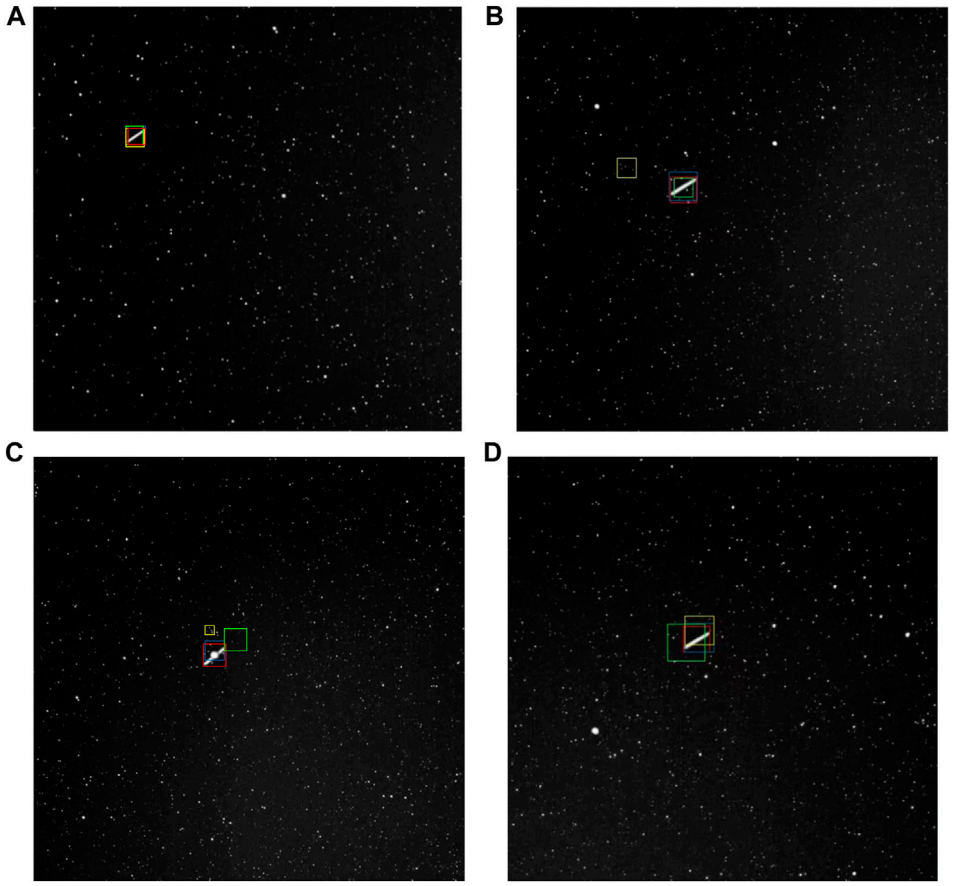

FIGURE 9
Visualisation of tracking results for selected video sequences **(A)** Frame 25 **(B)** Frame 37 **(C)** Frame 100 **(D)** Frame 123.

ground-based large-aperture optoelectronic detection detection equipment, at 25 frames, all three algorithms can successfully track the target, at 37 frames, due to the interference of stars around the target, the SiamFC algorithm loses the tracking target, at the same time, the SiamRPN and the proposed TrD-Siam successfully track the target, at 100 frames, due to the target being blocked by the stars, the SiamRPN and the SiamFC algorithms lose tracking the target and the algorithm of this article algorithm can successfully track the target when the target is obscured, at 123 frames, all algorithms realize to track the target, in summary, the algorithm of this article has better tracking performance.

4 Conclusion

In this article, we carry out research on the poor target tracking performance of optoelectronic detection system in space-based background, and propose a TrD-Siam algorithm based on image processing and transformer, which improves the overall quality of the star atlas by using image processing techniques for the problems of long distance and weak energy of space-based targets. For the problems of high false alarm rate and low efficiency of space-based target tracking, transformer is introduced into the SiamCAR framework to enhance the image feature extraction capability. Comparison experiments are conducted on OTB100 and VOT100 datasets respectively, and the experimental results prove that the algorithm in this article performs better in the three evaluation indexes of accuracy, robustness and EAO. And the TrD-Siam algorithm is verified by the data collected by the ground-based large aperture telescope, and compared with the comparison algorithm, the TrD-Siam algorithm has a better tracking performance and has a good application value.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding authors.

Author contributions

RZ: Writing–original draft, Writing–review and editing, Data curation. JSL: Methodology, Software, Writing–original draft. QF: Writing–review and editing, Data curation, Project administration. XYW: Writing–review and editing, Project administration. HC: Writing–review and editing, Investigation, Methodology. GYW: Writing–review and editing, Resources. TZ: Writing–review and editing, resources. HDS: Writing–review and editing, supervision.

YCL: Writing–review and editing, Supervision, Visualization. HLJ: Writing–review and editing, Visualization.

Funding

The author(s) declare financial support was received for the research, authorship, and/or publication of this article. Chinese Academy of Engineering (Nos. 2018-XY-11 and 2023-XY-10).

Acknowledgments

Thanks the Project of Chinese Academy of Engineering for help identifying collaborators for this work.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

1. Fu Q, Zhao F, Zhu R, Liu Z, Li Y. Research on the intersection angle measurement and positioning accuracy of a photoelectric theodolite. *Front Phys* (2023) 10:1121050. doi:10.3389/fphy.2022.1121050

2. Stramacchia M, Colombo C, Bernelli-Zazzera F. Distant retrograde orbits for space-based near earth objects detection. *Adv Space Res* (2016) 58(6):967–88. doi:10.1016/j.asr.2016.05.053

3. Li M, Yan C, Hu C, Liu C, Xu L. Space target detection in complicated situations for wide-field surveillance. *IEEE Access* (2019) 7:123658–70. doi:10.1109/ACCESS.2019.2938454

4. Zhang H, Zhang W, Jiang Z. Space object, high-resolution, optical imaging simulation of space-based systems. *Proc SPIE - Int Soc Opt Eng* (2012) 8385:290–6. doi:10.1117/12.918368

5. Zhang X, Xiang J, Zhang Y. Space object detection in video satellite images using motion information. *Int J Aerospace Eng* (2017) 2017:1–9. doi:10.1155/2017/1024529

6. Zhang B, Hou X, Yang Y, Zhou J, Xu S. Variational Bayesian cardinalized probability hypothesis density filter for robust underwater multi-target direction-of-arrival tracking with uncertain measurement noise. *Front Phys* (2023) 11:1142400. doi:10.3389/fphy.2023.1142400

7. Xi J, Wen D, Ersoy OK, Yi H, Yao D, Song Z, et al. Space debris detection in optical image sequences. *Appl Opt* (2016) 55(28):7929–40. doi:10.1364/AO.55.007929

8. Anderson JC, Downs GS, Trepagnier PC. <title>Signal processor for space-based visible sensing</title>. *Surveill Tech* (1991) 1479:78–92. doi:10.1117/12.44523

9. Tonissen SM, Evans RJ. Peformance of dynamic programming techniques for track-before-detect. *IEEE Trans aerospace Electron Syst* (1996) 32(4):1440–51. doi:10.1109/7.543865

10. Tzannes AP, Brooks DH. Temporal filters for point target detection in IR imagery. In: Proceedings of the Infrared Technology and Applications XXIII; August 1997; Orlando, FL, USA, 3061 (1997). p. 508–20. doi:10.1117/12.280370

11. Blostein SD, Huang TS. Detecting small, moving objects in image sequences using sequential hypothesis testing. *IEEE Transactions Signal Process.* (1991) 39(7):1611–29. doi:10.1109/78.134399

12. Zhang L, Gonzalez-Garcia A, Weijer JVD, Danelljan M, Khan FS Learning the model update for siamese trackers. In: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); October 2019; Seoul, Korea (2019). p. 4010–9. doi:10.1109/iccv.2019.00411

13. Li B, Yan J, Wu W, Zhu Z, Hu X High performance visual tracking with siamese region proposal network. In: Proceedings of the IEEE conference on computer vision and pattern recognition; June 2018; Salt Lake City, UT, USA (2018). p. 8971–80. doi:10.1109/cvpr.2018.00935

14. Bertinetto L, Valmadre J, Henriques JF, Vedaldi A, Torr PH, et al. Fully-convolutional siamese networks for object tracking. In: Proceedings of the Computer Vision–ECCV 2016 Workshops; October 2016; Amsterdam, The Netherlands. Springer International Publishing (2016). p. 850–65.

15. Li B, Wu W, Wang Q, Zhang F, Xing J, Yan J, et al. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; June 2019; Long Beach, CA, USA (2019). p. 4282–91. doi:10.1109/cvpr.2019.00441

16. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. *Adv Neural Inf Process Syst* (2017) 30. doi:10.48550/arXiv.1706.03762

17. Cohen M, Lu W. A diffusion-based method for removing background stars from astronomical images. *Astron Comput* (2021) 37:100507. doi:10.1016/j.ascom.2021.100507

18. Subhashini D, Dutt VSI. An innovative hybrid technique for road extraction from noisy satellite images. *Mater Today Proc* (2022) 60:1229–33. doi:10.1016/j.matpr.2021.08.114

19. Liu J, Duan J, Hao Y, Chen G, Zhang H, Zheng Y. Polarization image demosaicing and RGB image enhancement for a color polarization sparse focal plane array. *Opt Express* (2023) 31(14):23475–90. doi:10.1364/oe.494836

20. Guo D, Wang J, Cui Y, Wang Z, Chen S SiamCAR: Siamese fully convolutional classification and regression for visual tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; June 2020; Seattle, WA, USA (2020). p. 6269–77.

21. Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S End-to-end object detection with transformers. In: Proceedings of the European conference on computer vision; August 2020; Glasgow, UK. Springer International Publishing (2020). p. 213–29.

22. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. Imagenet large scale visual recognition challenge. *Int J Comput Vis* (2015) 115:211–52. doi:10.1007/s11263-015-0816-y

23. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft coco: Common objects in context. In: Proceedings of the Computer Vision–ECCV 2014: 13th European Conference; September 2014; Zurich, Switzerland. Springer International Publishing (2014). p. 740–55. doi:10.1007/978-3-319-10602-1_48

24. Real E, Shlens J, Mazzocchi S, Pan X, Vanhoucke V YouTube-BoundingBoxes: a large high-precision human -annotated data set for object detection in video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; July 2017; Hawaii, USA (2017). p. 7464–73. doi:10.1109/cvpr.2017.789

25. Fan H, Lin L, Yang F, Chu P, Deng G, Yu S, et al. Lasot: A high-quality benchmark for large-scale single object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; June 2019; Long Beach, CA, USA (2019). p. 5374–83.

26. Jung I, Son J, Baek M, Han B Real-time mdnet. In: Proceedings of the European conference on computer vision (ECCV); September 2018; Munich, Germany (2018). p. 83–98.

27. Zhang Z, Peng H. Deeper and wider siamese networks for real-time visual tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; June 2019; Long Beach, California (2019). p. 4591–600. doi:10.1109/cvpr.2019.00472

28. Xu Y, Wang Z, Li Z, Yuan Y, Yu G (2020). Siamfc++: Towards robust and accurate visual tracking with target estimation guidelines. In *Proc AAAI Conf Artif intelligence* 34, 7, 12549–56. doi:10.1609/aaai.v34i07.6944

29. Zhang Z, Peng H, Fu J, Li B, Hu W Ocean: Object-aware anchor-free tracking. In: Computer Vision–ECCV 2020: 16th European Conference; August 2020; Glasgow, UK. Springer International Publishing (2020). p. 771–87.

30. Chen Z, Zhong B, Li G, Zhang S, Ji R Siamese box adaptive network for visual tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; June 2020; Seattle, WA, USA (2020). p. 6668–77. doi:10.1109/cvpr42600.2020.00670

31. Yu Y, Xiong Y, Huang W, Scott MR Deformable siamese attention networks for visual object tracking. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; June 2020; Seattle, WA, USA (2020). p. 6728–37. doi:10.1109/cvpr42600.2020.00676