frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 11 January 2023
DOI 10.3389/fnbot.2022.1094892

OPEN ACCESS

EDITED BY
Hang Su,
Fondazione Politecnico di Milano, Italy

REVIEWED BY
Yang Yang,
Yunnan Normal University, China
Wen Qi,
Politecnico di Milano, Italy

*CORRESPONDENCE
Xian Wang
✉ 15111388435@163.com

RECEIVED 10 November 2022
ACCEPTED 23 December 2022
PUBLISHED 11 January 2023

CITATION
Sun D, Wang X, Man Y, Deng N and Peng Z (2023) A Siamese tracker with "dynamic–static" dual-template fusion and dynamic template adaptive update.
Front. Neurorobot. 16:1094892.
doi: 10.3389/fnbot.2022.1094892

A Siamese tracker with "dynamic–static" dual-template fusion and dynamic template adaptive update

Dongyue Sun, Xian Wang*, Yingjie Man, Ningdao Deng and Zhaoxin Peng

School of Mechanical Engineering, Hunan University of Science and Technology, Xiangtan, China

In recent years, visual tracking algorithms based on Siamese networks have attracted attention for their desirable balance between speed and accuracy. The performance of such tracking methods relies heavily on target templates. Static templates cannot cope with the adverse effects of target appearance change. The dynamic template method, with a template update mechanism, can adapt to the change in target appearance well, but it also causes new problems, which may lead the template to be polluted by noise. Based on the DaSiamRPN and UpdateNet template update networks, a Siamese tracker with "dynamic–static" dual-template fusion and dynamic template adaptive update is proposed in this paper. The new method combines a static template and a dynamic template that is updated in real time for object tracking. An adaptive update strategy was adopted when updating the dynamic template, which can not only help adjust to the changes in the object appearance, but also suppress the adverse effects of noise interference and contamination of the template. The experimental results showed that the robustness and EAO of the proposed method were 23% and 9.0% higher than those of the basic algorithm on the VOT2016 dataset, respectively, and that the precision and success were increased by 0.8 and 0.4% on the OTB100 dataset, respectively. The most comprehensive real-time tracking performance was obtained for the above two large public datasets.

KEYWORDS

object tracking, Siamese network, template update, dynamic–static dual-template fusion, deep learning

1. Introduction and motivation

Video object tracking, which refers to continuously tracking the state of an object in subsequent frame sequences by using the initial position and scale information of the object, is the basis for high-level visual tasks such as visual inspection, visual navigation, and visual servo (Nousi et al., 2020; Wang et al., 2020; Karakostas et al., 2021; Sun et al., 2021). In engineering practice, interference such as changes in the posture and scale of the object, noise interference, background occlusion, or variation of light conditions may lead to tracking failure, so object tracking remains a challenging task (Zhang et al., 2020; Zhang H. et al., 2021; Liu et al., 2022).

Object tracking methods can be roughly divided into generative methods and discriminative methods (Zhang Y. et al., 2021; Dunnhofer et al., 2022). Generative methods first build the model of the object and then search for the area which is most similar to the object in subsequent frames through iterating to achieve target positioning. Discriminative tracking algorithms transform the object tracking problem into a binary classification problem about the object and the background, and find the predicted object position by training a classifier to distinguish the object from the background. In general, generative methods do not rely on training samples and are easy to implement, while discriminative ones are stronger in robustness.

To handle various challenges, a significant amount of research has been focused on visual tracking in recent years. With the continuous advancement of machine learning and signal processing technology, algorithms based on correlation filters (Zhang J. et al., 2022; Ly et al., 2021) and deep learning (Haisheng et al., 2019; Voigtlaender et al., 2020; Tan et al., 2021; Zhang X. et al., 2022) have gradually replaced traditional methods as the mainstream object tracking algorithms. Both of these methods are discriminative methods. The basic idea of correlation filter tracking is to use a designed filter template for the correlation operation with the target candidate area, and the position where the maximum output response is located is the target position of the current frame. Before the advent of object tracking algorithms based on correlation filters, all tracking operations were completed in the time domain with a large amount of data and a long period of calculation time. However, the object tracking algorithms based on correlation filters convert the operation from the time domain to the frequency domain, reducing the amount of computation while ensuring the integrity of data. Early correlation filter object tracking algorithms include the MOSSE (Minimum Output Sum of Squared Error Filter) algorithm (Bolme et al., 2010), the CSK (Circulant Structure with Kernels) algorithm (Henriques et al., 2012), the KCF (Kernelized Correlation Filter) algorithm (Henriques et al., 2015), and the SAMF (Scale Adaptive Multiple Feature) algorithm (Li and Zhu, 2014), etc. All the above algorithms use manual features. Later, people began to introduce deep learning into correlation filtering, mining more robust depth features from the original data to replace traditional manual features, thereby further improving the robustness of correlation filtering algorithms. Typical object tracking algorithms which combine correlation filtering with deep learning include the Deep SRDCF (Convolutional Features for Correlation Filter Based Visual Tracking) algorithm (Danelljan et al., 2015a), the C-COT (Continuous Convolution Operators) algorithm (Danelljan et al., 2016a), and the ECO (Efficient Convolution Operators for Tracking) algorithm (Danelljan et al., 2017a), etc.

The deep learning object tracking algorithm based on the Siamese network has received extensive attention due to its good performance in testing various benchmark tracking datasets. SiamFC (Fully Convolutional Siamese Networks for Object Tracking; Bertinetto et al., 2016b), an earlier object tracking algorithm based on a fully convolutional Siamese network, uses a fully convolutional network structure to learn the similarity measurement between the target area and the search area, thus viewing tracking as a problem of searching for target objects across the entire image. Based on SiamFC, Bo et al. (2018) proposed SiamRPN (High Performance Visual Tracking with Siamese Region Proposal Network), an object tracking algorithm based on the region proposal network. The method, with a classification network for foreground and background estimation and a regression network for anchor bounding box correction included, estimates the position and size of the object through a bounding box with a variable aspect ratio, so that a more accurate bounding box can be obtained. Zhu et al. (2018) proposed a Siamese network tracking algorithm based on distractor aware, DaSiamRPN (Distractor-Aware Siamese Networks for Visual Object Tracking), which improves the discrimination ability of the model by introducing a distractor-aware module. Based on SiamFC or SiamRPN, Zhang and Peng (2019) proposed the SiamDW (Deeper and Wider Siamese Networks for Real-Time Visual Tracking) algorithm, which further improves tracking accuracy and robustness by introducing internal clipping units of residual blocks into deeper and wider networks. The SiamMask (Fast Online Object Tracking and Segmentation: A Unifying Approach) algorithm proposed by Wang et al. (2019) can simultaneously implement video object tracking and video object segmentation, and a predictive bounding box of an adaptive mask can be obtained during object tracking, which greatly improves the accuracy of tracking.

The above tracking algorithms based on the Siamese network achieved the optimal performance at that time, but all these algorithms use the position of the object in the first frame image as a fixed template throughout the tracking process, which makes them unable to deal well with the adverse effects of a change in the appearance of the object. In response to this problem, a template update mechanism was introduced into object tracking. Danelljan et al. (2015b) improved tracking efficiency by building a Gaussian hybrid model of the training sample together with a conservative template update strategy (updated every five frames). Galoogahi et al. (2017) proposed a background-aware correlation filter algorithm that achieves object tracking with high accuracy and real-time performance by intensively extracting real negative samples from the background to update the filter. The above studies use linear interpolation for template updating, which are prone to causing tracking drift, resulting in tracking failures. The UpdateNet (Learning the Model Update for Siamese Trackers) algorithm proposed by Zhang et al. (2019) implements template tracking through a trained convolutional neural network, which greatly improves the tracking performance. Wang W. et al.

(2020) introduced a sparse update mechanism in the tracking framework, which could help adaptively select the appropriate level for object tracking. Such an update strategy reduces the complexity of the model to a certain extent. Huang et al. (2019) proposed a correlation filter tracker based on transfer learning, which updates the model by migrating historical filter data to improve the robustness of the tracker. Han et al. (2020) proposed a spatial regularization update method with content perception, which adjusts the weight distribution map by optimizing the constraint problem to better adapt to the changes in the object and background, so that reliable tracking can be achieved. Zhang Y. et al. (2022) proposed a dual-stream collaborative tracking algorithm combined with reliable memory updates, which realizes real-time tracking speed and a superior tracking performance.

The introduction of the template update mechanism effectively improves the performance of the Siamese network object tracking algorithm. However, there are various noise interferences in actual application situations. The template update mechanism brings new problems while better adapting to the changes in the appearance of the target, causing the template to be polluted by noise interference (Su et al., 2022).

In order to further improve the performance of object tracking algorithms, a Siamese tracker with the "dynamic–static" dual-template fusion and dynamic template adaptive update is proposed based on the DaSiamRPN and UpdateNet template update network in this paper. The new method combines a static template and a dynamic template that is updated in real time for object tracking. An adaptive update strategy is applied when updating the dynamic template, which helps decide whether to update the dynamic template by using the similarity between the tracking result of the current frame and the dynamic template, as well as the no reference evaluation results of the current frame image, to judge the necessity of updating the template and the possibility of template pollution if the template is updated. In this way, it can suppress the adverse effects of noise interference and pollution of the template while adapting to changes in the object appearance. The new method achieved the best comprehensive and real-time tracking performance on the two large public datasets, VOT2016 and OTB100.

2. Proposed method

Based on the DaSiamRPN algorithm, the proposed method introduces the "dynamic–static" dual-template fusion and dynamic template adaptive update mechanism. Its overall network architecture is shown in Figure 1, including the feature extraction module, RPN (Region Proposal Network), dynamic template update module, and the "dynamic–static" dual-template fusion module. In order to ensure the real-time performance of target tracking, the AlexNet shallow network used by the DaSiamRPN algorithm was also adopted by the feature extraction module. RPN consists of a classification network for foreground-background estimation and a regression network for anchor bounding box correction. It uses a bounding box with variable aspect ratio to estimate the position and size of the object, thus obtaining a more accurate bounding box. Based on the UpdateNet network, the dynamic template update module designed an adaptive update strategy which determined whether to implement template update network according to two quantitative indicators, so as to provide the optimal template for the next frame: one was the similarity between the tracking result of the current frame and the dynamic template; the other was the no reference evaluation result of the current frame image. The dual-template fusion module takes into account the advantages of the initial template and the dynamic template through the weighted fusion of the static template and dynamic template, which reduces the robustness of the tracker.

2.1. The feature extraction module and RPN

Two neural networks that shared weights formed a Siamese network for object feature extraction: One was the template branch for extracting features from the target template frame *TF* and the other was the search branch for extracting the characteristics of the search frame *SF*. The two branches shared parameters in a convolutional neural network.

The feature extraction module applied the AlexNet network used by the DaSiamRPN algorithm, and the RPN was also consistent with that of the DaSiamRPN algorithm. The feature map obtained by the feature extraction module was the input of RPN. RPN was used to obtain a more accurate target candidate box, including the classification branch $\varphi_{\text{cls}}(\cdot)$ and the regression branch $\varphi_{reg}(\cdot)$.

Firstly, the first frame image of the video was taken as the target template (as shown in the blue bounding box in the video frame in the upper left corner of Figure 1), then the anchor box was generated by the RPN, and the Softmax classifier was used to extract the positive anchors to obtain more accurate $A^{cls}_{\text{W}\times H\times 2K}$ in the classification branch. The classification branch determined whether each location was foreground or background:

$$A^{cls}_{\text{W}\times H\times 2K}\varphi_{\text{cls}}(SF) * \varphi_{cls}(TF) \tag{1}$$

In Equation 1, $A^{cls}_{\text{W}\times H\times 2K}$ represents the classification feature map where the target and background score information of each predefined anchor box is stored. W and H represent the width and height of the feature map, respectively; K is the number of anchor boxes; *SF* represents the search frame; *TF* represents the template frame; and $*$ refers to the cross-correlation operation.

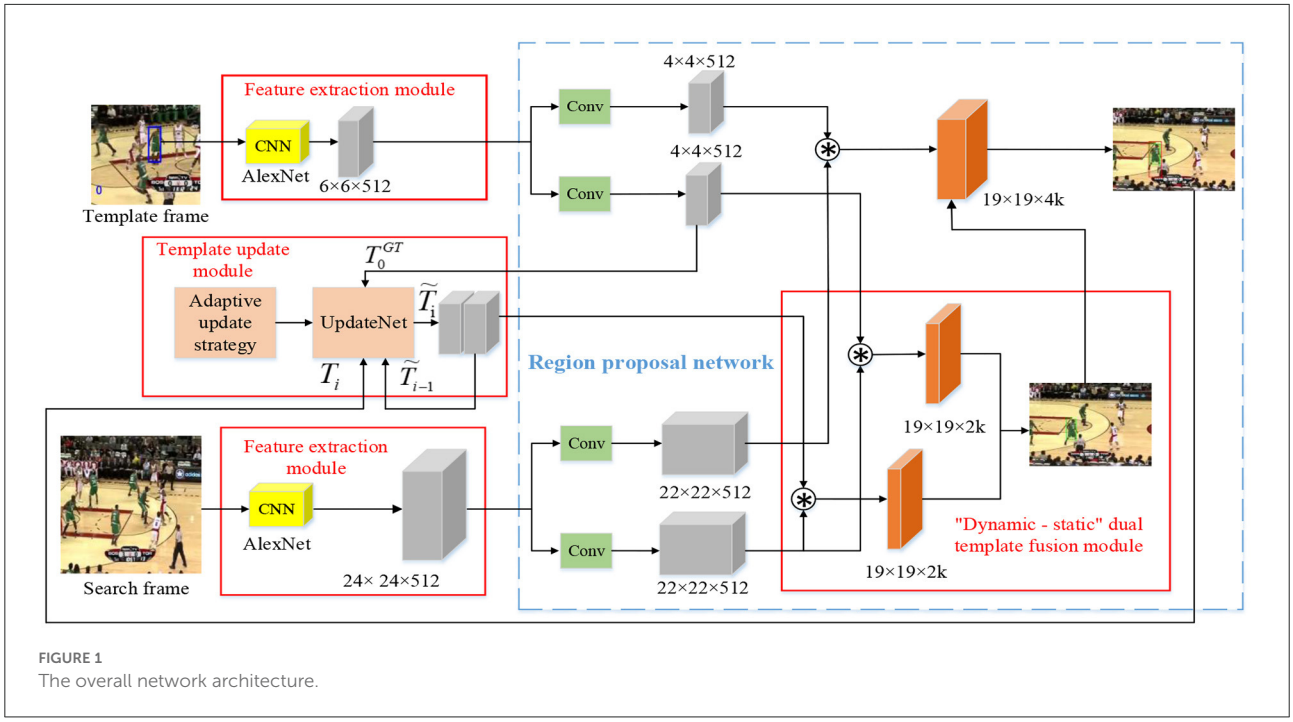

FIGURE 1
The overall network architecture.

Secondly, the non-maximum suppression (NMS) is used to determine the predefined anchor box (as shown in the green bounding box in the video frame at the lower right corner of Figure 1) from the feature map $A^{cls}_{W\times H\times 2K}$ obtained in the classification branch. x^{an}, y^{an}, w^{an}, and h^{an} represent the center coordinates, width, and height of the predefined anchor box, respectively.

Then, the coordinate offset of the center point of the corresponding anchor box (dx^{reg},dy^{reg}) and the length and width ratio of this anchor box to the real target box (dw^{reg}, dh^{reg}) are selected from the feature map obtained in the regression branch. The regression branch calculates all the target bounding boxes that may exist at each location:

$$A^{reg}_{W\times H\times 4K}\varphi_{\text{reg}}(SF) * \varphi_{reg}(TF) \tag{2}$$

In Equation 2, $A^{reg}_{W\times H\times 4K}$ represents the regression feature map which stores the information such as the coordinate offset of the center point of the predefined anchor box and the width and height ratio of the predefined anchor box to the real target box.

Finally, the prediction box (as shown in the red bounding box in the video frame in the upper right corner of Figure 1) is obtained through the bounding box coordinate regression of the predefined anchor box. x^{pre} represents the abscissa of the center coordinate of the prediction box:

$$x^{pre}x^{an} + dx^{reg} \times w^{an} \tag{3}$$

y^{pre} represents the ordinate of the center coordinate of the prediction box:

$$y^{pre}y^{an} + dy^{reg} \times h^{an} \tag{4}$$

w^{pre} represents the width of the prediction box:

$$w^{pre}w^{an} \times e^{dw^{reg}} \tag{5}$$

h^{pre} represents the height of the prediction box:

$$h^{pre}h^{an} \times e^{dh^{reg}} \tag{6}$$

2.2. The dynamic template update module

2.2.1. The UpdateNet network

The convolutional neural network of UpdateNet was used to implement the dynamic template update, and its network framework is shown in Figure 2 and can be represented by the following formula:

$$\tilde{T}_i = \zeta\left(T_0^{GT}, \tilde{T}_{i-1}, T_i\right) \tag{7}$$

where $\tilde{T}_i$ is the cumulative template updated after the current frame tracking finished; $\zeta(\cdot)$ is the function of the UpdateNet network; T_0^{GT} is the template given in the first frame; $\tilde{T}_{i-1}$ is the dynamic template updated after the previous frame image target tracking finished; and T_i is the object tracking result for

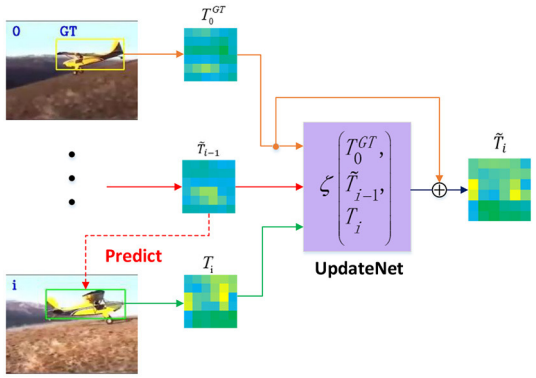

FIGURE 2
The network framework of UpdateNet.

the current frame. A residual learning strategy is adopted by this network, which adds a jump connection from the template given in the first frame to the output. For the first frame of the object tracking image sequence, both T_i and $\tilde{T}_{i-1}$ are set to T_0^{GT}. New tracking results are used by UpdateNet to update the template to better adapt to the changes in the object appearance.

2.2.2. Adaptive update strategy

The adaptive update strategy proposed in this paper determined whether to update dynamic templates by judging the necessity of updating the template and the possibility of template pollution if the template was updated according to two quantitative indicators: one was the similarity between each image region and the other was the no reference evaluation result of the image quality. In this way, it was able to suppress the adverse effects of noise interference and pollution of the template while adapting to the changes in the object appearance. The similarity between two image blocks was quantified by the $L1$ norm between them:

$$S = \|\mathrm{M} - \mathrm{N}\|_1 \tag{8}$$

In Equation 8, $\|\cdot\|$ is the $L1$ norm; M and N are the two image blocks whose similarity is compared; and S is the quantitative results of the similarity between the two image blocks. The smaller the value of S, the more similar the two image blocks are.

In this paper, NIQE (Natural Image Quality Evaluator; Mittal et al., 2013) was used to evaluate the image quality. The Mahalanobis distance between the MVG (Multivariate Gaussian Model) obtained through natural image feature fitting and that obtained through the feature fitting of the image to be measured is used by this method to indicate the image quality:

$$D = \sqrt{(\mathrm{v}_1 - v_2)^T \left(\frac{\sum_1 + \sum_2}{2}\right)^{-1} (\mathrm{v}_1 - v_2)} \tag{9}$$

where D represents the quantitative evaluation result of the image quality, and the smaller its value is, the better the image quality is. v_1 and v_2 represent the mean vectors of the MVG model of the natural image and that of the image to be tested, respectively. $\sum_1$ and $\sum_2$ are the covariance matrices of the MVG model of the natural image and that of the image to be tested, respectively.

With the above two quantitative evaluation indicators, the whole process of the proposed adaptive update strategy is shown in Figure 3. In order to accurately determine whether the template should be updated, a total of three similarity thresholds, λ_1, λ_2, λ_3, and an image quality threshold, δ, are set. Among them, λ_1 is the abnormal threshold value of the target appearance, and when the similarity between the tracking object of the current frame and the dynamic template S_c is $> \lambda_1$, it indicates that the target appearance characteristics have changed beyond the normal range. λ_2 is the obvious-change threshold value of the target appearance, and when S_c is $> \lambda_2$, it indicates that the target appearance characteristics have changed significantly. λ_3 is the changing threshold value of the target appearance and when S_c is $> \lambda_3$, it indicates that there are some changes in the target appearance characteristics that have a certain impact on the target tracking, and conversely, the target appearance characteristics are almost unchanged. δ is the good image quality threshold, and when the quality evaluation result of the current target sub-image block D_c is $<\delta$, it indicates that the target sub-image block of the current frame is high in quality and that the noise interference almost does not adversely affect the image acquisition.

In this process, the threshold value λ_1 is first used to judge whether there is a serious abnormity in the current image acquisition and target tracking. If S_c is $\geq\lambda_1$, the dynamic template will not be updated to avoid the template being contaminated by abnormal data.

If S_c is $< \lambda_1$, it is supposed to further judge whether it is $\geq\lambda_2$. If so, it indicates that the object appearance has been changed obviously under normal circumstances and in order to cope with these changes, the template must be updated.

If S_c is $< \lambda_2$, it needs to continue to judge whether S_c is $\geq\lambda_3$. At this time, if S_c is smaller than λ_3, it indicates that the object appearance feature is almost unchanged so there is no need to update the dynamic template. If S_c is $\geq\lambda_3$, it indicates that there are some changes in the object appearance under normal circumstances that have a certain impact on the target tracking, thought the changes are not significant enough to update the template. Then, there is a need to judge whether D_c is $\geq\delta$. If so, it indicates that the image is poor in quality so the dynamic template will not be updated to avoid template pollution, and conversely, the dynamic template will be updated.

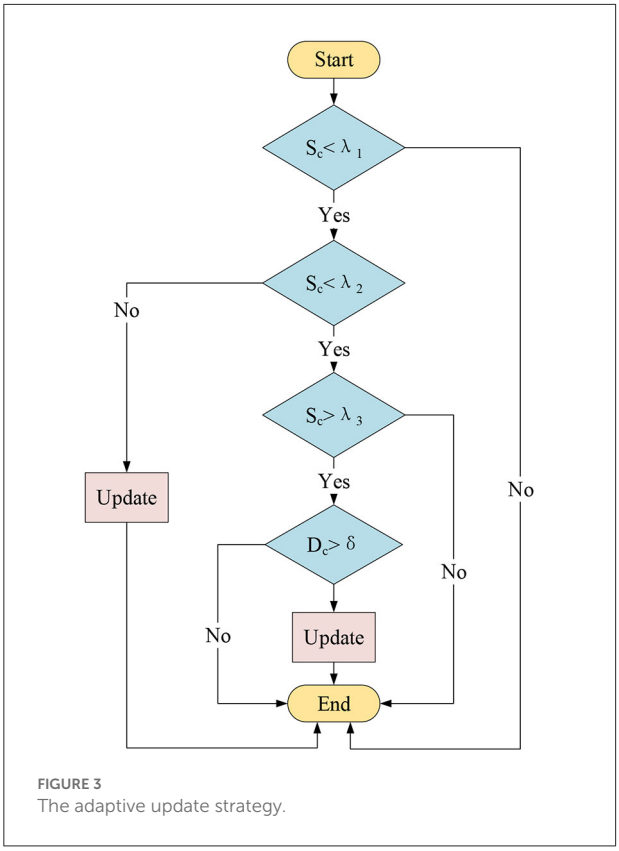

FIGURE 3
The adaptive update strategy.

2.3. "Dynamic-static" dual-template fusion module

The method in this paper fuses the feature map obtained by using static and dynamic templates in the RPN classification branch.

$$A = \alpha A_{W\times H\times 2K}{}^{cls} + (1-\alpha)\, A'_{W\times H\times 2K}{}^{cls} \tag{10}$$

In Equation 10, A is the feature map of the classification branch after fusion and $A'_{W\times H\times 2K}{}^{cls}$ represents the feature map obtained from the static template in the classification branch:

$$A'^{cls}_{W\times H\times 2K}\varphi_{\text{cls}}\,(SF) * \varphi_{cls}\,(TF_0) \tag{11}$$

where $\varphi_{cls}\,(TF_0)$ represents the feature of the initial frame in the classification branch, and W, H, K, and SF have the same meaning as in Equation 11.

$\alpha \epsilon [0, 1]$, which represents the weight coefficient during fusion:

$$\alpha = \frac{S_o}{S_d + S_o} \tag{12}$$

where S_d represents the quantization result of the similarity between the dynamic template of the previous frame and the object obtained by the previous frame tracking, and S_o is the similarity between the object obtained by the previous frame tracking and the templates of the initial frame.

The introduction of this module can give the object tracking algorithm the advantage of using static and dynamic templates, and make it suppress the adverse effects of noise interference and contamination of the template while adapting to the changes in the object appearance.

3. Experiment results and analysis

3.1. Implementation details and parameters

All the algorithms in this experiment were carried out using the Pytorch 1.7.1 deep learning platform on the deep learning workstation. The deep learning workstation uses the Windows 10 operating system with Intel® Xeon(R) Gold 6139M CPU, 128 GB RAM, and the Nvidia GTX3080 GPU, which is equipped to perform parallel computing.

The ImageNet VID (Russakovsky et al., 2015), YouTube—BoundingBoxes (Real et al., 2017), ImageNet DET, and COCO (Lin et al., 2014) datasets were used as training sets to train the feature extraction network to learn the similarity between objects. Ten sequences were randomly selected from the LaSOT (Fan et al., 2019) dataset as training sets to train the template update network. A batch of 64 samples were trained in each training phase, with a total of 50 epochs, and the learning rate of each epoch decreases logarithmically from 10^{-7} to 10^{-8}. The stochastic gradient descent method with momentum was used for optimization, and in the actual optimization process, the difficulty of optimization gradually increased as the depth of feature extraction increased. In order to prevent the loss function of each layer from gradient explosion, the momentum was set to 0.9; the weight decay parameter was set to 0.0005; and the size of the template image and the search image were set to 127 × 127 and 271 × 271, respectively.

In addition, the average similarity between the tracking object of the current frame and the dynamic template S_c obtained by the training template update network was 0.0017, the variance was 0.0023, the standard deviation was 0.048, and the maximum value was 7. After many studies, we found that the threshold value of appearance abnormality λ_1 should be more than four times the standard deviation. The threshold of appearance variation λ_2 can be about 0.01 times the standard deviation. The appearance change threshold λ_3 can be about 0.005 times the standard deviation. By calculating the image quality of more than 20,000 frames, we found that the image quality below 15 is significantly better. Therefore, when using the method in this paper, the thresholds λ_1, λ_2, λ_3, and δ were set to 0.5, 6×10^{-4}, 1×10^{-4}, and 15.

3.2. Evaluation on benchmarks

3.2.1. VOT2016 benchmark experiment

The VOT series dataset is one of the most used datasets for visual object tracking. The VOT2016 dataset includes 60 video sequences with multiple challenges, each of which is labeled with the visual properties of that sequence. There are six properties: camera motion, variation of light condition, occlusion, size change, motion change, and no degradation. The performance evaluation indicators used by the dataset include Expected Average Overlap (EAO), accuracy (A, Accuracy), and robustness (R, Robustness). Among them, EAO can better reflect the comprehensive performance of the tracking algorithm and is generally considered as the most important indicator. The accuracy, which describes the average overlap score, is obtained by calculating the ratio of intersection over union (IOU) of the prediction box and the truth box. Robustness is an indicator used to evaluate the stability of the tracking algorithm and the smaller its value is, the more stable the algorithm is.

The performance of this method and the current mainstream object tracking algorithms (SiamFC, SiamRPN, SiamDW-RPN, SiamMask, C-COT, DaSiamRPN, and UpdateNet) in the VOT2016 dataset is shown in Table 1. As can be seen from the table, the robustness of this method has been greatly improved compared with other algorithms, at 0.183. This is 23% higher than the basic algorithm DaSiamRPN and 8.9% higher than the second-ranked UpdateNet algorithm. Its EAO score, which reflects the comprehensive performance of the object tracking algorithm, is also the highest, with a score of 0.449. It is 9.0% higher than the DaSiamRPN algorithm and 0.2% higher than the second-ranked UpdateNet algorithm. The accuracy rate ranks third among all algorithms, which is 1.8% lower than the DaSiamRPN algorithm and 3.4% lower than the SiamMask algorithm with the highest accuracy. Above all, the method in this paper has the best comprehensive performance among all the compared algorithms, especially in terms of robustness in complex environments. Figure 4 shows the visual sorting plot of accuracy and robustness, and Figure 5 shows the score sorting diagram of average overlap expectations.

3.2.2. OTB100 benchmark experiment

The OTB100 dataset contains 100 video sequences, and the tracking scenes involved in these video sequences can be divided into 11 labeled properties. The two indicators, precision and success of tracking, are used by the dataset to evaluate the performance of the algorithm. Tracking precision refers to the ratio of the estimated number of frames with center position error to the total number of frames within 20 pixels, and the success refers to the percentage of the number of frames where the intersection over union of the target prediction box and the real bounding box is >0.5 to the total number of frames. The proposed method was compared with the current

TABLE 1 Performance comparison on the VOT2016 benchmark.

Algorithms	EAO↑	A↑	R↓
DaSiamRPN	0.412	0.620	0.238
UpdateNet	0.448	0.609	0.201
SiamFC	0.238	0.539	0.522
SiamRPN	0.303	0.581	0.387
SiamDW-RPN	0.376	0.565	0.281
SiamMask	0.430	0.630	0.206
C-COT	0.329	0.529	0.276
Ours	0.449	0.609	0.183

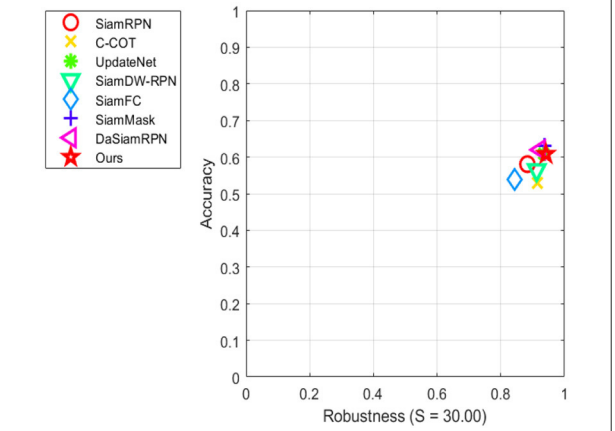

FIGURE 4
Accuracy-robustness ranking plots on the VOT2016 benchmark.

mainstream object tracking algorithms [SiamFC, SiamRPN, Staple (Bertinetto et al., 2016a), SRDCF (Danelljan et al., 2016b), CFNet (Valmadre et al., 2017), fDSST (Danelljan et al., 2017b), DaSiamRPN, and UpdateNet] on the OTB100 dataset, and Figure 6 shows the precision and success of each tracking algorithm. Among all the compared tracking algorithms, the proposed method ranks first, with a precision score of 86.4% and a success score of 64.7%. Compared with the basic algorithm DaSiamRPN, the precision and success of the proposed method increased by 0.8 and 0.4%, respectively. Compared with the UpdateNet algorithm, the precision and success of the proposed method increased by 0.6 and 0.9%, respectively. Overall, the performance of the proposed method was better than that of the current mainstream object tracking algorithms.

In particular, we present the comparison curves for deformation, occlusion, and out of plane rotation attributes on the OTB100 dataset in Figure 7. Obviously, in the precision plots, our tracking method outperformed all compared competitors on the deformation and out of plane rotation attributes. Compared with the UpdateNet algorithm, the

precision increased by 0.8 and 1.4% and compared with the DaSiamRPN algorithm, the precision increased by 2.7 and 1.3%, respectively. However, in the occlusion attribute of the precision plots, our tracking method was 0.4% lower than the UpdateNet algorithm, but 2.6% higher than the DaSiamRPN algorithm. In the success plots, our tracking method obtained the highest success score on the deformation, occlusion, and out of plane rotation attributes. Compared with the UpdateNet algorithm, the precision increased by 2.1, 0.5, and 2.1%, and compared with the DaSiamRPN algorithm, the precision increased by 3.0 1.9, and 1.1%, respectively. It can be seen that our method can overcome challenges in object tracking.

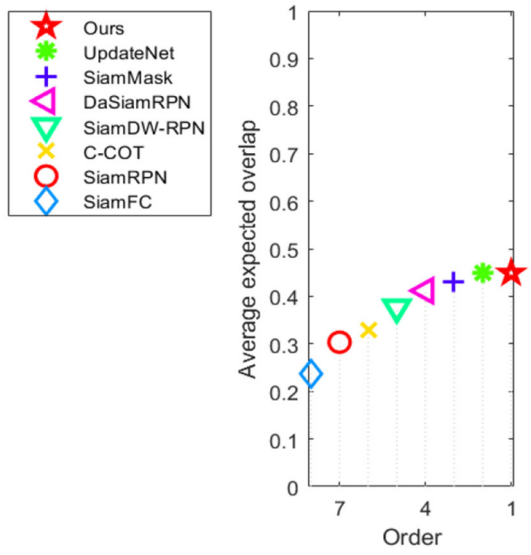

FIGURE 5
Expected overlap curves on the VOT2016 benchmark.

3.3. Qualitative evaluation

In order to visually display the tracking effect, several video sequences with various tracking difficulties were selected from the VOT2016 dataset and compared with the DaSiamRPN and UpdateNet algorithms to verify the experiment, as shown in Figure 8.

The main difficulty of tracking fish2 video sequences lies in the similar object interference. In the 92nd frame, the rapid movement of the object caused blurring, so the DaSiamRPN algorithm failed to track, while the method in this paper and the UpdateNet algorithm could track the object more accurately. In the 275th frame, there is an interfering object which is similar to the target, and the DaSiamRPN algorithm misjudged the analog as the tracking target. However, under the same conditions, the algorithm in this paper overcame the interference and achieved robust tracking. In the 309th frame, an interfering object which is similar to the target appears again, and the proposed method could track the target more accurately than the DaSiamRPN and UpdateNet algorithms.

The main difficulties of tracking handball2 video sequences are target occlusion, similar object interference, and posture changes. In the 279th frame, the tracking object is blocked by another moving object and due to the blurring of the target caused by its rapid movement, the DaSiamRPN algorithm tracked the target mistakenly. However, the proposed method

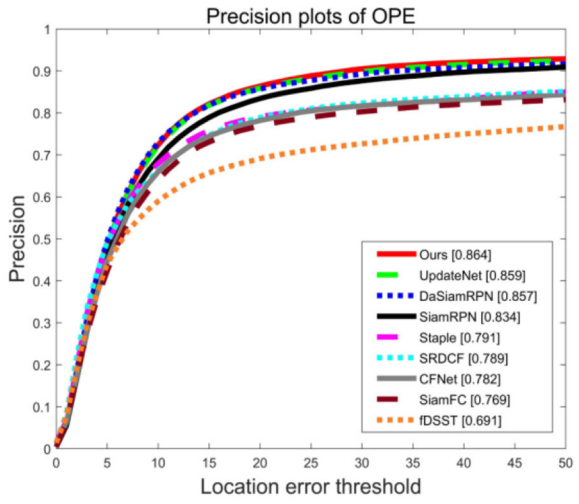

FIGURE 6
Precision and success plots of all compared trackers on OTB100.

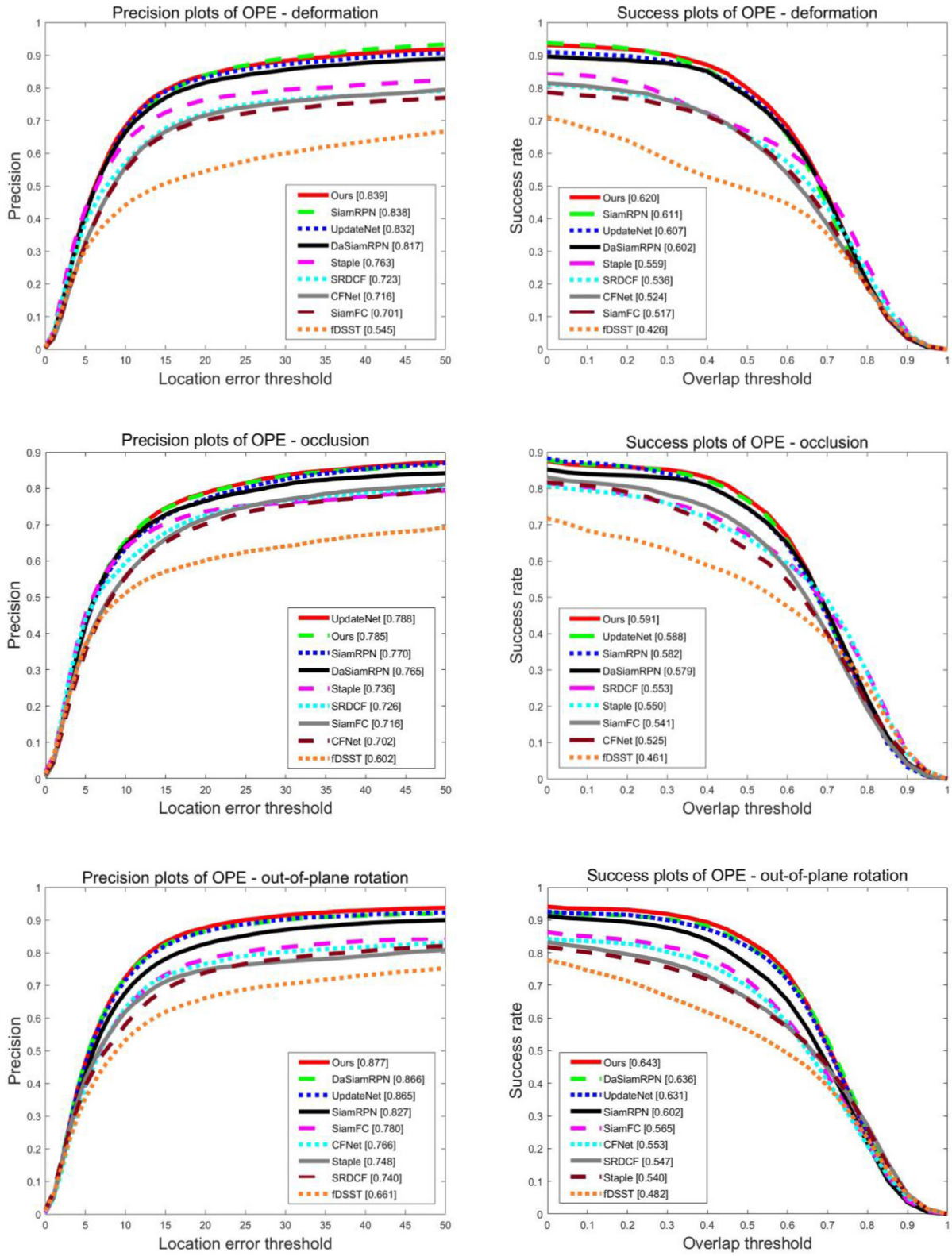

FIGURE 7
Precision and success plots of all compared trackers on deformation, occlusion, and out of plane rotation attributes on OTB100.

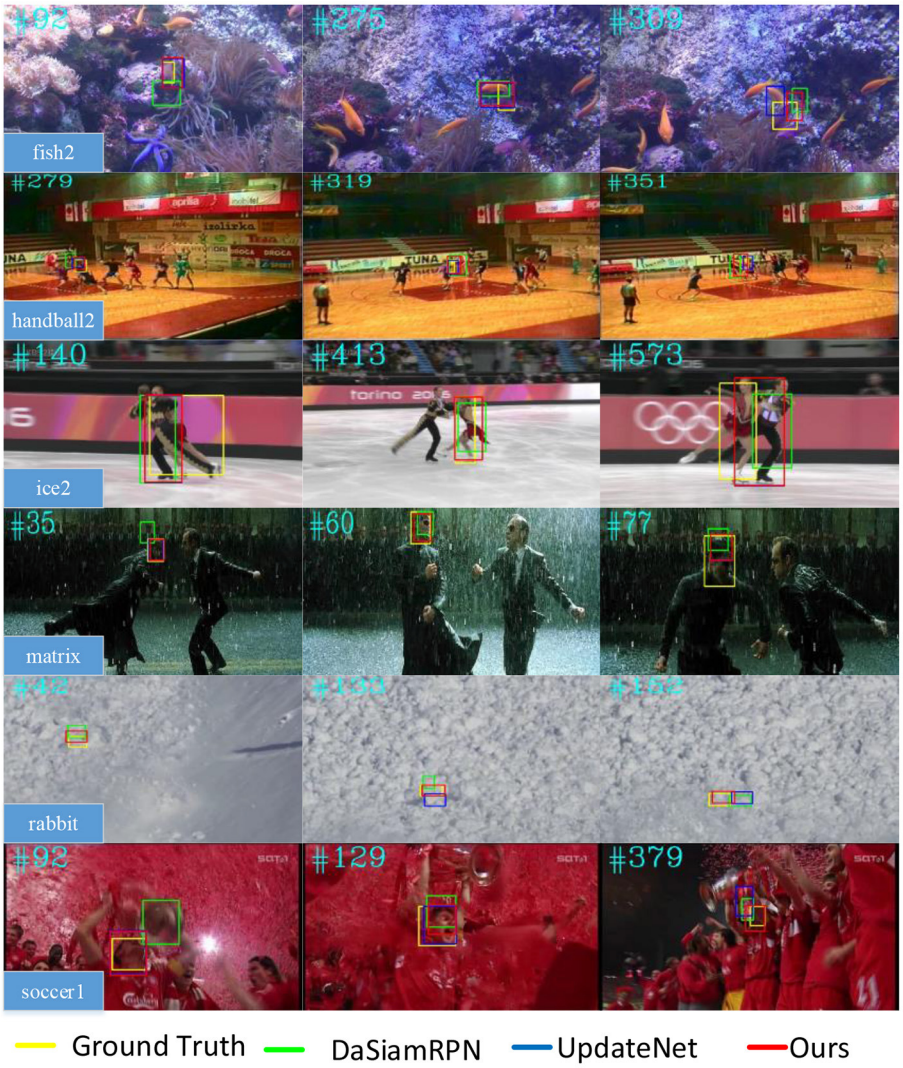

FIGURE 8
Qualitative evaluation on six challenging video sequences.

could still track the object despite the occlusion caused by another moving object.

The main difficulties of tracking matrix video sequences are similar object interference and changes in light conditions. In the 35th frame, similar object interference appears, so the DaSiamRPN algorithm tracked the target mistakenly. In the 77th frame, the DaSiamRPN algorithm had a tracking drift due to the complex background and the significant changes in light conditions. However, the proposed algorithm could track the object more steadily in spite of the above difficulties.

The main difficulties of tracking the rabbit video sequences are the small size of the tracking object and similar background interference. In the 133rd and 152nd frames, both the DaSiamRPN algorithm and the UpdateNet algorithm failed to track the object, while the proposed method in this paper still located the object stably.

Under tracking object occlusion, similar target interference, and a complicated environment, the DaSiamRPN algorithm always uses the predefined target in the first frame as the template of the tracking in the subsequent frames. The feature information easily gets lost because of the changes in the object appearance, resulting in the inability to accurately obtain the position of the object. The UpdateNet algorithm uses the template update mechanism to update the templates of each frame, which can introduce noise polluting templates that then results in tracking failure. On the other hand, the "dynamic–static" dual-template fusion and dynamic template adaptive update method proposed in this paper has the advantages of

TABLE 2 FPS comparison on the VOT2016 and OTB100 benchmark.

Algorithms datesets	DaSiamRPN	UpdateNet	Ours
VOT2016	110	76	87
OTB100	176	79	81

using static and dynamic templates, and it can still present a good tracking performance despite target occlusion, similar object interference, image blurring, and changes in light conditions, etc.

3.4. Real-timeliness of tracking

The tracking speed of the proposed method and the DaSiamRPN and UpdateNet algorithms on the VOT2016 and OTB100 datasets are shown in Table 2. The DaSiamRPN algorithm can run at 110 frames per second (FPS) and 176 FPS on the VOT2016 and OTB100 datasets, respectively. The UpdateNet algorithm can only run at 76 and 79 FPS. Because the UpdateNet algorithm introduces the template update mechanism to improve performance, the tracking speed is significantly lower than DaSiamRPN algorithms, but it can still meet the requirements of real-time tracking. Our method achieved a running speed of 87 and 81 FPS on VOT2016 and OTB100 datasets, respectively, and the tracking speed was slightly higher than that of the UpdateNet algorithm. Unlike the UpdateNet algorithm, the adaptive update strategy proposed in this paper does not update the template dynamically in every frame, so it has better real-time performance.

4. Conclusion

Deep learning object tracking algorithms based on Siamese networks have received extensive attention for their good performance in the testing of various benchmark tracking datasets. However, the earliest Siamese network object tracking algorithm uses the object position in the first frame image as a fixed template throughout the tracking, which cannot cope with the adverse effects of a change in the object appearance. The introduction of the template update mechanism makes the subsequent Siamese network better adapted to any change in the object appearance, though it also brings new problems, namely, causing the tracking template to be contaminated by external noise interference.

In order to further improve the performance of the target tracking algorithm, a Siamese tracker with "dynamic–static" dual-template fusion and dynamic template adaptive update is proposed in this paper, based on the DaSiamRPN Siamese network's object tracking and the UpdateNet template update network. The new method combines a static template and a dynamic one that is updated in real time for target tracking. An adaptive update strategy is applied when updating the dynamic template, which helps determine whether to update dynamic templates by judging the necessity of updating the template and the possibility of template pollution if the template is updated according to two quantitative indicators: one is the similarity between the tracking result of the current frame and the dynamic template; the other is the no reference evaluation result of the current frame image. In this way, it can suppress the adverse effects of noise interference and pollution of the template while adapting to the change in the target appearance. The experimental results showed that the robustness and EAO of the proposed method were 23 and 9.0% higher than those of the basic algorithm on the VOT2016 dataset, respectively, and that the precision and success were increased by 0.8 and 0.4% on the OTB100 dataset, respectively. The best comprehensive and real-time tracking performance was obtained on the above two large public datasets.

There are several important thresholds in the adaptive template update module of this method that make it necessary to collect samples in its application to statistical determination, which increases the difficulty of deploying the new method in the actual system. Therefore, the next step is to introduce an unsupervised machine learning algorithm so that self-adaption can be achieved.

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 author.

Author contributions

DS: writing—original draft, data curation, and visualization. XW: supervision, conceptualization, and writing—review and editing. YM: methodology and software. ND and ZP: investigation and validation. All authors contributed to the article and approved the submitted version.

Funding

This work was supported by State Key R&D Program (2018YFB1308200), the National Nature Science Foundation of China (Number 51405154), the Natural Science Foundation of Hunan Provincial (Numbers 2021JJ30251; 2018JJ3167), Excellent Youth Project of Hunan Education Department (Number 18B217), and the Visiting Scholar Program of China Scholarship Council (202008430103).

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

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnbot.2022.1094892/full#supplementary-material

References

Bertinetto, L., Valmadre, J., Golodetz, S., Miksik, O., and Torr, P. H. S. (2016a). "Staple: Complementary learners for real-time tracking," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, (Washington, DC: IEEE Computer Society), 1401–1409. doi: 10.1109/CVPR.2016.156

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H. S. (2016b). *Fully-Convolutional Siamese Networks for Object Tracking*. Cham: Springer. doi: 10.1007/978-3-319-48881-3_56

Bo, L., Yan, J., Wu, W., Zhu, Z., and Hu, X. (2018). "High performance visual tracking with siamese region proposal network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT: IEEE.

Bolme, D. S., Beveridge, J. R., Draper, B. A., and Lui, Y. M. (2010). "Visual object tracking using adaptive correlation filters," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), June 13-18, 2010*. (San Francisco, CA; New York, NY: IEEE), 2544–2550. doi: 10.1109/CVPR.2010.5539960

Danelljan, M., Bhat, G., Khan, F. S., and Felsberg, M. (2017a). "ECO: Efficient convolution operators for tracking," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017*. (Honolulu, HI; New York: IEEE), 6638–6646. doi: 10.1109/CVPR.2017.733

Danelljan, M., Hager, G., Khan, F. S., and Felsberg, M. (2015a). "Convolutional features for correlation filter based visual tracking," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*. (IEEE: Santiago), 58–66. doi: 10.1109/ICCVW.2015.84

Danelljan, M., Häger, G., Khan, F. S., and Felsberg, M. (2015b). "Learning spatially regularized correlation filters for visual tracking," in *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago: IEEE. doi: 10.1109/ICCV.2015.490

Danelljan, M., Hager, G., Khan, F. S., and Felsberg, M. (2016b). Adaptive decontamination of the training set: A unified formulation for discriminative visual tracking. *IEEE*, 1430–1438. doi: 10.1109/CVPR.2016.159

Danelljan, M., Hager, G., Khan, F. S., and Felsberg, M. (2017b). Discriminative scale space tracking. *IEEE Trans. Patt. Anal. Machine Intell.* 39, 1561–1575. doi: 10.1109/TPAMI.2016.2609928

Danelljan, M., Robinson, A., Khan, F. S., and Felsberg, M. (2016a). "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision, ECCV*, 472–488. doi: 10.1007/978-3-319-46454-1_29

Dunnhofer, M., Simonato, K., and Micheloni, C. (2022). Combining complementary trackers for enhanced long-term visual object tracking - ScienceDirect. *Image Vis. Comput.* 122, 104448. doi: 10.1016/j.imavis.2022.104448

Fan, H., Ling, L., Yang, F., Chu, P., Deng, G., Yu, S., et al. (2019). "LaSOT: A high-quality benchmark for large-scale single object tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (Long Beach, CA), 5369–5378. doi: 10.1109/CVPR.2019.00552

Galoogahi, H. K., Fagg, A., and Lucey, S. (2017). Learning background-aware correlation filters for visual tracking. *IEEE Comput. Soc.* 2017, 129. doi: 10.1109/ICCV.2017.129

Haisheng, L., Zheng, Y., Cao, J., and Cai, Q. (2019). Multi-view-based siamese convolutional neural network for 3D object retrieval. *Comput. Electr. Eng.* 78, 22. doi: 10.1016/j.compeleceng.2019.06.022

Han, R., Feng, W., and Wang, S. (2020). Fast learning of spatially regularized and content aware correlation filter for visual tracking. *IEEE Trans. Image Process* 29, 7128–7140. doi: 10.1109/TIP.2020.2998978

Henriques, J. F., Caseiro, R., Martins, P., and Batista, J. (2012). "Exploiting the circulant structure of tracking-by-detection with kernels," in *Proceedings of the European Conference on Computer Vision (ECCV), October 7-13, 2012, Florence, Italy* (Cham: Springer), 702–715. doi: 10.1007/978-3-642-33765-9_50

Henriques, J. F., Caseiro, R., Martins, P., and Batista, J. (2015). Highspeed tracking with kernelized correlation filters. *TPAMI* 37, 583–596. doi: 10.1109/TPAMI.2014.2345390

Huang, B., Xu, T., Li, J., Shen, Z., and Chen, Y. (2019). Transfer learning-based discriminative correlation filter for visual tracking. *Pattern Recognit.* 100, 107157. doi: 10.1016/j.patcog.2019.107157

Karakostas, I., Mygdalis, V., Tefas, A., and Pitas, I. (2021). Occlusion detection and drift-avoidance framework for 2D visual object tracking. *Sign. Process.* 90, 116011. doi: 10.1016/j.image.2020.116011

Li, Y., and Zhu, J. (2014). "A scale adaptive kernel correlation filter tracker with feature integration," in *Proceedings of European Conference on Computer Vision Workshops - Zurich, Switzerland, September 6-7 and 12 volume 8926*. (Zurich), 254–265. doi: 10.1007/978-3-319-16181-5_18

Lin, T. Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., et al. (2014). "Microsoft COCO: Common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV)*. (Zurich), 740–755. doi: 10.1007/978-3-319-10602-1_48

Liu, Q., Chen, D., Chu, Q., Yuan, L., Liu, B., Zhang, L., et al. (2022). Online multi-object tracking with unsupervised re-identification learning and occlusion estimation. *arXiv*. 1, 8. doi: 10.1016/j.neucom.2022.01.008

Ly, A., Yang, L., Kong, C., Chang, X., Zhao, S., Cao, Y., et al. (2021). Correlation filters with adaptive convolution response fusion for object tracking. *Knowl. Bas. Syst.* 228, 107314. doi: 10.1016/j.knosys.2021.107314

Mittal, A, Saad, M. A, and Bovik, A. C. (2013). Making a "completely blind" image quality analyzer. *IEEE Sign. Process. Lett.* 20, 209–212. doi: 10.1109/LSP.2012.2227726

Nousi, P., Triantafyllidou, D., Tefas, A., and Pitas, I. (2020). Re-identification framework for long term visual object tracking based on object detection and classification. *Sign. Process.* 88, 115969. doi: 10.1016/j.image.2020.115969

Real, E., Shlens, J., Mazzocchi, S., Pan, X., and Vanhoucke, V. (2017). "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*. (Hawaii), 7464–7473. doi: 10.1109/CVPR.2017.789

Russakovsky, O., Deng, J., Su, H., Karuse, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.* 115, 211–252. doi: 10.1007/s11263-015-0816-y

Su, H., Qi, W., Schmirander, Y., Ovur, S. E., Cai, S., and Xiong, X. (2022). A human activity-aware shared control solution for medical human-robot interaction. *Assembly Automation*. 42, 388–394. doi: 10.1108/AA-12-2021-0174

Sun, D., Wang, X., Lin, Y., Yang, T., and Wu, S. (2021). Introducing depth information into generative target tracking. *Front. Neurorobot.* 15, 718681. doi: 10.3389/fnbot.2021.718681

Tan, K., Xu, T. B., and Wei, Z. (2021). Learning complementary Siamese networks for real-time high-performance visual tracking. *J. Vis. Commun. Image Represent.* 80, 103299. doi: 10.1016/j.jvcir.2021.103299

Valmadre, J., Bertinetto, L., Henriques, J. F., Vedaldi, A., and Torr, P. H. S. (2017). "End-to-end representation learning for correlation filter based tracking," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR* (IEEE), 5000–5008. doi: 10.1109/CVPR.2017.531

Voigtlaender, P., Luiten, J., Torr, P. H., and Leibe, B. (2020). "Siam R-CNN: Visual tracking by re-detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE). doi: 10.1109/CVPR42600.2020.00661

Wang, J., Liu, W., Xing, W., Wang, L., and Zhang, S. (2020). Attention shake siamese network with auxiliary relocation branch for visual object tracking. *Neurocomputing* 400, 120. doi: 10.1016/j.neucom.2020.02.120

Wang, Q., Zhang, L., Bertinetto, L., Hu, W., and Torr, P. H. S. (2019). "Fast online object trackingand segmentation: A unifying approach," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR* (IEEE), 1328–1338 doi: 10.1109/CVPR.2019.00142

Wang, W., Zhang, K., Lv, M., and Wang, J. (2020). Hierarchical spatiotemporal context-aware correlation filters for visual tracking. *IEEE Trans. Cybern.* 2020, 1–14. doi: 10.1109/TCYB.2020.2964757

Zhang, H., Chen, J., Nie, G., and Hu, S. (2020). Uncertain motion tracking based on convolutional net with semantics estimation and region proposals. *Patt. Recogn.* 102, 107232. doi: 10.1016/j.patcog.2020.107232

Zhang, H., Cheng, L., Zhang, J., Huang, W, Liu, X., and Yu, J. (2021). Structural pixel-wise target attention for robust object tracking. *Digit. Sign. Process.* 1, 103139. doi: 10.1016/j.dsp.2021.103139

Zhang, J., Sun, J., Wang, J., Li, Z., and Chen, X. (2022). An object tracking framework with recapture based on correlation filters and Siamese networks. *Comput. Electr. Eng.* 98, 107730. doi: 10.1016/j.compeleceng.2022.107730

Zhang, L., Gonzalez-Garcia, A., van de Weijer, J., Danelljan, M., and Khan, F. S. (2019). "Learning the model update for siamese trackers," in *IEEE International Conference on Computer Vision (ICCV)*, 4009–4018. doi: 10.1109/ICCV.2019.00411

Zhang, X., Ma, J., Liu, H., Hu, H. M., and Yang, P. (2022). Dual attentional Siamese network for visual tracking. *Displays* 74, 102205. doi: 10.1016/j.displa.2022.102205

Zhang, Y., Liu G., Haung, H., Xiong, R., and Zhang, H. (2022). Dual-stream collaborative tracking algorithm combined with reliable memory based update. *Neurocomputing* 480, 39–60. doi: 10.1016/j.neucom.2022.01.046

Zhang, Y., Wang, T., Liu, K., Zhang, B., and Chen, L. (2021). Recent advances of single-object tracking methods: A brief survey. *Neurocomputing* 455, 1–11. doi: 10.1016/j.neucom.2021.05.011

Zhang, Z., and Peng, H. (2019). "Deeper and wider siamese networks for real-time visual tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (Long Beach, CA), 4591–4600. doi: 10.1109/CVPR.2019.00472

Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., and Hu, W. (2018). *Distractor-Aware Siamese Networks for Visual Object Tracking*. Cham: Springer. doi: 10.1007/978-3-030-01240-3_7