Yang *et al.*
*EURASIP Journal on Advances in Signal Processing* (2022) 2022:28
https://doi.org/10.1186/s13634-022-00860-9

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

# Research on deep correlation filter tracking based on channel importance

Guosheng Yang, Chunting Li* and Honglin Chen

*Correspondence: 19301539@muc.edu.cn
School of Information Engineering, Minzu University of China, Beijing 100081, China

**Abstract**

Correlation filter tracking requires little prior knowledge of the tracking target (e.g., the shape, and the posture) but has a fast-tracking speed. The deep features extracted by the deep convolutional neural network have strong representation ability, so the tracking method based on the combination of correlation filter and deep convolutional neural network, named as deep correlation filter tracking, is a hot issue in the field of target tracking at present. However, the deep convolutional neural network largely restricts the real-time performance of the deep correlation filter tracking because of its complex network structure and heavy computation burden. To balance the contradiction between tracking speed and tracking accuracy, a new channel importance is defined and the channel importance based method of how to select the important channels is given in this paper. And then, a deep correlation filter tracking method based on channel importance is proposed to lighten the feature network, reduce the computation load and improve the tracking speed under the premise of ensuring the tracking accuracy. In the process of tracking, the structural similarity index measurement (SSIM) of the predicted tracking target in two consecutive frames is calculated in real-time. Based on the SSIM, determine whether the feature network needs to be updated, and decide whether the tracking fails. If the feature network needs to be updated, the feature network will be updated online while the tracking is on. If the tracking fails, the target will be searched again, and the tracking is recovered from the failure. The tracking algorithm proposed in this paper is tested on the OTB2013 data set, and the experiment shows that the tracking algorithm designed in this paper can improve the real-time performance while meeting the requirement of tracking accuracy. The online update of the feature network can make the network adapt to the complex background and target changes to improve tracking accuracy; In the case of tracking failure, the re-tracking module can search for the target again and resume tracking given that the target is always present.

**Keywords:** Object tracking, Deep correlation filter, Channel importance, SSIM

## 1 Introduction

Visual object tracking (VOT) can be described as the process of confirming the target defined in the first frame and estimating its motion trajectory in the subsequent frames, under the condition that the state information (position, size, etc.) of the target is given in the first frame of tracking sequence [1–6]. VOT is the product

of multi-domain cross fusion, mainly involving image processing, machine learning, optimization, and other fields. Furthermore, VOT is the premise and basis for completing higher-level image understanding tasks [1, 2, 7, 8]. At present, it has been successfully applied in many fields, such as intelligent video surveillance [9, 10], unmanned driving [11, 12], human–computer interaction [13], intelligent robot [14], unmanned aerial vehicle [15], ocean detection [16], and etc. It is no doubt that VOT is an interdisciplinary, widely used, open, and attractive research field of computer vision.

The early work was mainly focused on the traditional visual object tracking algorithms, such as Mean Shift [17], Lucas Kanade [18], Particle filter [19], and so on. In 2010, Minimum Output Sum of Squared Error filter (MOSSE) [20] is introduced to the visual tracking for the first time. Since then, a large number of improved studies from different perspectives have been performed on the correlation filter based visual tracking algorithms to generate many kinds of variant algorithms, such as Circulant Structure of Tracking-by-Detection with Kernels (CSK) [21], Kernelized Correlation Filters (KCF) [22], Scale Adaptive Multiple Feature (SAMF) [23], Discriminative Scale Space Tracker (DSST) [24], etc. Compared with the traditional tracking algorithm, the tracking accuracy and speed of these variant algorithms have been significantly improved.

With the successful application of deep learning (especially deep convolutional neural network) in image classification and object detection, deep learning has also begun to be widely used in target tracking algorithms, and various target tracking algorithms based on deep convolutional neural networks have been developed, such as deep convolutional neural network + correlation filter (DCNN + CF), deep convolutional neural network + particle filter, Siamese neural network-based methods, and etc. Especially, DCNN + CF is the most typical one among them. On the one hand, DCNN + CF takes the deep convolutional neural network as the feature extractor, which plays its powerful feature representation ability to extract the deep features that can fully represent the target from the sequential images to ensure the tracking accuracy. On the other hand, it makes full use of the speed advantage of correlation filter to carry out feature association, and predicts the target position from the response of correlation filter to ensure real-time tracking. Because of these two remarkable characteristics (powerful feature representation ability and real time ability), DCNN + CF has become one of the hot research issues of visual tracking, meanwhile many successful applications have been achieved [25].

Although DCNN + CF visual tracking theory has made a lot of progress and has been successfully applied in related fields, there are still many challenges to be solved, which are manifested in:

i) Deep convolutional neural network gives full play to its powerful feature representation ability and extracts deep features that can fully represent the target from sequential images. Although it improves the tracking accuracy, it also increases the complexity of the algorithm and the computational burden at the same time, which will inevitably aggravate the real-time performance of the tracking algorithm. Therefore, how to balance tracking accuracy with real-time tracking is one of the challenges to implement DCNN + CF algorithm.

ii) Generally, the deep convolutional neural network of DCNN + CF is pre-trained offline, the structure and parameters of the deep convolutional neural network are kept unchanged, and the correlation filter is updated online during the real tracking process. In practice, however, video frames that do not appear in the training sample set are often encountered, such as illumination variation, scale variation, occlusion, deformation, motion blur, fast motion, in-plane rotation, out-of-plane rotation, out-of-view, background clutters, low resolution [5], and etc. These factors will greatly decrease the tracking accuracy and affect the tracking performance. One of the effective methods to solve this problem is to update the deep convolutional neural network online. Although some researchers suggested solutions using dynamic networks [26, 27], most of them only focus on specific interference factors. Therefore, completely considering these interference factors and updating the kernels of the deep convolution neural network in real time to improve the tracking performance is also one of the challenges to realize the DCNN + CF algorithm.

iii) The interference factors mentioned above will greatly decrease the tracking accuracy and affect the tracking performance and the most extreme case is tracking failure. How to detect the tracking failure and how to restore normal tracking from tracking failure are other challenge to implement the DCNN + CF algorithm.

In this paper, we propose a deep correlation filter tracking algorithm based on channel importance to solve problems mentioned above, which mainly includes the following three parts:

- According to the Gaussian output response of the visual tracking system, the importance score of each channel of the last layer of the feature network is defined (in the Sect. 3.2.1), and the importance score of each channel is calculated. Under the condition of maintaining tracking accuracy, we only reserve the channels with a larger importance score, and clip off the other channels with a smaller importance score in the last layer of the feature network during the tracking progress, which can reduce the network complexity, alleviate the computational burden, and improve the real-time performance of the tracking.
- Structural similarity index measurement (SSIM), a measure of the similarity of two images [55], is selected to represent the similarity of targets in two consecutive frames of the video sequence, because it has global statistical distribution characteristics, which can work well for a variety of interference factors and conforms to people's visual habits. It is indicated normal tracking when SSIM is higher than the pre-set threshold range. No matter what the reason is, if the similarity SSIM of the target in successive two frames falls within the pre-set threshold range, it indicates that the adaptability of the feature network decreases and the feature network can still work normally, but training is needed to improve the adaptability of the network. At this point, the feature network online update module is activated to train the feature network online with the current frame data. Meanwhile, real time tracking is continued. The feature network is updated in time to adapt to the changes of the target and environment, which enable the tracking system to improve the tracking accuracy under the constraint of ensuring real-time performance.

- No matter what the reason is, if the SSIM of the target in two consecutive frames is lower than the pre-set threshold range, it indicates that the tracking has failed. At this point, the re-tracking module is activated to re-search for the target in the given area of the current frame until the search in the given area is completed. If the target is found, then the tracking is recovered, and the next frame tracking is continued after the recovery. Otherwise, the next frame tracking is continued directly.

## 2 Related work

Object tracking has always been a hot research topic in the field of computer vision. The comprehensive reviews can be found in [3, 4]. In this section, some works highly associate with DCNN + CF is presented to highlight our motivation.

### 2.1 Correlation filter

In the field of signal processing, correlation filter is used to measure the degree of similarity between signals. The higher the signal similarity, the higher the correlation value of the filter output. The application of this method to tracking is to find the maximum value correlated with the tracking target in the output response of the filter. Combining correlation filter in signal processing with object tracking, Bolme et al. [20] proposed MOSSE algorithm to convert time domain calculations to frequency domain calculations, which greatly reduces computational complexity and enables tracking speeds to exceed 600 fps. Henriques et al. [21] introduced circular matrix and kernel function estimation into MOSSE algorithm, and proposed CSK algorithm by employing gray-scale features as used in the original MOSSE; Henriques et al. [22] further improved the CSK tracking algorithm by using the Histogram of Oriented features, and proposed the kernel correlation filter (KCF); The cyclic matrix is introduced to avoid matrix inversion, which reduces the computational complexity and improves the tracking speed; In addition, the tracking is taken as classification. By introducing kernel function, the non-linear classification problem is transformed into linear classification problem [8], which makes the algorithm more adaptable and the performance is further improved. Besides of the research works mentioned above, some researchers try to improve the correlation filter based tracking algorithms from different aspects. For example, SAMF [23], DSST [24] and ASRCF [28] (adaptive spatially regulated correlation filters) improved the correlation filter based tracking algorithm from the aspect of scale adaptation by setting the scale factor, the scale filter, and introducing the adaptive space regularization into objective function respectively. SRDCF [29] (spatially regulating discriminative correlation filters) effectively solved the boundary effect problem of correlation filter algorithm. MTCF (visual object multimodality tracker based on correlation filters) proposed by Yang et al. [30] can adapt to the changes of target translation, scale, and rotation at the same time.

Correlation filter has obvious advantages in tracking speed. However, due to the limitation of feature representation, the tracking accuracy is not satisfactory. Therefore, for correlation filter, improving tracking accuracy has become one of the important issues of visual tracking based on correlation filter.

### 2.2 Correlation filter tracking based on deep learning

Deep learning has strong feature representation ability, while the correlation filter algorithm has significant advantages in tracking speed. Therefore, the combination of both methods has become a potential research direction in the field of visual tracking.

#### *2.2.1 Off-line training of feature network*

Danelljan et al. [31] replaced the HOG (histogram of oriented gradient) feature in SRDCF with a single-layer convolution feature. Although the accuracy of the tracking is improved, the computational complexity is also increased, and the tracking speed is reduced as a result.

Ma et al. [32] proposed hierarchical convolutional features for visual tracking (HCFT). Considering the characteristics of different layer features, three-layer networks were used to train correlation filters respectively. The weighted sum of the outputs of the three associated filters is taken as the final output of the tracking system, corresponding to position of the target. Although the tracking accuracy is improved, the tracker is not effective in dealing with occlusion problems, and the robustness of long-term tracking is not high.

C-COT [33] (continuous convolution operator tracker) proposed a method of continuous convolution filter. The feature maps with different resolutions are interpolated into the continuous space domain through cubic interpolation, and then the Hessian matrix is applied to obtain the target position. This algorithm significantly improves the tracking accuracy and robustness, but it is difficult to ensure real-time performance due to the complexity of the calculation.

In addition to using the deep network as feature extraction, some researchers have tried to use a neural network to simulate the whole process of correlation filter based tracking. Bertinetto et al. [34] proposed SiamFC (fully continuous siamese networks) tracking algorithm based on the Siamese network structure. SiamFC has two branches. One branch is for extracting the target feature, while the other one is used to extract the detection window feature. The two kinds of features are correlated to obtain the corresponding predicted target position. A large number of subsequent Siamese network models are improved based on SiamFC. Siammask [35] (siamese network with mask) induced a Mask branch on SiamFC for direct target segmentation. A more accurate target box can be obtained through combining the tracking task with the semi-supervised segmentation task. Consequently, the accuracy of the tracker is largely improved. SiamDW [36] (deep and wider siamese networks) improved the tracking accuracy by designing a residual module to eliminate the negative impact of pooling layer the deep network.

Although the above methods use the deep features to more completely represent the target, most of the models are based on off-line training, which is difficult to adapt to the target changes (for example, appearance, scale, rotation, and etc.) and complex environmental changes.

#### 2.2.2 Feature network slimming

The powerful feature representation capabilities of deep learning methods can greatly improve tracking accuracy, but the complexity of deep neural networks also increases the computational burden and restricts real-time tracking. Because the features of different layers of the convolutional neural network have different characteristics, and their effects on tracking performance are also different. Some researchers began to pay attention to the feature network module and improved the real-time performance of the tracking by only selecting the features with more contributions to the tracking in order to simplify the network structure.

Li et al. [37] proposed a tracking model based on target-aware which means that the target with significant appearance variations is perceived. It defined the importance of filters as the GAP (global average pooling) of the gradient of the regression loss function. According to the output results of GAP, the most effective filter is selected to generate target perception features, which reduces the amount of calculation and improves the accuracy of tracking. Che et al. [38] proposed a response evaluation mechanism, AFER (average feature energy ratio), which calculated the ratio between the target area of each channel and the average feature value of the entire search area. The channels with AFER greater than the threshold were reserved, which enhanced the tracking speed. He et al. [39] proposed a two-step algorithm. The first step is to find representative channels based on LASSO (least absolute shrinkage and selection operator) regression to minimize the reconstruction error of the output feature map. Another step is to reconstruct the output of the remained channels using linear least squares. This method reduces the computational burden and ensures that the output difference between before and after channel change is minimal.

Liu et al. [40] proposed a network slimming method. It introduced a scaling factor $\gamma$ for each channel. The scaling factor was trained along with the network weights. The channels whose scaling factors were close to 0 were clipped out directly. This method simplifies the model and reduces the calculation cost without an obvious loss of precision. Ye et al. [41] proposed an OT (optimal thresholding) method that calculated the global optimal threshold by calculating the sum of the cumulative squares of the scaling factors in the BN layer. The training time of the model is shortened while the accuracy is guaranteed.

These methods simplify the network structure and reduce the computational burden while ensuring accuracy. But most of the above methods are to establish an abstract optimization function from a mathematical point of view (such as $L_1$ norm) for optimization which needed a lot of adjustment parameters. Adjustment and optimization are more complex and not very practical. At the same time, the physical concepts are not easy to be understand. Moreover, retraining is generally required after being lightened, which adds additional computational cost. From the perspective of the nature of correlation filter (Gaussian response), this paper defines the importance of channels based on the proportional relationship between the total Gaussian distribution of the target response and the target response Gaussian distribution of the single-channel. According to the importance of channels, the important channels were selected to lighten the feature network, and enhance the tracking speed while the tracking accuracy is guaranteed. The method is simple, consumes less resources,

requires less computation, is easy to understand the physical concept, and does not require retraining after being lightened.

#### 2.2.3 On-line training feature network

Once the pre-training of the feature network is completed, it generally does not change in the actual tracking. If there are instances that do not appear in the training set in the actual tracking application, the performance of the network will be degraded and the generalization is not strong. One of the effective ways to solve this problem is to optimize the parameters of the feature network by online adjustment.

Nam et al. [42] used historical samples at regular intervals to make long-term updates to the network and use recent samples to make short-term updates to the network when tracking failures are detected, which greatly improved the tracking accuracy. Song et al. [43] proposed the CREST (convolutional residual learning scheme for visual tracking) algorithm to generate training blocks based on the target location of each frame prediction. The training blocks are input into the network every 2 frames for an online update, which improves the robustness and tracking accuracy of the model. Qiu et al. [44] proposed a method to update the network in real time. In their method, the first frame was used to make the network initially learn the characteristics of the current target. During the tracking process, high score samples with several frames interval are used to update network parameters. When tracking fails, the high score sample is used for updating. By introducing online updates to the network, the network's discriminant power for specific targets is effectively strengthened and the tracking accuracy is enhanced. Liu et al. [45] recorded the frames with tracking scores greater than the threshold as successful frames. The few frames that were recently tracked successfully is used for adaptive short-term update, while more frames were used for the long-term update. Combining the two online update methods can help the network to better adapt to the changes of target and scene and improve the tracking accuracy.

The above methods can improve the robustness and tracking accuracy of the model through the online update of the network. But, in most methods, multi-frame samples at fixed time interval are used to update the feature network online. Thus such as methods have poor flexibility and also increases the computing burden of the network. Consequently, it is difficult to ensure the flexibility and the real-time tracking. In addition, when the feature network needed to be trained on line is not clear. In this paper, SSIM, being consistent with human visual perception and comprehensively considering the interference factors that cause the tracking accuracy to decline, is used to judge the similarity of the target box in two consecutive frames. When the similarity SSIM decreases to a certain range, it indicates that the current network cannot adapt to the change of the target. The current frame is used to update the feature network to ensure the robustness of the model.

#### 2.2.4 Tracking failure detection

Occlusion, distortion, motion blurring, rotation, and other interference factors restrict the tracking accuracy. In extreme cases, they may even cause tracking failure. Detection of tracking failures and how to quickly resume normal tracking are also one of the major issues to be solved in the field of visual object tracking.

Ma et al. [46] set a stable threshold to learn a long-term filter, and the maximum output response value of the long-term filter was used as the confidence. When the confidence is below the threshold, it indicates the tracking fails. The target is recovered from the tracking failure by the SVM (support vector machines) re-detection module. Walsh et al. [47] used the mean and entropy of the response to determine the tracking state of the current frame. When both of mean and entropy are lower than the pre-set threshold, it is determined that the tracking failed and the target search area needed to be updated in order to recover the tracking. The model improves the reliability of the tracker in terms of fuzzy motion and abrupt occlusion, but it is strongly influenced by the background and has low robustness. Wang et al. [48] defined the ratio of the difference between the maximum and minimum response values and the standard deviation as the response stability, which was used to judge the tracking status. The average value of the response stability of all historical frames is regarded as the threshold. If the response stability is lower than the threshold, the re-tracking module is started to resume tracking. This method solves the problem of tracking failure effectively and improves the robustness of the tracker. Shin et al. [49] determined whether a tracking failure has occurred by calculating the correlation between the maximum output response of the target predicted box and the one of the adjacent area of the predicted box in current frame. When tracking failure is detected, the target is re-searched in the adjacent area of the current frame to recover the tracking. This method makes up for the defects of the original model in occlusion and fuzzy motion. Li et al. [50] determined whether a tracking failure occurs by calculating the PSR (peak to sidelobe ratio) ratio of the current frame output response to the first frame output response. When the ratio is less than a pre-set threshold, the tracking is determined to fail, and the SVM detector searches for the target again to recover the tracking. This algorithm effectively improves robustness of the tracker for occlusion problems.

The above methods can solve the problem of tracking failure to a certain extent, but most of them judge the current tracking state from the perspective of an output response. The output corresponding is the result of the correlation between the target feature and the filter, which reflects the position information of the target. The nature of the tracking failure is that the real features of the target can not be extracted because of the various interference factors. Thus, one of the good ways to detecting tacking failure is proceeded from the target features. From the global statistical perspective of target features, SSIM is used to judge the similarity of the target box in two consecutive frames in this paper. When SSIM is less than a pre-set threshold, the tracking is determined to fail, and the recovery module is started to recover the tracking.

## 3 Methodology

DCNN + CF makes full use of the obvious advantages of correlation filter and deep neural network. On the one hand, it can give play to the speed advantage of correlation filter and ensure the real-time tracking; on the other hand, the deep convolutional neural network has powerful feature representation ability, which can better represent the features of the target and ensure the tracking accuracy. However, the introduction of deep learning into correlation filter inevitably increases the calculation burden, and it is difficult to balance the tracking accuracy and speed.

In addition, in the actual tracking process, due to the influence of environmental illumination, target scaling, target rotation and other interference factors, the target features extracted by the pre-offline training network are difficult to accurately describe the disturbed target. In other words, the feature network of pre-offline training is not robust enough. In such a case, it is difficult to accurately predict the position of the target, when the extracted features are associated with the subsequent tracking filter, and it is inevitable to result in target position drift in different degrees. As a result, the tracking accuracy is degraded in some extent. In the case of severe drift, the predicted position of the target is likely to move out of the tracking window, resulting in tracking failure. To address such issues, inspired by the documents[37, 51, 52], we think that the channel of the feature network with more contributions to the output response of the tracking system is taken as the important channel which plays a decisive role in the output response of the tracking system. All of the important channels are reserved while the other channels are pruned off to lighten the feature network and decrease the computational load and enhance the real time performance. To do so, channel importance is defined as decision criterion for selecting important channels. Based on such a motivation, firstly we define the importance score of a channel from the perspective of the nature of correlation filter (Gaussian output response), and then take the importance score of the channel as a decision criterion for selecting important channels to generate a lightened feature network, enhancing the tracking speed under the condition of meeting the tracking accuracy. Furthermore, because of the global statistical distribution characteristics, SSIM can work well for many kinds of interference factors. More importantly, SSIM conforms to people's visual habits. So SSIM is employed to calculate the similarity of

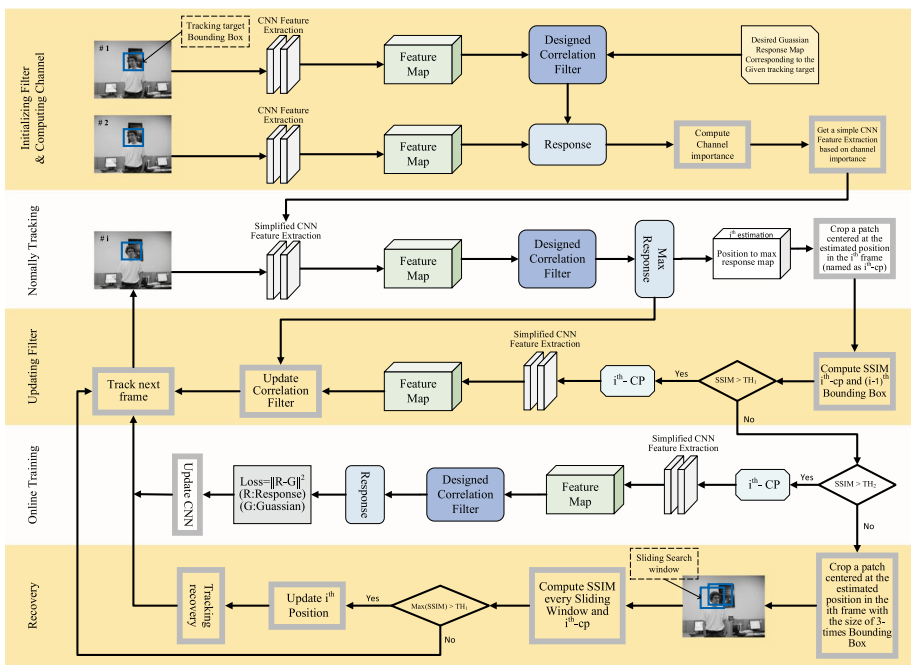


**Fig. 1** The overall framework of deep correlation filter tracking based on channel importance

two consecutive frames of the video sequence in this paper, and is taken as a decision criterion to make decisions on the update of feature network, the failure detection and recovery of the tracking network.

In summary, we propose a deep correlation filter tracking model based on the importance of the channel, and its overall structure is shown in Fig. 1.

The overall structure is divided into five functional modules. The first module named as initializing filter and computing channel has two functions. One is responsible for initializing the tracking filter by use of the first frame of video sequence; the other is for calculating the importance score of the channel by use of the output response of the filter correlated with the target feature maps of the second frame of sequence, and then selecting the important channels based on the importance score to generate the lightened feature network. The second module named as normally tracking is in charge of normal tracking. The third module named as updating filter is responsible for tracking filter updating online. The fourth module named as online training is responsible for training the feature network online. The fifth module named as recovery is in charge of tracking failure detection and tracking recovery.

To complete these functions, two thresholds $TH_1$ and $TH_2$ are set to define different tracking states. During the tracking process, the SSIM value of the target boxes of two consecutive frames is calculated. When SSIM is greater than the threshold $TH_1$, normal tracking is performed. When SSIM is within the range of $(TH_1, TH_2)$, the network is updated while tracking. When SSIM is less than the threshold $TH_2$, tracking failure is judged and re-tracking is performed. A detailed description of each functional module will be given in the following sections.

### 3.1 Correlation filter tracking

#### *3.1.1 Initialize the filter*

Suppose the size and the tracking box T centered on the target is given in the first frame, and the ideal Gaussian response corresponding to this tracking box is $G \in R^{M\times N}$, with its peak point corresponds to the position of the tracking target. Furthermore, suppose that the feature network has been trained offline. The number of convolution kernels (or the number of channels) of the output layer is $D$, the size of the convolution kernel is $m \times n$, and the output feature map is represented by $\varphi(T)$, and its size is $M \times N$, i.e., $\varphi(T) \in R^{M\times N\times D}$. According to the literature [24, 29, 33, 53, 54], the tracking filter can be initialized as:

$$W^l = \frac{\widehat{\varphi}^l(T) \odot \widehat{G}^*}{\sum_{l=1}^{D} \widehat{\varphi}^l(T) \odot (\widehat{\varphi}^l(T))^* + \lambda} \quad (l = 1, 2, \ldots D) \tag{1}$$

Here, $\odot$ denotes Hadamard product, * represents complex conjugate, $\wedge$ means Fourier transform, $W^l$ refers to the correlation filter of the $l$th feature channel. $\widehat{\varphi}^l(T)$ represents the Fourier transform of the feature output by the $l$th feature channel. $\widehat{G}^*$ represents the complex conjugate of the Fourier transform of the ideal Gaussian $G$. And the constant $\lambda \geq 0$ is regularization coefficient.

#### *3.1.2 Target location prediction*

In the tracking process, a search box $S$ centered on target with the same size as $T$ is constructed. The feature network is applied to $S$, and the extracted features are expressed as $\varphi^l(S)(l = 1, 2, \ldots, D)$. Then the tracking response $\left(r_{i,j}\right)_{M\times N}$ can be calculated by the following equation [54]:

$$g^l = \left(\widehat{W}^l\right)^* \odot \widehat{\varphi}^l(S) \quad (l = 1, 2, \ldots D) \tag{2}$$

$$\left(r_{i,j}\right)_{M\times N} = \mathcal{F}^{-1}\left(\sum_{l=1}^{D} g^l\right) \quad (l = 1, 2, \ldots D) \tag{3}$$

Here, $g^l$ represents the frequency output response of the $l$th tracking filter, $\mathcal{F}^{-1}$ represents the inverse Fourier transform, $r_{i,j} \in R^{M\times N}$ represents the total output response of all channels. Based on Eqs. (2) and (3), the coordinates $(x, y)$ of the predicted position of the target can be obtained as:

$$(x, y) = \max_{R^{M\times N}} \left(r_{i,j}\right)_{M\times N} \tag{4}$$

#### *3.1.3 On-line filter update*

During tracking process, the feature map $\varphi^l_{t-1}(S)$ of the target in the $t-1$ frame and the $W^{l-1}$ are employed to predict the position $(x_t, y_t)$ of the target in the $t$th frame by using Eqs. (2), (3), and (4). In the $t$th frame, construct a search box $S_t$ centered on $(x_t, y_t)$ with the same size as $T$. The ideal Gaussian response corresponding to this tracking box $S_t$ is configured as $G_t \in R^{M\times N}$. The feature network is applied to $S_t$, and the extracted feature is denoted as $\varphi(S_l)^l$. The filter $W^l_t$ is updated online by the following equation [54].

$$W^l_t = \frac{\widehat{\varphi}^l_t(S_t) \odot \left(\widehat{G_t}\right)^*}{\sum_{l=1}^{D} \widehat{\varphi}^l_t(S_t) \odot \left(\widehat{\varphi}^l_t(S_t)\right)^* + \lambda} \quad (l = 1, 2, \ldots D) \tag{5}$$

Here, $\widehat{\varphi}^l_t(S_t)$ represents the Fourier transform of the feature output $\varphi(S_l)^l$ of the $l$th feature channel, and $\left(\widehat{G_t}\right)^*$ represents the complex conjugate of the Fourier transform of the ideal Gaussian response $G_t$.

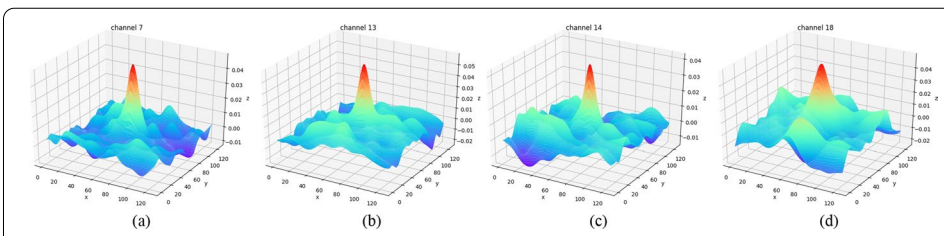


**Fig. 2** Typical channel output response. **a** The output response of channel 7; **b** The output response of channel 13; **c** The output response of channel 14; **d** The output response of channel 18

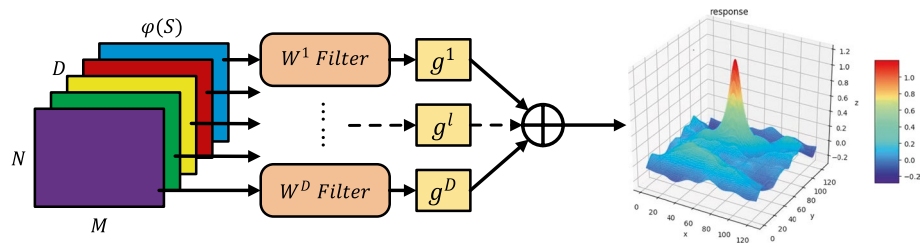


**Fig. 3** Intuitive diagram of multi-channel correlation filter tracking based on deep feature

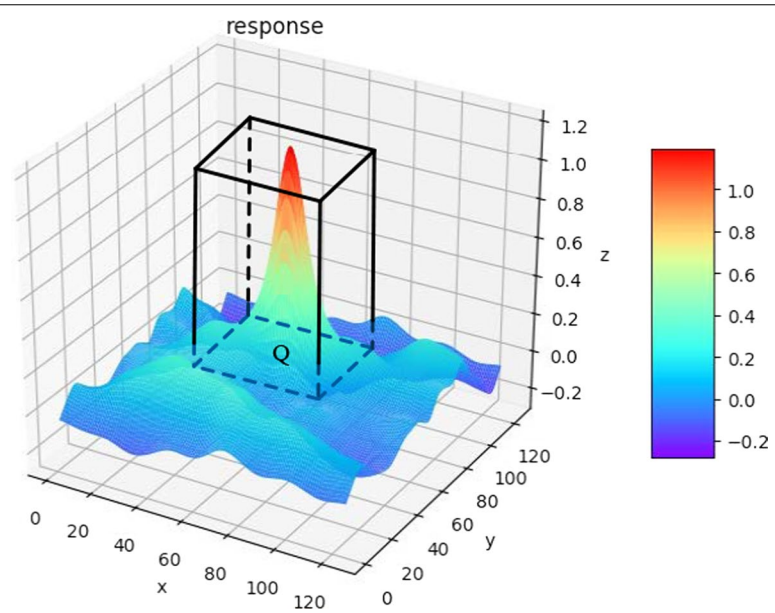


**Fig. 4** Intuitive diagram of tracking target total response

## 3.2 Important feature channel selection

### *3.2.1 Definition of channel importance*

The output layer of the feature network has $D$ output channels and outputs $D$ feature maps. According to Eqs. (2) and(3), each feature map is associated with its corresponding tracking filter and the output of the filter is a Gaussian response. Some typical outputs of all filters are shown in Fig. 2.

According to Eqs. (3) and (4), the final output response of the tracking system is the superposition of the output responses of all channels, and the peak value of the final output response corresponds to the predicted position of the target. The intuitive explanation of the whole process is shown in Fig. 3.

In terms of Eqs. (1), (2), and (3), and from Figs. 2 and 3, it can be seen that the final output response of the tracking system can be taken as an approximate two-dimensional Gaussian response. Therefore, the peak value of the two-dimensional Gaussian response corresponds to the predicted position $(x, y)$ of the target. According to the characteristics of the two-dimensional normal distribution function [60], about 95.5% of the data are concentrated in a rectangular box $Q$, with a length of $2 \times 2.58\sigma_x$ and a

width of $2 \times 2.58\sigma_y$, centered on $(\mu_x, \mu_y)$, as shown in Fig. 4. Here, $\mu_x$ and $\mu_y$ are the mean values of the two-dimensional normal distribution, and $\sigma_x$ and $\sigma_y$ are the variances of the two-dimensional normal distribution. Obviously,$(x, y) = (\mu_x, \mu_y)$.

The final output response of the tracking system is the superposition of the output responses of all channels. However, as can be seen from Figs. 2 and 3, the contribution of each channel to the final output response of the tracking system is completely different because the statistical distribution of the output response of each channel is different. The more the channel output response falls into this rectangular box, the greater the contribution to the total response. In other words, the channel is more important. As a result, it is natural to take the channel with more contributions as the important one which is reserved to lighten the feature network.

Thus, in term of contribution of the channel to the final output response of the tracking system, the channel importance is defined as the following.

The sum center$^l$ of the output response of the channel $l$ falling into the rectangular box $Q$ is defined as:

$$\left.\begin{array}{l} \text{center}^l = \sum_i \sum_j \mathcal{F}^{-1}\left(g_{i,j}^l\right) \\ i \in (x - 2.58\sigma_x, x + 2.58\sigma_x), j \in \left(y - 2.58\sigma_y, y + 2.58\sigma_y\right) \end{array}\right\} \tag{6}$$

The total sum around$^l$ of all the output response of the channel $l$ falling inside and outside the rectangular box $Q$ is defined as:

$$\text{around}^l = \sum_{i=1}^{M} \sum_{j=1}^{N} \mathcal{F}^{-1}\left(g_{i,j}^l\right) \tag{7}$$

The importance of the channel is defined as:

$$\text{score}^l = \frac{\text{center}^l}{\text{around}^l} \quad (l = 1, 2, \ldots, D) \tag{8}$$

The higher the score$^l$ is, the more the contribution of the channel response to the final response is, and the more important the channel is. Arrange score$^l$ in descending order, and take the channels corresponding to the first k score$^l$ as important channels for follow-up tracking.

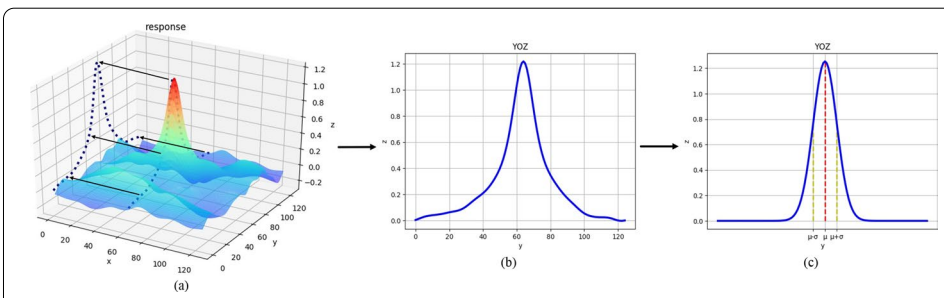


**Fig. 5** Schematic diagram of the projection and fitting of the total response on the YOZ plane

#### *3.2.2 Computing of $\sigma_x$ and $\sigma_y$*

Taking computing $\sigma_y$ as an example, describes the procedure for computing the variance of a two-dimensional Gaussian response.

The peak position of the target final response is $(x, y)$, which can be regarded as the mean value of the total response, i.e., $(x, y) = (\mu_x, \mu_y)$, and the variance of the final response is taken as $(\sigma_x, \sigma_y)$. Project the final response $(r_{i,j})_{M\times N}$ to the YOZ plane, as shown in Fig. 5a. It means that project the final response $(r_{i,j})_{M\times N}$ to YOZ is approximatively equivalent to $\left((r_{i,j})_{M\times N}|i = x\right)$, and the result is shown in Fig. 5b. For the projection curve shown in Fig. 5b, a one-dimensional Gaussian function can be used to fit the projection curve, shown in Fig. 5c.

Let $f(y_j) = r_{i,j}|i = x, j = 1, 2, \ldots, N$. The one-dimensional Gaussian fitting function is $f(y_j) = \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{(y_j-y)^2}{2\sigma_y^2}}$, where $y$ is the known mean and $\sigma_y$ is the parameter to be estimated. First, construct the likelihood function [61]:

$$L\left(\sigma_y^2\right) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{(y_i-y)^2}{2\sigma^2}} = \left(2\pi\sigma_y^2\right)^{-\frac{N}{2}} e^{-\frac{1}{2\sigma_y^2}\sum_{i=1}^{N}(y_i-y)^2} \tag{9}$$

Take the logarithm on both sides of Eq. (9) to get:

$$\ln L\left(\sigma_y^2\right) = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln\left(\sigma_y^2\right) - \frac{1}{2\sigma_y^2}\sum_{i=1}^{N}(y_i - y)^2 \tag{10}$$

Let the derivative of Eq. (10) for $\sigma_y^2$ be zero to get:

$$\frac{\partial \ln L\left(\sigma_y^2\right)}{\partial \sigma_y^2} = -\frac{N}{\sigma_y^2} + \frac{1}{2\left(\sigma_y^2\right)^2}\sum_{i=1}^{N}(y_i - y)^2 = 0 \tag{11}$$

The estimated value of $\sigma_y^2$ can be solved by Eq. (11):

$$\overline{\sigma}_y^2 = \frac{1}{N}\sum_{i=1}^{N}(y_i - y)^2 \tag{12}$$

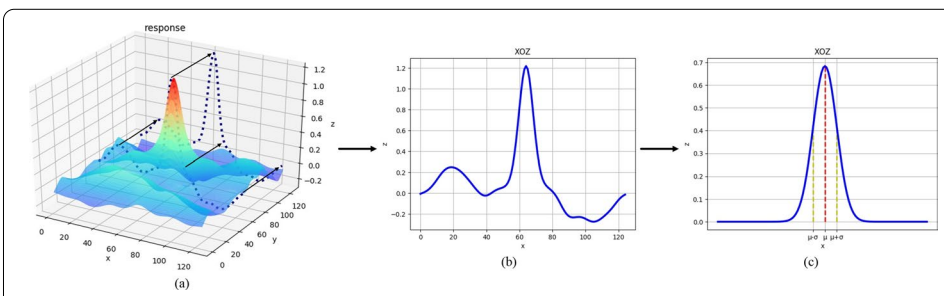


**Fig. 6** Schematic diagram of the projection and fitting of the total response on the *XOZ* plane

Then, the one-dimensional Gaussian fitting function is $f\left(y_j\right) = \frac{1}{\sqrt{2\pi}\bar{\sigma}_y} e^{-\frac{\left(y_j - y\right)^2}{2\bar{\sigma}_y^2}}$, as shown in Fig. 5c.

For computing of $\sigma_x$, project the final response $\left(r_{i,j}\right)_{M\times N}$ to the $XOZ$, as shown in Fig. 6a. It means that project the final response $\left(r_{i,j}\right)_{M\times N}$ to $XOZ$ is approximatively equivalent to $\left(\left(r_{i,j}\right)_{M\times N} | i = x\right)$, and the result is shown in Fig. 6b. For the projection curve shown in Fig. 6b, a one-dimensional Gaussian function can be used to fit the projection curve, shown in Fig. 6c. The computing method is the same as that of $\sigma_y$, and the result is shown in Fig. 6c.

In addition, if there are similar objects in an image, but not within the current search window, this case will not affect the overall performance of the tracking system. If there are similar objects in an image, and within the current search window, there may be more than one peak in Figs. 5 and 6. This case will affect the overall performance of the tracking system. However, such a case is not discussed in this paper because we only discussed the single target tracking.

### 3.3 On-line feature network update

The design of the feature network generally adopts off-line training and on-line fine-tuning strategy. In actual tracking applications, it is very possible for a target to have a variety of interference factors in the tracking video sequence, such as changes in ambient lighting, target scaling, target rotation, and etc. Therefore, the target features extracted by the off-line training network cannot accurately describe the disturbed target. It is difficult to accurately predict the position of the target, which restricts the improvement of the performance of the tracking system. To solve this problem, many feature network update strategies have been proposed around the similarity of targets in continuous tracking video sequences. Typical strategies mainly include similarity learning methods based on full convolution, peak signal-to-noise ratio (PSNR) method, SSIM method, and etc. The similarity learning method based on full convolution takes the learned similarity function as the similarity criterion, traverses all possible positions of the target, and takes the candidate position with the largest similar function value as the final predicted position of the target [34]. This method affects the real-time tracking performance because of the traversal calculation. In addition, another limitation to this method is that the candidate target with the largest similarity value is not necessarily the most similar (for example, the maximum similarity value with normalization is not greater than 0.5). PSNR is a widely used objective evaluation index based on error-sensitive images. However, since the visual characteristics of the human eyes are not taken into consideration, it is difficult to ensure that the evaluation results are completely consistent with the visual quality seen by the human eyes. Natural images are highly structured, and there are strong correlations among the pixels of the image. These correlations carry important information about the structure of objects in the visual scene. The Laboratory for Image and Video Engineering of the University of Texas at Austin proposed SSIM to measure the structural similarity of two images [55]. Taking into account the fuzzy changes of image structure information in human perception, SSIM measures image similarity from brightness, contrast, and structure respectively, and is better than PSNR in the

evaluation of image similarity [56]. In addition, SSIM has global statistical distribution characteristics, which can work well for a variety of interference factors and conforms to people's visual habits. Therefore, SSIM is selected as the criterion to calculate the similarity of two consecutive frames of the video sequence, and make decisions on the update of the decision feature network, the failure detection and tracking recovery of the tracking network in the paper.

#### 3.3.1 SSIM similarity criterion

Assuming that the two input images are $a$ and $b$ respectively, define SSIM as [55]:

$$\mathrm{SSIM}(a,b) = [l(a,b)]^{\alpha}[c(a,b)]^{\beta}[s(a,b)]^{\gamma} \tag{13}$$

$$l(a,b) = \frac{2\mu_a\mu_b + C_1}{\mu_a^2 + \mu_b^2 + C_1} \tag{14}$$

$$c(a,b) = \frac{2\sigma_{ab} + C_2}{\sigma_a^2 + \sigma_b^2 + C_2} \tag{15}$$

$$s(a,b) = \frac{\sigma_{ab} + C_3}{\sigma_a\sigma_b + C_3} \tag{16}$$

Here, $\alpha > 0$, $\beta > 0$ and $\gamma > 0$. $l(a,b)$ represents brightness comparison, $c(a,b)$ represents contrast comparison, and $s(a,b)$ represents structure comparison. $\mu_a$ and $\mu_b$ represent the mean value of image $a$ and image $b$ respectively. $\sigma_a$ and $\sigma_b$ represent the standard deviation of image a and image b respectively. $\sigma_{ab}$ represents the covariance of image $a$ and image $b$. $C_1, C_2$ and $C_3$ are constants, in order to avoid zero in the denominator.

In actual calculations, generally take $\alpha = \beta = \gamma = 1$, and $C_3 = C_2/2$. Substitute these parameters into Eqs. (13)–(16), reduce and merge to get[55]:

$$\mathrm{SSIM}(a,b) = \frac{(2\mu_a\mu_b + C_1)(\sigma_{ab} + C_2)}{\left(\mu_a^2 + \mu_b^2 + C_1\right)\left(\sigma_a^2 + \sigma_b^2 + C_2\right)} \tag{17}$$

Choose an appropriate threshold $\mathrm{TH}_1$, the SSIM similarity criterion is:

$$\left.\begin{array}{ll} \mathrm{SSIM}(a,b) \geq \mathrm{TH}_1, & a \text{ similar to } b \\ \mathrm{SSIM}(a,b) < \mathrm{TH}_1, & a \text{ not similar to } b \end{array}\right\} \tag{18}$$

#### 3.3.2 Strategy for updating feature network

Suppose the position of the tracking target in frame $t-1$ is $(x_{t-1}, y_{t-1})$, and the predicted position of the target in frame t is $(x_t, y_t)$. Construct two rectangular boxes $S_{t-1}$ and $S_t$ centered at $(x_{t-1}, y_{t-1})$ in frame $t-1$ and centered at $(x_t, y_t)$ in frame t respectively, all of them with the same size as $T$. Use Eq. (17) to calculate the SSIM of $S_{t-1}$ and $S_t$, and determine the similarity of $S_{t-1}$ and $S_t$ according to Eq. (18). To end this, two thresholds $\mathrm{TH}_1$ and $\mathrm{TH}_2$ are respectively set for SSIM, and $\mathrm{TH}_1 > \mathrm{TH}_2$. When $\mathrm{TH}_1 < \mathrm{SSIM}$, it indicates that the similarity between $S_{t-1}$ and $S_t$ is high, the tracking effect is better, and the tracking is kept as normal. In such a tracking process, the correlation filter is updated

but the feature network is not updated. When $TH_2 < SSIM < TH_1$, it indicates that the similarity between $S_{t-1}$ and $S_t$ is reduced because of the various interference factors such as illumination changes, occlusion, target zoom, target rotation, and etc. Although the tracking can be maintained, the tracking accuracy is reduced. It means that it is necessary for the feature network to be updated to increase the tracking accuracy. At this time, the parameters of the feature network of the $t-1$ frame are used as initial values, and the feature network is iteratively updated using the feature of the current frame target. The specific update process is as follows:

Step 1: Calculate the ideal Gaussian response graph corresponding to the predicted target position $(x_t, y_t)$, denoted as $G_{M\times N} = (g_{i,j})_{M\times N}$;

Step 2: Let the feature network of the $t-1$ frame perform on $S_t$, and extract the feature map of $S_t$ as $\varphi(S_t)$;

Step 3: Use Eqs. (2) and (3) to get the predicted response $(r_{i,j})_{M\times N}$ to the target at frame $t-1$;

Step 4: Construct the loss function loss as:

$$\text{loss} = \sum_{i=1}^{M}\sum_{j=1}^{N} \|r_{i,j} - g_{i,j}\|^2 \tag{19}$$

Step 5: Let $\frac{\partial \text{loss}}{\partial \varphi} = 0$, calculate the gradient, and back propagate to update the convolutional core parameters of each channel layer by layer. Finally obtain the update parameters of the feature network.

Step 6: Set $t \to (t-1)$, $(t+1) \to t$ to continue tracking in the next frame.

In addition, When the SSIM is between $TH_1$ and $TH_2$, take the parameters of the previous feature network as the initial values, and take the sum of squared errors between the Gaussian output response corresponding to the target predicted position and the ideal Gaussian response as the loss function, the feature network is trained by use of the stochastic gradient descent method so that it can capture the changes of target and environment and extract appropriate features. If there are similar objects in an image, but not within the current search window, the update will not lead to the tracking of an incorrect object. If there are similar objects in an image, and within the current search window, it is very possible that the update will lead to the tracking of an incorrect object. However, such a case is not discussed in this paper because we only focused on the single target tracking.

### 3.4 Re-tracking of targets

When $SSIM < TH_2$, it is considered that the tracking target has drifted out of the field of view, and it is judged that the tracking has failed. At this time, the tracking target should be searched in the neighborhood around the rectangular prediction box in the current frame or the subsequent frame in order to recover tracking from the failure. The specific re-tracking process is as follows:

Step 1: Taking the predicted position $(x_t, y_t)$ of the target in the $t-1$th frame as the center, a target search area $A$ which is n times larger than $T$ is constructed in the $t$th frame, where $n = \left(2, 3, \ldots, \text{Around}\left(\frac{W\times H}{M\times N}\right)\right)$, $W$ represents the width of the image, $H$ represents the height of the image, and Around(.) represents rounding;

Generally, $n = \text{Around}\left(\frac{W \times H}{M \times N}\right)$ is not taken. If so, it means that searching is performed within the entire image calculation is too heavy, which is not conducive to real-time tracking. In addition, during the movement of the target, the displacement between two consecutive frames is generally not too large. So it is not necessary to search within the whole image, as long as an appropriate n is selected according to the actual situation.

Step 2: Let a sliding window SW with the size of width $M$ and height $H$, start the sliding search with a sliding step of 1 from the upper left corner of $A$ to the lower right corner. In the entire search process, the SSIM between SW and $S_{t-1}$ is calculated according to Eq. (17) for each sliding search. Record the maximum value of all SSIMs in the search process as MaxSSIM;

Step 3: If MaxSSIM $>$ $\text{TH}_1$, it is considered that the target has been searched, and the window position corresponding to MaxSSIM is the predicted position of the target. And use this position to update $(x_t, y_t)$, jump to Step 5;

Step 4: If MaxSSIM $\not> \text{TH}_1$, it means there may be no target to be tracked in the current frame, and we need to continue searching in subsequent frames. Let $(t-1) \rightarrow (t-1)$, $(t+1) \rightarrow t$, jump to Step 6;

Step 5: Let $t \rightarrow (t-1), (t+1) \rightarrow t$;

Step 6: Continue to tracking or searching in the next frame.

## 4 Experiment results and analysis

### 4.1 Experiment environment

To verify the effectiveness of the method proposed in this paper, special software and hardware experimental environment is constructed. The hardware environment consists of a terminal (Intel(R) Core(TM) i5-3470), a server (X10DRG-Q) and a NVIDIA Corporation GP102 [TITAN Xp] GPU (video memory size is 12 GB). The software environment is as follows: Ubuntu 18.04 is selected as the operating system, Python is used for programming, and Pytorch based on deep learning is used as the development platform. Based on the above-mentioned software and hardware experimental environment and discriminant correlation filters network for tracking (DCFNet) framework [54], five experimental tasks are designed. It mainly includes (1) feature network pre-training with the training data set (VID) officially provided by the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [57] competition in 2015; (2) necessity of this work for tracking, with the data set OTB2013 [3]; (3) important feature selection with the data set OTB2013; (4) on-line update of the feature network with the data set OTB2013; (5) comparison study between our research work and other similar ones, with the data set OTB2013; (6) re-tracking, with the tracking test data set is OTB2013. The OTB2013 dataset is a sequence of 51 tracking videos used by Wu Y, Lim J, Yang M H. et al. in their article [3], which contains common interferences such as Scale Variation, Occlusion, Deformation, Fast Motion, etc. in the tracking task.

### 4.2 Feature network pre-training

The shallow features of the deep neural network describe more target spatial information, which is conducive to the spatial positioning of the target; while the deep features represent more semantic information of the target, which is conducive to target

classification. The shallow features of deep neural network are beneficial to target visual tracking because the key problem to be solved in target tracking is the prediction of target spatial position. Therefore, based on the DCFNet tracking framework, 2-layers convolutional neural network is selected as the feature network. Each layer has 32 convolution kernels, and the size is $3 \times 3$. Each initial weight w of the convolution kernel obeys uniform distribution $u(-\sqrt{k}, \sqrt{k})$ [59]. The value of k satisfies the Eq. (20):

$$k = \frac{1}{C_{\text{in}} \times \text{kernel}_{\text{size}[0]} \times \text{kernel}_{\text{size}[1]}} \tag{20}$$

Here, $C_{\text{in}}$ is the number of channels of the input tensor, $\text{kernel}_{\text{size}[0]} \times \text{kernel}_{\text{size}[1]}$ is the dimension of the convolution kernel ($3 \times 3$).

The activation function of the first layer of the network is ReLU, as shown in the following Eq. (21).

$$f(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \tag{21}$$

In addition, to ensure that the feature information of the target is as complete as possible, the feature network does not use the PooLing layer. Instead, the local response normalization (LRN) is used to replace the PooLing layer at the end of the convolutional layer (after the end of convolutional layer).

A video sequence is arbitrarily selected from the VID data set, and a rectangular tracking box with the 3 times size of the width $\times$ height of the tracking target is cropped centering on the determined tracking target in every frame of the video sequence. The rectangular box is resized to $125 \times 125$ as the network input. The feature network is applied to the rectangular box, and the extracted feature map is correlated with the correlation filter to get the final Gaussian output response. The overall network structure is shown in Fig. 7. The loss function is the sum of squared errors between the Gaussian output response and the ideal Gaussian response, and the training method adopts the stochastic gradient descent method (SGD). In the experiment, padding = 0, stride = 1, momentum = 0.9, learning rate = 0.01, weight decay = 0.00005, and mini-batch size set to 32. After 50 epoch training, the final feature network is obtained.

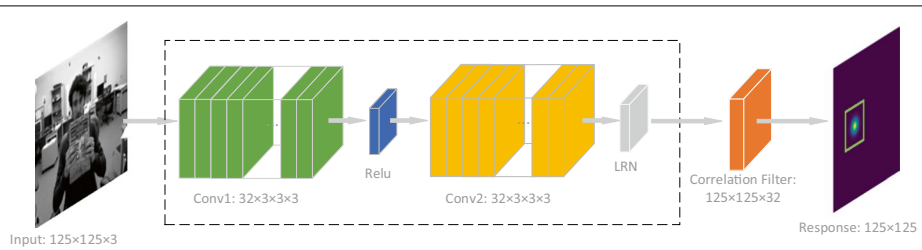


**Fig. 7** The overall network structure

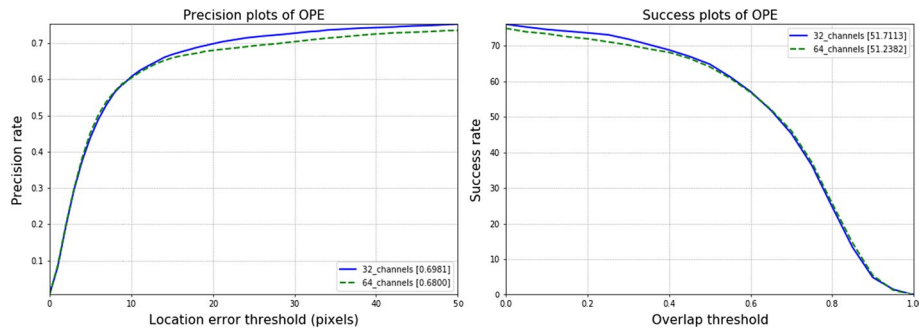


**Fig. 8** Success and precision plots on OTB2013 datasets with different network structure

**Table 1** Success and precision plots on OTB2013 datasets with different network structure

| Network structure | AOR | APE | Speed |
|---|---|---|---|
| 32 channels | 51.7113 | 0.6981 | 63.2 |
| 64 channels | 51.2382 | 0.6800 | 23.5 |

### 4.3 Necessity of this work for tracking

Two experiments were designed to illustrate the necessity of lightweight and online updating feature network for deep correlation filter tracking, as described below.

#### *4.3.1 Necessity of network slimming*

Based on the tracking network with 32-channels feature network cascading 32-channels filter network designed in Sect. 4.2, a new tracking network with 64-channels feature network cascading 64-channels filter network is constructed. These two tracking networks are respectively applied to OTB2013 data set, and the tracking results including tracking speed (FPS), tracking accuracy (Average Pixel Error (APE)) and tracking success rate (average overlap rate (AOR)) are shown in Fig. 8 and Table 1. As can be seen from Fig. 8 and Table 1, the tracking accuracy of the 64-channels tracking network does not change significantly compared with that of the 32-channels tracking network, while the tracking speed is greatly reduced to 23.5fps, which is lower than the basic requirement of real-time tracking 25fps. It can be seen that the number of channels in the feature network increases, but the tracking speed decreases. Therefore, it is necessary to simplify the feature network and improve the tracking speed for deep correlation filter tracking.

#### *4.3.2 Necessity of on-line training feature network*

Two tracking video sequences (Car4 and Liquor) are randomly selected from OTB2013 training data set. Taking channel 28 as an example, tracking experiments are conducted on tracking networks with and without updating networks respectively. The results (red box represents label or ground truth, green box for tracking result) are shown in Fig. 9, where Fig. 9a and c represent the tracking results of the tracking network without updating feature network, while Fig. 9b and d represent

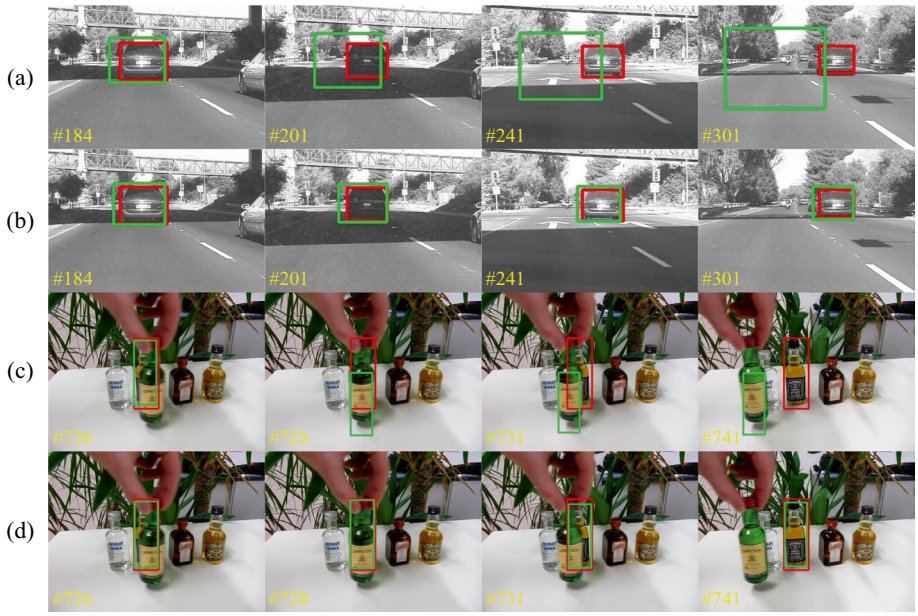


**Fig. 9** Comparison of online or non-online training the network

the tracking results of the tracking network with updating feature network. It can be clearly seen in video Car4 that the target tracking box drifts due to changes in illumination and target scale after the driving car passes shadow occlusion, i.e. tracking fails, as shown in Fig. 9a. In the tracking network with feature updating network, the vehicle can be still tracked stably after passing on the shadowed road surface, and no target box drift occurs, as shown in Fig. 9b. Similarly, in video Liquor, due to similar targets and occlusion, the tracking frame drifts from the target to the occlusion and the tracking fails, as shown in Fig. 9c. In the tracking network with feature updating network, when the shielding is removed, the tracking frame does not drift and can still track the target correctly, as shown in Fig. 9d. Therefore, it can be concluded that, in the tracking process, the pre-trained network cannot adapt to the changes of target and background well, so online update of feature network should be carried out according to the actual situation to ensure the tracking accuracy.

### 4.4 Important channel selection

A tracking video sequence is arbitrarily selected from the OTB2013 training data set, and a rectangular tracking box with the size of the width × height of the tracking target ($34 \times 81$, the value is the Basketball sequence in the OTB2013 data set) is constructed centering on the determined tracking target in the first frame of the video sequence. By applying a pre-trained feature network to this rectangular tracking box, a 32-channel feature map is obtained at the last layer of the feature network, which is written as $\mathrm{CHBU}_i$ (channel before updating), where $i = 1, 2, \ldots, 32$. Equations (6)–(8) and (12) were used to calculate the importance of each channel $score^i$ of each channel, and 32 channels were rearranged in descending order according to $\mathrm{score}^i$. Ablation experiments were performed on the rearranged 32 channels, i.e., 32, 30, 28, 26, 24, 22, and

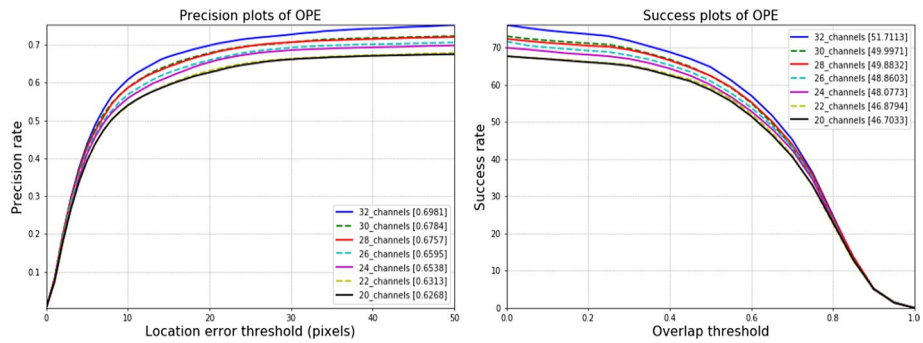


**Fig.10** Success and precision plots on the OTB2013 datasets with different channels

**Table 2** Success and precision as well as speed on the OTB2013 datasets with different channels

| Channel numbers | 32 | 30 | 28 | 26 | 24 | 22 | 20 |
|---|---|---|---|---|---|---|---|
| AOR | 51.7113 | 49.7113 | 49.8832 | 48.8603 | 48.0773 | 46.8794 | 46.7033 |
| APE | 0.6981 | 0.6784 | 0.6757 | 0.6595 | 0.6538 | 0.6313 | 0.6268 |
| Speed | 63.2 | 81.9 | 82.9 | 84.1 | 84.7 | 86.1 | 87.0 |

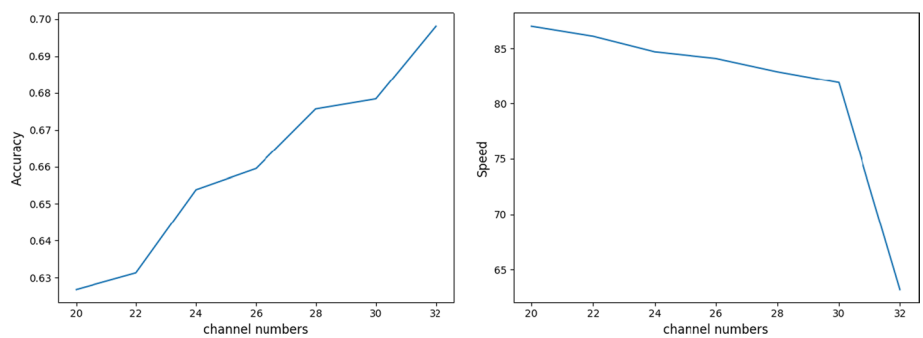


**Fig. 11** Accuracy and speed on the OTB2013 datasets with different channels

20 channels were taken as the important channels respectively, and target tracking was implemented using Eqs. (1)–(5). Tracking results are shown in Fig. 10 and Table 2, where one pass evaluation (OPE) means the first frame of the tracking sequence is initialized with the target ground truth position, and then the tracking algorithm is run to obtain the average tracking accuracy and success rate.

To more clearly observe the change of tracking accuracy and tracking speed with the change of channel number, we draw the change trend chart of tracking accuracy and channel number as well as the change trend chart of tracking speed and channel number according to Fig. 10 and Table 2, as shown in Fig. 11, where the speed means the average time taken by the tracking system for all tracking sequences from the first frame to the last frame, and the unit is frames per second (fps).

It can be seen from Fig. 11 that the tracking accuracy increases with the increase of the channel, but the tracking speed decreases with the increase of the channel. This is a

pair of irreconcilable contradictions, which can only be compromised in practical applications. We hope to reduce the number of channels as much as possible and increase the tracking speed under the premise of meeting the tracking accuracy. Based on this point of view, we select the 28 most important channels for follow-up tracking. It also can be seen from Fig. 11 and Table 2 that the tracking speed and the tracking accuracy of the original network are 0.6981 and 63.2 respectively, while the tracking speed and the tracking accuracy of our proposed network are 0.6757 and 82.9 respectively. Thus, a conclusion can be made that the tracking speed has been significantly improved, and the tracking accuracy has not dropped much compared with the original network, and the minimum accuracy can reach about (the tracking accuracy of our proposed network 0.6757/the tracking accuracy of the original network 0.6981) 95% of the original network. In addition, as can be seen in Fig. 11, the tracking speed starts to drop sharply when the number of channels is 30. Therefore, the number of channels should be selected in the range of less than 30 to ensure the tracking speed. While selecting the number of channels, the tracking accuracy should be taken into account. Therefore, in this paper, 28 channels are selected under the precondition of ensuring the system has high tracking accuracy. If higher tracking accuracy is not required, less than 28 channels can be selected to further increase the tracking speed. Therefore, choosing 30 channels may not be better.

In fact, the tracking speed decreases as the number of channels increases. However, this mapping relationship is a complex nonlinear relationship. It is possible that there is a certain critical point which the tracking speed suddenly decreases when the number of channels exceeds this critical point. In Fig. 9, 30 is a inflection point. When the number of channels exceeds this point, the tracking speed is drastically decreased to 65. This experimental result illustrates the above analysis.

In addition, the above data were adjusted by trials on the specific OTB2013 dataset. So, it is possible that the optimal number may be significantly changed if we change the dataset.

### 4.5 On-line feature network update

At each tracking step, Eqs. (13)–(18) are used to calculate the SSIM of the tracking box in the previous frame and the predicted tracking box of the current frame. When $TH_2 < SSIM < TH_1$, it indicates that it is necessary for the feature network to be updated online to enhance the tracking accuracy while tracking. When the TH *value* = 0.5, it indicates that the similarity of two images is in a critical state. When the TH value is lower than 0.5, it means that the two images are not similar and the target cannot be identified. This situation is judged as tracking failure and requires retrack, so the TH value for retrack is set to 0.5. When TH is higher than 0.5 and gradually

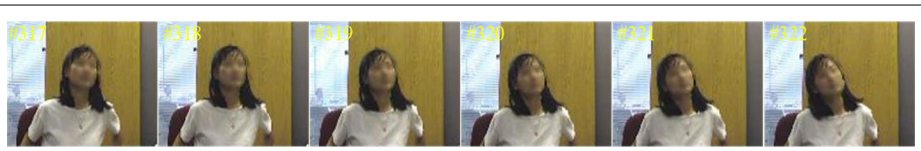


**Fig. 12** The SSIM of tracking video sequence girl gradually decrease

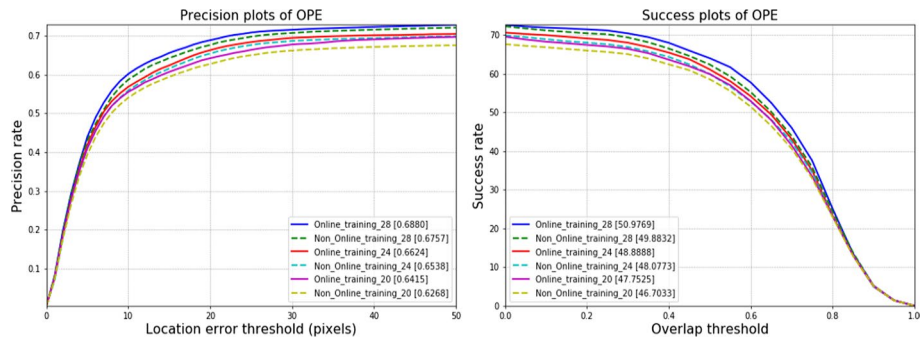


**Fig.13** Success and precision plots of different channels on OTB2013 datasets with Online or Non-online training

**Table 3** Success and precision plots of different channels on OTB2013 datasets with online or non-online training

| Channel numbers | Online training | AOR | APE | Speed |
|---|---|---|---|---|
| 28 | No | 49.8832 | 0.6757 | 76.9 |
| | Yes | 50.9769 | 0.6880 | 45.4 |
| 24 | No | 48.0773 | 0.6538 | 77.7 |
| | Yes | 48.8888 | 0.6624 | 46.5 |
| 20 | No | 46.7033 | 0.6268 | 78.9 |
| | Yes | 47.7525 | 0.6415 | 46.8 |

approaches to 1, it means that the confidence of determining the target is getting higher. Therefore another threshold value is chosen between 0.5 and 1, which determines when to update the feature network. If the selected threshold is closer to 1, the feature network needs to be updated more frequently, thus leading to an increased computational burden. Therefore, for experimental convenience, an intermediate value between 0.5 and 1 was chosen, i.e. $\mathrm{TH} = 0.7$. Select a tracking video sequence arbitrarily from the OTB2013 training data set. When $\mathrm{TH}_1 < \mathrm{SSIM}$, normal tracking; when $0.5 < \mathrm{SSIM} < 0.7$, update the network, i.e., from frame 317 to frame 322 of the selected tracking video sequence Girl, as shown in Fig. 12, the SSIM began to decrease because of the target's appearance (posture) changes. At this time, the parameters of the previous frame's feature network are taken as the initial values, Eq. (19) is taken as the loss function, and the feature network online training is carried out based on the predicted tracking box of the current frame and its corresponding ideal Gaussian response; meanwhile, target tracking is continued to complete. Taking the lightened feature network with different channels as an example, comparative experiments were carried out on the update and non-update feature networks of the tracking network respectively. The experimental results are shown in Fig. 13 and Table 3.

It can be seen from Fig. 13 and Table 3 that both AOR and APE of the online update network with different channels are significantly improved, but the tracking speed is significantly reduced, compared to non-on-line updating network of different channels. Although the tracking speed has been reduced, it can still meet the engineering

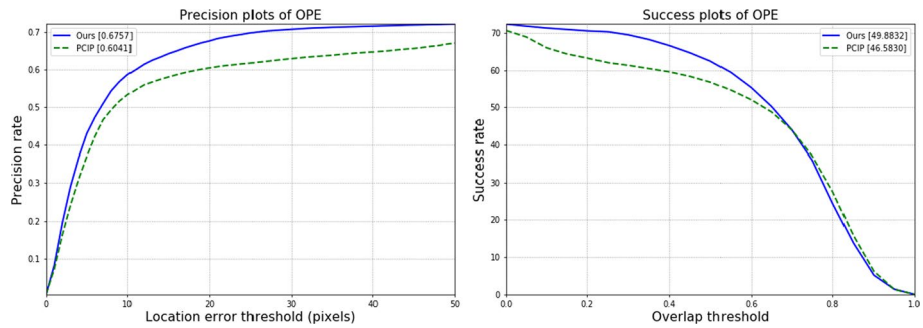


**Fig.14** Success and precision plots of different feature network slimming methods on OTB2013 datasets

**Table 4** Success and precision plots of different feature network slimming methods on OTB2013 datasets

| Method | AOR | APE | Speed |
|---|---|---|---|
| Ours | 49.8832 | 0.6757 | 76.9 |
| PCIP | 46.5830 | 0.6041 | 78.2 |

requirements, in other words, the tracking speed can reach the speed more than the minimum 25 fps. Thus, we can conclude that, on the premise of meeting the requirements of engineering tracking speed, the online update of feature network should be considered according to the actual situation, so that the tracking system can adapt to the ever-changing environment.

### 4.6 Comparison study between our research work and other similar ones

Two experiments are designed to illustrate the effectiveness and novelty of the proposed deep correlation filter tracking algorithm, which is composed of lightweight and online updating feature networks, compared with other similar research work. Details are described below.

#### *4.6.1 Comparison of different feature network slimming methods*

Using the pruning deep feature networks using channel importance propagation (PCIP) [59] channel pruning method, the pre-trained feature network is first pruned into a light-weight feature network with 28 channels. Then the light-weight feature network is cascaded with relevant filters to form a tracking network. Take the tracking results of this tracking network on the OTB2013 dataset as baseline. Using the light-weight feature network method proposed by us, a 28-channel tracking network with the same structure is obtained. The tracking results on OTB2013 dataset are compared with the baseline, and the results are shown in Fig. 14 and Table 4. As can be seen from Fig. 14 and Table 4, the tracking speed of our method is not much different from that of PCIP method, but the AOR and APE of our method are slightly improved compared with that of PCIP method, indicating that our lightweight feature network method and PCIP channel pruning method have very close tracking effect.

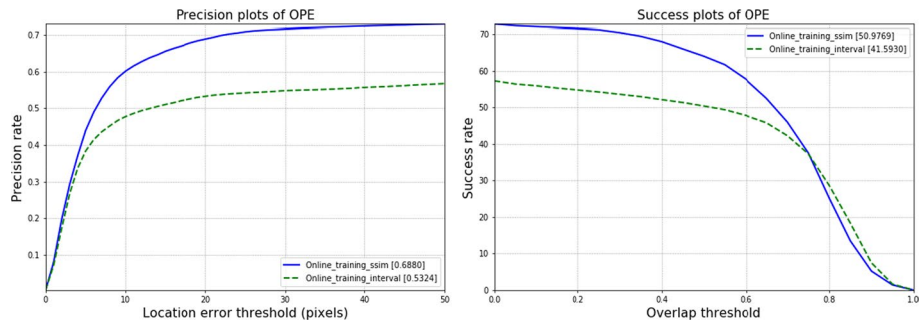


**Fig.15** Success and precision plots on OTB2013 datasets with different online training methods

**Table 5** Success and precision plots on OTB2013 datasets with different online training methods

| Online training | AOR | APE | Speed |
|---|---|---|---|
| SSIM | 50.9769 | 0.6880 | 45.4 |
| Interval | 41.5930 | 0.5324 | 16.7 |

In addition, more importantly, the tracking system constructed by the light-weight feature network method proposed by us has obvious physical interpretability, which facilitates the adjustment of tracking model parameters.

#### *4.6.2 Comparison of different on-line training feature network methods*

The feature network update strategy based on SSIM and the feature network update strategy based on the fixed interval (e.g. 2 frames) update strategy were selected [43]. The proposed 28-channels tracking networks with different update strategies were respectively applied on the OTB2013 data set. The tracking results are shown in Fig. 15 and Table 5. As can be seen from Fig. 15 and Table 5, the tracking system based on SSIM update strategy proposed by us is far superior to the tracking system based on fixed interval update strategy in terms of tracking accuracy and tracking speed, thus meeting the basic requirements of real-time tracking. In conclusion, the SSIM updating strategy proposed by us can make the network adapt to the changes of target and environment in the tracking process, and improve the tracking accuracy while ensuring the tracking speed.

### 4.7 Re-tracking after tracking failure

At each tracking step, Eqs. (13)–(18) are used to calculate the SSIM of the tracking box in the previous frame and the predicted tracking box in the current frame. When $TH_2 > SSIM$, it is judged that the tracking has failed. At this time, the tracking target should be searched in the neighborhood around the rectangular prediction box in the current frame or the subsequent frame in order to recover tracking from the failure. The specific process has been described in detail in Sect. 3.4. To verify the effectiveness of the strategy proposed in this paper, we constructed an artificial simulation environment. Three tracking video sequences (Boy, David2 and Suv) are arbitrarily

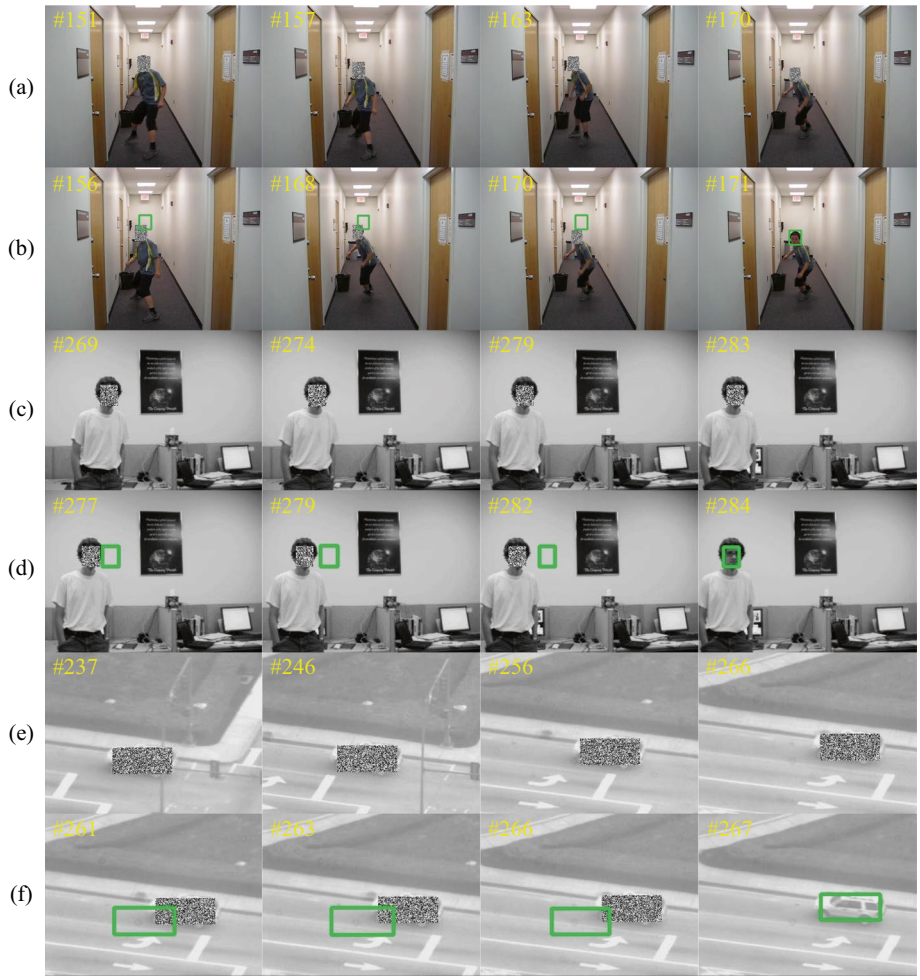


**Fig. 16** **a** Tracking video sequence boy with an artificial mask; **b** recovery tracking result based on the tracking video sequence boy with an artificial mask; **c** tracking video sequence David2 with an artificial mask; **d** recovery tracking result based on the tracking video sequence David2 with artificial mask; **e** tracking video sequence Suv with an artificial mask; **f** recovery tracking result based on the tracking video sequence Suv with artificial mask

selected from the OTB2013 training data set. Boy starts from frame 151 until the end of frame 170, David2 starts from frame 269 to the end of frame 283, and Suv strats from frame 237 to 266. The tracking box is partly covered with mosaics, as shown in Fig. 16a, c, e.

During the experiment, the re-tracking method described in Sect. 3.4 is used, and let $n = 3$. We re-track the simulation tracking sequence constructed in Fig. 16a, c, e, and the results are shown in Fig. 16b, d, f.

It can be seen from Fig. 16b, d, f that when the tracking target drifts out of the tracking window or field of view, it still appears near the prediction window of the current frame or in the subsequent frames, the re-tracking strategy proposed in this paper keeps searching the tracking target in the neighborhood around the rectangular prediction box in the current frame or the subsequent frame till the target is found in a certain frame (frame 171 in the Boy video sequence, frame 284 in the David2 video sequence, and frame 267 in

the Suv video sequence). Therefore, it is easy for the re-tracking strategy proposed in this paper to search for the tracking target, relocate the target, and re-track the target.

In addition, if the target drift range is relatively large, one of the effective ways to solving the problem is to increase the $n$ described in Sect. 3.4, i.e., to increase the search space. This will result in decreasing the tracking speed.

## 5 Conclusion

In this paper, we propose a deep correlation filter algorithm based on the importance of the channel, which lightens the feature network and improves the tracking speed on the premise of ensuring tracking accuracy. In the tracking process, the SSIM of the predicted tracking target in two consecutive frames of the video sequence is calculated in real-time. Based on the SSIM, determine whether the feature network needs to be updated, and decide whether the tracking fails. The experimental results on the OTB benchmark data set show that (1) Our model can improve the tracking speed and achieve a balance between accuracy and speed while ensuring the tracking accuracy; (2) The online update of the feature network can make the network adapt to the complex background and target changes and improve tracking accuracy; (3) In the case of tracking failure (the target drifts out of the field of view), the re-tracking module can search for the target again and resume tracking. (4) The model is simple and physically interpretable. All of these advantages will provide a new effective tracking method to the field of the single target tracking research. But there are some limitations of this method. One is that if there are similar objects in an images, and within the current search window, there may be more than one peak in the output response of the tracking system. This case will affect the overall performance of the tracking system. Another is how to adaptively select the threshold for updating the feature network of the tracking system. These limitations are our future work to be further researched.

**Abbreviations**

SSIM: Structural similarity index; VOT: Visual object tracking; MOSSE: Minimum output sum of squared error filter; CSK: Circulant structure of tracking-by-detection with kernels; KCF: Kernelized correlation filters; SAMF: Scale adaptive multiple feature; DSST: Discriminative scale space tracker; DCNN + CF: Deep convolutional neural network + correlation filter; ASRCF: Adaptive spatially regulated correlation filters; SRDCF: Spatially regulating discriminative correlation filters; MTCF: Visual object multimodality tracker based on correlation filters; HOG: Histogram of oriented gradient; HCFT: Hierarchical convolutional features for visual tracking; C-COT: Continuous convolution operator tracker; SiamFC: Fully continuous siamese networks; Siammask: Siamese network with mask; SiamDW: Deep and wider siamese networks; GAP: Global average pooling; AFER: Average feature energy ratio; LASSO: Least absolute shrinkage and selection operator; OT: Optimal thresholding; CREST: Convolutional residual learning scheme for visual tracking; SVM: Support vector machines; PSR: Peak to sidelobe ratio; PSNR: Peak signal-to-noise ratio; DCFNet: Discriminant correlation filters network for tracking; VID: Object detection from video; ILSVRC: ImageNet large scale visual recognition challenge; LRN: Local response normalization; SGD: Stochastic gradient descent; APE: Average pixel error; AOR: Average overlap rate; PCIP: Pruning deep feature networks using channel importance propagation.

**Acknowledgements**

Not applicable.

**Authors' contributions**

GY and CL proposed the original idea of the full text. CL designed and implemented the simulation experiments and analyzed the results. GY, CL, and HC wrote the manuscript. All authors read and approved the final manuscript.

**Funding**

This work was supported in part by the National Key R&D Program of China under Grant 2020YFB1406702-3.

**Availability of data and materials**

All the source codes and related pictures will be available from the corresponding author upon request.

## Declarations

**Ethics approval and consent to participate**
Not applicable.

**Competing interests**
The authors declare that they have no competing interests.

Received: 6 August 2021 Accepted: 2 March 2022
Published online: 28 March 2022

**References**

1. X. Li, Y. Zha, T. Zhang, Z. Cui, W. Zuo, Z. Hou, H. Lu, H. Wang, Survey of visual object tracking algorithms based on deep learning. J. Image Graph. **24**(12), 2057–2080 (2019). (**in Chinese**)
2. H. Lu, P. Li, D. Wang, Visual object tracking: a survey. Pattern Recogn. Artif. Intell. **31**(1), 61–76 (2018). (**in Chinese**)
3. Y. Wu, L. Jongwoo, M.-H. Yang, Online object tracking: a benchmark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), pp. 2411–2418
4. S.M. Marvasti-Zadeh, L.C. Seyed, H. Ghanei-Yakhdan, S. Kasaei, Deep learning for visual tracking: a comprehensive survey. arXiv e-prints arXiv-1912 (2019)
5. P. Li, D. Wang, L. Wang, Lu. Huchuan, Deep visual tracking: review and experimental comparison. Pattern Recogn. **76**, 323–338 (2018)
6. X.-Q. Zhang, R.-H. Jiang, C.-X. Fan, T.-Y. Tong, T. Wang, P.-C. Huang, Advances in deep learning methods for visual tracking: literature review and fundamentals. Int. J. Autom. Comput. **18**, 311–333 (2021)
7. Y. Tang, Y. Liu, H. Huang, Survey of single-target visual tracking algorithms. Meas. Control Technol. **39**(8), 21–34 (2020). (**in Chinese**)
8. L. Meng, X. Yang, A survey of object tracking algorithms. Acta Autom. Sin. **45**(07), 1244–1260 (2019). (**in Chinese**)
9. J. Zhu, Y. Lao, Y.F. Zheng, Object tracking in structured environments for video surveillance applications. IEEE Trans. Circuits Syst. Video Technol. **20**(2), 223–235 (2009)
10. N. Buch, S.A. Velastin, J. Orwell, A review of computer vision techniques for the analysis of urban traffic. IEEE Trans. Intell. Transp. Syst. **12**(3), 920–939 (2011)
11. M. Brown, J. Funke, S. Erlien, J.C. Gerdes, Safe driving envelopes for path tracking in autonomous vehicles. Control Eng. Pract. **61**, 307–316 (2017)
12. J. Xin, X. Du, J. Zhang, Deep learning for robust outdoor vehicle visual tracking. In *2017 IEEE International Conference on Multimedia and Expo (ICME)* (IEEE, 2017), pp. 613–618
13. S.S. Rautaray, A. Agrawal, Vision based hand gesture recognition for human computer interaction: a survey. Artif. Intell. Rev. **43**(1), 1–54 (2015)
14. J.M.B. Onate, D.J.M. Chipantasi, N. del Rocio Velasco Erazo, Tracking objects using artificial neural networks and wireless connection for robotics. J. Telecommun. Electron. Comput. Eng. (JTEC) **9**(1–3), 161–164 (2017)
15. J. Hao, Y. Zhou, G. Zhang, Q. Lv, Q. Wu, A review of target tracking algorithm based on UAV. In *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)* (IEEE, 2018), pp. 328–333
16. J. Luo, Y. Han, L. Fan, Underwater acoustic target tracking: a review. Sensors **18**(1), 112 (2018)
17. D. Comaniciu, P. Meer, Mean shift: a robust approach toward feature space analysis. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 603–619 (2002)
18. J. Shi, Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 1994), pp. 593–600
19. K. Nummiaro, E. Koller-Meier, L. Van Gool, An adaptive color-based particle filter. Image Vis. Comput. **21**(1), 99–110 (2003)
20. D.S. Bolme, J. Ross Beveridge, B.A. Draper, Y.M. Lui, Visual object tracking using adaptive correlation filters. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (IEEE, 2010), pp. 2544–2550
21. J.F. Henriques, R. Caseiro, P. Martins, J. Batista, Exploiting the circulant structure of tracking-by-detection with kernels. In *European Conference on Computer Vision* (Springer, Berlin, 2012), pp. 702–715
22. J.F. Henriques, R. Caseiro, P. Martins, J. Batista, High-speed tracking with kernelized correlation filters. IEEE Trans. Pattern Anal. Mach. Intell. **37**(3), 583–596 (2014)
23. Y. Li, J. Zhu, A scale adaptive kernel correlation filter tracker with feature integration. In *European Conference on Computer Vision* (Springer, Cham, 2014), pp. 254–265
24. M. Danelljan, G. Häger, F. Khan, M. Felsberg, Accurate scale estimation for robust visual tracking. In *British Machine Vision Conference, Nottingham*, September 1–5, 2014 (Bmva Press, 2014)
25. C. Fu, B. Li, F. Ding, F. Lin, G. Lu, Correlation filters for unmanned aerial vehicle-based aerial tracking: a review and experimental evaluation. arXiv preprint arXiv:2010.06255 (2020)
26. Y. Li, L. Song, Y. Chen, Z. Li, X. Zhang, X. Wang, J. Sun, Learning dynamic routing for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2020), pp. 8553–8562
27. X. Jia, B. De Brabandere, T. Tuytelaars, L.V. Gool, Dynamic filter networks. Adv. Neural Inf. Process. Syst. **29**, 667–675 (2016)
28. K. Dai, D. Wang, H. Lu, C. Sun, J. Li, Visual tracking via adaptive spatially-regularized correlation filters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 4670–4679
29. M. Danelljan, G. Hager, F.S. Khan, M. Felsberg, Learning spatially regularized correlation filters for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 4310–4318
30. G. Yang, Q. Wei, Visual object multimodality tracking based on correlation filters for edge computing. Secur. Commun. Netw. **2020**, 1–13 (2020)

31. M. Danelljan, G. Hager, F.S. Khan, M. Felsberg, Convolutional features for correlation filter based visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision Workshops* (2015), pp. 58–66
32. C. Ma, J.-B. Huang, X. Yang, M.-H. Yang, Hierarchical convolutional features for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 3074–3082
33. M. Danelljan, A. Robinson, F.S. Khan, M. Felsberg, Beyond correlation filters: learning continuous convolution operators for visual tracking. In *European Conference on Computer Vision* (Springer, Cham, 2016), pp. 472–488
34. L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, P.H.S. Torr, Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision* (Springer, Cham, 2016), pp. 850–865
35. Q. Wang, L. Zhang, L. Bertinetto, W. Hu, P.H.S. Torr, Fast online object tracking and segmentation: a unifying approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 1328–1338
36. Z. Zhang, H. Peng, Deeper and wider siamese networks for real-time visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 4591–4600
37. X. Li, C. Ma, B. Wu, Z. He, M.-H. Yang, Target-aware deep tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019), pp. 1369–1378
38. M. Che, R. Wang, Y. Lu, Y. Li, H. Zhi, C. Xiong, Channel pruning for visual tracking. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops* (2018)
39. Y. He, X. Zhang, J. Sun, Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 1389–1397
40. Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, C. Zhang, Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE International Conference on Computer Vision (2017)*, pp. 2736–2744
41. Y. Ye, G. You, J.-K. Fwu, X. Zhu, Q. Yang, Y. Zhu, Channel pruning via optimal thresholding. In *International Conference on Neural Information Processing* (Springer, Cham, 2020), pp. 508–516
42. H. Nam, B. Han, Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 4293–4302
43. Y. Song, C. Ma, L. Gong, J. Zhang, R.W.H. Lau, M.-H. Yang, Crest: convolutional residual learning for visual tracking. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 2555–2564
44. Z. Qiu, Y. Zha, P. Zhu, M. Wu, Visual tracking algorithm based on online feature discrimination with siamese network. Acta Opt. Sin. **39**(09), 253–261 (2019). (**in Chinese**)
45. X. Liu, Y. Zhou, Online tracking with convolutional neural networks. In *International Conference on Neural Information Processing* (Springer, Cham, 2017), pp. 208–216
46. C. Ma, J.-B. Huang, X. Yang, M.-H. Yang, Adaptive correlation filters with long-term and short-term memory for object tracking. Int. J. Comput. Vis. **126**(8), 771–796 (2018)
47. R. Walsh, H. Medeiros, Detecting tracking failures from correlation response maps. In *International Symposium on Visual Computing* (Springer, Cham, 2016), pp. 125–135
48. Y. Wang, X. Luo, L. Ding, S. Fu, X. Wei, Robust visual tracking based on response stability. Eng. Appl. Artif. Intell. **85**, 137–149 (2019)
49. J. Shin, H. Kim, D. Kim, J. Paik, Fast and robust object tracking using tracking failure detection in kernelized correlation filter. Appl. Sci. **10**(2), 713 (2020)
50. S. Li, J. Chu, G. Zhong, L. Leng, J. Miao, Robust visual tracking with occlusion judgment and re-detection. IEEE Access **8**, 122772–122781 (2020)
51. B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, A. Torralba, Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2921–2929
52. R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 618–626
53. M. Danelljan, G. Bhat, F.S. Khan, M. Felsberg, Eco: efficient convolution operators for tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017) , pp. 6638–6646
54. Q. Wang, J. Gao, J. Xing, M. Zhang, W. Hu, Dcfnet: discriminant correlation filters network for visual tracking. arXiv preprint arXiv:1704.04057 (2017).
55. Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
56. Data-Mining, The Difference between PSNR and SSIM. Last modified May 29, 2018. https://blog.csdn.net/liuzehn/article/details/80495763
57. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang et al., Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
58. X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. JMLR Workshop and Conference Proceedings* (2010), pp. 249–256
59. H. Chen, C. Li, Pruning deep feature networks using channel importance propagarion. In *2021 2nd International Conference on Computer Science and Management Technology* (in press)
60. D. Curran-Everett, S. Taylor, K. Kafadar, Fundamental concepts in statistics: elucidation and illustration. J. Appl. Physiol. **85**(3), 775–786 (1998)
61. S. Nadarajah, A generalized normal distribution. J. Appl. Stat. **32**(7), 685–694 (2005)


## Publisher's Note