Target Tracking Based on Deep Optimization Features

Tingting Yang¹, Haiping Wei¹*, Yuanchen Qi¹ and Naixiao Cui²

¹School of Computer and Communication Engineering, Liaoning Shihua University, Fushun 113001, Liaoning, China
²Bank of Communications Jinzhou Branch, Jinzhou 121000, Liaoning, China

*yytt5185@163.com

Abstract. Target tracking algorithm based on deep learning generally adopts offline trained deep network to extract features, but with unsatisfactory separation effect of the target and the background. A target tracking algorithm based on deep optimization features is proposed to solve this problem. First, the Conv3-1 and Conv4-3 layers are selected from the offline trained convolutional neural network model VGGNet16 for basic feature extraction. Then the loss function is used to quantitatively describe the effectiveness of different filter positioning target in the basic feature layer. Finally, the selected high-efficiency filters are fused to extract deep optimization features and achieve effective tracking under the correlation filtering framework. The experimental results show that compared with other related algorithms, this algorithm can better recognize the target and present better results.

1. Introduction

Target tracking is one of the important research directions in the field of computer vision, and plays an important role in the fields of precision guidance, intelligent video surveillance, human-computer interaction, and public safety, etc. After giving a target object specified by the bounding box of the first frame, target tracking aims to accurately locate the target object in subsequent frames. Although the research on target tracking has made great progress in recent years, it still faces many challenges, for example: the target object undergoes deformation, sudden movement, background clutter, occlusion and even temporarily leaves the field of vision, etc., causing the tracking to drift. Therefore, how to make the tracking algorithm more robust to meet the above challenges is still the core issue of current target tracking research.

Because of its good real-time and robustness, correlation filtering has gradually become a research hotspot in the field of target tracking, and a series of algorithms have been derived. Bolme et al. [1] propose the method of Minimum Output Sum of Squared Error filter (MOSSE). This method introduces correlation filtering to the field of target tracking for the first time. The target is located by calculating the maximum response value of the filter and the current frame image. Next Henriques et al. [2] propose the algorithm of Discriminative Correlation Filter (DCF), which use cyclic shifting for dense sampling, adopt a multi-feature fusion method and optimize the kernel function.

With the deepening of deep learning theory research, deep learning-based target tracking algorithms have also achieved good results. Li et al. [3] propose the Target-aware deep tracking Algorithm (TADT), which calculates the gradient value of the loss to guide the selection of channels, and integrating with the Siamese framework, effectively improving Tracking efficiency. Ma et al. [4] propose a hierarchical convolutional feature tracking algorithm (HCF), which takes into account that
convolution features of latter layers contain more semantic information to search the approximate range of the target and earlier layers contain more spatial information to locate the target accurately. Although HCF has a better tracking effect, the CNN model used to extract hierarchical convolutional features is trained offline, it is still some drawbacks. First, the target to be tracked may be arbitrary forms, but the training set used in training the off-line model may not contain the target; second, even if the training set contains the target, the depth feature extracted by the off-line model usually retains the information which is not very effective to distinguish the target from the background. In this paper, the HCF tracker with off-line training network is improved by adding optimization functions [5,6,7,8]. A deep optimization feature extractor is constructed to select the most effective feature for distinguishing specific target, so the tracking accuracy can be improved effectively.

2. Tracking algorithm based on depth optimization feature

2.1. Deeply optimized feature extraction

Deep models trained offline have poor discrimination and contain redundant information which leads to overfitting. For this reason, this paper improves on the basis of HCF tracker details as follows:

First of all, HCF selected the three-layer convolution features of Conv3-4, Conv4-4 and Conv5-4. But through a large number of experiments, this article found that only extracting the optimization features on the convolution layers of Conv3-1 and Conv4-3 can obtain good results.

The algorithm for deep optimal feature extraction will be introduced in detail. Because the convolutional layer of the convolutional neural network is composed of a series of convolution filters, choosing the most effective combination of convolution filters is the key to extracting the optimization features. And the basic idea of CNN training is to calculate the loss between the actual output and the ideal output through back propagation, and then use the gradient descent method to continuously adjust the parameters of the convolution filter, which is to minimize the loss, and finally a suitable combination of convolution filters is obtained. Therefore, the effectivenessβ of each convolution filter for a specific target can be calculated by the gradient of the loss, so that a combination of high-efficiency filters is selected to achieve feature optimization. To this end, this paper introduces the ridge regression loss function to extract the most effective convolution filter combination for the specific target discriminability for two-layer convolution features of Conv3-1 and Conv4-3, respectively. The ridge regression loss function minimizes the loss by obtaining a suitable combination weight M, treats the Gaussian function \( Y(i,j) \) as the ideal output, takes the input feature samples in the image block aligned with the center of the target as \( h_i \), takes the convolution of \( h_i \) and \( M \) as the actual output \( H_{out} \), and introduces \( L_2 \) regular term \( \mu \| M \|_2 \) to prevent overfitting. Its formula is as follows:

\[
L_{red} = \| Y(i,j) - M \ast h_i \|_2^2 + \mu \| M \|_2
\]

(1)

Where \((i, j)\) is the offset from the target center, and \(\mu\) is the regularization parameter. The effectiveness of each filter can be based on its effect on effectively reducing the loss, that is, the gradient of the regression loss. The formula is as follows:

\[
\frac{\partial L_{red}}{\partial H_m} = \sum_{i,j} \frac{\partial L_{red}}{\partial H_m(i,j)} \times \frac{\partial H_m(i,j)}{\partial H_m(i,j)} = \sum_{i,j} 2(Y(i,j) - H_m(i,j)) \times M
\]

(2)

Calculate the effectivenessβ of the i-th convolution filter for a specific target by the gradient of the loss. The formula is as follows:

\[
\beta_i = G_{AP} \left( \frac{\partial L}{\partial h_i} \right)
\]

(3)

Where \(G_{AP}\) represents the global average pooling function, that is, the average value is calculated for all pixels in the feature map of each layer. \(L\) is the loss function, \(h_i\) represents the output features of the i-th filter.
Based on the effectiveness $\beta$, the filters that can easily distinguish the target from the background interference can be selected, that is, the most effective filter combination $x = \hat{f}(h; \beta)$ for distinguishing the target, and $f$ is the function that selects the most efficient channel.

In addition, because the low-level convolution features have higher spatial information, this paper also needs to extract the features with high robustness to scale transformation on the lower-level Conv3-1 convolution features. First, we use the convolution feature to generate training samples of different sizes, and arrange the training samples to form paired training sample sets. Then we take a novel pairwise ranking loss function [9], filters that are robust to scale transformations extracted by the effectiveness of the ranking loss gradient. The pairwise ranking loss formula is as follows:

$$L_{\text{ran}} = \log(1 + \sum_{(h_i, h_j) \in \mathcal{L}} \exp(f(h_i) - f(h_j)))$$  \hspace{1cm} (4)

Where $(h_i, h_j)$ is an ordered paired training sample, and the size of $h_j$ is closer to the target size than $h_i$, and $f(h; w)$ is the prediction model. The derivation formula of $L_{\text{ran}}$ with respect to $f(h)$ is as follows:

$$\frac{\partial L_{\text{ran}}}{\partial f(h)} = -\frac{1}{L_{\text{ran}}} \sum_{i} \beta_{z_i} \exp(-f(h)\beta_{z_i})$$  \hspace{1cm} (5)

where $\beta_{z_j} = z - z_i$ and $z_i$ is a one-hot vector. Similar to formula (6), the gradient formula of ranking loss can be derived as follows:

$$\frac{\partial L_{\text{ran}}}{\partial H_{in}} = \frac{\partial L_{\text{ran}}}{\partial \hat{H}_{in}} \times \frac{\partial \hat{H}_{in}}{\partial H_{in}} = \frac{\partial L_{\text{ran}}}{\partial f(H_{in})} \times W$$  \hspace{1cm} (6)

Where $W$ is the filter weight. Through the loss gradient and equation (3) above, a filter combination with high robustness to scale transformation can be obtained, and the intersection of the filter combination with the above highly discriminative filter combination is the deep optimization feature.

### 2.2. Target tracking based on Discriminative Correlation Filter framework

This paper adopts the DCF framework that is improved based on MOSSE for efficient tracking. The deep optimization feature $x$ is input into the DCF. DCF uses the ridge regression in kernel space to find the kernel correlation function as follows:

$$K^{xx} = \sum_{d} \hat{x}_d e^{\hat{x}_d}$$

$$\hat{K}^{xx} = \sum_{d} \hat{x}_d e^{\hat{x}_d}$$  \hspace{1cm} (7)

where $e^{\hat{x}}$ represents the Fourier transform of the corresponding element, and $d$ represents the number of channels included in the feature. The ridge regression can be simplified through the Fourier transform and the diagonalization properties of the cyclic matrix, so as to obtain the detector $\hat{\alpha}$. The formula is as follows:

$$\hat{\alpha} = \frac{\hat{y}}{\hat{K}^{xx} + \lambda}$$  \hspace{1cm} (8)

Where $\gamma$ represents a Gaussian label, and $\lambda$ is a regularization parameter that controls overfitting. The response function $f(z) = F^{-1}(\hat{\alpha} e^{\hat{K}^{xx}})$ is obtained. $F^{-1}$ represents the inverse Fourier transform. We find the maximum value of the response function, that is, the center position of the target, to determine the offset of the target from frame $t+1$ with respect to the $t$ frame, and update the position coordinates of the target. At the same time, the template needs to be updated continuously during the tracking process, the formula is as follows:

$$\alpha_i = (1 - \eta)\alpha_{i-1} + \eta \alpha_i$$

$$x_i^d = (1 - \eta)x_{i-1}^d + \eta x_i^d$$  \hspace{1cm} (9)

Where $t$ is the frame index and $\eta$ is the learning rate.

### 3. Implementation details

The implementation details of the proposed algorithm are as follows:

1. We use VGGNet16 [10] trained on the large data set imagenet[11] to extract the activation output of the Conv4-3 layer and the Conv3-1 layer as the basic convolution features, set the size of the
convolution features of each layer to \( \frac{M}{4} \times \frac{N}{4} \) (M \times N is the size of search window). And use the cosine window to weight the extracted convolution features to eliminate boundary discontinuities.

2) We construct a deep optimization feature extractor in the first frame. First of all, we select the first 300 important filters from the Conv4-3 layer and use the gradient of the ridge regression loss to extract the filter combination with high distinguishability for the target. And then we select the top 100 important filters from the Conv3-1 layer and use the ridge regression loss and ranking loss to select the filter combination that is robust to scale transformation. Finally, we combine these two filter combinations with a certain weight, that is, the deep optimization feature extractor, obtain the deep optimization features by which.

3) We first transfer the extracted depth-optimized features into the DCF framework, keeping the parameters of the training related filters unchanged, and construct a detector (where the regularization parameter \( \lambda \) of equation (8) is \( 10^{-4} \)). Then we obtain the maximum response position through the correlation Operation between the detector and the cyclically shifted samples, and update the target image for the next frame. Finally, we use the learning rate \( \eta = 0.01 \) to continuously update the classifier to achieve complete tracking.

4. Experimental results

The tracker proposed in this paper experiments with the convolutional neural network toolbox matconvnet [12]. The algorithm is programmed using MatlabR2019a. The test hardware environment is Intel i7-9750 2.60 GHz CPU. The computer memory is configured with 16.00GB of RAM.

To evaluate the proposed tracker, it is compared with other related trackers (DCF, STAPLE_CA [13] and HCF). We select 5 representative videos from the benchmark dataset [14, 15]: Freeman3, Trans, Bird1, Jump and Rubik to evaluate the proposed method.

4.1. Qualitative analysis

Figure 1 shows the Tracking Results of proposed tracker and other related trackers (DCF, HCF, STAPLE_CA) in 5 challenging video sequences (from top to bottom: Freeman3, Trans, Bird1, Jump and Rubik). The following analysis is performed on the tracking result of Figure 1:

(1) The DCF tracker uses the hog feature for tracking, and it only performs well in sequences (Rubik) with scale changes and rotation challenges. However, when the target object undergoes such large scale transformations, fast motion, motion blur, and occlusion (Bird1, Jump), the tracking fails.

(2) HCF tracker uses offline training CNN model to extract convolution features to achieve tracking. It performs well in sequences (Rubik) with scale changes and rotation challenges, but when
the target object suffers from severe occlusion, motion blur, deformation, lighting changes, and fast motion (Freeman3, Trans, Bird1, Jump), tracker drifts.

(3) The STAPLE_CA tracker incorporates context awareness approach based on correlation filtering approach. It works well in sequences (Rubik) with challenges of scale changes and rotation changes, but it does not work well when video sequences (Freeman3, Bird1, Jump) with challenges of fast Motion, occlusion, motion blur, and large-scale deformation.

(4) The tracker proposed in this paper adds a feature optimization function on the basis of HCF, effectively improving the problem that HCF has a poor performance, When the target object suffers from severe occlusion, motion blur, deformation, lighting changes, and fast motion (Freeman3, Trans, Bird1, Jump). Compared with the other three trackers, the tracker proposed in this paper shows better results in 5 sequences with multiple video challenges.

4.2. Quantitative analysis
The tracking accuracy of the proposed tracker and the other three related trackers (DCF, HCF, STAPLE_CA) is evaluated in 5 video sequences respectively. Figure 2 shows that the proposed tracker shows higher accuracy and better tracking results than other trackers in 5 video sequences with multiple challenges. Among them, distance precision DP (20px) represents the percentage of the image with the center location error (CLE) less than the threshold 20 pixels in the entire video sequence.

![Figure 2. DP plots and DP(20px) of 4 trackers](image_url)

Based on the GPU environment driven by GeForce GTX 1660Ti, the average frame rate (FPS) of the algorithm in this paper has reached 8.26 frames per second, which can basically guarantee the
fluency and rapidity of the video tracking process.

5. Conclusion
This article first extracts the basic convolutional features through the offline training network VGGNet16, and then extracts the deep optimized features based on the basic convolutional features, effectively reducing the feature dimensions and eliminating the interference of redundant features, and improving the problem that the offline training depth model to distinguish target objects Aspects of poor results. Finally, the deep optimization feature is integrated with the DCF tracking framework, which improves the utilization rate of the deep optimization feature and achieves effective tracking. Experimental results show that our algorithm performs well in terms of accuracy and robustness.

Acknowledgment
This work is supported by National Natural Science Foundation (NNSF) of China under Grant 61702247; This work is supported by Key Project of the 13th Five-Year Plan for Economic Science in Liaoning Province under Grant JG18DA013;

References
[1] Bolme D S, Beveridge J R, Draper B A and Lui Y M. Visual object tracking using adaptive correlation filters[C]/Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition. Washington D.C., USA: IEEE Press, 2010: 2544-2550.
[2] Henriques J F, Caseiro R, Martins P, et al. Highspeed tracking with kernelized correlation filters[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(3): 583-596.
[3] Li X, Ma C, Wu B Y, et al. Target-aware deep tracking[J]. International Conference on Computer Vision and Pattern Recognition, 2019.
[4] Ma C, Huang J B, Yang X K, et al. Hierarchical convolutional features for visual tracking[C]/Proceedings of 2015 IEEE International Conference on Computer Vision. Santiago, Chile: IEEE Press, 2015: 3074-3082.
[5] Guo X W, Liu S X, Zhou M C and Tian G D. Disassembly sequence optimization for large-scale products with multiresource constraints using scatter search and petri nets[J]. IEEE Transactions on Cybernetics, 2016, 46(11): 2435-2246.
[6] Guo X W, Liu S X, Zhou M C and Tian G D. Dual-objective program and scatter search for the optimization of disassembly sequences subject to multiresource constraints[J]. IEEE Transactions on Automation Science and Engineering, 2018, 15(3): 1091-1103.
[7] Guo X W, Zhou M C, Liu S X, et al. Lexicographic multiobjective scatter search for the optimization of sequence-dependent selective disassembly subject to multiresource constraints[J]. IEEE Transactions on Cybernetics, 2019.
[8] Zhao Z Y, Liu S X, Zhou M C, Guo X W and Qi L. Decomposition method for new single-machine scheduling problems from steel production systems[J]. IEEE Transactions on Automation Science and Engineering, 2019.
[9] Li Y C, Song Y L and Luo J B. Improving pairwise ranking for multi-label image classification[C]/International Conference on Computer Vision and Pattern Recognition, 2017.
[10] Simonyan K and Zisserman A. Very deep convolutional networks for large-scale image recognition[EB/OL]. http://arxiv.org/abs/1409.1556.
[11] Deng J, Dong W, Socher R, et al. ImageNet: a largescale hierarchical image database[C]/Proceedings of the IEEE Computer Vision and Pattern Recognition. Miami, USA: IEEE Press, 2009:248-255.
[12] Vedaldi A and Lenc K. MatConvNet: convolutional neural networks for MATLAB[C]/Proceedings of the 23rd ACM International Conference on Multimedia. Brisbane, Australia: ACM, 2015: 689-692.
[13] Mueller M, Smith N and Ghanem B. Context-aware correlation filter tracking[C]//Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, USA: IEEE Press, 2017: 1387-1395.

[14] Wu Y, Lim J and Yang M H. Online object tracking: a benchmark[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. IEEE Press, 2013: 2411-2418.

[15] Wu Y, Lim J and Yang M H. Object tracking benchmark[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1834-1848.