Improving the Performance of Target Tracker by Deepening Neural Network

Yifeng Zhao¹, Weimin Lang¹, Xiangyu Peng¹ and Bin Li²

¹School of information and communication, National University of Defense Technology, Wuhan, China
²National Key Laboratory of National Defense Technology for Integrated Ship Power Technology, Naval University of Engineering, Wuhan, China

*Corresponding author e-mail: 1451004992@qq.com

Abstract. Visual target tracking is an important topic in the field of vision research. Its main task is to automatically process and analyze the input video information by computer, so as to obtain the position and key motion information of the target in the input video sequence, and to facilitate the subsequent research and analysis of the track and behavior of the tracking target. At present, researchers in the field of visual target tracking have proposed a series of excellent algorithms and frameworks. However, in view of the problems of deformation, occlusion, illumination change and random motion in the process of target tracking, it is still a great challenge to ensure the real-time, accuracy and robustness of tracking algorithms. In recent years, with the excellent performance of depth neural network in many tasks, the fusion of depth features based on relevant filtering algorithms to achieve target tracking tasks has become a hot research topic. In this paper, an improved target tracking algorithm is proposed based on the framework of classical target tracking methods: (1) Integrating deeper ResNet-101 deep convolution neural network model to detect target features more accurately. (2) Improve the model training method to save the network training cost. Finally, the performance of the proposed tracking algorithm is tested on the OTB target tracking standard dataset, and compared with other mainstream algorithms. The results show that the improved method proposed in this paper can effectively improve the tracking effectiveness and robustness.

1. Introduction

The existing tracking methods are mainly based on the appearance of the target to separate the foreground and background of the target. One of the major drawbacks of this method is that the tracker relies on low-level manual features, but manual features can not effectively express the semantic information of the target. Unlike handmade features, convolutional neural networks can acquire rich high-level semantic features from a large number of labeled object classes and image data. Moreover, it has innate advantages in judging the classification of targets. In the face of different dataset, the generalization performance of convolution network is also very strong, and the robustness of network is also very high when the appearance of the target changes [1]. Therefore, the features learned by convolutional neural network have potential application value in target tracking task.

However, after the introduction of the depth of the high dimension characteristic, target appearance model parameters are also a large increase, training a large number of parameters is bound to increase

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd
the computational overhead tracking algorithm, affect the tracking speed of tracking[2]. Therefore, while improving the tracking accuracy of the network, we also need to optimize the network to reduce the time overhead of the tracker to improve the tracking speed of the tracker. Related work.

2. Related work

Visual target tracking technology is put forward in the 1950s. In 1955, Wax first proposed the basic concept of multi-target tracking in his theoretical research, and then from 1960 to 1964, Kalman filtering algorithm and bayesian target trajectory correlation method were proposed successively to further lay the foundation of target tracking technology[3].

In the research of integrating depth features into target tracking task, Zhang et al. extracted a large number of video samples by online training to train bilevel convolutional neural network, and then realized target tracking by particle filter framework. On the basis of correlation filtering and incorporating depth features, Martin et al. proposed a c-cot target tracking algorithm, which used vgg-net to extract target features, and then transformed feature graphs of discrete domain into continuous domain through cubic interpolation. Finally, Hessian matrix is applied to locate the target position with sub-pixel precision. On the basis of c-cot, Martin further improved the ECO target tracking algorithm, which includes the use of factorization convolution operation to reduce the complexity of neural network model and save time, screen more representative correlation filters to prevent overfitting, and select more diverse samples to reduce redundancy[4, 5].

3. Depth feature selection

3.1. Target tracking task feature analysis

In Fig. 1, we simply show the target characteristics of different layers extracted based on imagenet-vgg-m-2048 convolutional neural network commonly used in current tracking tasks.

\[\text{Figure 1. Output characteristics of different convolution layers}\]

From left to right, the original input image, the target features of the shallow convolution layer and the target features of the deep convolution layer are in turn. As we can see from the picture, The deep convolution layer near the right extracts higher-level and more abstract semantic features of the target, which can make the system better classify different targets, and the changes of deformation and occlusion of the target will not significantly affect the target features learned. The shallow convolution layer near the left learns more image features such as contour and line of the target, which makes it easier to distinguish the target from other interference items of the same kind, but the shallow convolution layer features can not effectively respond to the appearance changes of the target. In order to effectively use the target features learned by convolution neural network, we need to selectively select the target features learned by different convolution layers and filter out the noise and task-independent features that may affect the accuracy of the target.

3.2. Convolutional Layer Selection

The convolution network model used in this paper is ResNet-101 network model. The network consists of five parts: conv1, res2x, res3x, res4x and res5x. When we select the output features of the network model, we mainly select them on the output feature map of these five parts. But we are not sure which level of feature output we need to use. Considering that in different blocks, the feature difference of the target extracted by the adjacent layer is very small, and if the target feature extracted by the adjacent layer is selected for tracking tasks, it will also produce feature redundancy and
unnecessary interference, so we output the feature map of convolution layer in the last layer of conv1, res2x, res3x, res4x and res5x blocks to judge.

Like most other methods, we retain the output of the first convolution layer [6], the conv1 layer, as one of the features. For the output of res2c layer, res3b3 layer, res4b22 layer and res5c layer, we need to further verify their functions. Because of the redundancy of features, we do not need to apply all features extracted from res2c, res3b3, res4b22 and res5c layers to target tracking tasks. Therefore, we further select several combinations of convolution layers to verify the performance of the feature mapping of the output of the convolution layer. Based on the above characteristics, we selected four convolutional layer assemblages \( \{C, C_1, C_2, C_3\} \). The convolution layer combination represented by \( C_1 \) is \{conv1, res2c, res5c\}, The convolution layer combination represented by \( C_2 \) is \{conv1, res3b3, res4b22\}, The convolution layer combination represented by \( C_3 \) is \{conv1, res3b3, res4b22\}, and The convolution layer combination represented by \( C_4 \) is \{conv1, res3b3, res5c\}. We simply validate the performance of these four combinations in the tracker. The test results are shown in Fig. 2.

![Figure 2. Average overlap rate of different combinations](image)

Through the above experimental analysis, we select the output target feature mapping of Conv1 layer, res3b3 layer and res5c layer in ResNet-101 network model as the feature input of target tracking task.

4. Model optimization

4.1. Sample collection

Sample collection is a key step to update the model online in target tracking algorithm. The current method of sample data acquisition is to collect sample data once per frame of target video sequence. In view of the problems existing in the current sample collection strategy [7], in this paper, we propose a method to judge whether to collect the image of the current frame as the sample image according to the target similarity of adjacent frames. The key point of this method is to compute the similarity of the target region of adjacent frames as a criterion for judging whether the current frame is taken as the sampling object. We introduce the perceptual hash algorithm to calculate the similarity between adjacent target images. Generally speaking, we can see a picture as a two-dimensional signal, this signal contains different frequency characteristics, Cosine transform in discrete domain is used to obtain the low-frequency features of input objects to calculate the similarity between images. Cosine
transform in discrete domain can convert the pixel values of images into frequency domain for calculation. The specific expression is as follows:

\[ F(y) = c(y) \sum_{x=0}^{N-1} f(x) \cos \left( \frac{(x + 0.5)\pi}{N} y \right) \]  

(1)

\[ c(y) = \begin{cases} 
\frac{1}{N}, & y = 0 \\
\frac{2}{N}, & y \neq 0 
\end{cases} \]  

(2)

Where \( f(x) \) is the original input signal, \( F(y) \) is the frequency coefficient after the discrete domain cosine transform, \( N \) is the number of points of the input signal, \( c(y) \) is the compensation function. Because the image usually has a large amount of redundant information, only a small part of the coefficients of the frequency components are not zero after the image is transformed by the cosine transform in the discrete domain. The specific steps of using perceptual hashing algorithm to calculate image similarity are as follows: Firstly, the size of the target image is reduced to \( 8 \times 8 \) size, Secondly, the color image is changed into grey image, Thirdly, the target image is converted by cosine transform in discrete domain, and the frequency coefficient matrix of the image is obtained, Fourthly, according to the position of the energy concentration point of the image analyzed above, the \( 32 \times 32 \) coefficient matrix is reduced, and the low-frequency component of the image is extracted by retaining only the coefficient matrix part whose upper left corner size is \( 8 \times 8 \), Then, on the basis of the coefficient matrix of \( 8 \times 8 \) extracted, the average \( M \) of all 64 elements of the matrix is calculated. For the elements \( x_{ij} \) in the matrix, the following mapping operations are adopted:

\[ y_{ij} = \begin{cases} 
1, & x_{ij} \geq M \\
0, & x_{ij} < M 
\end{cases} \]  

(3)

That is to say, for elements larger than or equal to the average value in the matrix, we replace them with 1 and the rest with 0. Then we combine all the values in a certain order to form a 64-bit hash value. When measuring the similarity between two adjacent frames, we calculate the hash values of two target images, and then compare the Hamming distance \( d \) between the two hash values. Let's assume that \( \lambda \) is a similarity parameter of two images, and the parameter values are expressed by Hamming distance. From this, we can get that the value space of similarity parameter \( \lambda \) is \([0, 64]\). The higher the value of similarity parameter is, the higher the correlation between the two images is. In addition, we give a similarity threshold \( \tau \), through which we can decide whether to collect the image of the current frame as a sample image. In this paper, we set the value of the threshold parameter to be 10. When the similarity parameter of two images is greater than the threshold value, the image of the current frame is sampled as the sample image. Through such a sampling strategy, images with large sample differences can be effectively selected to update the model, so as to avoid large Numbers of samples with high similarity squeezing the memory space and increasing the computational overhead of the network.

4.2. Model updating

An important part associated with sample collection is model update. After collecting effective sample data, the parameters of the tracker model need to be updated in time, so that the tracker can effectively respond to the appearance change of the target. In the previous section, we used the method of sampling image according to certain rules in the sampling strategy of sample dataset. In the model and new strategy, we are also ready to change to a sparse model updating method. That is to say, we set an update parameter \( G_N \), where \( N \) represents the \( N \) frame. When the target image of the \( N \) frame in the sample sampling strategy is selected
for sampling, the value of $G_N$ is 1, otherwise it is 0. $G_N=1$ means updating various parameters in the model at the beginning of the current frame optimization process, and $G_N=0$ means that the current frame does not update the model.

5. Experimental analysis

5.1. Experimental Dataset

In this paper, we use OTB2015 dataset, which contains 100 video frame sequences, one quarter of which are gray image sequences. In video tracking task, OTB starts with random frames of video sequence or by applying random interference to the tracking frame. If the target drift occurs in the tracking process, i.e. the tracking failure, the data set will directly determine the target loss and will not re-track the target.

5.2. Evaluation criteria

In this paper, we mainly use two evaluation indicators to analyze the performance of the algorithm, one is tracking accuracy, the other is tracking success rate. Tracking accuracy is usually calculated by the Euclidean distance between the central position point of the target and the actual central position point of the target calculated by the tracking algorithm. Usually, a Euclidean distance threshold is set, and then the proportion of the number of video frames less than the threshold in the complete video sequence to the total number of video frames is calculated. This ratio is the tracking accuracy. We set the threshold to 20 pixels. The success rate of tracking is usually calculated by the overlap area between the target frame of tracking algorithm and the actual target frame. Usually, a threshold of overlap rate is set. Then, the ratio of the number of video frames larger than the threshold to the total number of video frames in the complete video sequence is calculated. That is, the success rate of tracking. We set the threshold to 50%.

5.3. Experimental results

In general, the accuracy and success rate of the proposed method are significantly improved. The precision of the proposed method is 0.87, and the success rate is 0.814.

6. Conclusion

On the basis of existing tracking algorithms, we further mine the improvement of tracking performance by depth features. The convolution neural network used in target tracking algorithm can be up to 50 layers. On this basis, we further deepen the network depth, introduce 101 layers of convolution neural network, and through experimental analysis, extract the appropriate feature mapping of convolution layer for target tracking. At the same time, as the number of network layers deepens, the parameters of the model will increase greatly, and the dimension of feature mapping will
also increase. The impact is that the computational overhead of the tracker will increase dramatically, and the speed of tracking the target will be serious. In view of this, we improve the strategy of sample sampling and model updating to reduce the number of samples and the frequency of model updating by effectively controlling the number of samples and the frequency of model updating. The computational overhead of the tracker is proved by experiments. The proposed method improves the tracking accuracy, robustness and tracking speed in target tracking tasks.

7. Acknowledgments

This paper was supported by National University of Defense Technology Project Research on Edge Computing Data Deduplication Mechanism Supporting Ownership Dynamic Management and Privacy Protection” (No. ZK18-03-2).

References

[1] H. Grabner, M. Grabner, and H. Bischof. Real-Time Tracking via On-line Boosting, In BMVC, 2006.
[2] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding, In ACM Multimedia, 2014 , pp. 675-678.
[3] N. Wang, D. Yeung. Learning a deep compact image representation for visual tracking, Advances in Neural Processing Systems (NIPS), 2013, pp. 809-817.
[4] M. Danelljan, G. Hager, F. Shahbaz Khan, and 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.
[5] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg. Eco: efficient convolution operators for tracking. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 21–26.
[6] L. Wang, W. Ouyang, X. Wang, Visual Tracking with Fully Convolutional Networks, IEEE International Conference on Computer Vision, 2016.
[7] J. Z. Y. D. H. B. hiqun He, Yingruo Fan, Correlation filters with weighted convolution responses, IEEE International Conference on Computer Vision, 2017.