Target tracking based on neural network depth feature and texture fusion

Yongzhan Cao¹, Meiju Liu¹, Shanykui Yang², Guodong Yang³, Peng Chen², Shuyun Zhu², Zhuang Ge², and Yuwang Liu²*

¹ Information and Control Engineering Faculty, Shenyang Jianzhu University, Shenyang, Liaoning 110168, China;
² State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China
³ School of Mechanical Engineering and Automation, Northeastern University 3 Wenhua Street, Shenyang 110819, China

*Corresponding author’s e-mail: liuyuwang@sia.cn

Abstract. This paper presents a method of target tracking based on convolution neural network and texture feature fusion. The lower layer of the convolutional neural network can extract some spatial structure, shape and other features of the target. High-level level can extract relatively abstract semantic information. In this paper, vgg-m convolutional neural network is adopted to realize tracking by adaptive fusion of the extracted depth features of Conv2 and Conv5 with the texture features extracted by two-dimensional Gabor filtering. In this paper, the experimental analysis of this method is carried out on the OTB2013 data set, and the results show that this method can achieve more accurate positioning of the target and has a strong timeliness.

1. Introduction

Since the publication of major research work on deep trust networks by Hinton et al in 2006, deep learning has become a new direction for machine learning, and has played an important role in many important issues in the field of artificial intelligence. So far, several deep learning frameworks, such as deep neural networks, convolutional neural networks, deep belief networks, and recurrent neural networks, have been successfully applied in computer vision, speech recognition, natural language processing, audio recognition, and bioinformatics. Excellent result. Target tracking algorithms based on deep learning have become an indispensable part of the target tracking field.

Although target tracking has been applied in many aspects, it still faces more challenges [1][2][3]. The video sequence is affected by illumination changes, occlusion, motion blur, background noise, and low-resolution factors.

Therefore, an ideal target tracking algorithm can not only cope with the influence of various factors, but also improve the tracking accuracy, and can track the time more quickly and in real time, so as to achieve real-time and accurate tracking of the target.

In this paper, the depth features extracted from the VGG-M neural network and the Gabor texture feature are regarded as two independent tracking cues. The adaptive fusion strategy of feature uncertainty is used to achieve accurate tracking of the target.

With the advent of deep CNN networks, the fully connected layer of the network has been commonly used for image representation [3][4]. The information displayed by the deep convolution layer is more...
2019 5th International Conference on Energy Equipment Science and Engineering
IOP Conf. Series: Earth and Environmental Science 461 (2020) 012021
doi:10.1088/1755-1315/461/1/012021

conducive to image classification [5][6]. On the other hand, the shallow convolution layer display is
more suitable for visual tracking than the deeper layer [7]. The deep convolution layer is discriminative
and has advanced visual information. In contrast, shallow layers contain low-level features with high
spatial resolution that facilitate positioning. Therefore, the scheme adopted in this paper is to extract the
depth features and texture features on the VGG-M convolutional neural networks Conv2 and Conv5 to
achieve the tracking purpose.

2. Texture feature expression

The Gabor transform is a kind of windowed short-time Fourier transform, and its window function is a
Gaussian kernel function. However, unlike the Fourier transform, the Gabor transform has good locality
in both the frequency domain and the spatial domain. It has been widely used in face recognition, edge
detection, texture segmentation, image representation and image compression.

The two-dimensional Gabor transform is a powerful tool for multi-scale analysis and representation
of images. The function of a two-dimensional Gabor filter can be expressed as:

\[ Q_{\mu,v}(x) = \frac{\|Q_{\mu,v}\|^2}{\sigma^2} \exp(-\frac{\|Q_{\mu,v}\|^2}{2\sigma^2})(\exp(iQ_{\mu,v}x) - \exp(-\frac{\sigma^2}{2})) \] (1)

\[ Q_{\mu,v} = \begin{pmatrix} Q_x \\ Q_y \end{pmatrix} = \begin{pmatrix} Q_x^\cos \phi_x \\ Q_y^\sin \phi_y \end{pmatrix} \] (2)

\( x \) is the spatial position coordinate, \( Q_x \) is the center frequency of the filter, \( \phi_x \) reflects the direction
selectivity of the filter, \( \frac{\|Q_{\mu,v}\|^2}{\sigma^2} \) is used to compensate the energy spectrum attenuation determined by
the frequency, \( \exp(-\frac{\|Q_{\mu,v}\|^2}{2\sigma^2}) \) is the Gaussian envelope function of the constrained plane wave,
\( \exp(iQ_{\mu,v}x) \) is The complex-valued plane wave has a real part of cosine plane wave \( \cos(Q_{\mu,v}x) \) and
an imaginary part of sinusoidal plane wave \( \sin(Q_{\mu,v}x) \). In order to eliminate the influence of the DC
component of the image on the two-dimensional Gabor transform, the real part of the complex-valued
plane wave is subtracted by \( \exp(-\frac{\sigma^2}{2}) \), which can avoid the influence of the absolute value of the
image grayscale on the Gabor transform, so that the Gabor transform does not change the illumination of the image. sensitive. The expressions for the real and imaginary parts of a two-dimensional Gabor transform are:

\[ \text{Re}(Q_{\mu,v}(x)) = \frac{\|Q_{\mu,v}\|^2}{\sigma^2} \exp\left(-\frac{\|Q_{\mu,v}\|^2}{2\sigma^2}\right) \left[\cos(Q_{\mu,v}x) - \exp\left(-\frac{\sigma^2}{2}\right)\right] \] (3)

\[ \text{Im}(Q_{\mu,v}(x)) = \frac{\|Q_{\mu,v}\|^2}{\sigma^2} \exp\left(-\frac{\|Q_{\mu,v}\|^2}{2\sigma^2}\right) \sin(Q_{\mu,v}x) \] (4)

The Gabor transform of the image is actually the gray feature of the region near the given point \( I(x) \)
on the image. The transformation process can be realized by convolving the given point \( I(x) \) with the
Gabor function on the image.
3. Feature fusion strategy
In order to be able to fuse texture features with depth features, we need to calculate the similarity between the two. The Bhattacharyya coefficient is used as a method of feature similarity measure [11]. The specific definition is as follows:

\[
d = \sqrt{1 - \rho[P, Q]}
\]  

(5)

Where \( P = \{p_u\}_{u=1,2,3,..,N} \) stands for candidate region eigenvalues, \( Q = \{q_u\}_{u=1,2,3,..,N} \) represents the goal template.

\[
\rho[P, Q] = \sum_{u=1}^{N} \sqrt{p_u q_u}
\]  

(6)

In the equation, the smaller \( d \) is, the larger \( p \) is, and the similar the candidate region and the target template are.

In order to improve the discriminative ability of the post-fusion likelihood function and overcome the influence of noise, we adopt adaptive fusion of multi-feature based on feature uncertainty metric [12]. The characteristic uncertainty expression is:

\[
T_{i+1}^v = \text{var}(p_i)
\]  

(7)

In the formula, the uncertainty of the feature at time \( t+1 \) is the positional variance of the particle at time \( t \), and the larger the positional variance, the more dispersed the particle. It is the entropy of the observed probability values of all particles under the \( i \)-th feature at time \( t \), indicating the degree of dispersion of the observed probability values on the particles, and entropy is a measure of uncertainty:

\[
K(p^i) = -\sum_{j=1}^{N} p(z^i | x_j) \log_2 p(z^i | x_j)
\]  

(8)

\[
p(z^1, z^2, z^3, \ldots, z^n | x) = \prod_{i=1}^{n} \left( \frac{p(z^i | x) + T^1 T(x)}{1 + T^1} \right)
\]  

(9)

The fusion rule when \( n \) is 2 is:

\[
p(z^1, z^2 | x) = \frac{1}{(1 + T^1)(1 + T^2)} \times
\left( p(z^1 | x) p(z^2 | x) + T^1 T(x) p(z^1 | x) + T^2 T(x) p(z^2 | x) + T^1 T^2(T(X))^2 \right)
\]  

(10)

In the formula, the uncertainty corresponding to the depth feature is represented \( T^1 \), \( T^2 \) indicating that the texture feature corresponds to the uncertainty. \( T_{i+1}^v = \frac{1}{N} \), \( N \) for the number of particles. The similarity between the candidate model and the target model can be represented by the observed probability value.

4. Algorithm implementation
In this paper, the target is tracked under the particle filter framework. The specific flow of the algorithm is as follows:

Step 1 Manually select the tracking target in the initial frame, and take the target frame as a positive sample, select a series of negative samples around it and send it to the VGG-M network.
Step 2 Obtain the texture features of the target model from the two-dimensional Gabor transform. And take N initial particles \( \{x'_i\}_{i=1}^N \) from the prior distribution, and the weights are set to \( \{\omega_{0,i}\}_{i=1}^N = \frac{1}{N} \).

Step 3 by \( x_t = Ax_{t-1} + N_t \). The predicted state of the current frame \( x_t \) is obtained based on the state of the previous frame \( x_{t-1} \).

Step 4 Calculate the observation probability of the depth feature and the texture feature and normalize the two. And obtain the entropy of the observation probability of these two features.

Step 5 Calculate the variance from the position coordinates of the particle space, and calculate the uncertainty of the texture feature and the uncertainty of the depth feature.

Step 6 Calculate the likelihood function of the feature fusion \( p(z^1, z^2 \mid x) \), and update the particle weight of the current frame, and then perform normalization.

Step 7 Determine whether sampling is required by the distribution of particle weights. Re-extract N particles, and set the weight of each particle to \( \frac{1}{N} \), otherwise do not process.

Step 8 Go to step 3 to track the next frame.

5. Experimental results and analysis
In order to test the performance of the tracking algorithm proposed in this paper, we compare the algorithm of this paper with the current five mainstream target tracking algorithms on the OTB2013 dataset. The evaluation indicators selected in this paper are the correct rate and success rate.

(1) correct rate
The specific calculation formula for the correct rate is as follows:

\[
d = \sqrt{(x_p - x_s)^2 + (y_p - y_s)^2}
\]  

(11)

Where, and are the coordinate positions of the tracking result and the real value in the original image, respectively. In the experiments in this paper we set the threshold to 20 pixels.

(2) Success rate
The specific calculation formula for the success rate is as follows:

\[
S = \frac{\text{Area} \left( D_T \cap D_G \right)}{D_T \cup D_G}
\]  

(12)

Represents a rectangular box that tracks the results, representing a rectangular box of true values. When the coincidence rate between the rectangular box of the tracking result and the rectangular box of the true value is greater than a predetermined threshold, we consider that the current video frame is successfully tracked, otherwise the tracking fails. In this experiment we set the threshold to 0.5.

Figure 1 and Figure 2 show the overall success rate and accuracy of the 51 video sequences in the OTB2013 data set, respectively, of the algorithm and five tracking comparison algorithms. As can be seen from the figure, the method proposed in this paper has achieved good results in tracking accuracy and accuracy. Significantly better than the other four tracking methods.
Five complex and challenging video sequences, Singer2, Trellis, Soccer, Car1 and Blueface, were selected for testing on the OTB2013 dataset and compared with the current five mainstream tracking algorithms. Figure 3 is the tracking result of some video frames. The algorithm in this paper has achieved good tracking effect. In Blueface video, the tracking target rotates and some frames appear fast moving, but the algorithm in this paper can still implement it. Very good tracking. In summary, the algorithm of this paper has good robustness and can accurately track the target.

6. Conclusion
This paper proposes a method based on the fusion of convolutional neural networks and texture features. Tracking is achieved by adaptively combining the depth features obtained by the Conv2 and Conv5 layers in the VGG-M convolutional neural network with the texture features obtained by the two-bit Gabor filtering, which further improves the tracking accuracy. The experimental results show that the proposed method can achieve effective tracking of targets and has high tracking efficiency.
References

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

[2] Wu Y, Lim J, Yang M H. Online object tracking: A benchmark[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2013: 2411-2418.

[3] Oquab M, Bottou L, Laptev I, et al. Learning and transferring mid-level image representations using convolutional neural networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 1717-1724.

[4] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[5] Cimpoi M, Maji S, Vedaldi A. Deep filter banks for texture recognition and segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3828-3836.

[6] Liu L, Shen C, van den Hengel A. The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015: 4749-4757.

[7] Danelljan M, Hager G, Shahbaz Khan F, et al. Convolutional features for correlation filter based visual tracking[C]//Proceedings of the IEEE International Conference on Computer Vision Workshops. 2015: 58-66.

[8] Lee T S. Image representation using 2D Gabor wavelets[J]. IEEE Transactions on pattern analysis and machine intelligence, 1996, 18(10): 959-971.

[9] Mehrotra R, Namuduri K R, Ranganathan N. Gabor filter-based edge detection[J]. Pattern recognition, 1992, 25(12): 1479-1494.

[10] Daugman J G. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters[J]. JOSA A, 1985, 2(7): 1160-1169.

[11] Nummiaro K, Koller-Meier E, Van Gool L. An adaptive color-based particle filter[J]. Image and vision computing, 2003, 21(1): 99-110.