Real-time High-speed Visual Tracking based on Deep Convolutional Neural Network

Rui Li [1] and Jirong Lian[2 *]

Lanzhou University of Technology, Lanzhou, Gansu Province, 730050, China

Email: 649094574@qq.com

[*]corresponding author

Keywords: Convolutional neural network; Visual tracking; Deep learning; Machine vision

Abstract: Aiming at the problem of real-time high-speed visual tracking and how to improve the success rate and accuracy of visual tracking without significantly increasing the computing performance requirements is the key to solving such problems. Also aiming at the problem that historical information of traditional visual tracking algorithms is easy to lose; this paper proposes an algorithm that uses tree structure and multi-convolutional neural network to jointly estimate the target state and dynamically updates the model. After experiments, the performance and efficiency of the algorithm are higher than traditional algorithms, and it can help to deal with high-speed visual tracking problems.

1. Introduction

As an important work and application field of machine vision, visual tracking has occupied a large proportion in major domestic and foreign journals and conferences in recent years. The essence of visual tracking is to perform target recognition on a series of ordered images, and calculate the motion parameters of the target to achieve tracking and prediction of moving objects. Traditional vision tracking algorithms can be roughly divided into generative and discriminative [1]. Among the representative algorithms of generative methods are kernel methods, sparse representations, and Gaussian mixture models. The discriminative algorithm starts with the idea of a classifier and uses support vector machines and random forest methods to separate the target and the background to achieve target tracking. However, the performance and performance of traditional visual tracking algorithms in non-ideal environments are not satisfactory. Factors such as occlusion, shadow, light, and attitude will affect target recognition and tracking results. On this basis, the academic community put forward the idea of combining deep learning and visual tracking, and CNNs have emerged in related research and applications [2].

Some literatures [3] combined the image network and CNN, and trained the tracking algorithm based on the deep convolutional neural network through the image network, which further improved the tracking accuracy. However, in practical applications, the detection of moving targets usually encounters problems such as the temporary loss of moving targets or the difficulty of tracking the target caused by the background, which directly leads to the interruption of target tracking. If the historical features of the target feature and state can be saved in the algorithm, and reference to the historical state of the target model in subsequent tracking will help solve this problem.

2. Online Visual Tracking based on Improved CNN

In high-speed visual tracking, the core problem is that high-speed moving objects require high frame processing speed, and need to deal with a series of image problems such as occlusion, distortion, high-speed blur, and low resolution. This paper explores an improved CNN-based online visual tracking method that can actively deal with such situations due to occlusion, distortion, high-speed blur, or temporary loss of targets, and can derive more models at a lower cost [5].

2.1 Structure of CNN

J Deng, W Dong, R Socher and others proposed a VGG-M network in the Image Net document. The structure described in this paper draws on the convolutional layer of this network [4]. As shown in Figure 1, the network consists of three convolutional layers and three fully connected layers. Conv3 uses two spaces for binary classification. The other two fully connected layers are both 512 in size. The input of the network is a 75x75 RGB image, the size is equal to the size of the single channel corresponding to the last conv3.

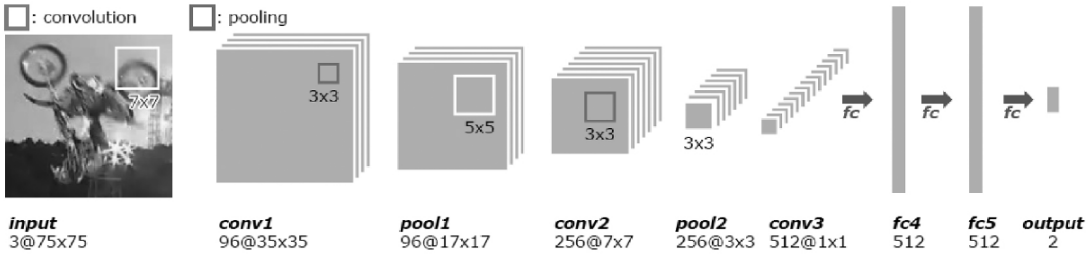

Figure 1 CNN structure

2.2 Management structure of vertical multi-objective appearance model

When managing the vertical multi-objective appearance model, in order to improve the maintenance performance, a tree structure is used to record and maintain it definition of $T=\{v,\varepsilon\}$ Is the tree structure, each of which corresponds to a CNN v In order to further express the association between the various networks in the book, this paper uses directed edges $\{u,v\}$ Perform it and $\{u,v\}$ The score can judge the relationship between the two end points [6]. The specific relationship can be seen in the following formula: $s(u,v)=\frac{1}{|F_v|}\sum_{t\in F_v}\varphi_u(x_t^*)$

In this formula, Fv is essentially a series of space-time-sequential video frames. In this algorithm, it is mainly used for training the required CNN. The CNN is associated with v. To identify the target x_t^* to complete the job,which t is the current frame sequence. state of the specified frame, use here

Another key is that u Positive prediction score φ_u

2.3 Application of multiple CNNs in target state estimation

We built a CNNs tree earlier in order to manage and maintain the vertical multi-objective appearance model. When the algorithm estimates the current state of the target in a specific frame, it needs to be combined with the historical CNN score in the tree to complete it. In order to improve the accuracy of the target state estimation, the state of the target needs to be obtained in real time, and the slave x_t^1 to x_t^n Acquisition of multiple targets. among them $v_+ \subseteq v$ CNNs are effective. At this point, the weighted average is used to calculate the score for each target:

$$H(x_t^i) = \sum_{v_+ \subseteq v} w_{v \to t} \varphi_v(x_t^i) \cdot \tag{1}$$

There is a key parameter in the above formula: it is used to identify the network weight of the $w_{v \to t}$ Meaning that, with this formula, the scores of target vertex in the current frame. Here we use all candidate targets can be calculated. In actual judgment, the candidate target with the largest value can be considered as the real target obtained by this calculation.

So far we have roughly sorted out the idea of the algorithm and made clear how to judge the best target among many targets. Mentioned above $w_{v \to t}$ It is a key parameter in the judgment process, so what needs to be solved next is how to define the weight. There are two factors directly related to the weight: one is the reliability of the CNN itself; the other is the affinity of the currently used architecture. Therefore, in the definition of weights, these two factors need to be considered at the same time, so as to obtain a relatively objective and effective weight [7].

Let's look at framework affinity first. The so-called frame affinity actually refers to the degree of influence of the CNN corresponding to many vertices on the final tracking effect. Here $a_{v \to t}$ Identify the frame affinity, which is calculated as follows:

$$a_{v \to t} = \max_{x_t^i} \varphi_v(x_t^i) \tag{2}$$

As can be seen from the above formula, affinity is essentially the maximum positive score of all candidate targets. In this way, the impact weight can be $w_{v \to t}$ The factor of frame affinity is determined. In order to further integrate the reliability into the judgment factors, the tree structured path constructed above can be used to update the CNN, and the correlation between CNNs and the corresponding v is used as the evaluation basis. β To represent this correlation, the calculation is done recursively as shown in the following formula: $\beta_v = \min(s(p_v, v), \beta_{pv})$

The establishment of the evaluation method of the above two factors can help us to establish a mixed weight calculation method, which can convert the weight $w_{v \to t}$ Expressed as:

$$w_{v \to t} = \frac{\min(a_{v \to t}, \beta_v)}{\sum_{v \in v_+} \min(a_{v \to t}, \beta_v)} \tag{3}$$

$a_{v\to t}$ as well as β_v For illustration, both are evaluation scores of CNN. The weight Especially here

of the CNN in determining the t-th frame is determined by the small value in the selected obstacle.

2.4 Boundary Regression and Model Update

The best target determined does not necessarily correspond to a tight bounding box. The main reason for this phenomenon is the inexact positioning characteristics of CNNs [8]. Therefore, here we consider using boundary regression to improve the effect and quality of target positioning.

First, regardless of model update, use the above method for frame target tracking

$\Delta(=10)$ At this time, a special node is created for the new CNN z The reliability of the new CNN

can be further enhanced by the parent. At this time, the corresponding vertex of the parent p_z It can

be expressed as follows: $p_z = \arg\max_{v\in V+} \min(\tilde{s}(v,z), \beta_v)$

Here comes the tentative edge (v,z) then the formula is $\tilde{s}(v,z)$ This is the score of this item, which can be calculated according to formula (1). The CNN corresponding to the newly added vertex z p_z CNN fine-tuning can be obtained. At this time, the tree structure is expanded to add the new z and corresponding edges (p_z,z) putting them together while passing V_+ The active CNN represented will also be extended to several CNNs in the tree.

To achieve the target state estimation from frame to frame, construct a (x,y,s) In space, 256 samples are selected from the original independent multivariate normal distribution with the target state as the core, and these samples are placed in the space. The three standard deviations are: $\sigma_x = 0.5l$ $\sigma_y = 0.5l$ $\sigma_s = 0.5$ among them is l Take the average of the target mine height and width, and for the sample bounding box, its scale should be determined by 1.5 times the initial target length and width.

After the frame tracking training is completed, the training data needs to be collected. For each frame, at least 50 positive samples and more than 200 negative samples were plotted, greater than 0.7 IoU or less than 0.5 IoU. Because there is no deviation from the layer features in the frame and the convolution operation does not need to be run multiple times, image patches are not normally used as training examples under normal circumstances, but the features of the conv layer conv3 need to be stored.

3. Experimental Analysis

3.1 Parameter setting

Aiming at this algorithm, it is planned to apply MatCnvNet in MATLAB for experiments. The experimental equipment hardware is a computer equipped with NVDIA Titan X display processing core and equipped with i7-7700k processor. In order to evaluate the algorithm, two datasets, the

online tracking benchmark and the visual object tracking 2015 benchmark were all tested.

3.2 Experimental analysis

There are many mainstream tracking algorithms. This paper chooses eight algorithms such as HCF, FCNT, CNN-SVM, SRDCF, MUSTer, MEEM, DSST, and Struck to compare with this algorithm.

By testing a variety of mainstream tracking algorithms, in order to obtain more objective comparison results, the tracker parameters are completely consistent and fixed. The data comparison between the algorithms in this paper and these tracker algorithms is based on OTB50 in OTB100 and 50 sequences out of 100 sequences. The data can be seen in the following table (where CNN * is the algorithm):

Table 1 Comparison of experimental data

OTB50 of OTB100			
	SuccessRate		Precision
CNN*	0.682	CNN*	0.937
MUSTer	0.641	HCF	0.891
SRDCF	0.619	MUSTer	0.885
HCF	0.605	CNN-SVM	0.852
FCNT	0.597	MEEM	0.840
CNN-SVM	0.586	DSST	0.826
MEEM	0.572	FCNT	0.823
DSST	0.557	DSST	0.747
Struck	0.474	Struck	0.656
sequence 50 of 100			
	SuccessRate		Precision
CNN*	0.654	CNN*	0.884
SRDCF	0.591	HCF	0.837
MUSTer	0.575	CNN-SVM	0.814
HCF	0.562	MEEM	0.786
CNN-SVM	0.555	SRDCF	0.776
FCNT	0.547	MUSTer	0.74
MEEM	0.533	FCNT	0.772
DSST	0.517	DSST	0.689
Struck	0.458	Struck	0.638

According to the above experimental data, it can be seen that the improved algorithm proposed in this paper has better results, which is mainly due to the learning features possessed by CNN itself, that is, it is good at capturing semantic features and capturing features. Grabbing is more effective [9]. On the premise of high-speed frame processing and hardware performance, accuracy and reliability are guaranteed. At the same time, because the traditional tracker is usually based on low-level feature capture, it has poor response to various challenges, such as distortion, occlusion, deformation, high-speed motion blur, and so on. Although CNN can cope with the above-mentioned anomalies, it still has limited support for moving objects and occlusion under strong light [10]. Although this algorithm improves CNN, there are also situations where the support for poor vision is not ideal. If you consider integrating all re-tested modules in the future instead of relying solely on local candidate samples, you may be able to play a certain optimization role.

In order to further evaluate the effectiveness of this algorithm, several additional cases are discussed and compared here: a linear single CNN, a linear mean CNN, a tree structure mean CNN, and a tree structure maximum CNN combined with this algorithm. The results can be seen in the table below:

Table 2 internal comparison of algorithms

	Accuracy	Success rate	AUC
Single Linear CNN	0.885	0.854	0.652
Linear mean of CNN	0.918	0.857	0.668
Tree structure mean CNN	0.919	0.866	0.676
Largest tree structure CNN	0.920	0.869	0.668
CNN*	0.935	0.880	0.682

According to the analysis results, it can be seen that the effectiveness of using multiple models is higher than that of single models, which indicates that the diversity of models has a positive effect on the effectiveness of algorithms. The number structure mean method is better than the linear structure mean method, which shows that CNN can enhance the algorithm performance in the tree. This is because the path can be selected when updating the CNN, and the update is more efficient and reliable. Comprehensive comparison, this paper proposes that CNN * has higher advantages than other comparisons, indicating that the idea of this paper is correct. In the scenario of high-speed target tracking, this algorithm can provide higher success rate and accuracy.

Conclusion

In this paper, the traditional target tracking algorithm based on convolutional neural network is improved. The CNN is dynamically updated using a tree structure. Multiple convolutional neural networks in the tree structure work together to complete the target state recognition. At the same time, the CNN in the tree is continuously updated, so that its multi-modal processing performance for object appearance is enhanced. This is of great help for the tracking of high-speed moving objects. Experiments show that the algorithm does not significantly increase the computing performance requirements, and at the same time can support faster recognition with higher accuracy.

Acknowledgements

1. Gansu Provincial Key R & D Plan-Industrial (18YF1GA060)
2. National Natural Science Foundation of China (61761028)

References

[1] VARAS D, MARQUES F.Region-Based Particle Filter for Video Object Segmentation[C]// Proceedings of the 2014IEEE Conference on Computer Vision and Pattern Recognition (CVPR) .2014:3470-3477.

[2] BENGIO Y, COURVILLE A, VINCENT P.Representation Learning: A Review and New Perspectives[J].IEEE Transactions on Pattern Analysis&Machine Intelligence, 2013, 35 (8) :1798-1828

[3] SZEGEDY C, LIU W, JIA Y, et al.Going deeper with convolutions[C]// Proceedings of the Computer Vision and Pattern Recognition.2015:1-9.

[4] CHATFIELD K, SIMONYAN K, VEDALDI A, et al.Return of the Devil in the Details: Delving Deep into Convolutional Nets[J].arXiv:1405.3531.

[5] FAN J, XU W, WU Y, et al.Human tracking using convolutional neural networks[J].IEEE Transactions on Neural Networks, 2010, 21 (10) :1610-1623.

[6] ZHANG Z, WONG K H.Pyramid-Based Visual Tracking Using Sparsity Represented Mean Transform[C]// Proceedings of the IEEE Conference on Computer Vision&Pattern Recognition.2012:1822-1829.

[7] JING L.Incremental Learning for Robust Visual Tracking[J].International Journal of Computer Vision, 2008, 77 (1-3) :125-141.

[8] NAM H, HAN B.Learning Multi-Domain Convolutional Neural Networks for Visual Tracking[J].arXiv:1510.07945.

[9] HONG S, YOU T, KWAK S, et al.Online Tracking by Learning Discriminative Saliency Map with Convolutional Neural Network[C]// Proceedings of The 32nd International Confernece on Machine Learning.2015:597-606.

[10] MA C, HUANG J, YANG X, et al.Hierarchical Convolutional Features for Visual Tracking[C]// Proceedings of the IEEE International Conference on Computer Vision.2016:3074-3082.