An Improved Vehicle Tracking Method Based on MDNet

Wang Jianwen, Li Aimin and Pang Yewen

January 10, 2020

An Improved Vehicle Tracking Method Based on MDNet

Jianwen Wang, Qilu University of Technology (Shandong Academy of Sciences), JiNan, ShanDong, China, wjw19960808@163.com

Aimin Li*,Qilu University of Technology (Shandong Academy of Sciences), JiNan, ShanDong, China, 974509313@qq.com

Yewen Pang,Qilu University of Technology (Shandong Academy of Sciences), JiNan, ShanDong, China,

pangyewen137@gmail.com

ABSTRACT

In the field of intelligent transportation, many factors can affect the tracking results of moving vehicles in the video, such as background complexity, illumination change, occlusion, scale transformation and so on. In order to solve the drift problem and improve the tracking accuracy in the vehicle target tracking process, this paper proposed an improved vehicle target tracking algorithm based on MDNet. By combining the instance segmentation method with the MDNet algorithm, the background and vehicle targets can be distinguished remarkably, which enhance tracking performance greatly. The proposed tracking algorithm is evaluated on the OTB dataset. We compared the tracking result of our method with eight mainstream target tracking algorithms. The experimental results illustrate outstanding performance. The target tracking accuracy and tracking success rate of our algorithm achieve good performance in many cases.

KEYWORDS

intelligent transportation; MDNet; target tracking; instance segmentation

1 Introduction

Vehicle target tracking is a key problem in the research field of intelligent transportation. The intelligent transportation system performs tasks such as traffic flow control, vehicle abnormal behavior detection based on the captured video images. Accurate detection and tracking of vehicle targets are of great significance for traffic safety and intelligent vehicle management [1-3]. Different algorithms for tracking moving vehicles have been proposed in the past few years, such as optical flow based target tracking [4], motion estimation based target tracking [5], recognition based target tracking, and deep learning based target tracking method. The difficulty in vehicle target tracking lies in how to ensure the robustness, real-time and accuracy of the algorithm. The existing tracking algorithms have a good effect on dealing with the problem of moving vehicle tracking in some special cases. However, due to the complexity of the target motion and the feature change of the target, when there is occlusion, rotation, scale change and background interference, the tracking result is bad.

In order to achieve more robust target tracking results, researchers have proposed many tracking algorithms in recent years. In 2016, Martin Danelljan et al. proposed the C-COT (Continous Convution Operators for Tracking) [6] algorithm. C-COT combined with deepSRDCF uses the deep neural network VGG-NET [7] to extract target features. The disadvantage of C-COT is that the training data and feature space are large, resulting in a low frame rate. In 2017, Martin Danelljan proposed the ECO [8] target tracking algorithm for the above C-COT problem. ECO mainly solves the problem of too large C-COT model. Speed up tracking by reducing correlated filtering parameters, simplifying training sets, compressing feature space, and reducing model update frequency.

MDNet (Multi-Domain Convolutional Neural Networks) is a tracking method based on convolutional neural network proposed by Nam H et al. [9] in 2016. This tracking algorithm has outstanding performance compared to the state-of-the-art techniques. The method learns the representation of object from some labeled video sequences to assist tracking. Each of these videos is treated as a separate domain. The proposed network has separate branches, namely: domain-specific layers for binary classification. Each domain of MDNet is trained separately, and the shared layer is updated iteratively each time. The target needs to be tracked throughout the video frames, so this method has problems such as time-co nsuming and over-fitting.

The improved algorithm proposed in this paper combines MDNet with Mask RCNN [10], a famous instance segmentation algorithm. Using instance segmentation can reduce the tracking range of the vehicle to be tracked. First, the video image is segmented to obtain some candidate target regions, and then the segmented foreground vehicle region is used as the input of the MDNet for subsequent target tracking processing. Because the candidate tracking region obtained by instance segmentation is small, the network structure we used can also be relatively small. This strategy not only helps to improve tracking accuracy by distinguishing background and foreground targets better, but also reduce Computational load. Our experiment demonstrates better tracking performance of our improved tracking algorithm compared to MDNet tracking algorithm. The effectiveness of our algorithm is verified by quantitative and qualitative analysis.

2 Related Work

The most popular method in current target tracking is to train a classifier online, and then find the tracking target from the basic information provided in the first frame of the video, and continuously update online in subsequent video frames. Bolme et al [11] proposed a fast and efficient tracking detection strategy with a minimum output squared error and (MOSSE) filter, which can run hundreds of frames per second. Due to the adoption of multi-channel formulas, spatial constraints and the introduction of depth features, the performance of the filter tracker has been significantly improved. Grabner et al. proposed an online adaptive enhancement tracking algorithm (Online AdaBoost, OAB) [12] , which can select the most discriminative features online to distinguish between target and background. These methods can achieve satisfactory tracking in a restricted environment, but they all have certain limitations. Because most methods only use low-level features such as color, these features are susceptible affecting by in illumination, occlusion, deformation, etc. under dynamic conditions. Convolutional neural networks have achieved great success in many areas of computer vision, but their application in target tracking is extremely limited. By performing pre-trained large datasets, Fast R-CNN[13], applies CNN(convolutional neural networks)[14] to object detection tasks with scarce data. Because the convolutional neural network trains offline before tracking and remains unchanged during the tracking process, early tracking algorithms based on deep learning can only process predefined target object. Hong S al. [15] proposed a target tracking method based on convolutional neural networks, but compared with the traditional method, its accuracy is not well due to its lack of training data. In 2019, Bo Li et al. proposed SiamRPN++ (Evolution of Siamese Visual Tracking with Very Deep Networks) [16]. The problem of applying a network such as a deep network ResNet[17], Inception[18-21] to a tracking network based on Siamese Network is solved.

MDNet tracking algorithm learns domain-independent representations from pre-training, and then captures information about specific domains through online learning during the tracking process. The entire network is offline pre-trained, including the fully connected layer and the specific layers of a single domain. Based on the new tracking sequence of the target, combined with the shared layer in the pre-trained CNN, a new network is constructed by using a new binary classification layer. MDNet is updated online, which is achieved by evaluating candidate windows randomly sampled around the previous target. The tracking result of MDNet is greatly improved compared to some other popular target tracking method.

3 An Improved MDNet Tracking Method Combined With Instance Segmentation

The framework of our improved MDNet tracking algorithm is shown in Figure 1. Mask RCNN is used to segment video frames into instances first, and then the candidate regions obtained from the instances is used as the input of our improved MDNet algorithm. Thereby strengthening the foreground tracking target, reducing the scope of our tracking, can more clearly distinguish the background and target. Training and testing are conducted online. The network we used receives 107x107 RGB inputs, and has five hidden layers. Three layers are convolutional layers (conv1-conv3) and two are globally connected layers (fc4-fc5). Even though the network we used is small, it can still get robust tracking result.

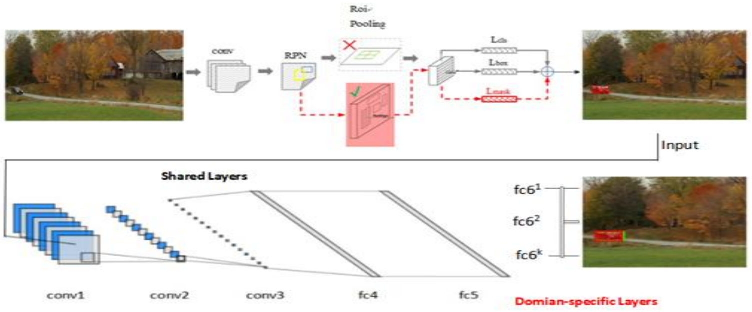

Figure 1:The overall work flow chart of this article.

3.1 Instance Segmentation

The framework of Mask RCNN proposed by Kaiming He is shown in Figure 2. Preprocessing operations such as data labeling on video frames is firstly performed, and then the labeled frames are inputted into the trained neural network.

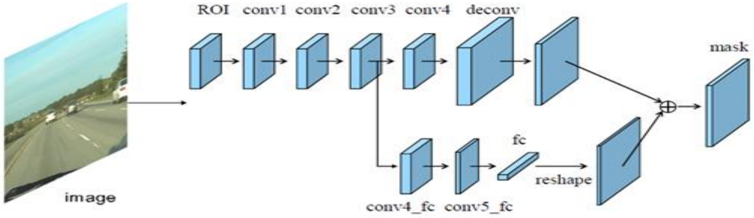

Figure 2:The Mask R-CNN framework for instance Segmentation.

The corresponding feature map is obtained, and then some candidate ROIs are acquired in the feature map. Next, these candidate ROIs are inputted into the RPN network for binary classification. At the same time some candidate ROIs are filtered out. The RPN network will output the coordinates of ROI as [x, y, w, h]. The coordinates are inputted ROI Pooling, and we can obtain a 7x7 size map for classification and positioning. The purpose of ROI Pooling is to adjust the ROIs of different sizes to a smaller feature map of 7X7. Next, the ROIAlign operation is performed to solve the problem of regional mismatch in the ROI Pooling operation. Mask RCNN uses average binary cross-entropy loss. The loss function of Mask R-CNN can be described as:

$$L_{final} = L(\{p_i\},\{t_i\} + (L_{cls} + L_{box} + L_{mask})) \qquad (1)$$

L_{cls} and *L_{box}* are used for classification and regression. *L_{mask}* classifies each pixel and contains the output of K * m * m dimension. K represents the number of categories and m * m is the size of ROI image extracted. Finally, these ROI are classified and Mask generated. The loss function of training RPN has the following description:

$$L(\{p_i,t_i\})=\frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i*)+\lambda\frac{1}{N_{reg}}\sum_i p_i*L_{reg}(t_i,t_i*) \qquad (2)$$

In the above formula, i is the sequence number of the anchor in each small batch, p is the probability of the anchor target, p* is the label, t is the four parameters of the prediction box, t* is the parameter of the calibration box, Lcls is the classification loss function; Lreg is the regression loss function. ROI Align back propagation can be described as follow:

$$\frac{\partial L}{\partial x_i}\sum_r\sum_j[d(i,i*(r,j)<1)](1\text{-}\Delta h)(1\text{-}\Delta w)\frac{\partial L}{\partial y_{rj}} \qquad (3)$$

d(.) represents the distance between two points, and Δh and Δw represent the difference between the abscissa and the ordinate of x_i and x_i*(r,j). The result of the example segmentation is shown in Figure 3. Through instance segmentation, we can basically get a relatively accurate region of the target to be tracked.

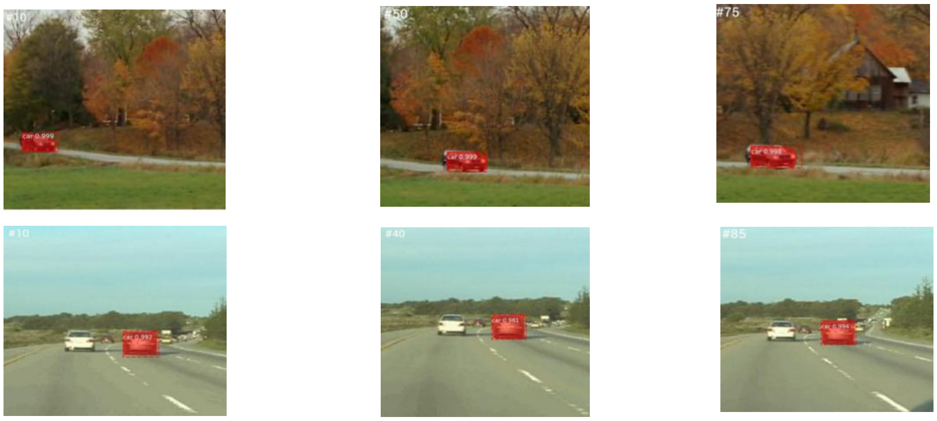

Figure 3:Instance Segmentation result.

3.2 Improved MDNet Vehicle Target Tracking Method

The main operation of target tracking is to distinguish the target from the background in the video images, which is simpler than the general target classification or target recognition problem, therefore a simpler network model can be adopted. Before applying MDNet to track the target, we used Mask RCNN described in 3.1 to segment the target in the video image to get the candidate region of the vehicle target. Tracking target in candidate regions makes it easier to improve tracking efficiency. Normally, the spatial position information of the target is diluted with the deepening of the network, which can deteriorate the tracking result or cause tracking failure. Our method can solve this problem. At the same time, the size of the vehicle candidate area we obtained by instance segmentation is much smaller than that of the original image, so we can achieve good tracking result with a smaller network.

The architecture of the network we used is shown in Figure 4. The input is a 107 x 107 RGB image. The network has five hidden layers, three convolutional layers (conv1-conv3) and two fully connected layers (fc4-fc5). The network architecture used is smaller than the target identification or target classification network, such as AlexNet [22], VGG-Net, etc.

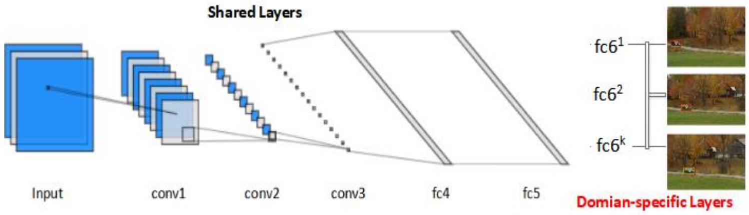

Figure 4:Algorithm Network Structure Diagram

MDNet uses the multi-domain learning framework to separate domain-independent information from domain-specific information. CNN is trained by a stochastic gradient descent (SGD) method in which each domain is specifically processed in each iteration. SGD method is used for training, in which the video sequence is disrupted. The original video sequence is arranged in frame order. In each iteration, 8 frames are taken in sequence, and then in these 8 frames, 4 positive samples (IOU>=0.7) and 12 negative samples (IOU<=0.5) are taken for each frame. Here IOU is the overlap ratio of the resulting candidate bound and ground truth bound. IOU is defined as follows:

$$\mathrm{IOU} = \frac{\mathrm{I(X)}}{\mathrm{U(X)}} \quad (4)$$

$$\mathrm{I(x)} = \sum_{\mathrm{v} \in \mathrm{V}} \mathrm{Xv} * \mathrm{Yv} \quad (5)$$

$$\mathrm{U(X)} = \sum_{\mathrm{v} \in \mathrm{V}} (\mathrm{Xv} + \mathrm{Yv} - \mathrm{Xv} * \mathrm{Yv}) \quad (6)$$

The segmentation result obtained by instance segmentation is unified into 107*107 as the input of the network. 32 positive samples and 96 negative samples form a mini-batch. In the process of target tracking, we always maintain a simple network.

According to the change speed of the target appearance, we adopt two update methods: long-term update and short-term updates. The long-term update is updated at regular intervals, and the short-term updates are adopted when a potential update fails, i.e., when the predicted target's positive score is less than 0.5. When we predict the state of each frame target, first distribute N templates around the object in the previous frame, then get the positive sample score $f^+(x^i)$ and the negative sample score $f^-(x^i)$ according to our network. We take the sample with the highest score as our current optimal target state X*:

$$X^* = \arg\max_{x^i} f^+(x^i) \quad (7)$$

256 candidate regions are generated around the predicted target position of the previous frame on each frame according to the Gaussian distribution. Each generated candidate box is expressed as (x, y, w, h). Then, the candidate boxes are cut from the original frame image, and resized into 107*107, which is used as the input of the network for calculation. The scores of the 256 candidate regions are calculated by forward propagation. The 5 candidate regions with the highest scores are selected. These candidate box regions are averaged to generate a target bounding box of the current frame. And the average of the candidate region scores is calculated as the score of target bounding box. We set a threshold and then compare the score with the threshold to determine if the tracking is successful. If the tracking is successful, the bounding box is fine-tuned. 50 positive sample regions (IOU>=0.7) are generated according to the target bounding box of the current frame prediction, and 200 negative sample regions (IOU<=0.3) are generated. The sample regions are then forwardly propagated separately, and finally the features of conv3 of these regions are preserved. If the number of video frames exceeds 100, the positive sample regions of the earliest frames are discarded. If the number of video frames exceeds 20, the negative sample regions of the earliest frames are discarded.

If the track fails, short-term update is performed. Select the positive and negative samples of the last 20 frames, and then perform iterative training for 15 rounds. The iterative process is the same as normal iteration. Each iteration randomly extracts 32 positive samples and 1024 negative samples to form a mini-batch. These 1024 negative samples are placed in the test model, and 4 cycles are performed. The score is calculated, and the result of the calculation is the target's score. Then 96 of the 1024 negative samples with the highest score were selected as the hard negative samples. Next, the training model is imported, and the scores of the positive samples (32) and the scores of the difficult negative samples (96) are calculated separately, and the loss is calculated by the forward propagation, and then the optimizer is optimized and the parameters are updated.

The use of ReLU [23] activation function in MDNet can effectively overcome the problem of gradient disappearance and speed up the training process. But if the learning rate is too high, 40% of the neurons in the network will be dead. We adopt RReLU [24] activation function in our improved MDNet algorithm. In RReLU, a_{ji} is a value that is evenly distributed in *U* (*l, u*) and randomly extracted. This value will be fixed in the testing process. The activation function we use is expressed as follows:

$$\mathrm{y\ ji} = \begin{cases} \mathrm{x}_{\mathrm{ji}} & \text{if } \mathrm{x}_{\mathrm{ji}} \geq 0 \\ \mathrm{a}_{\mathrm{ji}}\mathrm{x}_{\mathrm{ji}} & \text{if } \mathrm{x}_{\mathrm{ji}} < 0 \end{cases}$$

Where (8)

$$a_{ji} \sim U(l,u), l \prec u \text{ and } l,u \in [0,1)$$

How to set learning rate is also very important. If the learning rate is set too small, the entire network convergence process may become extremely slow; if the learning rate is set too large, the gradient may be near the minimum value, and may not even converge. Our learning rate is not fixed, but is dynamically set according to the number of training rounds. At the beginning of training process a relatively larger learning rate is used when the distance is farther from the optimal solution. As the number of iterations increases, the learning rate is gradually reduced in the process of approaching the optimal solution.

4 Experimental Verification

4.1 Evaluation on OTB

OTB[25-26] is a popular tracking benchmark that contains 100 fully annotated videos with substantial variations. The evaluation is based on two metrics: center location error and bounding box overlap ratio.We select the video sequence describing traffic to evaluate the proposed algorithm.The algorithm is implemented in Python, and runs at around 1 fps with 2.30GHz Intel(R) Xeon(R) Gold 5118 CPU.

4.2 Evaluation Comparision

To further validate the comprehensive performance of our improved algorithms, this section will show the experimental results improved and the experimental results of other trackers. The green box represents the Ground Truth of the target, and the red box represents the bounding Box of the tracking result of the target. Figure 5. shows the tracking effect of MDNet . Figure 6. is the tracking result of our improved method where the red solid rectangle represents the result of the instance segmentation:

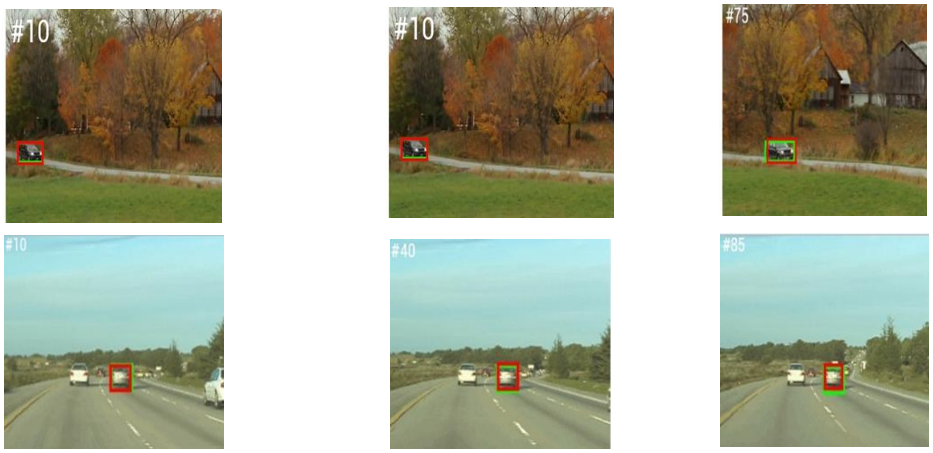

Figure 5:MDNet Tracking Effect.

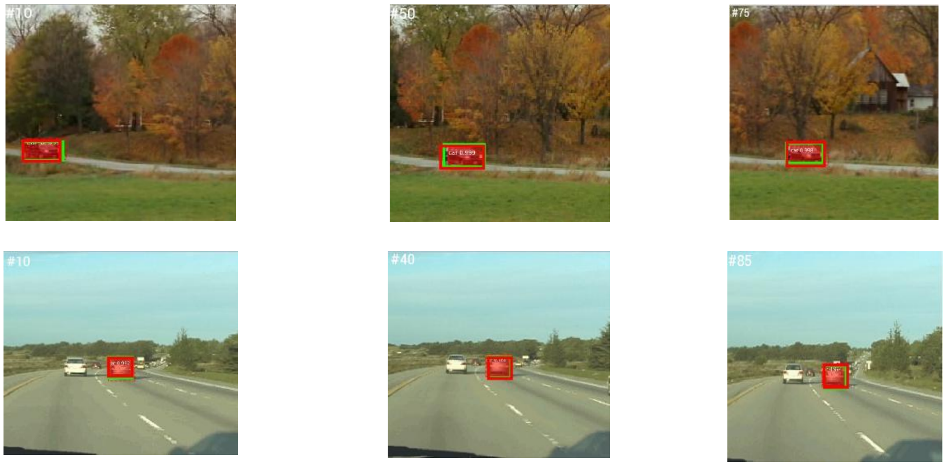

Figure 6: Our Algorithm Effect.

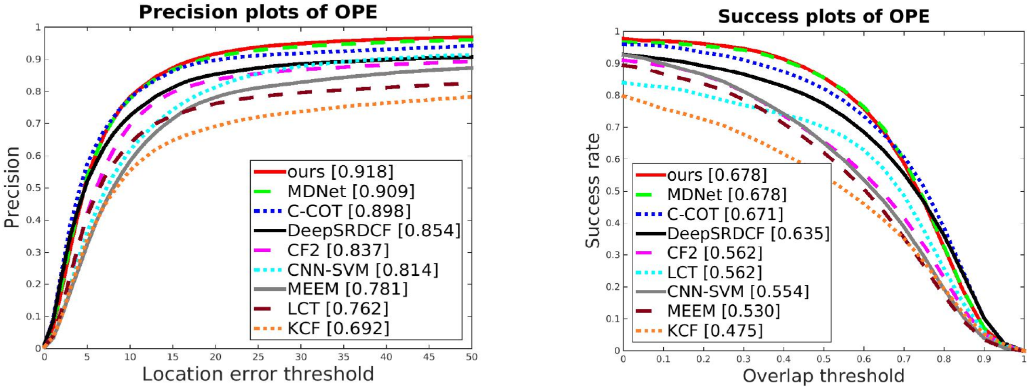

Figure 6: Comparison of optimization methods with other tracking results. Figure 7. (a) and Figure 7. (b) show the comparison of the precision and success plot of the improved algorithm with other algorithms on OTB. The numbers in the legend indicate the representative precisions at 20 pixels for precision plots, and the area-under-curve scores for success plots.

Figure 7 shows that the improved tracking success rate is better than most tracking algorithms. Comparative experimental data with other tracking algorithms are shown in Table I.

Table 1: Experimental result comparison

	Precision rate	Success rate
Ours method	**0.918**	**0.678**
MDNet	0.909	0.678
C-COT	0.898	0.671
DeepSRDCF	0.854	0.635
Cf2	0.837	0.562
CNN-SVM	0.814	0.562
MEEM	0.781	0.554
LCT	0.762	0.530
KCF	0.692	0.475

5 Conclusions

In order to improve the performance of the target vehicle tracking algorithm, we have improved the MDNet tracking method. Firstly, the candidate region of the target vehicle is obtained in the video image by the method of instance segmentation. MDNet tracking method is then applied to the candidate area, so that a small network size can be maintained during the tracking process to avoid the deterioration of the tracking result. The activation function and learning rate of MDNet were also changed to achieve improved accuracy of tracking result. The experimental results show that our method has an overall improvement in tracking performance compared to other methods. In the future, we will continue to study how to improve the tracking efficiency of MDNet, and improve the performance of the algorithm in order to obtain better tracking results.

REFERENCES

[1] Li He,Shiru Qu.Vehicle Detection Based on PLS-VIP Feature Dimensionality Reduction[J]. China Journal of Highway and Transport, 2014, 27(04): 98-105.]

[2] Liu Zhanwen, Zhao Xiangmo, Wang Wei, Gao Tao, Li Shuying. Vehicle Target Segmentation Method with Weak Contrast Based on Visual Attention Mechanism[J]. China Journal of Highway and Transport, 2016, 29(08): 124-133.

[3] Gao Wei, Liu Zhengguang, Yue Shihong, Zhang Jun. Motion Vehicle Tracking Algorithm for Intelligent Transportation[J]. China Journal of Highway and Transport, 2010, 23(03): 89-94.

[4] Choi I H, Pak J M, Ahn C K, et al. Arbitration algorithm of FIR filter and optical flow based on ANFIS for visual object tracking[J]. Measurement, 2015, 75: 338-353.

[5] Sun J, He F, Chen Y, et al. A multiple template approach for robust tracking of fast motion target[J]. Applied Mathematics-A Journal of Chinese Universities, 2016, 31(2): 177-197.

[6] Danelljan M, Robinson A, Khan F S, et al. Beyond correlation filters: Learning continuous convolution operators for visual tracking[C]//European Conference on Computer Vision. Springer, Cham, 2016: 472-488.

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

[8] Danelljan M, Bhat G, Shahbaz Khan F, et al. Eco: Efficient convolution operators for tracking[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 6638-6646.

[9] Nam H, Han B. Learning multi-domain convolutional neural networks for visual tracking[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 4293-4302.

[10] He K, Gkioxari G, Dollár P, et al. Mask r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2017: 2961-2969.

[11] Bolme D S, Beveridge J R, Draper B A, et al. Visual object tracking using adaptive correlation filters[C]//2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010: 2544-2550.

[12] Grabner H, Leistner C, Bischof H. Semi-supervised on-line boosting for robust tracking[C]//European conference on computer vision. Springer, Berlin, Heidelberg, 2008: 234-247.

[13] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[14] Shin H C, Roth H R, Gao M, et al. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning[J]. IEEE transactions on medical imaging, 2016, 35(5): 1285-1298.

[15] Hong S, You T, Kwak S, et al. Online tracking by learning discriminative saliency map with convolutional neural network[C]//International conference on machine learning. 2015: 597-606.

[16] Li B, Wu W, Wang Q, et al. Siamrpn++: Evolution of siamese visual tracking with very deep networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 4282-4291.

[17] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

[18] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 1-9.

[19] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[J]. arXiv preprint arXiv:1502.03167, 2015.

[20] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2818-2826.

[21] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[C]//Thirty-First AAAI Conference on Artificial Intelligence. 2017.

[22] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]//Advances in neural information processing systems. 2012: 1097-1105.

[23] Agarap A F. Deep learning using rectified linear units (relu)[J]. arXiv preprint arXiv:1803.08375, 2018.

[24] Xu B, Wang N, Chen T, et al. Empirical evaluation of rectified activations in convolutional network[J]. arXiv preprint arXiv:1505.00853, 2015.

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

[26] 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.