REAL-TIME VEHICLE DETECTION AND TRACKING USING DEEP NEURAL NETWORKS

XIAO-FENG GU[1], ZI-WEI CHEN[1], TING-SONG MA[1], FAN LI[1], LONG YAN[2]

[1] International Centre for Wavelet Analysis and Its Applications, School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 611731
[2] State Grid Inner Shizuishan Power Supply Company, Shizuishan, 753000, China
E-MAIL: 421525351@qq.com

Abstract:

Dynamic vehicle detection and tracking can provide essential data to solve the problem of road planning and traffic management. A method for real-time vehicle detection and tracking using deep neural networks is proposed in this paper and a complete network architecture is presented. Using our model, you can obtain vehicle candidates, vehicle probabilities, and their coordinates in real-time. The proposed model is trained on the PASCAL VOC 2007 and 2012 image set and tested on ImageNet dataset. By a carefully design, the detection speed of our model is fast enough to process streaming video. Experimental results show that proposed model is a real-time, accurate vehicle detector, making it ideal for computer vision application.

Keywords:

Vehicle detection; Vehicle tracking; OAUE; Deep neural networks

1. Introduction

In today's society, more and more vehicles are taking to the highways every year, which makes a push to monitor and control the traffic more efficiently. The real-time vehicle detection and tracing is essential for intelligent road routing, road traffic control, road planning and so on. Therefore, it is important to know the road traffic density real time, especially in mega cities for signal control and effective traffic management.

For a long time, several approaches[1,2] in the literature have been proposed to resolve the problem of various moving vehicles; Nevertheless, the aim of real-time fully-automatic detection of vehicle is far from being attained as it needs improvement in detection and tracking for accurate prediction with faster processing speed. Zheng et al. use brake lights detection through color segmentation method to generate vehicle candidates and verify them through a rule-based clustering approach. A tracking-by-detection scheme based on Harris-SIFT feature matching is then used to learn the template of the detected vehicle on line, localize and track the corresponding vehicle in live video [2]. It is a good measure to extract vehicle areas, however, it needs a relatively ideal background. Wei Wang et al. have presented a method of multi-vehicle tracking and counting using a fisheye camera based on simple feature points tracking, grouping and association. They integrates low level feature-point based tracking and higher level "identity appearance" and motion based real-time association [1]. However, the average processing time of it is around 750ms, which is not fast enough to achieve the real-time processing.

System based Convolutional Neural Networks (CNN) can provide the solution of many contemporary problems in vehicle detection and tracing. CNN currently outperform other techniques by a large margin in computer vision problems such as classification [3] and detection [4]. The training procedure of CNN automatically learn the weights of the filters, so that they are able to extract visual concepts from raw image content. Using the knowledge obtained through the analysis of the training set containing labelled vehicle and non-vehicle examples, vehicle can be identified in given images. In general, Convolutional Neural Networks show more promising results.

In this paper, we propose a method of real-time vehicles detection and tracking using Convolutional Neural

Networks. We present a network architecture, which create multiple vehicle candidates and predict vehicle probabilities in one evaluation. Our architecture uses features from the entire image to create vehicle candidates. Firstly, we use convolutional layers of the system to extract features from the raw image. Secondly, we use four kinds of inception modules. Thirdly, we add Spatial Pyramid Pooling (SPP) layer between convolutional layers and fully connected layers, which is able to resize any images into fixed size. Lastly, the fully connected layers predict the probability and coordinates of vehicles.

The rest of the paper is organized as follows. Section 2 presents the proposed vehicle detector based on convolutional neural networks and explain it in detail. In Section 3, experimentation process is described. Finally, conclusion of the work is given in Section 4.

2. Vehicle Detector

We present a network architecture on the basis of deep convolutional neural networks. Our network architecture is composed of convolutional layers, spatial pyramid pooling layer, inception modules, and fully connected layers as shown in Fig.1.

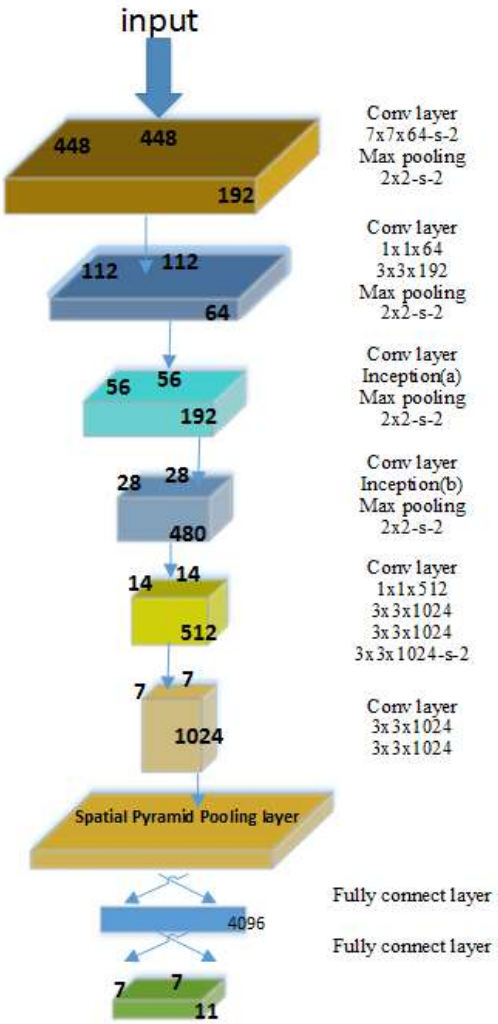

Fig.1 Network Architecture

Inception module is proved to be especially useful in the context of localization and object detection as the base network for [6] and [5]. It is able to increase the depth and width of the network while keeping the computational budget constant. The main idea of the inception architecture proposed firstly in GoogleNet [7] is to consider how an optimal local sparse structure of a convolutional vision network can be approximated and covered by readily available dense components. Therefore, four kinds of inception modules are added into our system with the purpose of broadening the network and reducing the number of parameters as shown in Fig.2.

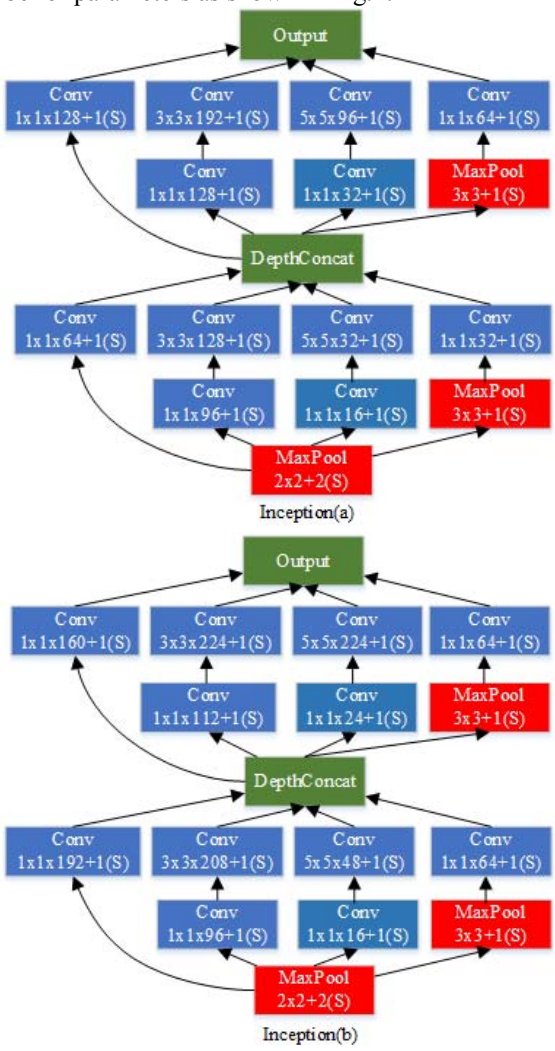

Fig.2 Inception Modules

We all know that fully connected layer needs fixed size of input image, but image sets always have different sizes and compression on the image inevitably results in image distortions. Spatial Pyramid Pooling is able to resize any image into fixed size. In spite of concerns that max-pooling layers lead to loss of accurate spatial information, the same convolution network architecture as

[8] has been successfully used for [8,9], object detection [9,10,11,12]. Therefore, to the problem of poor performance in unusual aspect ratios, in our network, we add Spatial Pyramid Pooling layer between convolutional layers and fully connected layers.

Table 1 illustrate convolutional layers, max-pooling layers, and inception layers used in our architecture. The first part is a 7×7 convolution with 64 filters, which learn convolved features from the image pixels and result in a 224×224×64 output. Then a max-pooling layer with 2×2 filter size and stride 2 halve the resolution of the grid. In order to reduce the features space from preceding layers, a 1×1 convolutional layer is exploited in the second part, then a 3×3 convolution with 192 filters learn convolved feature from the output of the last layer. The next part consist of inception modules and max-pooling layers result in a 14×14×512 output.

Table 1 Incarnation of our architecture

Type	Patch size	Stride	Output size	Depth
convolution	7 x 7	2	224x224x64	1
max pool	2 x 2	2	112x112x64	0
convolution	1 x 1	1	112x112x64	1
convolution	3 x 3	1	112x112x192	1
max pool	2 x 2	2	56x56x192	0
inception(a)			56x56x480	4
max pool	2 x 2	1	28x28x480	0
inception(b)			28x28x512	4
max pool	2 x 2	2	14x14x512	0
convolution			7x7x1024	6

Additional 6 convolutional layers are added to the network; they are used mainly as dimension reduction modules to remove computational bottlenecks and this allows for increasing the depth and width of our networks without a considerable performance penalty. Our final layer predicts predict both vehicle probabilities and their coordinates. We use a linear activation function for the final layer, and all the convolutions use rectified linear activation:

$$\varphi(x) = \begin{cases} x, & if\ x > 0 \\ 0.1x, & otherwise \end{cases} \quad (1)$$

3. Experiment

We train our network on the PASCAL VOC 2007 and 2012 [13], and in order to enhance the performance of detecting vehicle, we label them by ourselves. Our system is implemented on a build which consist of a 8-core 4GHz Intel I7 6700K, a NVIDIA GTX1080 GPU and 8GB of RAM and we train this network for approximately 5 days.

After obtaining the knowledge through the analysis of the training set, we test our network on ImageNet dataset [14]. As shown in Fig.3, our system create multiple vehicle candidates, predict vehicle probabilities and coordinates.

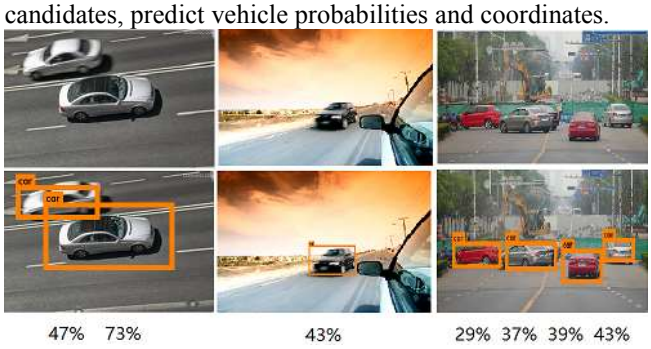

Fig.3 Qualitative Results on Our Model

In Table 2, we compare our detector with R-CNN, Fast R-CNN, Faster R-CNN, 100Hz DPM and UVA on PASCAL VOC 2012. They are trained on the same image sets. Our detector achieves a large improvement in mAP. The mAP of our network up to 80.5% in detecting vehicle class, which reach a state-of-the-art result among those methods.

Table 2 PASCAL VOC 2012 test

Method	mAP
Our Method	80.5
R-CNN	64.4
Faster R-CNN	76.6
Fast R-CNN	72.0
DPM	47.4
UVA	63.5

In Table 3, we compare our detection speed with other detection systems. Our networks runs at 46 frames per second on a GTX1080 GPU. This means it can process streaming video in read-time.

Table 3 Detection on PASCAL VOC 2007 and 2012

Model	Train set	mAP	FPS
Our method	VOC2007+2012	63.5	46
R-CNN	VOC2007	53.5	6
Fast R-CNN	VOC2007+ 2012	70.0	12
Faster R-CNN	VOC2007+ 2012	62.1	18
DPM	VOC2007	15.0	90

4. Conclusions

In our work, we propose a network architecture for vehicle detection and tracking in real-time. In general, our network includes 9 convolutional layers, 4 inception modules, one SPP layer and 2 fully connect layers. It is

ideal for computer vision application. This proposed system can be applied in the fields of intelligent road routing, road traffic control, road planning and so on.

However, there are limitations of our network architecture. Our system struggles with small objects and nearby object in groups. Future works will contain the improvement of that.

Acknowledgements

This paper was supported by the National Natural Science Foundation of China (Grant No. 61370073), the National High Technology Research and Development Program of China (Grant No. 2007AA01Z423), the project of Science and Technology Department of Sichuan Province, and the Scientific Research Project of Chengfei Group.

References

[1] Wang W, Gee T, Price J, et al. Real time multi-vehicle tracking and counting at intersections from a fisheye camera[C]//2015 IEEE Winter Conference on Applications of Computer Vision. IEEE, 2015: 17-24.

[2] Zheng Z H, Wang B. On-Road Vehicle Detection Based on Color Segmentation and Tracking Using Harris-SIFT[J]. Advanced Materials Research, 2012, 433-440:5334-5338.

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

[4] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[5] Erhan D, Szegedy C, Toshev A, et al. Scalable Object Detection Using Deep Neural Networks[J]. 2013:2155-2162.

[6] Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation [J]. Computer Science, 2014:580-587.

[7] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2014:1-9.

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

[9] Sermanet P, Eigen D, Zhang X, et al. Overfeat: Integrated recognition, localization and detection using convolutional networks[J]. arXiv preprint arXiv:1312.6229, 2013.

[10] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[11] Szegedy C, Toshev A, Erhan D. Deep neural networks for object detection[C]//Advances in Neural Information Processing Systems. 2013: 2553-2561.

[12] Erhan D, Szegedy C, Toshev A, et al. Scalable object detection using deep neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014: 2147-2154.

[13] Everingham, M., Eslami, S., Gool, L.: The Pascal Visual Object Classes Challenge: A Retrospective. Int. Interna-tional Journal of Computer Vision. 111(1), 98{136 (2015)

[14] Sadeghi M A, Forsyth D. 30hz object detection with dpm v5[C]//European Conference on Computer Vision. Springer International Publishing, 2014: 65-79.