ISBN 978-981-11-3671-9
Proceedings of 2017 the 7th International Workshop on Computer Science and Engineering (WCSE 2017)
Beijing, 25-27 June, 2017, pp. 267-271
doi: 10.18178/wcse.2017.06.046

Vehicle Detection from Static Images in Unrestricted Scenes Using Deep Convolutional Neural Network

Zhuo Yan [1 2 +], Cheng Cheng [1], Yi Xie [1], Jianting Fu [1 2], Peng Cheng [2], Yu Shi [1], Xiangdong Zhou [1 3] and Jiahu Yuan [1]

[1] Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China

[2] University of Chinese Academy of Sciences, Beijing, China

[3] Automated Reasoning and Cognition Key Laboratory of Chongqing, Chongqing, China

Abstract. Most of the traditional methods, which extract manual feature from data, are based on the particular scene or video source. In this paper, we propose a vehicle detection method that targets to the static images in unrestricted scenes. Firstly, we measure similarities of all initialization regions and merge them by some rules to get bounding boxes. Then the features of these bounding boxes are extracted by deep convolutional neural network (D-CNN) respectively. Finally, Lib-SVM classifier is employed to classify each bounding box and to complete vehicle detection. Compared with traditional method, the proposed strategy performs stronger robustness.

Keywords: vehicle detection, similarity measure, static images, deep convolutional neural network

1. Introduction

Vehicle detection method can be roughly classified into two categories. The first category is based on the physical methods, there are mainly three classifications which are the frame difference method [1], the optical flow method [2], and the background difference method [3]. The second category is based on the feature extraction method such as PCA, Harr [4], oriented gradients (HOG) feature [5], SIFT [6], support vector machine (SVM) [7] and Adaboost classifier. However, these methods have different application fields, and the generalization ability of them is poor.

Deep convolutional neural network is widely used in the field of object detection, because of its strong ability of depth feature extraction and representative for data source. He et al. [8] introduced the multi-task learning mechanism into the training process of D-CNN which was applied to the scene text detection. Pre-trained D-CNN was utilized by Nogueira et al. [9] to detect the activity of fingerprints. Zhan et al. [10] employed the D-CNN to extract the feature and completed classification to realize face detection. Furthermore, the combination of multi-task learning and D-CNN was carried out by Tang et al. [11] on facial landmark detection. A recent breakthrough in convolutional neural network is achieved by Alex and Geoffrey [12]. In order to strengthen the ability about nonlinear feature extraction and the generalization ability of network, Network in Network (NIN) architecture was proposed by Yan et al. [13] The related spatial pyramid pooling in deep convolutional networks (SPP-net), showed by He et al. [14], applied the spatial pyramid pooling layer on each candidate window and generated a fixed-length representation before fully-connected layers.

In this paper, we propose a vehicle detection method based on deep neural network from the static images in unrestricted scenes. This method could overcome the influence of the complex conditions effectively, and has strong practicability.

[+] Corresponding author. Tel.: +86-138-4003-9998.
E-mail address: yanzhuo@cigit.ac.cn.

2. Overview of Method

The method is composed of three parts. The first part is pre-processing. Exactly, all the 8 neighborhood pixels of an image are segmented into a large many regions, as initialization regions, based on the spacing between each other. Then, these regions are merged into bounding boxes of vehicle candidates according to the similarities of them, including: color, texture, size of region, intersection-over-union (IOU) of region, etc. In the second step, the feature of each the bounding box is extracted by the iterative convolution operation and maximum pooling process. In the third step, the LibSVM classifier is employed to judge every bounding box is a vehicle or not. This is the first time to apply this method to address the vehicle detection problem.

3. Detection Implementation

3.1. Pre-processing of static images

We define the region spacing is the maximum weight value of all the two adjacent points in one region. The spacing between two regions is the minimum weight value of all the two adjacent points which are belonging to different regions. Supposing that, the input image concludes n pixels and m edges, the output consists of a series of connected regions. All the edges of the input image are in descending order according to their weight values. It is assumed that there is an edge between two pixels as long as they are adjacent. The weight value of an edge is the absolute value of the difference between the two pixels, as shown in (1):

$$\omega\left(p_i, p_j\right) = \left|I\left(p_i\right) - I\left(p_j\right)\right| \tag{1}$$

where I(pi) and I(pj) represent the weight values of pixel i and pixel j, respectively. The initial segmentation is denoted as Seg0, in which each pixel is a region, and then Segk is constructed by Segk-1 according to definite rules. If the pi and pj in Segk-1 belong to different regions respectively, and the weight value of the edge k is smaller than any region spacing of these two regions, we merge the two regions, otherwise set Segk = Segk-1. The process is shown as (2):

$$MIN\left(R_a, R_b\right) = \min\left(In\left(R_a\right) + \tau\left(R_a\right), In\left(R_b\right) + \tau\left(R_b\right)\right) \tag{2}$$

where the MIN(Ra, Rb) means the smaller region spacing of region Ra and region Rb. In addition, every single pixel is a region R at the start of initialization, and each a pair of pixels will merge if the weight value between them is the same. To avoid over segmentation, a range about similarity should be set for every pixel. In the (3), the τ(R) is used to define a threshold of similarity, which is two areas are merged according to, and the |R| means the size of region R.

$$\tau(R) = k/\left|R\right| \tag{3}$$

Next, we find out two regions which have the minimum similarity between them. There are numbers of methods to obtain similarity of two regions, such as color, texture and the size of overlap. Considering the characteristic of vehicle, such as the texture of the front window, the texture of radiator, the distance ratio for the positions of headlights and fog lamp, and some others, this paper uses HOG features as similarity for comparing. As shown in (4):

$$S\left(r_i, r_j\right) = \sum_{k=1}^{n} \min\left(t_i^k, t_j^k\right) \tag{4}$$

in which *n* represents the product of interval number, bin number and color channel number, and *t* means the vector value of the corresponding interval. When the similarities of all regions in the image are calculated, these areas will be merged following $r_t = r_i \cup r_j$. In details, after region merging operation for one iteration, all the similarities $S(r_i, r^*)$ and $S(r^*, r_j)$ which are relevant to r_i and r_j should be removed from the set *S*. In the new set, similarities $S(r_t, r^*)$ between r_t and its adjacent regions are calculated, and then the new set is merged with set *R*. After the color histogram of new region is calculated, the regions related to set *S* are merged again until it is empty, and a new set *R* is got.

3.2. Feature extraction

In this paper, the D-CNN structure contains five convolutional layers (conv1 to conv5), two fully connected layers (fc6 and fc7), and one output layer for classification. Conv1 is the result of convolving its previous layer, the input image, with learned filters. It is similarly for conv2 to conv5, fc6, and fc7. Parameter setting of each layer is introduced in detail as the following: the number of feature map in Conv1

is 96, and the dimensions of feature map, convolution kernel and pooling layer are 27*27, 11*11 and 3*3, respectively; the number of feature map in Conv2 is 256, and the dimensions of feature map, convolution kernel and pooling layer are 12*12, 5*5 and 3*3, respectively; the number of feature map in Conv3 is 384, and the dimensions of feature map and convolution kernel are 11*11 and 3*3, respectively; the number of feature map in Conv4 is 384, and the dimensions of feature map and convolution kernel are 10*10 and 3*3, respectively; the number of feature map in Conv5 is 256, and the dimensions of feature map, convolution kernel and pooling layer are 5*5, 3*3 and 2*2, respectively; the dimensions of the last two fully connected layers are 4,096 and 1,024, respectively.

As the above introduced, convolutional layer and pooling layer are the major components of D-CNN. The input data are convolved with corresponding convolution kernel to obtain feature maps. And they are used as the input data for the subsequent pooling layer for dimensionality reduction. Then, repeat the above operations. Specifically, the input images in this work are uniformed to the size of 224*224 pixels. The size of the convolution kernel is uniformed to the 5*5 pixels, and the step size is 4. Due to the weight sharing strategy, there are 26 training parameters of each feature map, and 48,400 connection weight values (44*44*25) after Conv1. The calculation method of convolutional layer is following (5):

$$g\left(x, y\right) = f\left(x, y\right) * \left(c, u\right) + \varepsilon\left(b\right) \tag{5}$$

where $\varepsilon\left(b\right)$ means the bias, f represents the input image, and c is the convolution kernel.

Pooling layer which is following the convolutional layer actually plays the role of sub sampling and dimensionality reduction. The process of pooling: sum the pixel values of four adjacent pixels with weight vector W and bias b. Then, sigmoid activation function is employed to generate a feature map P which is reduced by four times. Repeat the above steps until the dimension of feature map becomes 1*1, before the fully connection is carried out. Finally, feature vectors of each bounding box are obtained.

3.3. Classification with LibSVM classifier

In last part of this detection method, all the feature vectors of bounding boxes of vehicle candidates will be judged by LibSVM classifier. Then the best results are extracted and output on the input image. In details, feature vector is multiplied by a trained SVM model, just like a linear kernel function, following (6):

$$g\left(x\right) = W \bullet f\left(x\right) + b, \tag{6}$$

in which W represents the discriminant model of SVM, f means the feature vector, and g is the final result score. After the scores of each bounding box are calculated, the bounding box with the maximum value will be selected as the final detection result, and output on the original image.

4. Results

4.1. Process of vehicle detection

Original image as input data are brought into D-CNN in pre-processing part which could generate a large number of merged regions with irregular size. These regions will be used for the next classification part. Here, the regions are the merging results from the pixels which are segmented, merged and extracted accordance to some certain rules, and these processes are shown in Fig. 1.

Fig. 1: The process about merging and extracting of segmented regions of vehicle.

4.2. Comparisons

In order to verify the effectiveness of the detection method proposed in this paper, HOG feature is chosen for comparison. We observe and compare the recall rates of these two methods in the situation of the same accuracy rate by adjusting the size of the bounding box as well as parameters of IOU. The results of experiment are shown in table 1. From the Fig. 2, we can see that our method performs significantly better than HOG in the recall rate.

Table 1: Comparisons results

Method	Accuracy Rate						
	0.95	0.92	0.88	0.85	0.8	0.77	0.75
	Recall Rate						
HOG	0.53	0.62	0.71	0.77	0.79	0.79	0.8
Ours	0.72	0.72	0.8	0.85	0.85	0.86	0.89

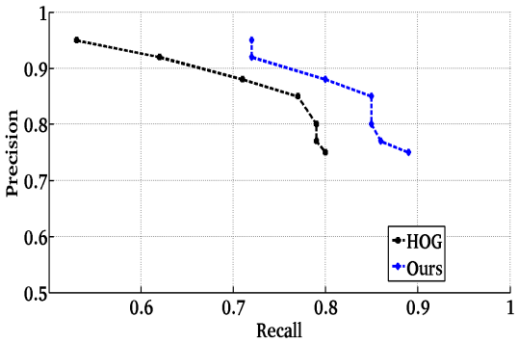

Fig. 2: Comparison of HOG and our method.

4.3. Complex conditions

In addition, this paper lays out experiments on the vehicle images under a variety of conditions, such as dark, exposure, multiple view angles and occlusion, which are as shown in Fig. 3. The left side of the figure is the performance of vehicle detection under the different occlusions and view angles. The right side of the figure is the detection results under the various light intensities. The results show that the proposed method could perform good robustness for vehicle detection under a variety of conditions.

Fig. 3: The results of vehicle detection under the various kinds of condition.

4.4. Various Types of Vehicle

Furthermore, we also carry out challenge experiments on various types of vehicle, including: cars, SUV, vans, buses, large trucks, etc., and achieve good performances. Fig. 4 shows the experimental results of a part of the test set. The first row is the detection results of cars, the second row is about vans, the third row is about buses and large trucks, and the last row shows the results of SUV.

Fig. 4: The results of vehicle detection in actual traffic road.

5. Conclusion

Deep convolutional neural network has been applied in many fields, but a few related works on vehicle detection. In this paper, we collect a large number of different types of vehicle images, which are under the various scenes, view angle, light, occlusion and other conditions, from the actual traffic road in Xinjiang Province. These images are segmented into bounding boxes of vehicle candidates in pre-processing. Then D-CNN is deployed to extract the deep features of the bounding boxes. Finally, the LibSVM classifier is employed to classify each bounding box whether it is a vehicle and to produce the final vehicle detection. We also compare two feature extraction methods which are based on HOG and D-CNN, respectively. The experimental results show that the number of bounding boxes of vehicle candidates in pre-processing is less, and the feature extracted by the D-CNN has stronger representative than traditional method. The vehicle detection method proposed in this work could effectively overcome a variety of conditions to achieve accurate detection for various types of vehicles. Therefore, it has a certain practical value and promotion.

6. Acknowledgements

This research was supported by the National Natural Science Foundation of China (No. 61502444), Strategic Priority Research Program of the Chinese Academy of Sciences (Grant XDA06040103) and Chongqing Municipal Science and Technology Commission (No. cstc2014jcyjA10036, cstc2015jcyjA10062 and cstc2016shms-2t2x0052-04).

7. References

[1] Bayona, Á., SanMiguel, J. C., and Mart ńez, J. M. Stationary foreground detection using background subtraction and temporal difference in video surveillance. *Proc. of 2010 IEEE International Conference on Image Processing (ICIP)*, IEEE. 2010, pp. 4657-4660.

[2] Ramirez, A., Ohn-Bar, E., and Trivedi, M. M. Go with the Flow: Improving Multi-view Vehicle Detection with Motion Cues. *Proc. of 22nd International Conference on Pattern Recognition (ICPR)*, IEEE. 2014, pp. 4140-4145.

[3] Qi, M. B., Yang, A. L., and Jiang, J. G. A vehicles detection and tracking algorithm based on improved codebook. *Journal of Image and Graphics*. 2011, 16 (3): 406-412.

[4] Viola, P., and Jones, M. J. Robust real-time face detection. *International journal of computer vision*. 2004, 57 (2): 137-154.

[5] Dalal, N., and Triggs, B. Histograms of oriented gradients for human detection. *Proc. of 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE. 2005, pp. 886-893.

[6] Chen, X., and Meng, Q. Vehicle detection from UAVs by using SIFT with implicit shape model. *Proc. of 2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE. 2013, pp. 3139-3144.

[7] Chang, C. C., and Lin, C. J. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*. 2011, 2 (3): 27.

[8] He, T., Huang, W., Qiao, Y., and Yao, J. Text-attentional convolutional neural network for scene text detection. *IEEE Transactions on Image Processing*. 2016, 25 (6): 2529-2541.

[9] Nogueira, R. F., de Alencar Lotufo, R., and Machado, R. C. Fingerprint Liveness Detection Using Convolutional Neural Networks. *IEEE Transactions on Information Forensics and Security*. 2016, 11 (6): 1206-1213.

[10] Zhan, S., Tao, Q. Q., and Li, X. H. Face detection using representation learning. *Neurocomputing*. 2016, 187: 19-26.

[11] Zhang, Z., Luo, P., Loy, C. C., and Tang, X. Facial landmark detection by deep multi-task learning. *Proc. of European Conference on Computer Vision (ECCV)*, Springer International Publishing. 2014, pp. 94-108.

[12] Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Proc. of Advances in Neural Information Processing Systems (NIPS)*. 2012, pp. 1097-1105.

[13] Lin, M., Chen, Q., and Yan, S. Network in network. arXiv:1312.4400, 2013.

[14] He, K., Zhang, X., Ren, S., and Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2015, 37 (9): 1904-1916.