Text Line Detection from Rectangle Traffic Panels of Natural Scene

Shiyuan Wang\textsuperscript{1,b}, Linlin Huang\textsuperscript{1,a} and Jian Hu\textsuperscript{1,c}

\textsuperscript{1}School of Electronics and Information Engineering, Beijing Jiaotong University, 100044, Beijing, China

E-mail: \textsuperscript{a}huangll@bjtu.edu.cn; \textsuperscript{b}15120030@bjtu.edu.cn; \textsuperscript{c}jhu@bjtu.edu.cn

Abstract. Traffic sign detection and recognition is very important for Intelligent Transportation. Among traffic signs, traffic panel contains rich information. However, due to low resolution and blur in the rectangular traffic panel, it is difficult to extract the character and symbols. In this paper, we propose a coarse-to-fine method to detect the Chinese character on traffic panels from natural scenes. Given a traffic panel Color Quantization is applied to extract candidate regions of Chinese characters. Second, a multi-stage filter based on learning is applied to discard the non-character regions. Third, we aggregate the characters for text lines by Distance Metric Learning method. Experimental results on real traffic images from Baidu Street View demonstrate the effectiveness of the proposed method.

1. Introduction
Traffic sign detection and recognition system is an important part of intelligent transportation systems. It plays an important role in drivers and pedestrian safety. While the detection and recognition of traffic sign are mainly focused on symbol, the detection of text within rectangular traffic panel is far less developed. However, text contained within rectangular traffic panel images can be of great semantic value, and so is an important step towards information retrieval. Text detection within rectangular traffic panel still remains a challenge in computer vision and has particularly important application value in the future transportation systems.

![Figure 1](image-url) Rectangular traffic panel (a) illumination effects (b) different of color, size, distribution etc

In the real situation, we get a variety of traffic panels that are taken in uncontrolled outdoor scenes. The various conditions such as illumination effects or weather conditions in the outdoor bring a huge challenge to our text detection. Traffic panels have a wide range of changes in color and size. Therefore many traditional methods have no effect on all traffic panels. Some samples with the small size and low resolution cause some text to blur. It is a huge obstacle in detecting text. In addition, the size and color of character is different, the distribution of characters in the traffic panel is not rule and
the distance between characters is also variation. These factors have a direct impact on text aggregation results. Figure 1 shows sample from our database.

2. Related work
The work of text detection has been studied by many scholars in recent years. It is mainly divided into texture-based methods [1], region-based methods [2] and edge-based methods [3]. Zhong et al. [4] used the spatial variance of horizontal direction to obtain the approximate position of the text, and then extracted the text by color segmentation in the color image. Anh-Nga Lai et al. [5] used the local Color quantization and K-means clustering to detection Korean text in the natural scene. The biggest problem of text detection using these methods is that the computational efficiency is low, and the method is reduced in the practical application.

There are some scholars who are studying the text within traffic panel. However English text is the main research language rather than Chinese text. Greenhalgh et al. [6] located many candidate regions by MSERs and saturation, hue and value color. Individual characters are grouped into text lines. X. Rong et al. [7] established a top-down network framework that was able to locate and identify text contained traffic panel in the natural scene. The authors build a two-stage cascade framework inspired by the You Only Look Once (YOLO) detector [8]. Yet the method of CNN has relatively high requirements to hardware and datasets.

3. Text line detection based on multi-stages
The diagram of the proposal method is shown in Figure 2. The region of interest (ROI) is character candidate region in Figure 2. We use multi-stage detection method in order to improve the detection speed as well as accuracy. First, we employ Color quantification (CQ) method to detect the character candidate regions, and discard the non-character candidate region by machine learning method. Second, we recall the lost character region. Finally, the candidate regions are grouped into text line by the Distance Metric Learning method [9].

![Figure 2. Text line extraction](image1)

![Figure 3. Character candidate region detection](image2)

3.1. Character candidate region detection
Due to the image with blurred and low resolution in our database, our proposed detection method must be able to solve these problems. Colour quantification (CQ) can solve the trouble of murky image and find the edge of the character. In order to save execution time and less memory, Octree structure is used. The diagram of character candidate region detection is shown in Figure 3.

CQ algorithm chooses K different colors as the initial color in the octree, and then sequentially reads the image data. When encountering a new color, the color is merged and the number of this color is stored. Finally, all the pixels are divided and have their own quantified categories. We get the different image layer according to the value of quantization. As shown in equation (1).

\[
g_k(x,y) = \begin{cases} 
255 & \text{if } l(x,y) == k \\
0 & \text{otherwise}
\end{cases}
\]

\(g_k(x,y)\) is the pixel value of \(x, y\) location in \(k\)th layer. \(k\) is the kinds of quantization, \(k = 1, \ldots, K\). \(l(x,y)\) represents the pixel value of \(x, y\) location in quantized image. We get many binary images in different layer. However, some layers of images do not contain characters; these layers will be abandoned by enacted rule. The rule is shown in equation (2).

\[
r_k = p_k/p
\]
\( r_k \) is the ratio in \( k \)th layer, \( p_k \) is the number of foreground point, \( p \) is the number of all pixels. When \( r_k \) < threshold, the \( k \)th layer is no longer used in the follow process. The multi-layers are shown in Figure 4. Surely, only partial layers are shown, these binary images are obtained according to equation (1). From the comparison of four images, we know each image contains the unique information and other images don’t have. In (a) and (c) graphics, we see that areas surrounded by these red boxes are special because they contain relatively clear character to compare to other layers in the same regions. Obviously characters in (b) are clearer than other layers. No characters exist in (d), so the binary layer will be discarded through the equation (2). We consider that the whole image is composed of different binary layers. In additional, the binary multi-layers are clearer and easier to detect character region. Finally, we employ connected components (CCs) method to extract character using all binary images.

![Figure 4](a) ![Figure 4](b) ![Figure 4](c) ![Figure 4](d)

Figure 4. Partial binary layers after quantization (a),(b),(c),(d) represent different layers

3.2. Multi-stage Filter

Candidate regions contain character, number and sign etc. but only the character is our detection target. We design a two-stage filter to discard the non-character candidate region. From Table 1, we choose the different feature and the classifier is Adaboost in the first filter, the result is shown. In order to extract more character regions, we apply the geometric features in the first filter. In first stage, we extract four parameters as the geometric features and use Adaboost classifier to classify. They are the ratio of width and height in the candidate boxes, the ratio of width in the candidate boxes and width in the panel images, the ratio of height in the candidate boxes and height in the panel images, the ratio of area in the candidate boxes and area in the panel images. We find this coarse filter is very efficient and the time-consuming is low. The obvious non-character areas are thrown away. But some number or the candidate regions that are similar to character region are still saved. The next stage will remove these candidate boxes.

| Feature     | Recall (%) | Precision (%) |
|-------------|------------|---------------|
| Geometric features | 90.3       | 30.1          |
| HOG         | 89.5       | 36.0          |
| Gabor       | 86.1       | 34.5          |
| LBP         | 84.2       | 38.7          |

Form Table 2, we know HOG feature has the good performance in recall rate. So we use the HOG feature to extract descriptors in the candidate regions and give the judgment by Adaboost classifier in second stage. As we all know, HOG feature can maintain a good invariance in the geometric and optical deformation of the image. The shape of character, number and sign is different; the main characteristic is employed to distinguish character and non-character. HOG feature shows superior performance in this stage.

| Feature | Recall (%) | Precision (%) |
|---------|------------|---------------|
| HOG     | 86.5       | 68.7          |
| Gabor   | 84.6       | 65.3          |
| LBP     | 82.8       | 70.6          |
3.3. Recall character regions
In the front of filter, some character regions must be misjudged as the non-character and be lost. So these character regions should be recalled to enhance the result in this situation. After filtering, each candidate boxes get the confidence. We employ the value of confidence to find lost character regions. Through our statistics and observation, characters are usually coming in the form of text lines. That is to say, we can find the character by this relation between characters. The diagram of recall character is shown in Figure 5.

![Figure 5. Recall Character regions](image)

The diagram shows the process of recalling character regions. \(C\) represents the confidence of character candidate regions. \(H_{TH}\) is the high threshold of confidence, \(L_{TH}\) is the low threshold of confidence. Region of interest (ROI) is also character candidate region. First, the confidence of ROI compares with \(L_{TH}\), if \(C < L_{TH}\), ROI is considered as non-character regions. If \(C > L_{TH}\) and \(C < H_{TH}\), we regard the ROI as character region. ROI will directly output. Surely, if \(C > H_{TH}\), we think ROI that is called seed region must be character region, and find the other lost character candidate regions surrounding it. If there are some lost ROIs in the left or right of seed regions, we judge the height between ROIs and the seed regions; if their height is almost the same, we think these ROIs are character regions and output to the next stage.

3.4. Text extraction on traffic panel
Through the previous steps, we have extracted character regions in the rectangular traffic panel. In the next steps, the extracted characters are merged into text lines. We apply the Distance Metric Learning method to set the condition of the aggregation, like the combined threshold between the characters and different row. The function of distance metric learning is defined as:

\[
    d(u, v; w) = w^T x_{u, v}
\]

\(u, v\) represent the character regions, \(d\) is the distance between \(u\) and \(v\). \(w\) is weight from the logistic regression. \(x\) is the feature vector and contains the relationship between \(u\) and \(v\). As follows:

\[
    d_x = \begin{cases} 
    \frac{abs(x_u - x_u - w_u)}{max(w_u, w_v)} & \text{if } x_u < x_u \\
    \frac{abs(x_u - x_u - w_u)}{max(w_u, w_v)} & \text{otherwise}
    \end{cases}
\]

\(d_{top} = \frac{abs(y_u - y_v)}{max(h_u, h_v)}\)

\(d_{down} = \frac{abs(y_u - y_v + h_u - h_v)}{max(h_u, h_v)}\)  \hspace{1cm} (5)

\(d_x\) is the adjacent distance, \(d_{top}, d_{down}\) are the different of alignment, \(d_w, d_h\) which are the difference of the width and height are obtained by equation (5) but the input is the width and height of \(u, v\). We use the weight of feature vector to obtain the \(d(u, v; w)\). In order to get the parameter of \(w\) in the logistic regression, we choose the maximum distance in the same row and the smallest distance in the adjacent row. If \(d(u, v; w)\) is less than zero, the \(u\) and \(v\) must be merged. The merger should be stop until no character needs to be merged. The result is shown in Figure 6.

![Figure 6. The result of text line extraction](image)

4. Experimental results
We evaluated our proposed method and faster R-CNN\cite{10} of the text detection on our dataset. The experiments were performed on computer with cpu i7 and GPU 1080Ti. Our dataset contains 1073 training images and 436 test images. The size of the rectangular traffic panel varies between 50×27 and 330×163 pixels. These images contain 11962 training characters and 5114 test characters, 4148 training strings and 1696 test strings.

First, we estimate Chinese character extraction. The result of Chinese character detection depends on the Intersection over Union (IoU) between our detection region and ground truths. We set the Threshold of overlap is 0.7. Chinese character detection results are given below.

| Method          | Recall (%) | Precision (%) |
|-----------------|------------|---------------|
| Our method      | 88.63      | 95.81         |
| Faster R-CNN    | 76.31      | 86.47         |

From Table 3, we know that we acquire the relatively good result in the Chinese character detection. The precision reachs up to 95.81% and the recall is 88.63%. At the same time, we also apply IoU to evaluate Chinese text line extraction. The IoU is also 0.7. The Chinese text line extraction results is shown as Table 4, the result also proves that our proposed method is efficient.

| Chinese text line detection results |
|-----------------------------------|
| Text line                         |
| Recall (%)                        |
| Precision (%)                     |
| 89.83                              | 93.24 |

5. Conclusion
In this paper, we propose an effective approach for detecting text in Chinese rectangular traffic panel image using multi-layered detection method based on Octree Quantification on our database; the proposed method has a good performance for rectangular traffic panel text extraction, which obtains a higher precision 93.24%. Compared to Faster R-CNN, the database is relatively small, so the deep learning result is not particularly good. But from the small dataset, the experimental results show the feasibility of our proposed algorithm.

Acknowledgements
This work was supported in part by the National Natural Science Foundation of China (NSFC) Grant 61271306.

References
[1] Yin, X. C., et al. Robust Text Detection in Natural Scene Images. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2013, 36 pp 1-1.
[2] Yi, C., Tian, Y., Arditi, A.: Portable camera-based assistive text and product label reading from hand-held objects for blind persons. IEEE/ASME Trans on Mechatronics. 2014, 19 pp 808-817.
[3] C. Yao, X. Bai, W. Liu, Y. Ma and Z. Tu. Detecting texts of arbitrary orientations in natural images. Computer Vision and Pattern Recognition, 2012, 157 pp 1083-1090.
[4] Zhong Y, Karu K, Jain A K. Locating text in complex color images. Proceedings of International Conference on Document Analysis and Recognition (ICDAR), 1995, 1 pp 146-149.
[5] Lai, Anh Nga, et al. Korean Text Extraction by Local Color Quantization and K-means Clustering in Natural Scene. Intelligent Information and Database Systems, 2009, 200 pp 138-143.
[6] Greenhalgh J, Mirmehdi M. Recognizing Text-Based Traffic Signs. IEEE Transactions on Intelligent Transportation Systems, 2015, 16 pp 1360-1369.
[7] Rong, Xuejian, C. Yi, and Y. Tian. Recognizing Text-Based Traffic Guide Panels with Cascaded Localization Network. European Conference on Computer Vision, 2016, pp 109-121.
[8] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. CVPR, 2015 pp 779-788.
[9] Nguyen, Bac, C. Morell, and B. D. Baets. Large-scale distance metric learning for k-nearest neighbors regression. Neurocomputing, 2016 214 pp 805-814.
[10] Ren, Shaoqing, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017 39 pp 1137.