2010 Second WRI Global Congress on Intelligent Systems

License Plate Location Based on Haar-like Cascade Classifiers and Edges

Yuanxing Zhao[1,2,3], Jing Gu[1,2], Chui Liu[1,2], Shumin Han[1,2], Yong Gao[1,2], Qingmao Hu[1,2]
1. Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China
2. The Chinese University of Hong Kong, Hong Kong
3. Hebei University of Technology

Abstract—**License plate recognition is a core module for intelligent transportation systems, while license plate location is an important part of it. Haar-like cascade classifier is good for face detection, but its application to license plate localization largely depends on selection of positive and negative samples. In this paper we studied on how to choose good samples for Haar-like cascade classifiers and image post-processing methods to achieve good location results. It is hoped that the study could be useful to guide sample preparation for other object detection using Haar-like cascade classifiers.**

Keywords: ***license plate location; Haar-like cascade classifier; sample preparation; image post-processing***

I. INTRODUCTION

License plate (LP) has been widely used to validate cars, therefore LP recognition (LPR) is one of the important parts in intelligent transportation, used in traffic monitoring, stolen car detection, portal controlling and so on. As such, many researchers have paid close attention to this field.

The main style of Chinese LPs is blue plate. Its width is 440mm and height is 140mm. It usually has 7 characters and a dot between the second and third characters from left. Although the geometry of an LP is stable, the complex environment of image acquisition can greatly change its image appearance.

Many problems have to be solved to develop a robust LPR system, some of which are listed below:

- Plate variations: there are more than 5 styles of LPs in China. They have different colors, sizes, character layouts, and adornments.
- Environment variations: different illuminations and weather conditions may change the quality of input image greatly and plate-like backgrounds can make localization difficult.
- Image variations: image collecting devices can produce images with different properties, such as different resolutions and different noises.

The rest of this paper is organized as follows. Related works about LP location are reviewed in section II. Our LP location algorithm is presented in section III. Experimental results are provided in section IV and conclusions in section V.

II. PREVIOUS WORK

In the past decades, many methods have been proposed for LP location. In this section, we will briefly review such methods.

Approaches for detecting LPs from images can be classified into two categories: image processing based methods and machine learning based methods.

A. Image processing methods

1) Edge based method: One of the significant features in plates is that they always have rich edge information. Techniques based on edge features [1] would have good results if the input images have high quality and do not have complex background like plates.

2) Color based method: Some methods based on color information have been considered [2] as LPs only have a few specified colors. These methods can have good results if illumination is desirable and the quality of images is high.

The techniques based on image processing are easy to implement, but they are sensitive to changes in environment. Therefore a robust LP location algorithm should not be merely based on the techniques described above.

B. Machine learning methods

1) Haar-like feature + AdaBoost: the method is originated from face detection [3] and has already been successful in some other object detection fields. Some researchers have already used Haar-like cascade classifiers to detect LPs. In [4], Sun et al compared the performance of Discrete AdaBoost, Real AdaBoost, and Gentle AdaBoost on LP location. In [5], Wu et al used edge information to remove large number of areas in images, then put the rest of areas into classifiers to produce candidate regions. The method showed an encouraging performance but having problems of choosing a good threshold for edge images with different qualities. In [6], *Dlagnekov* used 1500 images for both training and testing, using enhanced Haar-like features (derivative and variance of the image instead of the original image itself).

2) Color-texture feature + support vector machine (SVM): in [7] color textural patterns are fed directly to the SVM to derive a classifier for LPs which is then used to generate an image where each pixel represents the possibility of being part of a plate region.

The methods based on machine learning take object detection as a pattern recognition problem with two classes. They collect great number of samples to extract stable features, then use such features to train a classifier. At the location phase, features are extracted from the input image and put into classifier to decide if an object is detected. These methods are robust to changes in environment.

978-0-7695-4304-8/10 $26.00 © 2010 IEEE
DOI 10.1109/GCIS.2010.55

From the review we can see that lots of existing location methods can work well in special conditions while the performance will drop sharply when images are degraded.

III. THE FRAMEWORK OF THE ALGORITHM

In this section we will describe of our algorithm in detail. The flow chart is given in Fig .1.

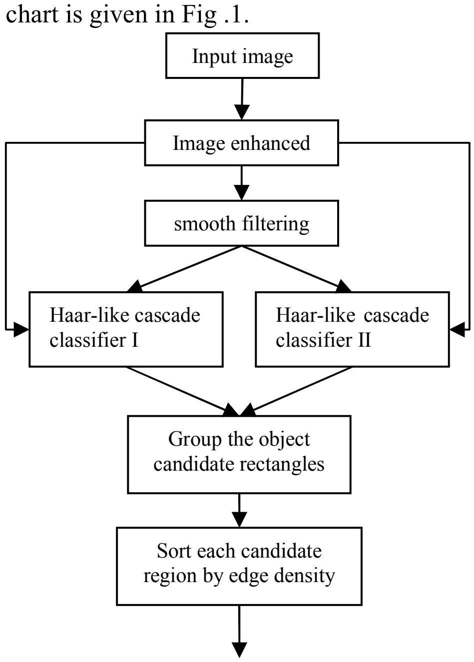


Figure 1. The flow chart of the algorithm.

A. *Image preprocessing*

The original color image is converted to gray as we do not use color information. To speed up the algorithm we resize the image to 1/4 of the original size. Then the histogram of the image is equalized to improve its contrast. Gaussian filter is applied to remove small detailed information.

B. *Haar-like cascade classifier*

Haar-like cascade classifier was first used in face detection. A number of Haar-like features in a default window are extracted. In our experiment we use the 15 features described in [8]. The exhaustive set of Haar-like features in a default windows is over-complete and much more than the number of pixels.

AdaBoost is an effective classifier to train from great number of features. It selects a small number of efficient features to build a weak classifier, stump classifier or classification and regression tree classifier. The process of building a stump weak classifier is described in table 1.

As described above we can see that the weak classifier largely depends on the selection of positive and negative samples. So how to choose samples is a core problem in AdaBoost's training.

TABLE I. CHANGE WEAK CLASSIFIER

- Put positive and negative samples together.
- Randomly choose a Haar-like feature, and calculate the feature value for each sample.
- Sort the samples by value.
- Choose a threshold to best divide positive and negative samples. We build a stump weak classifier in this way.
- If the detection rate of the classifier is more than 0.5, we used it to build a strong classifier.

Weak classifiers are then combined to form a strong classifier. The structure of a strong classifier is shown in Fig .2

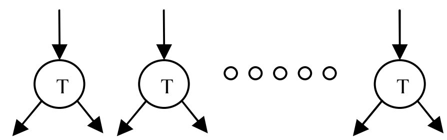


Figure 2. The structure of AdaBoost.

To reduce computation time in the detection stage a cascade structure is adopted. The total time is reduced radically by processing large areas of the input image via simple classifiers and processing the rest of the input image via complex classifiers. The structure of cascade classifier is shown in Fig .3.

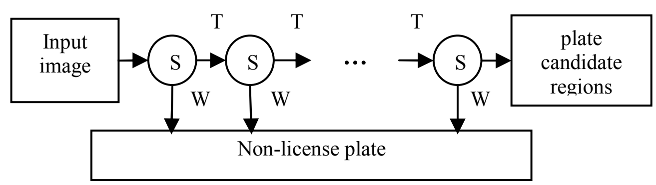


Figure 3. Cascade classifier structure.

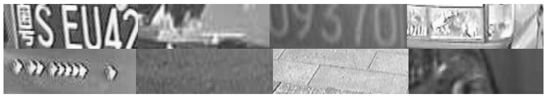
In our experiment we use two Haar-like cascade classifiers to form an ensemble classifier. Each classifier can detect more than one plate in an image. The two classifiers were trained using the same positive set with different negative set and different train parameters. In this way the accuracy was improved.

C. *Image post-processing*

There may be more than one candidate region outputted by a classifier. Some candidate regions outputted by a classifier are shown in Fig. 4. In this part we use some image processing methods with some prior knowledge to choose the best candidate region.

As Fig. 4 (b) shown above, the features of an LP can be determined as follows:

- The area of LPs is moderate. Here we defined that the width is larger than 50 pixels and smaller than 1/3 of image's width.
- The aspect ratio of the region (width/height) should be within a given range (2 to 3.5).



(a) False candidate region.

(b) True candidate region.

Figure 4. Examples of image windows.

- Different parts of the region of an LP may be chosen by classifiers many times.
- The region of an LP has rich vertical edge information.

TABLE II. THE ALGORITHM TO SORT CANDIDATE REGIONS

- The vertical edge of the candidate region is computed using Sobel operator.
- Calculate the area of each candidate region S
- Calculate the density of vertical edge in each region by

$$D = \frac{\sum_{R} D_V}{S}$$

Where D_v is the value in a candidate region of vertical edge image. D is the vertical edge density in the candidate region.

- Normalize each candidate region's area and density to [0,1]
- The final principle is

$$P=D+S$$

The larger the P is the likely the region is an LP.

First, LP candidate regions which do not conform to the first and second principles are removed.

Then candidate regions whose distance to each other lower than a threshold are merged, the distance was defined by (1).

$$D=(\min\{w_1,w_2\}+\min\{h_1,h_2\})*0.5. \qquad (1)$$

where D is the distance of two candidate regions, w_1 and w_2 are the width of the two regions, and h_1 and h_2 are their height, respectively. We choose 0.2D as the threshold to merge regions. If the difference of corresponding points between two rectangles is smaller than 0.2D, they would be thought as belonging to the same rectangle. Then an average region is derived by averaging their 4 vertices' coordinates.

At last we sort the remaining regions by the fourth principle. Because the false candidate regions such as Fig.4 top-left of (a) also have rich vertical edge information, we consider the area of the candidate region too. We define the sort principle in table 2.

IV. EXPERIMENT

A. Data set

Training and test data sets are selected frames from cameras of different parking lots and road crossings in China. Fig. 5 shows two typical frames captured from those devices. These frames have a spatial resolution of 720×576.

Figure 5. Two typical images used in experiment.

1) Positive samples

How to choose positive samples from all possible situations and achieve least information redundancy in such a large dataset is a big challenge in our experiment. In order to select data randomly we picked positive samples as follows:

- Put all images together.
- Sort mage names in a vector.
- Use a uniform distribution random number generator to randomize the images and repeat 3 times.
- Use another random number generator to select subscripts randomly. The one which has been selected was removed in the vector. Repeat this process 7000 times to build a training set.

2) Negative samples

We build the negative samples by randomly selecting diverse images from the internet.

3) Test set

We randomly selected 5000 images from the rest of images set after selected the positive set.

B. Result

The following situations have been evaluated:

- E1: put the primitive image and the image smoothed by Gaussian filter into the two classifiers which were trained by our positive and negative sample sets described above. Then use the method described in table 2 to find the region which is the best like LPs.
- E2: only put the primitive image into the two classifiers which were trained by our positive and negative sample sets described above. Then use the method described in table 2 to find the regions which is the best like LPs.
- E3: put the primitive image and the image smoothed by Gaussian filter into the two classifiers which were trained by our positive and negative sample sets

described above. Then combine the candidate regions by (1) without other image post-processing.

- E4: put the primitive image and image smoothed by Gaussian filter into the two classifiers which were trained by the positive sample set randomly selected from the dataset without randomizing the negative sample set. Then use the method described in table 2 to find the region which is the best like LPs.
- E5: put the primitive image and image smoothed by Gaussian filter into the two classifiers which were trained by the negative sample sets built from the primitive images with the LP removed and our positive sample set. Then use the method described in table 2 to find the region which is the best like LPs.
- E6: we explored a new method based on vertical edge density to locate license plates in the input images. First, the vertical edge of the image is computed using Sobel operator. Then, some pixels whose value is below a threshold would be eliminated. At last, using a box filter to form some region, the region whose geometry property is the same as license plates would be chose as a license plate.

The true and false positive rates were shown in table 3.

TABLE III. EXPERIMENT RESULTS

	true positive rate	**False positive rate**
E1	90.8%	5.9%
E2	81.5%	12.0%
E3	85.8%	13.8%
E4	86.8%	9.8%
E5	82.8%	15.7%
E6	67.3%	32.5%

C. *Failure classification*

The accuracy rate of our experiment is 89.5%. Some of the fail cases are shown in Fig. 10.

Figure 6. False candidate region.

We put the failure classification into negative sample set and retrain the classifier. The number of failure classification can be reduced.

V. DISCUSSION AND CONCLUSION

This paper explored how to use AdaBoost for a strong classifier to detect LPs, and demonstrated that Haar-like cascade classifier is an efficient method in LP location.

The key is the selection of positive and negative samples. Here we use random generator to randomize dataset's distribution and select samples randomly in the dataset. In this way we can get a positive dataset representing possible situations and achieved least information redundancy. Negative samples were selected randomly from the internet. The classifier trained by the positive and negative datasets described above is better than the classifier trained from samples without randomization (Table III).

Our experience shows that the proposed method based on machine learning is better than the method based on image processing, and combing appropriate image post-processing with some priori knowledge can improve the classifier's accuracy.

Our algorithm can be used in situations with complex background as well as degraded LP images.

We noticed that when the plates have a big angle with camera, the classifier may fail. Sometimes though the classifiers have detected the LP region, the aspect ratio of the region (width/height) will reject to cause failure. To handle the big angle, the range of aspect ratio needs to be enlarged and image processing should be explored.

We also noticed that when the lamps of the car were turned on at night, the contrast of the LP would be low, and the detection may fail.

Our future work involves putting other features which may go beyond Haar-like features, using other image processing methods to further improve the accuracy, and exploring other methods to choose samples for training classifiers.

REFERENCES

[1] D. Zheng, Y. Zhao, and J. Wang "An Effidient method of license plate location," Pattern Recognition Letter 26 (2005) 2431-2438.

[2] X. Shi, W. Zhao, and Y. Shen, "Automativ License Plate Recognition System Based on Color Image Processing," O. Gervasi et al., Ed. New York: Springer-Verlag, 2005, vol. 3483, pp. 1159-1168.

[3] P. Viola and M. J. Lee, "Robust real-time face detection," Int. J. Comput. Vis., vol. 57, no. 2, May 2004, pp. 137-154.

[4] J. Sun, D. Cui, D. Gu, H. Cai, and G. Liu, "Empirical Analysis of AdaBoost Algorithms on License Plate Detection," Proceedings of the 2009 IEEE International Conference on Mechatronics and Automation, August 9-12,Changchun, China, pp. 3497-3502.

[5] Q. Wu, H. Zhang, W. Jia, X. He, J. Yang, and T.Hintz, "Car plate detection using cascaded tree-style learner based on hybrid object features," in Proc. IEEE Int. Conf. AVSS, Sydney, Australia, 2006, pp. 15-20.

[6] L. Dlagnekov, "License Plate Detection Using AdaBoost," Mar. 2004, San Diego, CA: Comput. Sci. Eng. Dept. , Univ. California, San Diego. [Online]. Available: http://www.cse.ucsd.edu/classes/fa04/cse252c/projects/louka.pdf

[7] K. I. Kim, K. Jung, and J. H. Kim, "Color Texture-Based Pbject Detection: An Application to License Plate Localization," S.-W. Lee and A. Verri, Eds. New York: Springer-Verlag, vol. 2338, pp. 293-309.

[8] R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," In Proc. Of the IEEE Computer Society Conf. On Computer Vision and Pattern Recognition (2002), pp. 900-903.