

© [2006] IEEE. Reprinted, with permission, from [Huaifeng Zhang; Wenjing Jia; Xiangjian He; Qiang Wu, A Fast Algorithm for License Plate Detection in Various Conditions, Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on (Volume:3 ), 11 Oct. 2006]. This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Technology, Sydney's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to [pubs-permissions@ieee.org](mailto:pubs-permissions@ieee.org). By choosing to view this document, you agree to all provisions of the copyright laws protecting it

# A Fast Algorithm for License Plate Detection in Various Conditions

Huaifeng Zhang, *Member, IEEE*, Wenjing Jia, *Student Member, IEEE*,  
Xiangjian He, *Senior Member, IEEE*, and Qiang Wu, *Member, IEEE*

**Abstract**—This paper proposes a fast algorithm detecting license plates in various conditions. There are three main contributions in this paper. The first contribution is that we define a new vertical edge map, with which the license plate detection algorithm is extremely fast. The second contribution is that we construct a cascade classifier which is composed of two kinds of classifiers. The classifiers based on statistical features decrease the complexity of the system. They are followed by the classifiers based on Haar-features, which make it possible to detect license plate in various conditions. Our algorithm is robust to the variance of the illumination, view angle, the position, size and color of the license plates when working in complex environment. The third contribution is that we experimentally analyze the relations of the scaling factor with detection rate and processing time. On the basis of the analysis, we select the optimal scaling factor in our algorithm. In the experiments, both high detection rate (with low false positive rate) and high speed are achieved when the algorithm is used to detect license plates in various complex conditions.

## I. INTRODUCTION

LICENSE plate detection has been adopted widely into numerous applications such as unattended parking, security control of restricted areas and stolen vehicle verification. Because of ambient lighting conditions, image perspective distortion, interference characters, etc., it is difficult to efficiently detect license plates in complex conditions. Most of previous license plate detection algorithms are restricted to work under certain conditions, such as fixed backgrounds [1], known color [2], or designated ranges of the distance between camera and vehicle [3, 4].

In the previous years, some researchers have been working on license plate detection in complex conditions. Chang et al. [5] proposed a license plate detection algorithm using color edge and fuzzy disciplines. Their algorithm can only be used to detect license plates with specific colors. In [6], Matas and

Zimmermann proposed an algorithm to detect license plates and road signs in various conditions. They used character regions as basic units of license plate, which makes their algorithm robust to viewpoint and illumination. However, this algorithm can hardly distinguish interference characters from the true license plates. Kim et al. [3] proposed another license plate detection algorithm using both statistical features and license plate templates. After the statistical features were used to select the Regions of Interest (ROI), license plate templates were employed to match the ROI. In most cases, however, general license plate templates are very difficult to be constructed. Moreover, their algorithm can only work on a fixed scale. Hence the application of this algorithm is extremely restricted.

Recently, Haar-like features were widely used for object detection. A set of Haar-like features can represent the interior structure of an object invariant to some transformations. The classifiers [7, 8] based on Haar-like features can detect objects from complex background independent of the variance of the color, the illumination, the position and the size of the objects. However, one of the problems of these algorithms is that a large account of features were included in the classifier, which makes the system very complex and unstable. Chen and Yuille [9] constructed a simple cascade classifier for text detection using statistical features. Their algorithm could detect text regions in various natural scenes. However, only statistical features were used in their algorithm, which always results in high false positive rate in practice.

In this paper, we propose a fast algorithm for detecting license plate in various conditions. Both statistical features and Haar-like features are used in our algorithm. The classifiers based on statistical features are trained through simple learning procedures. Then AdaBoost learning procedure [10] is used to select important Haar-like features and construct classifiers. The final cascade classifier is obtained by combining the above two kinds of classifiers. The classifiers based on statistical features decrease the complexity of the system. They are followed by the classifiers based on Haar-features, which further improve the detection rate and lower down the false positive rate.

Vertical edge map has been used for license plate detection for many years [1, 11, 12]. The previous algorithms used a one-direction Sobel operator to generate the vertical edge map. However, a large amount of nearly-horizontal edges are kept in such vertical edge map. In this paper, we propose a new vertical edge map which only keeps the edge points with

Manuscript received March 30, 2006. This work was sponsored by the Australian Research Council under Large ARC-Discovery Grant DP0451666.

Huaifeng Zhang is with the Faculty of Information Technology, University of Technology, Sydney, PO Box 123, Broadway, NSW, 2007, Australia (phone: +61-2-9514 4771; fax: +61-2-9514 4535; e-mail: hfzhang@it.uts.edu.au).

Wenjing Jia is with the Faculty of Information Technology, University of Technology, Sydney, PO Box 123, Broadway, NSW, 2007, Australia (e-mail: wejia@it.uts.edu.au).

Xiangjian He is with the Faculty of Information Technology, University of Technology, Sydney, PO Box 123, Broadway, NSW, 2007, Australia (e-mail: sean@it.uts.edu.au).

Qiang Wu is with the Faculty of Information Technology, University of Technology, Sydney, PO Box 123, Broadway, NSW, 2007, Australia (e-mail: wuq@it.uts.edu.au).

vertical edge angles. Using the new vertical edge map, our algorithm eliminates more background regions from further detecting, which makes the algorithm very fast.

Our detection algorithm works on multiple scales to detect objects with different sizes. In multi-scale object detection, the size of the mask which goes through the image varies along the changes of the scaling factor. The scaling factor affects the performance of object detection greatly. In this paper, we quantitatively analyze the relations of the scaling factor with detection rate and average processing time. On the basis of the analysis, an optimal scaling factor is selected in our algorithm.

The rest of the paper is organized as follows. The framework of our algorithm is introduced in Section II. Vertical gradient map and two statistical features, Gradient Density and Density Variance, are defined in Section III. Then Haar-like features and AdaBoost algorithm are described in Section IV. Experimental analysis and results are presented in Section V. The paper is concluded in Section VI.

## II. THE FRAMEWORK OF THE ALGORITHM

In this paper, we construct a cascade classifier [8] to increase the detection speed, in which the first two layers are based on statistical features and the following layers are based on Haar-like features. In this section, we introduce the algorithm in two aspects: testing and training.

### A. Training

Positive samples and negative samples are needed in the training procedure. The positive samples are obtained through labeling the license plate regions from the vehicle images. The negative samples are randomly extracted from different images which do not contain any license plate. All of the samples are scaled to 48\*16 for the convenience of training. We compute the vertical edge maps on both positive samples and negative samples before extracting any features.

All the samples are input to the training algorithm. Firstly, for all the samples, the values of one of the statistical features, called Gradient Density, are calculated. A classifier is obtained by selecting the threshold which will classify all the positive samples as positive ones. All the samples, including the positive samples and negative samples, that are classified as positive ones are used to train the classifier of the second layer. This classifier is based on another statistical feature, called Density Variance. The input samples are classified again and the positive ones obtained in this layer are used to train the classifier of the third layer. Similarly, the samples classified as positive ones by the third layer are input to the fourth layer, and so on. The training finishes when the given false positive rate is reached. In our algorithm, we trained four layers of classifier based on Haar-like features and AdaBoost leaning procedures, which is the layer 3 through layer 6 in the final cascade classifier.

### B. Testing

When an image is input into the classifier, its vertical edge

map is computed first. Then a mask of 48\*16 is used to capture the same size of pixel block on the image. This mask will go through the whole image area. At each position, the cascade classifier is used to verify if the block covers a license plate. A cascade classifier can be taken as a degenerate decision tree as shown in Fig. 1. A positive result from the classifier on an upper layer triggers the classifier on the next layer. A negative outcome at any layer leads to an immediate rejection of the block. Then it slips to the next position and the same procedure is repeated.

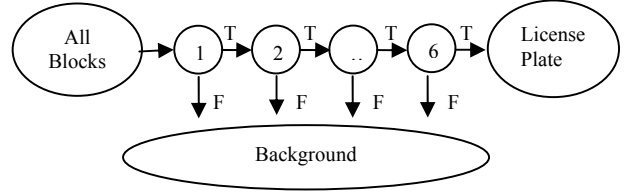


Fig. 1 The working flow of cascade classifier, where 1,2,...,6 represent the layers.

The detection is implemented on multiple scales. In order to detect license plates of variant sizes, the mask size is scaled up from 48\*16 to 240\*80. Suppose the scaling factor is 1.5, the mask of second scale is (48\*1.5) \* (16\*1.5), that is, 72\*24. And the mask of third scale is (72\*1.5)\*(24\*1.5), and so on.

## III. STATISTICAL GRADIENT FEATURES

Statistical analysis shows that the regions of license plates have some common characteristics. Firstly, a license plate region usually contains rich edge information. Secondly, most of the edges are vertical edges [12]. In this research, we assume the license plates fixed horizontally on vehicles. However, the view angle can vary in a scope. Thirdly, the edges are distributed relatively uniformly in a license plate region.

Based on the observation, we define two statistical features of the block of a license plate. Both of the features are constructed from the vertical edge map of the original image.

To simplify our algorithm, the gradient information is investigated rather than edge information because an efficient general-purpose edge detector is usually difficult to obtain in practice.

### A. Vertical Gradients

The Sobel operator is one of most widely used operators for computing gradients of an image. It uses two masks in a 3 by 3 region to compute the partial-derivatives of the central pixel on x-direction and y-direction respectively. The two masks are

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

The previous vertical edge map based license plate

detection algorithms [1, 11, 12] used  $S_x$  to calculate the vertical edges. When such vertical gradient map is used, not only vertical gradients but also a lot of nearly horizontal gradients are obtained.

In our algorithm, whether a gradient is vertical or not is judged by its edge angle. Suppose  $G_x$  is the gradient obtained from  $S_x$ ,  $G_y$  is the gradient obtained from  $S_y$ , respectively. The edge angle is defined as

$$\theta = \arctan \frac{G_x}{G_y} \quad (1)$$

In our algorithm, pixels with edge angles more than 45 degree and less than 135 degree are kept.

We use an example to illustrate the difference between our algorithm and the previous algorithms. Suppose a 3 by 3 image region, centered at image point  $C$ , has the following intensities:

$$\begin{bmatrix} 20 & 100 & 100 \\ 100 & C & 100 \\ 20 & 20 & 20 \end{bmatrix}$$

Obviously the edge around image point  $C$  is not a vertical one. However, using previous algorithms, the vertical gradient of  $C$  is computed with x-direction Sobel operator:

$$G_x = (100 + 100 \times 2 + 20) - (20 + 100 \times 2 + 20) = 80$$

In our algorithm, because the edge angle of image pixel  $C$  is

$$\theta = \arctan \frac{G_x}{G_y} = \arctan \left( \frac{80}{240} \right) = 18^\circ < 45^\circ,$$

the vertical gradient of image pixel  $C$  is zero no matter how big its gradient magnitude is.

The different gradient maps obtained from two algorithms are shown in Fig.2. From the figure we know, our vertical gradient algorithm eliminates more background information than the previous algorithm, while keeping most of the information of license plate regions.

### B. Gradient Density

The gradient density in a block is used to describe the edge density of the block using

$$D_G = \frac{1}{N} \sum_i \sum_j G(i, j) \quad (2)$$

where  $G(i, j)$  represents the gradient magnitude at location  $(i, j)$  and  $N$  is the number of pixels in the block.

The Sobel gradient operator is employed to produce gradient map, where the resulted gradient magnitudes are normalized by the maximum gradient strength in the image.

During the training procedure, the size of the block is fixed to the size of the sample images, which is 48\*16. During the testing procedure, the size of the block is changed depending on the scale of the searching block.

### C. Density Variance

Besides the abundant edge information, note that the foreground characters in a license plate are usually distributed with relatively even interval. As its consequence, the gradient in the block of a license plate is distributed more evenly in space with similar strength, compared to most of the areas with simple structures. Fig. 3 gives such an example.

Therefore, we modify and redefine the density variance feature [13] in order to discriminate license plates from background regions.

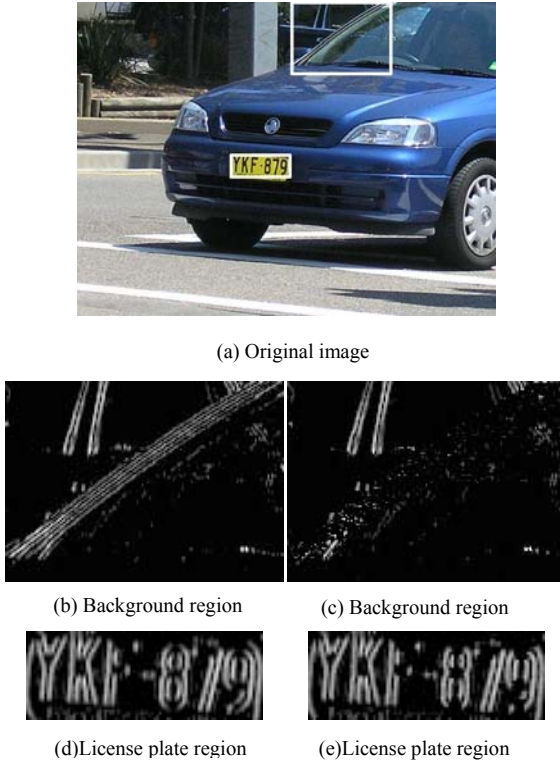


Fig. 2. The original image (a) and the vertical edge maps (b) through (e) of the two marked regions in (a), where (b) and (d) are obtained from previous algorithms. (c) and (e) are obtained from our algorithm

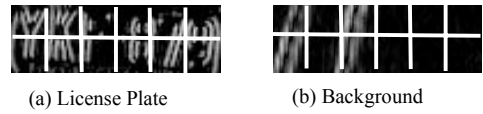


Fig. 3. Areas with different gradient distributions

To obtain the feature, a block is divided into 12 equal-sized sub-blocks, as shown in Fig. 3. Let  $g_i$  denote the mean value of the gradient strength at sub-block  $i$ , and  $g$  denote the mean value of the gradient strength of the whole block. Then, the density variance of the block, denoted as  $V_G$ , is defined as

$$V_G = \frac{\sum_{i=1}^n |g_i - g|}{n \cdot g} \quad (3)$$

where  $n$  is the number of the sub-blocks, e.g.,  $n = 12$  in above example.

The above defined density variance, which takes value

from 0 to 1, is a ratio to the mean gradient strength of the block. In this way, no matter whether the gradient is strong or weak, the density variance keeps low as long as there are similarly strong or weak gradient distributed evenly through the block.

#### IV. HAAR-LIKE FEATURE AND ADABOOST

The Haar-like features originate from Haar basis functions. They consist of a number of rectangles covering adjacent image regions (see Fig. 4). The value of a Haar-like feature is the difference between the average of the pixel values (in our algorithm, the gradient magnitude) in white rectangles and grey rectangles.

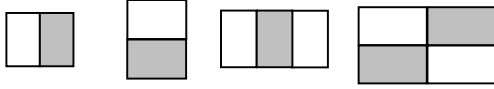


Fig. 4. Four types of Haar-like features

A Haar-like feature is determined by its type, the size and the position of the rectangles. The size and the position can be any as long as the feature is in the image block. Such Haar-like features dictionary can capture the interior structure of objects that are invariant to certain transformations. However the number of the features is too large in this features dictionary, e.g. there are hundreds of thousands features in a  $48 \times 16$  image block. It is prohibitively time-consuming to compute all the features.

AdaBoost algorithm [10] is a good choice to select a small number of features from a very large number of potential features. The classifier trained through the AdaBoost algorithm is the combination of a set of simple classifiers (called weak classifier), where each simple classifier uses one feature. The construction of weak classifier is independent of AdaBoost algorithm. In our algorithm, perceptron [14] is selected as the weak classifier, in which the classifying threshold is determined by the given detection rate.

The basic idea of the AdaBoost algorithm is as follows. After constructing a weak classifier, the samples are re-weighted in order to emphasize those which are incorrectly classified. Then the next weak classifier is trained with the re-weighted samples. A number of weak classifiers are trained in this way till the given false positive rate is reached. The final classifier (called strong classifier) is constructed by combining these weak classifiers using a set of weights. These weights are determined by classification error of each weak classifier. For the details of AdaBoost algorithm, please refer to [10].

#### V. EXPERIMENTS

We use 460 vehicle images in our experiments. 300 images are taken as training images, in which there are 305 visible license plates; the other 160 images are testing images, in which there are 169 visible license plates. The images used in

our experiments were taken in various circumstances with various illuminations and view angles. The colors and the styles of the license plates are different. Some examples of the license plates are shown in Fig. 5.



Fig. 5. Some examples of the license plates used in our experiments

The negative samples used to train the classifiers based on statistical features are collected by randomly selecting 28,000 sub-windows from 50 images which do not contain any license plate. The negative samples used in AdaBoost learning procedure are obtained from the incorrectly classified samples which are randomly extracted from 220 images that do not contain any license plate.

In the experiments, a six-layer cascade classifier is obtained. Each of the first two layers uses the two statistical features defined in Section III respectively. On the last four layers, the numbers of the features in the strong classifiers are 19, 34, 47 and 58 respectively. So our final cascade classifier has 6 layers and uses 160 features. This classifier is much simpler than Viola's classifier which has 38 layers and uses 6060 features [8].

In Section II we show that our detection algorithm works on multiple scales. The sizes of searching mask are scaled up from  $48 \times 16$  to  $240 \times 80$ . How to select the scaling factor is a problem related to the algorithm speed and the detection rate. In this paper, we studied the relations of scaling factor with detection rate and average processing time (shown in Fig. 6). The scaling factor varies from 1.01 to 1.5 in this experiment. Please note that, in the experiment, the false positive rate varies a little regarding to the variance of scaling factor. So we ignore the effect of the false positive rate when selecting the scaling factor. From Fig. 6 (a) we can see that the detection rate decreases when the scaling factor increases. And the decreasing rate (the slope of the curve) increases when the scaling factor increases. When the scaling factor is bigger than 1.2, the detection rate decreases sharply. So the scaling factor bigger than 1.2 is not recommended. From Fig. 6 (b) we know, the average processing time increases when the scaling factor decreases. When the scaling factor is smaller than 1.05, the processing time becomes much longer. In our algorithm, vertical edge map and statistical features can both speed up the system. Even when the scaling factor is 1.05, the average processing time is about 100ms, which meets the requirement of our detection system. So, in our algorithm, the scaling factor is set to 1.05.

In the experiments, among the 169 visible license plates in 160 testing images, 163 license plates are detected, with



detection rate 96.4%. At the same time, there are only 8 false positive regions. On a PC with Pentium 2.8GHz CPU, the detector can process a 648\*486 image in about 100ms.

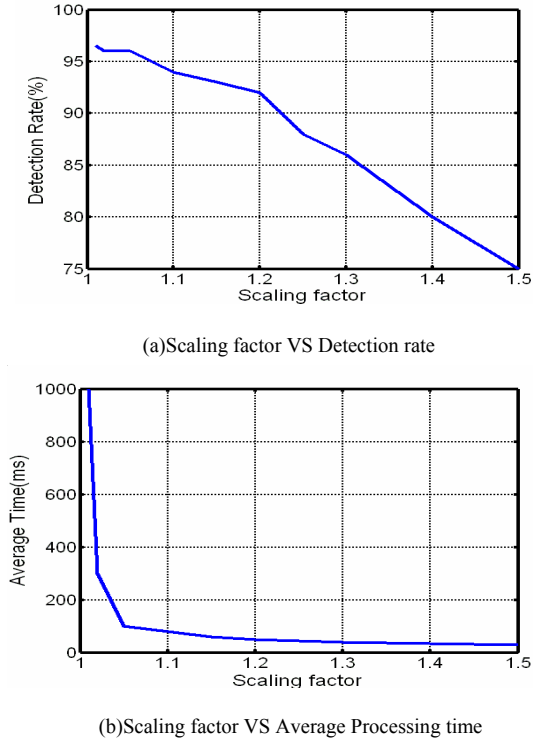


Fig. 6. The relations of scaling factor with detection rate and average processing time

Fig. 7 shows some of the detection results, where the license plates are circled by the boxes colored red and white. Fig. 7(a) and Fig.7(b) are the examples with complex background; Fig.7(c) and Fig.7(d) give the results of detecting multiple license plates in one image; From Fig. 7(e) and Fig.7(f), we can see that the algorithm can be used to detect license plates taken from very different view of angles; Fig. 7(g) and Fig. 7(h) are the examples of detecting license plates with different colors. Fig. 7(i) and Fig.7(j) are the results of license plate detection against interference characters; Fig.7(k) and Fig.7(l) show that the algorithm are robust to the variance of the illumination. From the examples, we may see that our algorithm can work under various background, various illuminations, and various view angles. The algorithm can detect the license plates with various sizes, positions and colors.

## VI. CONCLUSIONS

In this paper, we construct a cascade classifier for license plate detection. The classifier uses both statistical features and Haar-like features, which makes it possible to detect license plate from various complex background. The classifiers on the first two layers are based on statistical features. They can exclude more than 80% background regions from further training or testing. The classifiers on the

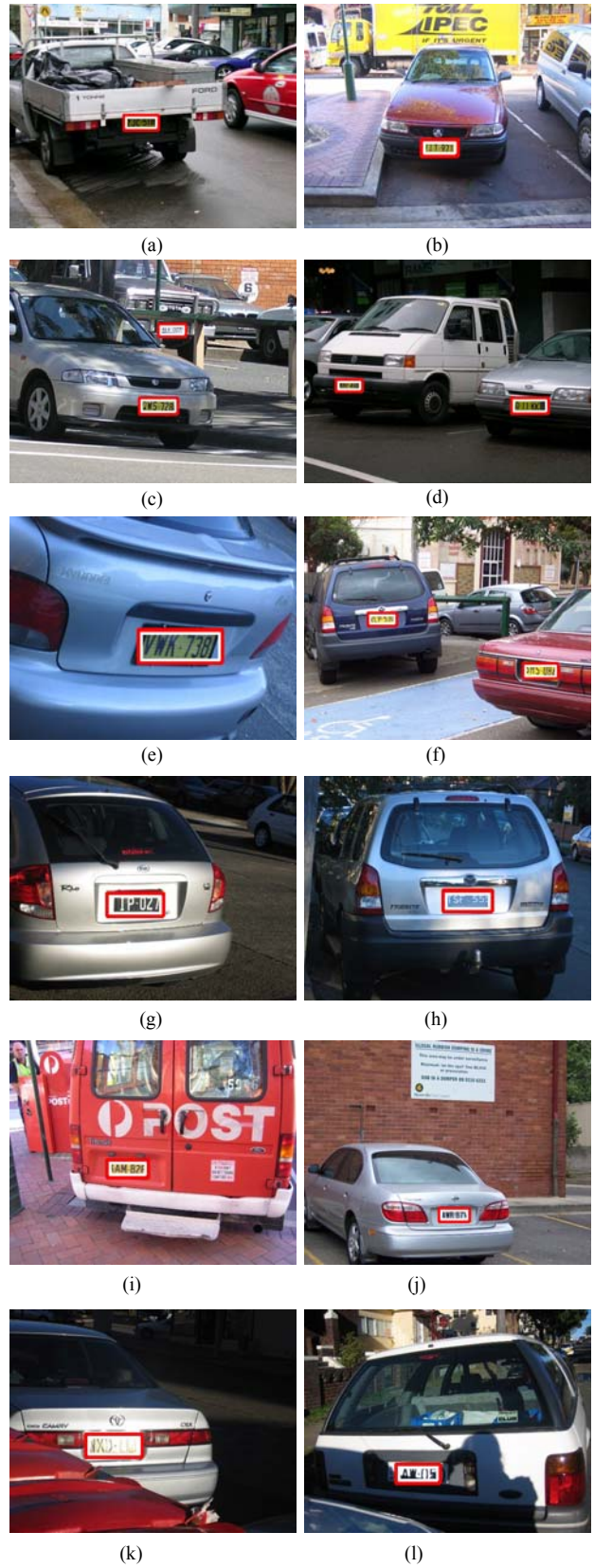


Fig. 7. Detection results of some vehicle images

next four layers, trained by AdaBoost learning procedure, are based on Haar-like features. In our algorithm, a new defined vertical gradient map is used to extract statistical features, which makes the detection speed extremely high. In the experiments, we analyze the relations of scaling factor with detection rate and processing rate. The optimal scaling factor is selected in our algorithm. Finally, 96.4% detection rate (with very low false positive rate) and high speed are obtained when the algorithm used to detect license plate in various complex conditions.

## VII. ACKNOWLEDGEMENT

This work was sponsored by the Australian Research Council under Large ARC-Discovery Grant DP0451666.

## REFERENCES

- [1] Bai, H., J. Zhu, and C. Liu. A Fast License Plate Extraction Method on Complex Background. in Proceedings of IEEE International conference on Intelligent Transportation Systems, pp. 985-987 vol.2, 2003.
- [2] Kim, S.K., D.W. Kim, and H.J. Kim. A Recognition of Vehicle License Plate Using a Genetic Algorithm Based Segmentation. in Proceedings of International Conference on Image Processing, pp. 661-664, 1996.
- [3] Kim, S., et al. A Robust License-plate Extraction Method under Complex Image Conditions. in Proceedings of 16th International Conference on Pattern Recognition, pp. 216-219 vol.3, 2002.
- [4] Jia, W., et al. Mean Shift for Accurate License Plate Localization. in Proceedings of International Conference on Intelligent Transportation Systems, pp. 566-571, 2005.
- [5] Chang, S.-L., et al., Automatic License Plate Recognition. IEEE Transactions on Intelligent Transportation Systems, 2004. 5(1): p. 42-53.
- [6] Matas, J. and K. Zimmermann. Unconstrained License Plate and Text Localization and Recognition. in Proceedings of IEEE International Conference on Intelligent Transportation Systems, pp. 572-577, 2005.
- [7] Mita, T., T. Kaneko, and O. Hori. Joint Haar-like Features for Face Detection. in Proceedings of the Tenth IEEE International Conference on Computer Vision, pp. 1619-1626, 2005.
- [8] Viola, P. and M.J. Jones, Robust Real-Time Face Detection. International Journal of Computer Vision, 2004. 57(2): p. 137-154.
- [9] Chen, X. and A.L. Yuille. Detecting and Reading Text in Natural Scenes. in Proceedings of the International Conference on Computer Vision and Pattern Recognition, pp. II-366-II-373 Vol.2, 2004.
- [10] Freund, Y., An Adaptive Version of the Boost by Majority Algorithm. Machine Learning, 2001. 43(3): p. 293-318.
- [11] Bai, H. and C. Liu. A Hybrid License Plate Extraction Method Based on Edge Statistics and Morphology. in Proceedings of the 17th International Conference on Pattern Recognition, pp. 831-834 Vol.2, 2004.
- [12] Yu, M. and Y.D. Kim. An Approach to Korean License Plate Recognition Based on Vertical Edge Matching. in Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, pp. 2975-2980 vol.4, 2000.
- [13] Chen, W.-Y. and S.-Y. Chen, Adaptive Page Segmentation for Color Technical Journals' Cover Images. Image and Vision Computing, 1998. 16(12-13): p. 855-877.
- [14] Duda, R.O., P.E. Hart, and D.G. Stork, Pattern Classification. 2nd ed. 2001, New York, NY, USA: Wiley.