Real-Timely Detecting License Plate under Various Conditions

Huaifeng Zhang, Wenjing Jia, Xiangjian He and Qiang Wu

Computer Vision Research Group,
University of Technology, Sydney
E-mail: {hfzhang, wejia, sean, wuq}@it.uts.edu.au

Abstract This paper proposes a learning-based algorithm for real-time license plate detection. Two kinds of features, statistical gradient features and Haar-like features, are used in the algorithm. Firstly, two statistical features are extracted from vertical gradient images. Classifiers based on these two features are constructed through simple learning procedures respectively. Using these classifiers, more than 80% of background area can be excluded from further training or testing. Then the AdaBoost learning procedure is used to build up the other classifiers based on selected Haar-like features. Combining the classifiers using the statistical features and the Haar-like features, we obtain a cascade classifier which can real-timely detect license plates from various complex backgrounds. In the experiments, high detection rate and low positive false rate are achieved when the algorithm is used to detect license plates from images taken in various complex environments.

1. Introduction

License plate recognition (LPR) 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 under various complex conditions. Therefore, in the LPR system, license plate detection is the most crucial step. Most of previous license plate detection algorithms are restricted in certain working conditions, such as fixed backgrounds [1], known color [2], or designated ranges of the distance between camera and vehicle [3, 4].
In these years, there were some researchers working on license plate detection under various conditions. Chang et al. [5] proposed a license plate detection algorithm using color edge and fuzzy disciplines. Their algorithm only can be used to detect the license plates with specific colors. In [6], Matas and Zimmermann proposed an algorithm to detect license plate and road sign in various conditions. They used character regions as basic units of license plate, which makes their algorithm robust to viewpoint and illumination. However, it 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 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 plate templates are very difficult to be constructed. Moreover, the sizes of the statistical features used in their algorithm were fixed. Hence the application of this algorithm is restricted extremely.

Recently, Haar-like features were widely used in object detection [7, 8]. One of the problems of these algorithms is that too many features were used in the classifier, which makes the system too complex. Chen and Yuille [9] constructed a simple cascade classifier for text detection using statistical features. However, only statistical features were used in their algorithm, which always results in high false positive rate in practice.

In this paper, we use both statistical features and Haar-like features in the 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 low down the false positive rate. Moreover, in our algorithm, the statistical features are extracted from vertical gradient image, which makes algorithm extremely fast.

The rest of the paper is organized as follows. The framework of our algorithm is introduced in Section 2. Vertical gradient image and two statistical features, Gradient Density and Density Variance, are defined in Section 3. Then Haar-like features and AdaBoost algorithm are described in Section 4. Experimental results are presented in Section 5. At last the paper is concluded in Section 6.
2. The Framework of the Algorithm

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

2.1. 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 license plate. All of the samples are scaled to 48*16 for the convenience of training.

Firstly, for all the samples, the values of one of the statistical features, Gradient Density, are calculated. A classifier is obtained by selecting the threshold which classifies all the positive samples as positive ones. Then, all the samples, including positive samples and negative samples, that are classified as positive ones are used to train the classifier on the second layer. This classifier is based on the other statistical feature, Density Variance. The input samples are classified again and the positive ones are used to train the classifier on the third layer. Similarly, the samples classified as positive ones by the third layer are input to the fourth layer, and so on. The classifier on the third layer and the following layers are based on local Haar-like features and AdaBoost leaning procedures. In our algorithm, we trained four layers of such classifier. Finally, a six-layer cascade classifier is constructed.

2.2. Testing

When an image is input into the classifier, 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.
The detection is implemented in multiple scales. In order to detect license plates of variant sizes, the block size is scaled up from 48*16 to 240*80, with a scaling factor of 1.2.

![Fig. 1. The working flow of cascade classifier, where 1,2,…,6 represent the layers, T is true and F is False](image)

3. 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 [1, 11]. 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 vertical edge image.

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

3.1. Vertical Gradients Image

The vertical gradient image is generated by the convolution of the original image and the x-direction Sobel operator

\[
S = \begin{bmatrix}
1 & 0 & -1 \\
2 & 0 & -2 \\
1 & 0 & -1
\end{bmatrix}
\] (1)

The vertical gradient image emphasizes the differences on x-direction and diminishes the differences on y-direction. Because most of the edges in license plate regions are vertical edges, the vertical edge image keeps most of the edge information in license
plate region, at the same time, eliminates a large amount of edge information in background regions as shown in Fig.2.

![Fig. 2. The gradient images of a vehicle image](image)

(a) Normal gradient image; (b) Vertical gradient image

### 3.2. Gradient Density

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

\[
D_G = \frac{1}{N} \sum_i \sum_j G(i, j),
\]

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

The x-direction 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.

### 3.3. 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 [12] in order to discriminate license plates from background regions.

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 \( \Delta V_g \), is defined as

\[
\Delta V_g = \frac{\sum_{i=1}^{n} |g_i - g|}{n \cdot g},
\]

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.

### 4. Haar-like Feature and AdaBoost

The Haar-like features originate from Haar basis functions [13]. 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.

![Four types of Haar-like features](image)
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*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 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.

5. 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.

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 one of the statistical features defined in Section 3. 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. Compared to Viola’s classifier having 38 layers and using 6060 features [8], our classifier is much simpler and efficient.

In our experiment, among the 169 visible license plates in 160 testing images, 156 license plates are detected, with detection rate 92.3%. At the same time, there are only 7 false positive regions. On a PC with Pentium 2.8GHz CPU, the detector can process a 648*486 image in about 50ms.

Fig. 6 shows some of the detection results, where the license plates are circled by white boxes. From the examples, we may see that our algorithm can work under various complex environments, various illuminations, and various view angles. The algorithm can detect the license plates with various sizes, positions and colors. Fig. 6(a) is an example with complex background; Fig. 6(b) shows the license plate detection against interference characters; Fig. 6(c) shows the result of detecting multiple license plates in one image; Fig. 6(d) shows that the algorithm can be used to detect very large size and blurred license plate.

6. Conclusions

In this paper, we construct a cascade classifier for license plate detection using both statistical and Haar-like features. 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 next four layers, trained by AdaBoost
learning procedure, are based on Haar-like features. In our algorithm, the training and testing are both extremely fast. With much less features, we obtain 92.3% detection rate and very low false positive rate when the license plate detection algorithm works in various complex environments.

![Detection results of some vehicle images](image)

**Fig. 6. Detection results of some vehicle images**

7. References


