

A Fast and Robust License Plate Detection Algorithm Based on Two-stage Cascade AdaBoost

Md. Mostafa Kamal Sarker¹, Sook Yoon² and Dong Sun Park^{1,3}

¹Division of Electronics Engineering, Chonbuk National University, Jeonbuk, Korea
[e-mail: rajusust62@gmail.com, dspark@jbnu.ac.kr]

²Department of Multimedia Engineering, Mokpo National University, Jeonnam, Korea
[e-mail: syoon@mokpo.ac.kr]

³IT Convergence Research Center, Chonbuk National University, Jeonbuk, Korea

*Corresponding author: Sook Yoon

Received May 20, 2014; revised August 9, 2014; accepted August 22 2014; published October 31, 2014

Abstract

License plate detection (LPD) is one of the most important aspects of an automatic license plate recognition system. Although there have been some successful license plate recognition (LPR) methods in past decades, it is still a challenging problem because of the diversity of plate formats and outdoor illumination conditions in image acquisition. Because the accurate detection of license plates under different conditions directly affects overall recognition system accuracy, different methods have been developed for LPD systems. In this paper, we propose a license plate detection method that is rapid and robust against variation, especially variations in illumination conditions. Taking the aspects of accuracy and speed into consideration, the proposed system consists of two stages. For each stage, Haar-like features are used to compute and select features from license plate images and a cascade classifier based on the concatenation of classifiers where each classifier is trained by an AdaBoost algorithm is used to classify parts of an image within a search window as either license plate or non-license plate. And it is followed by connected component analysis (CCA) for eliminating false positives. The two stages use different image preprocessing blocks: image preprocessing without adaptive thresholding for the first stage and image preprocessing with adaptive thresholding for the second stage. The method is faster and more accurate than most existing methods used in LPD. Experimental results demonstrate that the LPD rate is 98.38% and the average computational time is 54.64 ms.

Keywords: license plate detection, cascade classifier, Haar-like features, AdaBoost, adaptive thresholding

This work is supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2013778).

<http://dx.doi.org/10.3837/tiis.2014.10.012>

1. Introduction

Recently, many applications such as intelligent traffic management systems have required license plate recognition (LPR) systems, as the number of vehicles on the roads increase continuously. In a LPR system, license plate detection (LPD) is one of the most crucial steps. Moreover, it can be very difficult to detect license plates from cluttered backgrounds efficiently due to variations in plate style, the environment, cameras, and car speeds. Most previous LPD algorithms have restricted to use in certain working conditions, such as fixed backgrounds [1], known colors [2], or fixed sizes of license plates [3][4]. Thus, it is still a challenging problem to detect the many complex license plates variants under all conditions.

In a license plate detection system (LPDS), we need to deal with a large variety of license plates, especially in South Korea, as shown in Fig. 1. In Korea, there are license plates with various colors and sizes and different patterns and formats of numbers and characters. There are three different sizes of license plates in Korea: large (520 mm × 110 mm), medium (440 mm × 200 mm) and small (335 mm × 170 mm or 155 mm).



Fig. 1. Different types and sizes of Korean license plate

As shown in Fig. 2, the variations under consideration can be summarized as follows [5]:

1. Plate variation	
(i) Location	Plates may exist in different locations in an input image.
(ii) Quantity	An input image may contain several or no plates.
(iii) Size	Plates with different sizes may exist in an image or different images.
(iv) Colors of plate characters and backgrounds	Plates may have various characters and background colors due to different plate types (e.g., taxis, private cars) or capturing device characteristics.
(v) Others	In addition to characters, a plate may contain adornments, such as frames and screws.
2. Environment variation	
(i) Illumination	Different types of illumination may occur in input images, mainly due to environmental lighting and vehicle headlights.
(ii) Plate-like background patterns	A background may contain patterns similar to plates, such as numbers stamped on a vehicle, bumpers with vertical patterns, and textured floors.
(iii) Noise in image	The captured images have many noises when it is snowing or raining.

3. Camera variation	Significant variations in camera perspectives when license plate images are captured may make big differences between plates.
4. Car speed variation	Significant variations in car speed when the license plate images are captured may distort captured images.



Fig. 2. Examples of captured license plate images

For automatic license plate detection systems, researchers have used many diverse methods, including corner template matching [6], Hough transform [7][8][9], histogram analysis [10][11][12], and morphological operations [13]. And sophisticated mechanisms, such as vector quantization (VQ) [14], neural networks [15][16], mean shift [4], support vector machine [16] and likelihood ratio [17], have been developed to detect plates based on the features of plate characteristics or statistical representations including shape [11][17], symmetry [18], aspect ratio [10], color of HSI [16][19], color of YC_bC_r [18], and edges [10][17].

Viola et al. [20] proposed a new method for face detection, in which they introduced a new method known as the integral image for fast feature calculation and they used Haar-like features [21] and an AdaBoost learning algorithm for a high detection rate and fast processing time. The AdaBoost learning algorithm, first proposed by Freund and Schapire [22], selects simple Haar-like feature sets called weak classifiers from the complete features set and builds a strong classifier using a combination of weak classifiers by adapting the distribution of the training samples.

One of major concerns about LPD is unevenly illuminated input images, mainly due to variable environmental lighting and vehicle headlights. And an adaptive thresholding technique is commonly useful to eliminate the effect of variant illumination, but it is time-consuming. For better performance of LPD system in speed and accuracy, we need to seek the method which minimizes its processing time but maximizes its detection rate by reducing the unwanted illumination effect. Fortunately, in real world, because not all input images are unevenly illuminated, and indeed such cases may be relatively infrequent, we do not need preprocessing with adaptive thresholding for all input images. When we investigated our image data set obtained from the real road, the preprocessing with adaptive thresholding was not necessary for 87.94% of all images. In this paper, we propose a license plate detection method that is rapid and robust against variation in illumination. Instead of using one stage for dealing with all cases, the proposed method consists of two stages: one for dealing with normal illumination cases and the other for dealing with abnormal illumination cases. Each stage uses a cascade AdaBoost based on Haar-like features, validation check with CCA for eliminating false positives, and two different image preprocessing techniques: a preprocessing without an

adaptive thresholding technique for normal illumination cases and a preprocessing with an adaptive thresholding technique for abnormal illumination cases.

This paper is organized as follows. The proposed license plate detection method is described in Section 2. The experimental results in Section 3 show that the proposed method can provide rapid license plate detection with high accuracy. Finally, our conclusions are provided in Section 4.

2. Proposed System

In this paper, LPD system proposed for real-time processing and robustness against uneven-illumination consists of two stages, as shown in Fig. 3. Each stage consists of three main blocks: Image Preprocessing, Cascade AdaBoost and Validation Check using CCA. The two stages use different image preprocessing blocks: image preprocessing without adaptive thresholding for the first stage and image preprocessing with adaptive thresholding for the second stage. To further reduce time complexity, the integral image method is used for adaptive thresholding.

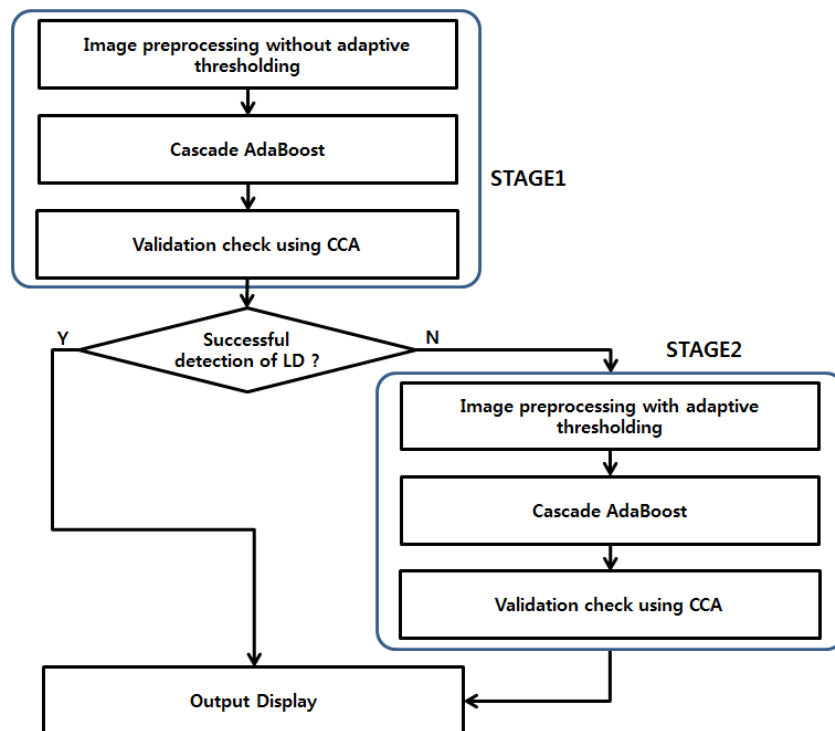


Fig. 3. Framework of proposed LPD system with two stages

2.1 Cascade AdaBoost

A cascade classifier is a concatenation of strong classifiers in a cascade. The AdaBoost algorithm is used to train each strong classifier, which is a series of weak classifiers, and is built to accept positive samples while rejecting negative samples. It forms a strong classifier through selecting and ensembling the weak classifiers. The image windows that are not rejected by a strong classifier participate in building the following strong classifier. The cascade AdaBoost for the LPD system consists of two phases: an off-line training phase and an

on-line detection phase, as shown in Fig. 4.

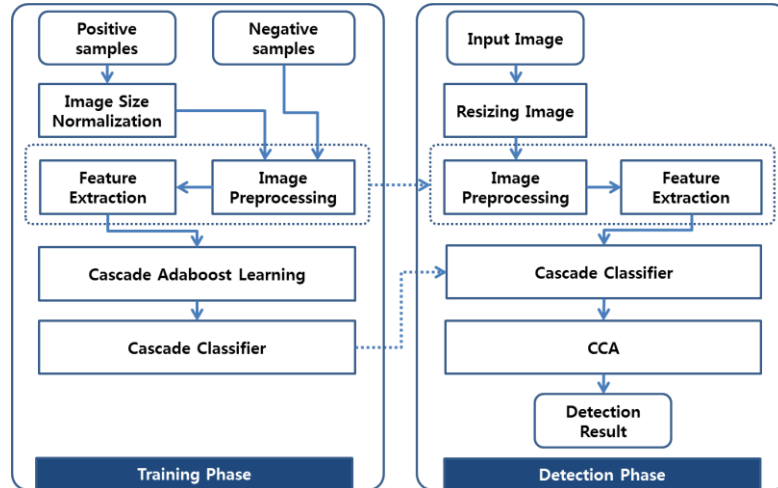


Fig. 4. Off-line training phase and on-line detection phase of Cascade AdaBoost

2.1.1 Off-line Training Phase

The purpose of the off-line training phase is to train weak classifiers and a combination of weak classifiers dubbed a strong classifier using the AdaBoost training algorithm and to build up a cascade classifier for the LPD, as shown in Fig. 5. A series of weak classifiers (critical features) with their weights are extracted first after being trained with a large number of positive and negative examples. Then, some strong classifiers are selected from the weak classifiers according to their weights. Strong classifiers in a cascade structure are then used to construct a detector for the on-line detection phase.

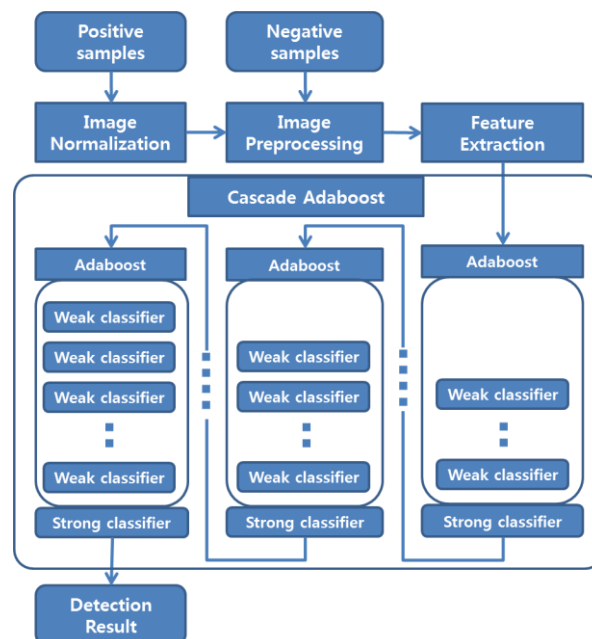


Fig. 5. Structure of the off-line training phase

The four steps for the off-line training procedure are presented below.

a) Training Samples

For the off-line training, positive sample images and negative sample images are required. The positive sample images contain only license plate images and negative sample images contain background images with no license plate image. The Korean license plate has three different sizes: large (520mm × 110mm), medium (440mm × 200mm), and small (335mm × 170mm). In total, 15,000 positive sample images (6,000 large, 3,000 medium and 6,000 small) are used for the training. Fig. 6 shows some positive sample images of Korean license plate.



Fig. 6. Some positive sample images of Korean license plates

To equalize the size of positive sample images required for the AdaBoost training algorithm, all types of Korean license plate images were normalized to one equal size using boundary padding or boundary pixel extension [23]. Thus 40×14 was selected as the image size for our database images through experiments. Fig. 7 shows examples of normalized license plate images with boundary padding or boundary pixel extension.

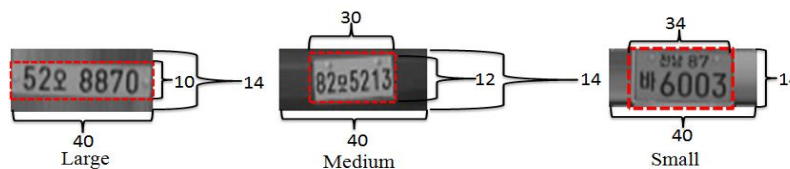


Fig. 7. Examples of normalized license plate images

Negative sample images with no license plate should include some parts of cars, the road, or and trees, for example. In our training procedure, 25,000 images were used for negative samples. Fig. 8 shows some negative sample images.

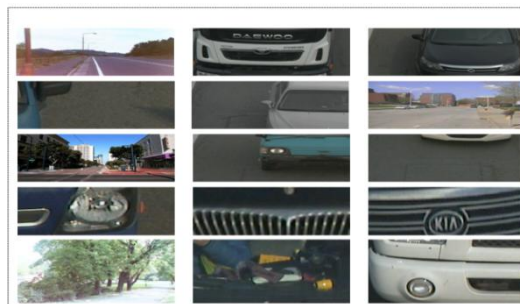


Fig. 8. Some negative sample images

b) Image Pre-processing

In the image pre-processing step, image converting, filtering, and edge detection are used for training with the sample images, as described below.

First, 24-bit RGB images are converted into 8-bit gray-level images using Eq. (1).

$$\text{Gray value} = 0.3 \times \text{Red} + 0.59 \times \text{Green} + 0.11 \times \text{Blue} \quad (1)$$

Second, a Gaussian filter is used for image filtering. A Gaussian filter [24] is a weighted averaging of neighboring pixels using weights chosen according to the shape of a Gaussian function, which is

$$g[i, j] = e^{-\frac{(i^2+j^2)}{2\sigma^2}} \quad (2)$$

where i is the distance from the origin in the horizontal axis, j is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. Fig. 9 shows examples of images filtered with the Gaussian filter.



Fig. 9. Examples of images filtered with the Gaussian filter: (a) Positives samples and (b) Negative samples

Thirdly, for edge detection or edge image, the Sobel vertical edge operator [25] is applied. Fig. 10 shows the convolution mask of Sobel vertical edge operator and edge images after using Sobel vertical edge operator.

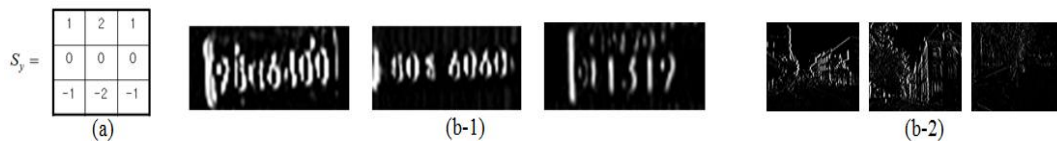


Fig. 10. Edge detection : (a) convolution mask of Sobel vertical edge operator and examples of edge images obtained by (a): (b-1) Positive samples and (b-2) Negative samples

c) Feature extraction

The simple features used in the proposed system are an extended set of Haar-like features, presented by Rainer Lienhart and Jochen Maydt [26], shown in Fig. 11. They are useful to extract principal features for LPD. Details of its computation are shown in [26]. All the features can be computed very quickly in a constant time for any size.

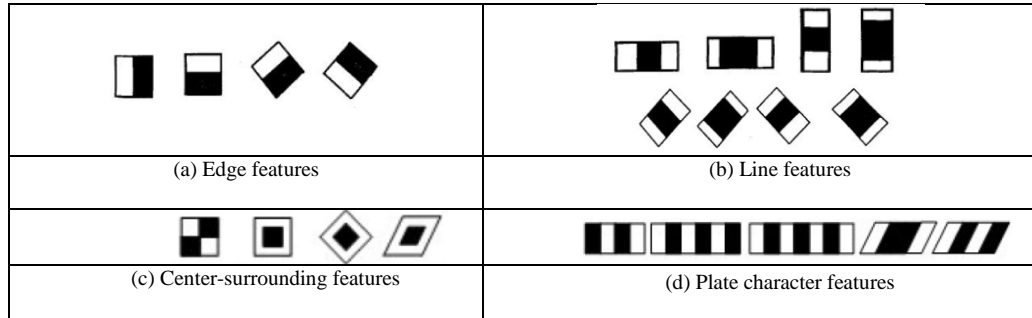


Fig. 11. Haar-like prototypes used in the proposed algorithm

d) *AdaBoost algorithm for training:*

When we have determined a feature set from a training set of positive and negative samples, as discussed earlier, a machine learning algorithm could be used to learn its classification function. In the proposed system, a cascade classifier based on an AdaBoost training algorithm was used to detect any license plate. A cascade classifier is a concatenation of strong classifiers and a strong classifier is based on a large set of weak classifiers that classify the objects. It is a degenerated decision tree where, at each layer, a classifier is trained to detect all of license plates in the interest image scope while rejecting a certain fraction of non-license-plate patterns. At each layer, a strong classifier is trained to achieve the predetermined hit rate (or true positive rate), h , and a false alarm (or false positive) rate, f , using the Discrete AdaBoost algorithm. The learning algorithm for a weak classifier is designed to select a single rectangular feature that best separates the positive and negative samples. For each feature, the weak learner determines the optimal threshold classification function, so that the minimum number of samples is misclassified. A weak classifier $h_j(x)$ thus is represented by a feature (f_j), a threshold (θ_j), and a polarity (p_j), indicating the direction of the inequality sign [27] :

$$h_j(x) = h(x, f_j, p_j, \theta_j) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where x is a sub-window of an image. The following is for a summary of the AdaBoost algorithm for classifier learning [27].

1. Prepare sample images $(x_1, y_1), \dots, (x_n, y_n)$, where $y_i = 0, 1$ for negative and positive examples, respectively.
2. Initialize weights $\omega_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the numbers of negatives and positives respectively.
3. For $t = 1, \dots, T$:

- a) Normalize the weights,

$$\omega_{t,i} = \frac{\omega_{t,i}}{\sum_{j=1}^n \omega_{t,j}} \quad (4)$$

So that ω_t is a probability distribution.

- b) For each feature j , calculate the weighted error,

$$\epsilon_j = \sum \omega_i |h_j(x_i) - y_i| \quad (5)$$

and choose the classifier, $h_t(x_i) = h(x_i, f_t, p_t, \theta_t)$, where f_t, p_t , and θ_t are the minimizers of ϵ_j .

- c) Update the weights:

$$\omega_{t+1,i} = \omega_{t,i} \beta_t^{1-e_i} \quad (6)$$

where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and

$$\beta_t = \frac{\epsilon_t}{1-\epsilon_t}.$$

4. The final strong classifier is:

$$C(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$\text{where } \alpha_t = \log \frac{1}{\beta_t}.$$

For the Cascade AdaBoost training[28], The background (BG) threshold is 80, the maximum (Max) intensity deviation is 40, the minimum (Min) hit rate is 0.995, the Max false alarm rate is 0.5, the Min number of positive sample per cluster is 500, the number of training stages is 14 and the total number of used features is 61,789. After finishing the training procedure of the AdaBoost algorithm, a cascade classifier is created with a XML file. This XML file contains a strong classifier with features. The AdaBoost algorithm for license plate detection then uses this XML file in the on-line detection procedure of our experiment.

2.1.2 On-line Detection Phase

The number of test images used for the on-line license plate detection was 52,000 (the database used will be explained in the experimental results and discussion later) and each image has a resolution of 1,280×720. The procedure for the on-line license plate detection is as follows.

a) Resizing Input Image:

The size of the images in our databases is very large. However, high image resolution requires more computational time, so each original test image (1,280×720) was resized to a size of 320×180 to speed up the detection.

b) Image pre-processing:

The same image pre-processing technique as explained in the off-line training phase was used: converting an image from RGB to gray scale, image filtering with the Gaussian filter, and edge detection (edge image) by a Sobel vertical edge operator. Fig. 12 shows the results of image pre-processing.



Fig. 12. Examples of image pre-processing results: (a) results after using the Gaussian filter (b) results after using the Sobel vertical operator

c) Cascade classifier for detecting license plate:

Weak classifiers are combined with each other to form a strong classifier. The strong classifier in the first layer allows a great majority of the image regions to enter to the next layer and rejects as many as possible negative samples. The strong classifier in the next layer becomes more complex (i.e., has more weak classifiers). Thus, it has a stronger classification ability and the final result is more likely to be a license plate. **Fig. 13** shows the cascade structure of the proposed license plate detection system.

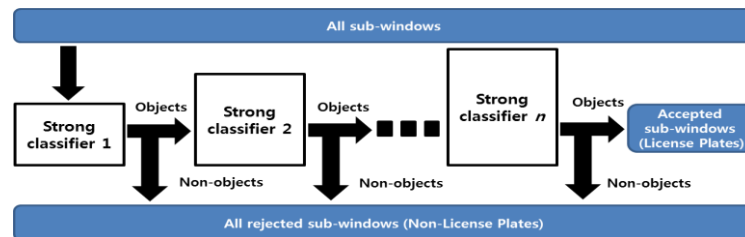


Fig. 13. Cascade AdaBoost structure for license plate detection

In the combination process, a strong classifier, which consists of more important features and easier structures is positioned in the front of the whole cascade classifier so that the system can exclude many false samples and speed up the detection. **Fig. 14** shows the results of license plate detection using our proposed Cascade AdaBoost algorithm.

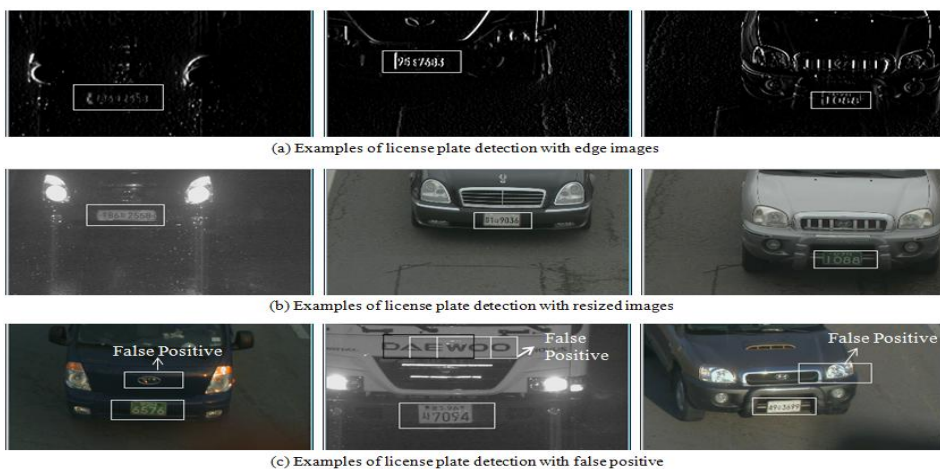


Fig. 14. Results of license plate detection using Cascade AdaBoost

d) *Image check using Connected Component Analysis (CCA)*

There are many false-positive areas detected as license plate regions using the Cascade AdaBoost algorithm. To ignore these false-positives, a connected component analysis (CCA) is used. CCA was used in computer vision to detect connected regions in binary digital images, although color images and data with higher dimensionality can also be processed [29][30]. When integrated into an image recognition system or human-computer interaction interface, connected component labeling can operate on a variety of information [31][32]. Blob extraction is generally performed on the resulting binary image from a thresholding step. Blobs may be counted, filtered, and tracked. Using the number of blobs remaining, we can reject the false positive region images and find the only license plate region images and save them, as shown in Fig. 15. The method for checking whether an area is a license plate region is as follows.

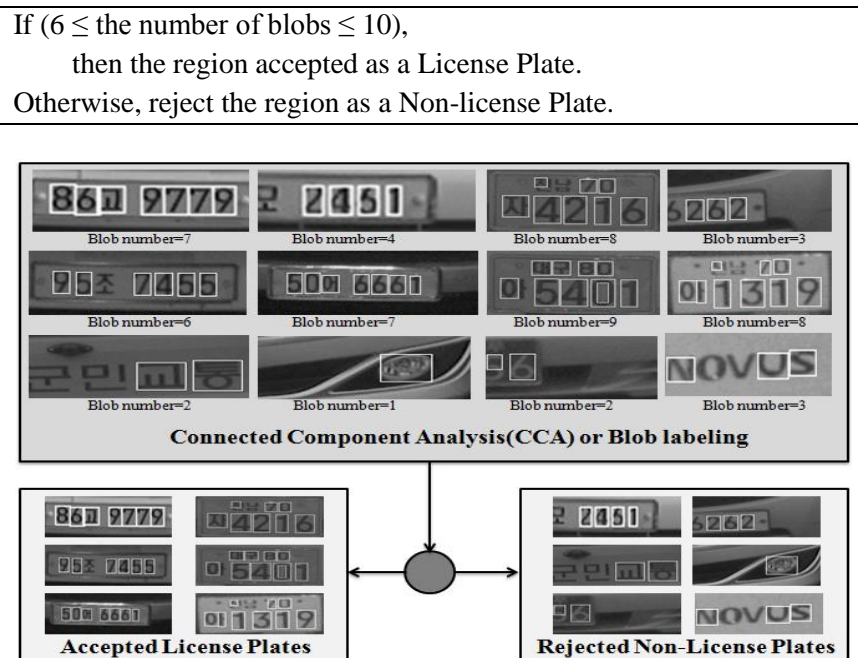


Fig. 15. Check the images with CCA and save the license plate images

2.2 Adaptive Thresholding Using the Integral Image

After analyzing license plates missed from the first stage using the normal image processing, Cascade Adaboost and the CCA technique, it was found that most of them were from unevenly illuminated images. However, as mentioned earlier, it is not efficient to apply a technique for reducing this effect to all cases because typical uneven illumination-handling techniques are time-consuming. Thus, from considerations of accuracy and speed, we built up a two-stage model. In the second stage of our proposed system, the procedure for the training and detection phase using Cascade AdaBoost is the same as in the first stage, while using an image preprocessing technique different from the first one. We used adaptive thresholding using an integral image for the image preprocessing in the second stage.

The adaptive thresholding technique used in the proposed system is the technique using an integral image proposed by D. Bradley et al. [33], which is an extension of Wellner's

method [34]. Using the integral image, it computes the average of a $s \times s$ window of pixels centered on each pixel. The integral image is calculated in the first pass through the input image. In a second pass, the $s \times s$ average is computed using the integral image for each pixel in constant time and then the comparison is performed. If the value of the current pixel is t percent less than this average, then it is set to black, otherwise it is set to white. Fig. 16 shows some examples of license plate detection results of the second stage including adaptive thresholding using integral image in the image preprocessing



Fig. 16. License plate detection results after applying adaptive thresholding using the integral image

3. Experimental Results and Discussion

3.1 Databases

For our test experiment, we used our own database to calculate the detection rate and computational time. The total number of images was 1800. All images in our database were captured with a CMOS camera on a pole placed on a real road under different weather conditions, so they include rotated or illuminated images. Fig. 17 shows a description of the image-capturing environment.

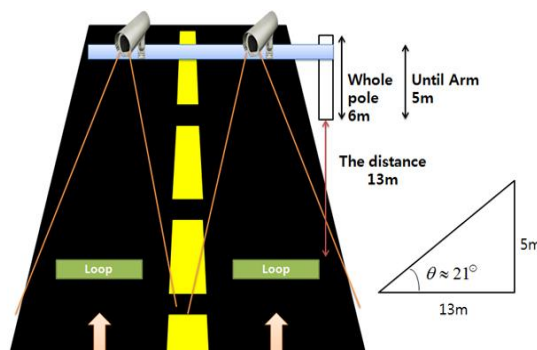


Fig. 17. Image-capturing environment

The height of the pole was 6 m, but the height to the arm where image-capturing camera was fixed was 5 m. The distance from the bottom of the pole to the loop (car position) was 13 m. Thus, the angle between the camera and loop is 21° . When the images were captured, the speed of the moving cars was around 100 - 120 km/h.

3.2 Experimental results of the proposed method

The experiment was based on a PC with 3.10-GHz Intel Core i3-2100 with 4 GB of RAM, running in Microsoft Visual Studio 2010 with the Opencv (Open Source Computer Vision) library. The window size used as input of the proposed detection system was 40×14 . When only the first stage was applied to the database, 1,583 among 1,800 images were detected correctly and there were 306 false-positives before applying the CCA technique. However, after applying CCA technique, no false-positive remained. The 217 images that were not detected successfully at the first stage were taken as input images for the second stage with the Adaptive Thresholding using an Integral Image. From the second stage, the number of total detected images was 188 from the 217 images and 35 false-positives. After applying CCA, no false-positive remained. The number of final missed images was 29: twenty-four images with strong dazzle lights at night, one image with a dark big shadow at day, and four images with strong reflected lights at day. All of them have severe light conditions. Fig. 18 shows some of 29 missed images. [Table 1](#) shows the license plate detection results of the proposed two-stage system.

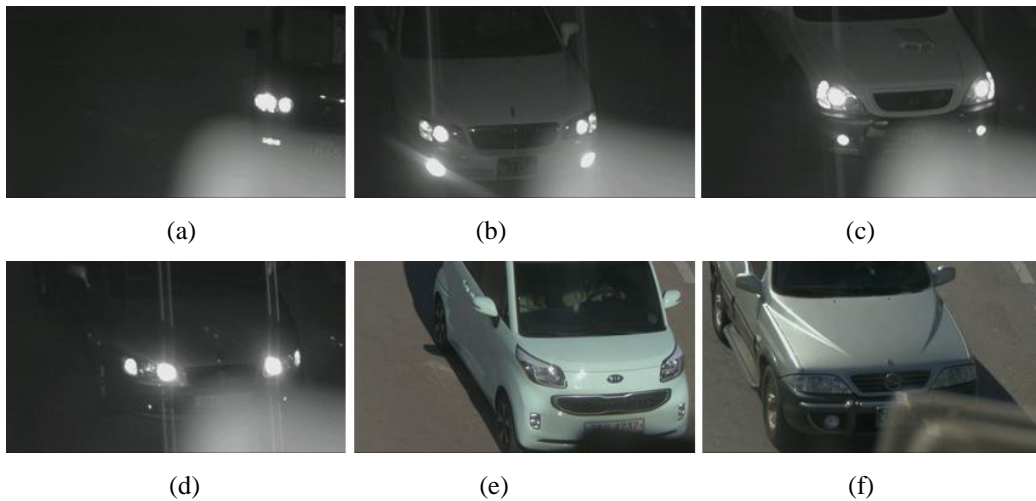


Fig. 18. Examples of 29 missed images: (a)-(d) images with dazzle lights at night, (e) an image with a dark and big shadow at day, and (f) an image with a strong reflected light at day

Table 1. License plate detection result of the proposed two-stage system

Proposed system	Number of test images	Number of missed images	Detection rate	Average time per image
Stage I	1800	217	87.94%	45ms
Stage II	217	29	86.64%	80ms
Total	1800	29	98.38%	54.64ms

Together, the number of total test images was 1,800 and the number of image detected successfully through the proposed two-stage system was 1,771. The number of missed images was 29. Thus, the average detection rate of the proposed method using the two stages was 98.38% and the computational time was 54.64 ms (45 ms for 1583 images and 125ms for the remaining 217 images). The proposed method has a lower time complexity than the method which apply adative thresholding technique for all input images and higher accuracy than the method which does not use adaptive thresholding and the experimental results showed that it worked as originally intended. By applying the adaptive thresholding, the detection rate was increased by 11.87% from 87.97% (obtained from the first stage without adaptive thresholding) to 98.38%. By applying the two-stage system, the processing time could be reduced by 31.7%, from 80 ms (which was required to apply only the second stage including the adaptive thresholding to all images) to 54.64 ms. Also, as shown in Fig. 19, the proposed method showed the good performance under different conditions due to the location and rotation state of the cars as well as uneven illumination.



Fig. 19. Examples of successfully detecting license plates from cars under different conditions using the proposed method

Table 2 shows a performance comparison of the proposed method with some typical ALPR systems. The proposed method is faster and has higher detection performance than most other methods.

Table 2. Performance Comparison of Some Typical ALPR Systems for License Plate Detection (LPD)

Methods	Main Procedures for License plate detection	Database size	Image conditions	LPD rate	Processing time	Real time	Plate format
[35]	Block-based processing	180 pairs of images	Multiplates with occlusion and different sizes	94.4%	75 ms	Yes	Taiwanese plates
[36]	Hough Transform and contour algorithm	805 images	800 × 600 pixels, different rotation and lighting condition	98.8%	0.65 s	No	Vietnamese plates
[37]	GST	330 images	Various viewing directions	93.6%	1.3 s	No	Korean plates
[38]	Edge detection and vertical and horizontal	12 s video	320 × 240 pixels	85.5%	100 ms	Yes	Taiwanese plates

	projections						
[4]	Color, Mean Shift	57 images	324x243 pixels	97.6%	6s	No	Australian Plates
[39]	Gradient analysis	2340 images	Different weather and illumination conditions	98.6%	1.1s	NO	Italian plates
[40]	Scan line, texture properties, color, and Hough transform	332 images	867 × 623 pixels (various illumination and different distances)	97.1%	0.53s	No	Taiwanese plates
Our Proposed Method	Cascade AdaBoost Algorithm and Adaptive Thresholding Using the Integral Image	1800 images	1280×720 pixels, various weather condition and different illumination	98.38%	54.64 ms	Yes	Korean Plates

4. Conclusions

In this paper, a license plate detection (LPD) system was proposed for real-time processing and robustness even in uneven illumination situation. It consists of two stages and each stage consists of three main blocks: Image Preprocessing, Cascade AdaBoost, and Validation check using CCA. The two stages use different image preprocessing blocks: image preprocessing without adaptive thresholding for the first stage and image preprocessing with adaptive thresholding for the second stage. Experimental results show that the test accuracy was 98.38% and the average computational time was 54.64 ms, which is faster than existing methods. The proposed method also has better performance than existing methods. Most of the techniques are very complex and not suited to real-time applications but the proposed algorithm is suitable for real-time use. Regarding the proposed methods, further research will be needed before the proposed method can put to practical use and character recognition of the license plate is our next project.

References

- [1] H. Bai and C. Liu, "A Hybrid License Plate Extraction Method Based on Edge Statistics and Morphology," in *Proc. of the International Conference on Pattern Recognition*, vol. 2, pp. 831-834, August 23-26, 2004. [Article \(CrossRef Link\)](#)
- [2] S. K. Kim, D. W. Kim and H. J. Kim, "A Recognition of Vehicle License Plate Using a Genetic Algorithm Based Segmentation," in *Proc. of International Conference on Image Processing*, vol. 2, pp. 661-664, September 19, 1996. [Article \(CrossRef Link\)](#)
- [3] S. Kim, D. Kim, Y. Ryu and G. Kim, "A Robust License-plate Extraction Method under Complex Image Conditions," in *Proc. of International Conference on Pattern Recognition*, vol. 3, pp. 216-219, August 11-15, 2002. [Article \(CrossRef Link\)](#)
- [4] W. Jia, H. Zhang, X. He and M. Piccardi, "Mean Shift for Accurate License Plate Localization," in *Proc. of International Conference on Intelligent Transportation Systems*, pp. 566-571, September 13-15, 2005. [Article \(CrossRef Link\)](#)
- [5] S. Du, M. Ibrahim, M. Shehata, W. Badawy, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 23, no. 2, pp. 311-325, February, 2013. [Article \(CrossRef Link\)](#)
- [6] H. A. Hegt, R. Haye, and N. A. Khan, "A high performance license plate recognition system," in *Proc. of IEEE Int. Conf. Systems, Man, Cybern.*, vol. 5, pp. 4357-4362, October 14, 1998. [Article \(CrossRef Link\)](#)

- [7] V. Kamat and S. Ganesan, "An efficient implementation of the hough transform for detecting vehicle license plate using dsp's," in *Proc. of IEEE Real-Time Technology and Applications Symp.*, pp. 58, May 15-17, 1995. [Article \(CrossRef Link\)](#)
- [8] Y. Yanamura et al., "Extraction and tracking of the license plate using hough transform and voted block matching," in *Proc. of IEEE Int. Conf. Intelligent Vehicles Symp.*, pp. 243-246, June 9-11, 2003. [Article \(CrossRef Link\)](#)
- [9] K. M. Kim, B. J. Lee, K. Lyou, and G. T. Park, "The automatic recognition of the plate of vehicle using the correlation coefficient and hough transform," *Journal of Control, Automation and Systems Engineering*, vol. 3, no. 5, pp. 511-519, 1997.
- [10] S.-Z. Wang and H.-J. Lee, "Detection and recognition of license plate characters with different appearances," in *Proc. of IEEE Int. Conf. Intelligent Transportation Systems*, vol. 2, pp. 979-984 October 12-15, 2003. [Article \(CrossRef Link\)](#)
- [11] Y. Amit, D. Geman, and X. Fan, "A coarse-to-fine strategy for multiclass shape detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 12, pp. 1606-1621, December, 2004. [Article \(CrossRef Link\)](#)
- [12] D. U. Cho and Y. H. Cho, "Implementation of preprocessing independent of environment and recognition of car number plate using histogram and template matching," *J. Korean Comm. Sci.*, vol. 23, no. 1, pp. 94-100, January, 1998.
- [13] P. Xiang, Y. Xiuzi, and Z. Sanyuan, "A hybrid method for robust car plate character recognition," in *Proc. IEEE of Int. Conf. on Systems, Man, and Cybernetics*, vol. 5, pp. 4733-4737, October 10-13, 2004. [Article \(CrossRef Link\)](#)
- [14] R. Zunino and S. Rovetta, "Vector quantization for license plate location and image coding," *IEEE Trans. Ind. Electron.*, vol. 47, no. 1, pp. 159-167, February, 2000. [Article \(CrossRef Link\)](#)
- [15] S. Draghici, "A neural network based artificial vision system for license plate recognition," *Int. J. Neural Syst.*, vol. 8, pp. 113-126, February, 1997. [Article \(CrossRef Link\)](#)
- [16] K. Kim, K. Jung, and H. Kim, "Fast color texture-based object detection in images: Application to license plate localization," *J. Lecture Notes Comput. Sci.*, vol. 177, pp. 297-320, 2005. [Article \(CrossRef Link\)](#)
- [17] Y. Amit, D. Geman, and X. Fan, "A coarse-to-fine strategy for multiclass shape detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 12, pp. 1606-1621, December, 2004. [Article \(CrossRef Link\)](#)
- [18] S. Yoshimori et al., "License plate detection using hereditary threshold determine method," *J. Lecture Notes Comput. Sci.*, vol. 2773, pp. 585-593, 2003. [Article \(CrossRef Link\)](#)
- [19] S. L. Chang et al., "Automatic license plate recognition," *IEEE Trans. Intell. Transport. Syst.*, vol. 5, no. 1, pp. 42-53, March, 2004. [Article \(CrossRef Link\)](#)
- [20] P. Viola, and M. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137-154, May, 2004. [Article \(CrossRef Link\)](#)
- [21] C. P. Papageorgiou and T. Poggio, "A Trainable Object Detection System: Car Detection in Static Image," *A.I. Memo No 1673, C.B.C.L Paper No 180. Massachusetts Institute of Technology*, 1999. [Article \(CrossRef Link\)](#)
- [22] Y. Freund and R. E. Schapire, "Experiments with a New Boosting Algorithm," in *Proc. of the 13th International Conference on Machine Learning*, pp. 148-156, July 3-6, 1996.
- [23] Richard Szeliski. "Computer Vision: Algorithms and Application," SPRINGER, ISBN 978-1-84882-934-3, pp. 101-102, 2011. [Article \(CrossRef Link\)](#)
- [24] David A. Forsyth and Jean Ponce. "Computer Vision: A Modern Approach," Pearson Education, Inc. ISBN 0-13-191193-7, pages: 136-139, 2003.
- [25] Milan Sonka, Vaclav Hlavac and Roger Boyle. "Image Processing, Analysis, and Machine Vision," International Student Edition(Third Edition), THOMSON, ISBN: 10: 0-495-24438-4, pages:136-137.
- [26] Rainer Lienhart and Jochen Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," in *Proc. of IEEE International Conference on Image Processing*, vol. 1, pp. 900-903, September 22-25, 2002. [Article \(CrossRef Link\)](#)
- [27] P. Viola, M. Jones "Rapid object detection using a boosted cascade of simple features," in *Proc. of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 511-518, December 8-14, 2001. [Article \(CrossRef Link\)](#)
- [28] W-C Cheng, D-M Jhan, "A cascade classifier using Adaboost algorithm and support vector machine for pedestrian detection," in *Proc. of the 2011 IEEE International Conference on Systems, Man, and Cybernetics*, pp.1430-1435, October 9-12, 2011. [Article \(CrossRef Link\)](#)

- [29] H. Samet and M. Tamminen, "Efficient Component Labeling of Images of Arbitrary Dimension Represented by Linear Bintreees," *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol.10, no. 4, pp. 579-586, 1988. [Article \(CrossRef Link\)](#)
- [30] Michael B. Dillencourt and Hannan Samet and Markku Tamminen, "A general approach to connected-component labeling for arbitrary image representations," *Journal of the ACM*, vol. 39, no. 2, pp. 253-280, April 1992. [Article \(CrossRef Link\)](#)
- [31] Weijie Chen, Maryellen L. Giger and Ulrich Bick, "A Fuzzy C-Means (FCM)-Based Approach for Computerized Segmentation of Breast Lesions in Dynamic Contrast-Enhanced MR Images," *Academic radiology* , vol. 13, no. 1, pp. 63-72, 2006. [Article \(CrossRef Link\)](#)
- [32] Kesheng Wu, Wendy Koegler, Jacqueline Chen and Arie Shoshani, "Using Bitmap Index for Interactive Exploration of Large Datasets," in *Proc. of SSDBM 2003*, pp. 99-108, July 9-11, 2003. [Article \(CrossRef Link\)](#)
- [33] D. Bradely, G. Ross, "Adaptive Thresholding using the Integral Image," *Journal of Graphics Tools*, vol. 12, no. 2, pp. 13-21, 2007. [Article \(CrossRef Link\)](#)
- [34] P. D. Wellner, "Adaptive thresholding for the digitaldesk," Tech. Rep. EPC-93-110, EuroPARC, 1993. [Article \(CrossRef Link\)](#)
- [35] H.-J. Lee, S.-Y. Chen, and S.-Z. Wang, "Extraction and recognition of license plates of motorcycles and vehicles on highways," in *Proc. of Int. Conf. Pattern Recognit.*, vo. 4, pp. 356-359, August 23-26, 2004. [Article \(CrossRef Link\)](#)
- [36] T. D. Duan, D. A. Duc, and T. L. H. Du, "Combining Hough transform and contour algorithm for detecting vehicles' license-plates," in *Proc. of Int. Symp. Intell. Multimedia Video Speech Process.*, pp. 747-750, October 20-22, 2004. [Article \(CrossRef Link\)](#)
- [37] D.-S. Kim and S.-I. Chien, "Automatic car license plate extraction using modified generalized symmetry transform and image warping," in *Proc. of IEEE Int. Symp. Ind. Electron.*, vol. 3, pp. 2022-2027, June 12-16, 2001. [Article \(CrossRef Link\)](#)
- [38] K. Deb and K.-H. Jo, "A vehicle license plate detection method for intelligent transportation system applications," *Int. J. Cybern. Syst.*, vol. 40, no. 8, pp. 689-705, 2009. [Article \(CrossRef Link\)](#)
- [39] P. Comelli, P. Ferragina, M. N. Granieri, and F. Stabile, "Optical recognition of motor vehicle license plates," *IEEE Trans. Veh. Tech.*, vol. 44, no. 4, pp. 790-799, November, 1995. [Article \(CrossRef Link\)](#)
- [40] J.-M. Guo and Y.-F. Liu, "License plate localization and character segmentation with feedback self-learning and hybrid binarization techniques," *IEEE Trans. Veh. Tech.*, vol. 57, no. 3, pp. 1417-1424, May 2008. [Article \(CrossRef Link\)](#)



Md. Mostafa Kamal Sarker received his B.Sc degree from Shahjalal University of Science and Technology, Sylhet, Bangladesh in 2008. And in 2013 he received his M.S degree from Chonbuk National University, Republic of Korea. His research interest includes Image Processing, Digital Image Processing, and Machine Learning. His research area includes Artificial Intelligence. Email: rajusust62@gmail.com



Sook Yoon received the B.S., M.S., and Ph.D. degrees in engineering from Chonbuk National University, Jeonbuk, Korea, in 1993, 1995, and 2003, respectively. Until June 2006, she conducted her postdoctoral research work in electrical engineering at the University of California, Berkeley. She is presently an associate professor at Department of Multimedia Engineering, Mokpo National University, Jeonnam, Korea. Her current research interests include image processing, pattern recognition, machine learning, and multimedia computing. Email: syoon@mokpo.ac.kr



Dong Sun Park received the B.S. degree from Korea University, Seoul, Korea, in 1979, and the M.S. and Ph.D. degrees from the University of Missouri, Columbia, in 1984 and 1990, respectively. He is currently a Professor with the Division of Electronics Engineering, Chonbuk National University, Jeonbuk, Korea. His current research interests include image processing, pattern recognition, computer vision, and artificial intelligence. Email: dspark@jbnu.ac.kr