

Research Article

Vehicle Plate Detection in Car Black Box Video

Dongjin Park and Kyungkoo Jun

Department of Embedded Systems Engineering, Incheon National University, Incheon, Republic of Korea

Correspondence should be addressed to Kyungkoo Jun; kjun@inu.ac.kr

Received 26 May 2017; Revised 28 September 2017; Accepted 8 November 2017; Published 28 November 2017

Academic Editor: Constantine Kotropoulos

Copyright © 2017 Dongjin Park and Kyungkoo Jun. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Internet services that share vehicle black box videos need a way to obfuscate license plates in uploaded videos because of privacy issues. Thus, plate detection is one of the critical functions that such services rely on. Even though various types of detection methods are available, they are not suitable for black box videos because no assumption about size, number of plates, and lighting conditions can be made. We propose a method to detect Korean vehicle plates from black box videos. It works in two stages: the first stage aims to locate a set of candidate plate regions and the second stage identifies only actual plates from candidates by using a support vector machine classifier. The first stage consists of five sequential substeps. At first, it produces candidate regions by combining single character areas and then eliminates candidate regions that fail to meet plate conditions through the remaining substeps. For the second stage, we propose a feature vector that captures the characteristics of plates in texture and color. For performance evaluation, we compiled our dataset which contains 2,627 positive and negative images. The evaluation results show that the proposed method improves accuracy and sensitivity by at least 5% and is 30 times faster compared with an existing method.

1. Introduction

Internet services that share user created contents including videos and images have long become a part of people's everyday information and entertainment elements. Proliferation of such services accompanies side effects. Because of heightened awareness about privacy, the exposure of personal information without consent has drawn people's attention more than ever before. For example, it is certainly undesirable if vehicle plates of some people are exposed without their permission in the Internet services such as Google street view [1] and black box video sharing sites [2–4]. Many countries ban sharing of personal information captured in black box videos. For instance, Germany and USA prohibit the distribution of images or videos containing faces and plates without written permission. Thus, it is required to delete or at least obfuscate privacy related data before making them available online [5].

However, such work to eliminate personal information is impracticable if it is performed manually without automation considering the quantity of images and videos that are newly available every day. The privacy information that this paper deals with is vehicle plates captured in black box videos.

The services sharing such videos need to remove plates from uploaded videos. To automate such work, methods for detecting plates play a central role; plate should be located correctly before being removed.

Methods for vehicle plate detection have long been used in various fields such as security control, parking management, and automatic toll systems as a vital prestep before recognizing plate numbers. However, existing methods have limitations by constraints and assumptions particularly regarding location of plates within images, plate sizes, and lighting conditions. Detecting license plates in street view and black box videos provides various challenges: road signs and billboards are similar to plates, plates have different sizes depending on distance and are sometimes rotated or slanted, and their colors change according to lighting.

In this paper, we propose a novel method to detect Korean plates in videos captured by black box cameras. It detects six different types of Korean plates as shown in Figure 1. We aim to develop a scheme which works without assumption about plate locations within images, sizes, and illumination. We design it to be able to detect multiple instances of plates. One of the challenges that we should overcome is the case where boundary of plates is indistinguishable from vehicle



FIGURE 1: Six different types of Korean vehicle plates.

color. Some of existing methods exploit assumption that rectangular boundaries of plates are distinct from background. However, it is not always true. To work with such cases, we propose a bottom-up way. We first detect characters that might constitute plates and then combine them together to form plate-like regions which are later classified.

The paper is organized as follows. Section 2 surveys a list of research efforts for plate detection. Section 3 describes the proposed method in detail and Section 4 presents the performance evaluation results. Section 5 concludes the paper.

2. Related Works

Methods for detecting vehicle license plates from images have been studied in various literature because of their wide applicability. The methods can be largely divided into five groups. In the remaining section, we describe characteristics of each group in detail and discuss their strength and weakness, respectively.

Edge-based methods [7–12] are one of the simplest approaches. They scan images to find areas on which vertical and horizontal edges overlap and, at the same time, of which shapes are rectangles with the ratio of width to height close to those of plates. The methods depend on Sobel filter to detect edges. When searching such areas, Hough transform is adopted in [9, 10] to detect even rotated plates. To determine boundaries of rectangles, connected component analysis (CCA) [11] or template matching [12] is used. In general, the edge-based approaches are simple to implement but require that all edge pixels are connected. Otherwise, disconnected parts of edges are discarded as noise. Other constraints are that plates should have different colors than vehicle and the ranges of plate sizes in images and illumination condition should be known in advance.

Texture-analysis-based methods [13–15] are motivated by an observation that edges from characters appearing on plates have texture characteristics. Thus, detection processes compare edges for possible matching with a set of predefined textures derived from actual plates. For texture matching, various methods are used such as vector quantization [13], Gabor filter [14], and wavelet transform [15].

Multistage approaches [16, 17] consist of a series of steps along which the number of candidate areas which might be plates decreases and only areas with high probability of being plates are left at the end of the steps. Selection processes along steps use feature vectors derived from areas such as Haar [16], covariance descriptor, and HoG descriptor [17]. Use of such feature vectors lifts the constraint that requires that plate colors should be different from vehicle.

Color-based approaches are inspired by an observation that color combination used in plates including characters on them is hardly found in streets. A simple method [18] selects areas of which color combination matches with those of plates by dividing pixels in areas into 13 categories under HLS color model. There are other approaches; neural network is used to classify color distribution [19] and boundaries of plates are detected by colors [20]. These color-based methods are robust to rotation and perspective transformation plates. However, they are subject to lighting and not applicable when colors of vehicles and plates are similar. Regarding such limitation, use of average and standard deviation of hue value distribution of areas is proposed [21].

Character-based methods work in a bottom-up way; they infer plate areas by using information of detected characters. A direct method [22] finds areas containing patterns like digit characters and then use a neural network to determine their likelihood of being plates. A similar method [23] improves accuracy by limiting the range of pattern sizes. For efficiency, another method employs an extra step to exclude nonplate areas by limiting the ratio of width to height [24]. There are other modified methods; a Laplacian filter is used to strengthen character edges [25] and multiple classifiers such as Adaboost and support vector machine (SVM) over Haar feature are employed [6]. This type of methods is susceptible to false positives: nonplate objects that are rectangles with character patterns.

We propose a method that is a hybrid of the multistage approach and the character-based one. A similar one to our method is Ho et al.'s work [6] which shares two-stage structure: a first stage selects candidate regions which have high probability of being plates and a second stage determines actual plates among candidates by using a machine learning

classifier. Another similarity can be found in a two-stage method [26] of Google. It was developed to protect privacy by blurring plates captured in street view images. It has a layer of convolutional neural network (CNN) to detect vehicles and another layer of neural network to locate plates from detected vehicles.

However, similarity ends here. Our method uses a different approach for selecting candidate regions than Ho et al.'s work and is more efficient in terms of the number of candidate regions. Also, we use different feature vectors for a SVM classifier. In terms of complexity, our method is different and simpler than Google's work which employs two CNNs.

3. Vehicle Plate Detection

We develop a novel method to detect vehicle plates from the videos captured by car black boxes. Our approach is motivated by the fact that plates have letter and numeric characters on them. We search in an image to locate characters and merge the character regions to form a plate-like rectangular area. Only the areas that satisfy certain conditions are classified as plates. Such approach is reasonable because Korean vehicle plates have several digit characters evenly spaced. However, black box videos have many nonplate objects that show similar features of digit characters, for example, road signs and billboards. Moreover, the task is challenging because plates have different sizes; some of them are slanted at an angle and are under different illumination.

The proposed method consists of two sequential steps. The first step selects candidate regions which are most likely to be plates; thus it is called a candidate selection step. The second step, a decision step, determines which candidate regions are actual plates by using a machine learning-based classifier, SVM.

Figure 2 shows intermediate results while images are processed through the two steps. From the results, we explain what substeps consist of each step and the overview. Details will be discussed later. Given an input image, character regions are emphasized by strengthening edges as shown in Figure 2(b). Then only the edges that have high probability to belong to character regions are selected as in Figure 2(c). The next step merges neighboring character regions together to form rectangular areas. Among the areas, only those that satisfy shape conditions that characterize license plates are chosen as in Figure 2(d), which are then fed into SVM to determine actual plates as in Figure 2(e). The white rectangles mark final results of plate detection.

We now describe the first step, the candidate selection step, in detail. It consists of five substeps. As the substeps proceed, the number of candidate regions decreases by filtering out nonplate areas through the operations such as morphology, connected component analysis, and region merging. The first substep detects edges from an input image and then sharpens them by convoluting with the Laplacian filter, leaving only pixels on which zero crossing occurs. By this filtering, the edges become more distinct. It can prevent the loss of character regions in the following steps.

The second substep is to remove noise and connect edges by two morphology operations as shown in Figure 3. It

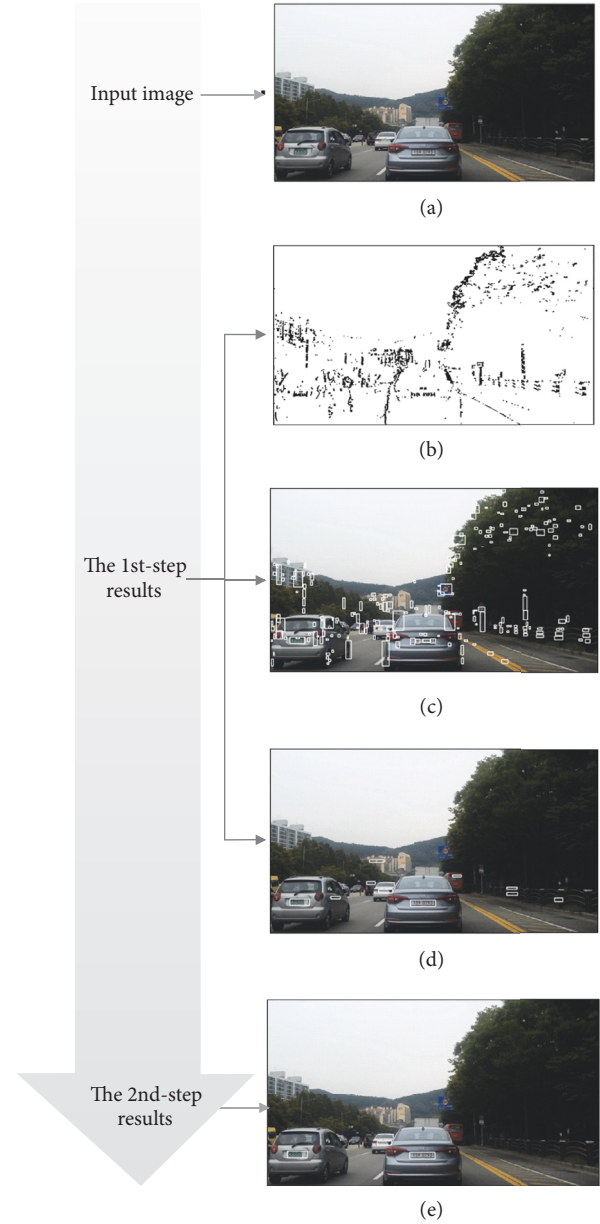


FIGURE 2: The processing sequence of the proposed method.

should be noted that the previous step strengthened not only character edges but also noise. Noise is in general too short edges that cannot constitute character regions. We use two operations in order: opening and closing. The opening removes short edges by erosion and expansion, while the closing merges disconnected but neighboring edges together by expansion and erosion.

In the third substep of Figure 4, we select some of rectangular areas that have high probability that they correspond to a single character region. It is achieved by connecting neighboring edge pixels and finding a minimum rectangular area that includes all the pixels. For this, we use connected component analysis (CCA). However, it returns other rectangular areas as well as actual character regions. Thus, extra

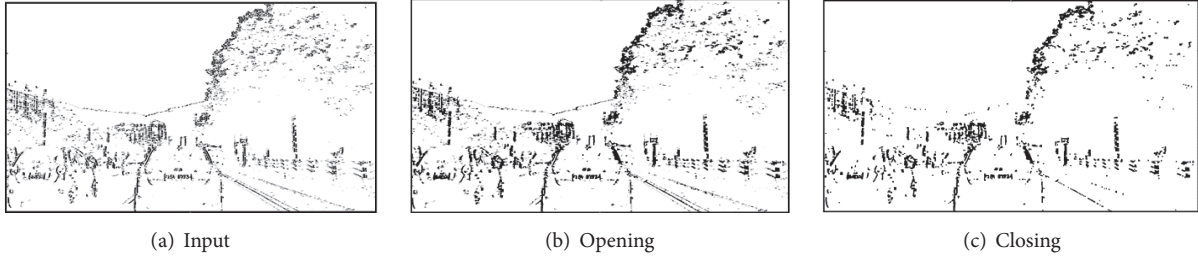


FIGURE 3: Morphology operations to remove noise and strengthen character areas.

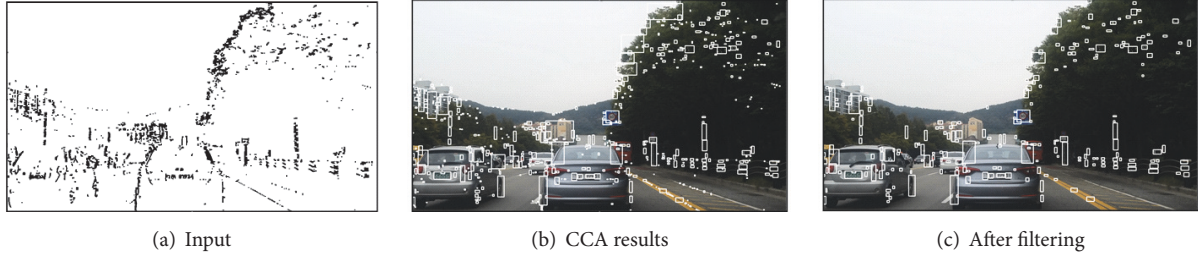


FIGURE 4: Detection of character areas by CCA and filtering.

regions need to be identified. Given a rectangular region, C_i , it is considered as a character region if the following condition is met:

$$R_{\min} \leq \frac{e_i}{A_i} \leq R_{\max}, \quad (1)$$

where e_i is the number of edge pixels in C_i and A_i is its area in pixel. Otherwise, it is classified as noncharacter regions and removed from the results as in Figure 4(c). We determine R_{\min} and R_{\max} from the statistics of actual character regions gathered from our own dataset. For example, we use the threshold range of $[0.2, 0.8]$ which are derived from all possible characters of license plates.

The fourth substep merges neighboring rectangular areas together to form a bigger rectangle that contains all of them. The rationale behind this is that plates consist of a set of neighboring character areas. The decision whether to merge regions is based on the following three conditions. If either one of them is not met, they are not merged. The first condition states that all the regions, c_i , in a set, R_k , are mergeable if the following is met:

$$\frac{|c_i(h) - c_m(h)|}{c_m(h)} < H_{th}, \quad \forall c_i \in R_k, \quad (2)$$

where $c_i(h)$ is the height of a region c_i , $c_m(h)$ is an average height of all regions in a set R_k , and H_{th} is a threshold. By this condition, only the regions with similar heights are merged.

The second condition requires that vertical coordinates of center points should satisfy the following condition:

$$\frac{|c_i[y] - c_m[y]|}{c_m(h)} < V_{th}, \quad \forall c_i \in R_k, \quad (3)$$

where $c_i[y]$ is a vertical center coordinate of a region c_i and V_{th} is a threshold. This condition checks whether regions are

horizontally aligned. And the third condition is that any of two closest regions should be apart by less than a threshold G_{th} times narrower width of two regions as follows:

$$\frac{|c_i[x] - c_j[x]|}{\min(c_i(w), c_j(w))} < G_{th}, \quad i \neq j, \quad (4)$$

$$D(c_i, c_j) < D(c_i, c_m), \quad \forall m \neq j,$$

where $c_i(w)$ is the width of c_i , $c_i[x]$ is the horizontal center coordinate of c_i , and $D(c_i, c_j)$ is the distance between centers of c_i and c_j . This condition checks whether any one of the regions is apart from others in a set. After checking the three conditions, Figure 5(b), for example, shows that the rectangular regions on the license plates in the center bottom area of the image were merged into one bigger rectangle.

The condition checking of the fourth substep, in theory, should be performed on all possible combinations of regions. It, however, causes the resulting computing time to be prohibitively enormous. Thus, we rely on two heuristics to limit the number of possible sets to be checked. Firstly, we limit the size of a set to be minimum two and maximum ten; we assume no plate is larger than ten merged regions. Secondly, we use a backtracking algorithm in such a way that we extend sets, which met the three conditions, by adding a new region and checking if it still holds.

The final substep selects only plate-like regions as shown in Figure 6(b). We define the plate-likeness quantitatively by the following three conditions. Regions that fail to satisfy all the conditions are discarded. Firstly, the ratio of width to height of a region should be within a range of $[K_a, K_b]$ as follows:

$$K_a \leq \frac{c_i(w)}{c_i(h)} \leq K_b, \quad (5)$$

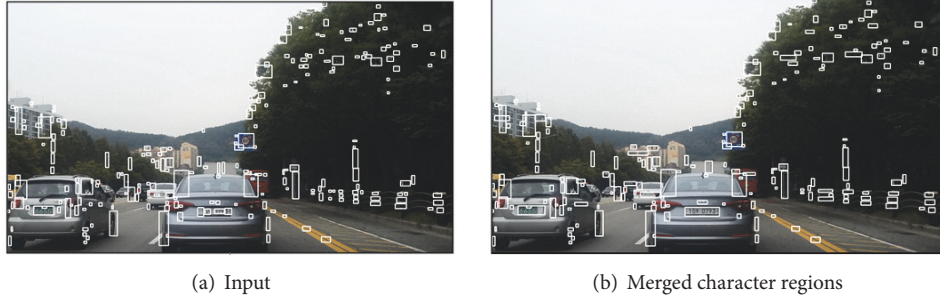


FIGURE 5: Candidate plate regions from merging character areas.

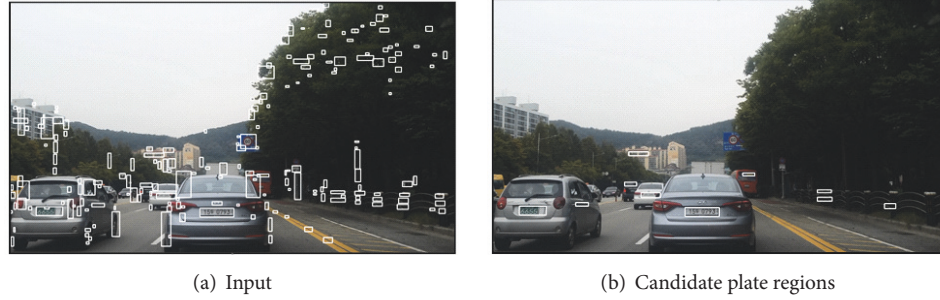


FIGURE 6: Candidate plate regions at the end of the first stage.

where $c_i(W)$ and $c_i(H)$ are width and height of candidate region c_i . Secondly, the ratio of region size to whole image should be within a range of $[M_{\min}, M_{\max}]$ as follows:

$$M_{\min} \leq \frac{c_i(W) \cdot c_i(H)}{I_{sz}} \leq M_{\max}, \quad (6)$$

where I_{sz} is image size. This is because plates in black box videos are limited in their maximum and minimum sizes because of distance to camera and resolution. Thirdly, a center point of a region should be in lower two-thirds segments when dividing image horizontally into three equal segments as follows:

$$c_i[y] \geq \frac{1}{3} \cdot I(H), \quad (7)$$

where $c_i[y]$ is vertical coordinate of center of a candidate region c_i and $I(H)$ is the image height. This is because of an angle by which black box cameras capture vehicles. We determine the range thresholds K_a , K_b , M_{\min} , M_{\max} from the statistics of six different types of plates from our own dataset.

Candidate regions obtained at the end of the first stage are then fed into the second stage, the decision step, which uses a machine learning-based classifier to select only actual plates. For this stage, we use a nonlinear support vector machine (SVM) which takes as input feature vectors derived from candidate regions.

Feature vector is designed to represent edge density distribution and dominant color of region. It has $k (= n + m)$ dimensions where n dimensions are for edge density and m for dominant colors as shown in Figure 7. Before retrieving feature vector, regions which have different sizes

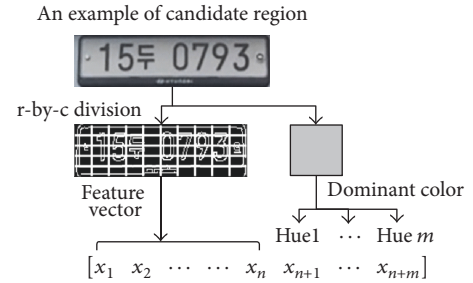


FIGURE 7: An example to show derivation of feature vectors.

are normalized so that they have the width of w pixels while their aspect ratio is kept. Also, the color model of regions is changed to HSV from RGB. This is because HSV is more robust than RGB in the case of illumination change; using the hue component makes algorithms less sensitive to lighting variations.

To calculate the first n values of feature vector, a region is divided into n rectangular blocks, each of which has the same size and, for each block, the ratio of the number of edge pixels to the total number of pixels in a block is calculated. A total of n resulting ratios comprise the first n dimensions. For example, the region in Figure 7 is divided into $n (= r \times c)$ blocks arranged in r rows and c columns. The ratios are stored in feature vector in a row-wise way. The hyperparameter of r and c should be determined empirically by considering both the image resolution and the ratio of width to height of plates. In the experiments of our paper, we used the value of $r = 5$ and $c = 10$.



FIGURE 8: Example images for SVM training.

TABLE 1: Parameter and threshold values used during experiments.

Parameters	Values
R_{\min}, R_{\max}	0.2, 0.8
$H_{\text{th}}, V_{\text{th}}, G_{\text{th}}$	0.25, 0.4, 0.5
K_a, K_b	1.3, 4.1
M_{\min}, M_{\max}	$0.1 \times 10^{-3}, 0.25 \times 10^{-2}$
n, m	50, 3
# of bins, w	256, 200

The remaining m dimensions of feature vector are used to represent m most dominant colors of region. We retrieve dominant colors by using a histogram which has 256 bins representing the entire range of hue values. We choose m bins with the highest frequencies and their corresponding hue values which are $[0, 255]$ are retrieved in frequency order to fill m values of feature vector. In theory, two ($m = 2$) dominant colors are sufficient because plates are composed of two colors: background and characters. However, adding one more piece of color information ($m = 3$) is necessary due to illumination irregularity caused by partial shadow, ambient, diffuse, and specular lights.

After obtaining feature vectors of regions, they are fed into a pretrained SVM to perform binary classification. Given a set of training examples, each labeled as belonging to plates or not, an SVM training algorithm builds a model that can later assign regions to one category or the other, making it a nonprobabilistic binary linear classifier. More details about the SVM are discussed in the next section.

4. Performance Evaluation

We present the results of performance evaluation of the proposed method along with comparison with the work of [6]. For the training of SVM used in our method, we used a total of 86 actual plate images captured from black box videos. The images were selected in such a way that they represent six different types of Korean plates. Also, a total of 137 nonplate images were used from the same videos for the training. Training images have rectangular shape, of which width ranges from 22 to 168. It should be noted that images are resized in such a way that width becomes w before feature vector is retrieved. Examples of training images are shown in Figure 8. The algorithm parameters used in our method are listed in Table 1.

TABLE 2: Confusion matrices from experiment results.

	(a)	
	Prediction	
Our method	Positive ($n_d \geq 1$)	Negative ($n_d = 0$)
Truth		
Positive ($n_p \geq 1$)	1,374	253
Negative ($n_p = 0$)	130	870
	(b)	
	Prediction	
Ho et al.'s work [6]	Positive ($n_d \geq 1$)	Negative ($n_d = 0$)
Truth		
Positive ($n_p \geq 1$)	1,270	357
Negative ($n_p = 0$)	150	850

As testing data, we used two sets of data: positive and negative. The positive data consists of a total of 1,627 driver-view images as shown in Figure 9 that contain at least two vehicles with distinguishable license plates. The negative data is a total of 1,000 images that have unrecognizable license plate or no vehicle at all. Both positive and negative images [27] were captured from six different black box videos having at least 1280-by-720 resolution. The positive images were labeled with coordinates of actual plates.

A confusion matrix is used to analyze the classification performance of the proposed method. We build the matrix in such a way that, given positive data, if the number of detected plates, n_d , and coordinates match its label, we consider it as true positive by increasing the count of true positive (TP) by one. Otherwise false negative (FN) is increased. On the contrary, in the case of negative data, we increase the count of true negative (TN) when n_d is zero. Otherwise, false positive (FP) is increased.

For comparison purpose, we implemented the work of Ho [6] and had it run on the same set of the positive and negative dataset. We chose it because it not only claims over 0.9 of recall rate but also shares a similar two-stage structure to ours; it uses Adaboost to select a set of candidate regions, which are then classified by SVM in the second stage.

Table 2 shows confusion matrices which are the results of experimenting with the test data; the result of the proposed method is on the left and Ho et al.'s work is on the right where n_p and n_d are the numbers of actual and detected plates, respectively. We derive from the matrices a list of performance metrics as shown in Figure 10. The improvement percentages in accuracy, precision, sensitivity, and specificity by the proposed method are 5.22%, 3.12%, 8.15%, and 2.35%, respectively. The largest improvement in sensitivity implies that the ability of our method that detects plates if any is more advanced than Ho et al.'s work.



FIGURE 9: Examples of training and test images used for performance evaluation.

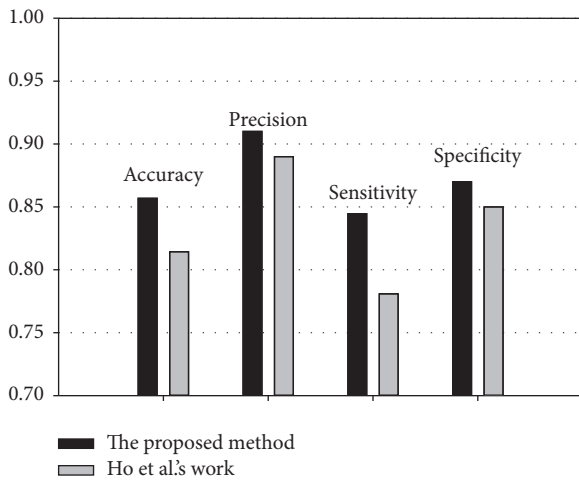


FIGURE 10: Comparison based on various confusion matrix metrics.

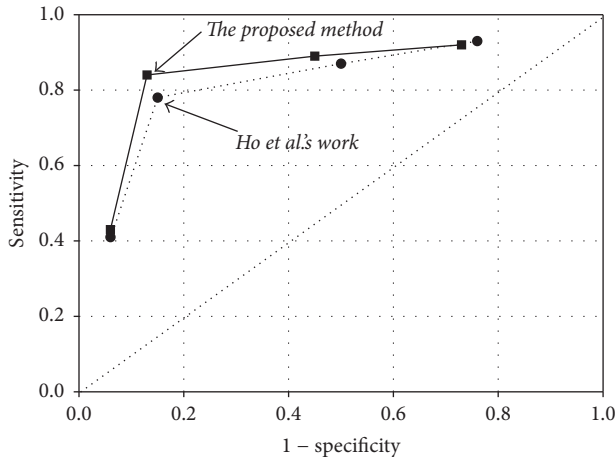


FIGURE 11: Comparison of two methods on receiver operating characteristic (ROC) curve.

More intuitive comparison between the methods comes from receiver operating characteristic (ROC) curve. Figure 11 shows where both methods are positioned within the region of ROC curve. In theory, the closer the position is to the top

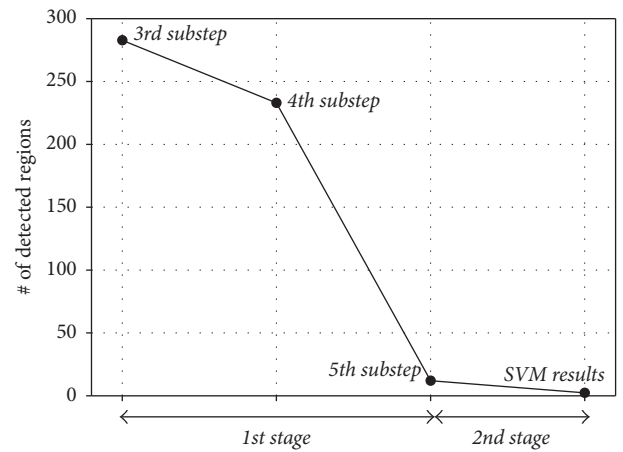


FIGURE 12: The number of detected regions after each substep of the first stage.

left corner, the better the classification performance is. Thus, it is evident that the proposed method is superior to Ho et al.'s work. In the future work, complete ROC curve will be explored by using possible combinations of adjustable threshold parameters. Then, more optimal configuration of parameters can be searched so that it enhances performance further.

We compare how fast the algorithms work. To this end, we measure the elapsed times from the moment when an input image is given until the detection ends. The proposed method takes 0.58 sec on average, while that of Ho et al.'s work takes 17.9 sec; our work is approximately 30 times faster. Possible reasons of such gap are in part because of the difference in the number of candidate regions produced at the end of the first stage. The proposed method yields 12 candidate regions on average, while that of Ho et al.'s work yields over 400. This implies that Ho et al.'s work has 33 times more load than the proposed method. Also, a sliding window that Ho et al.'s work uses to scan over images repeatedly while changing its sizes is another reason for the gap.

We now analyze how the number of intermediate results decreases along the sequential processes of the proposed method. It helps us to catch a glimpse of narrowing-down nature of our method. Figure 12 shows the average numbers

of extracted regions after the substeps when the proposed method works on the test data. The third substep, which is to detect character regions, produces 282.8 regions on average. The fourth substep for merging the regions reduces them to 233. The final fifth step which checks the plate-likeness selects only 12 among them. It implies that the plate-likeness checking is an effective way to filter out nonplate regions. After the second stage involving the SVM classification, the number of detected regions drops down to 1.5, which falls within the range of actual true plate numbers in the test data, [0, 3].

5. Conclusions

We proposed the two-stage method for detecting vehicle plates from car black box videos. The first stage finds a set of candidate regions which have high probability of being plates and the second stage identifies actual plates among candidates by using a binary machine learning classifier, SVM. Our proposed method works in a bottom-up way in the sense that candidate regions are constructed from a set of single character areas. The performance evaluation results showed that our method improves overall detection accuracy, efficiency, and performance compared with an existing work which has similar multistage structure.

In future works, we would improve the method to become less susceptible to rotation or transformation of plates. For this, current scheme using thresholding for ratios or alignment-based filtering will be reexamined. Other further works are related to real-time performance. A quantitative goal is to detect at least five plates in an image of 1280-by-720 resolution in less than 10 msec on an embedded platform with the hardware specification of Raspberry PI 3. We expect such real-time performance to widen application ranges of the proposed method such as unmanned self-driving vehicles and automatic toll systems.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Incheon National University Research Grant in 2013.

References

- [1] L. Vincent, "Taking online maps down to street level," *The Computer Journal*, vol. 40, no. 12, pp. 118–120, 2007.
- [2] <http://www.iatom.co.kr>.
- [3] <https://www.facebook.com/blackboxkorea/>.
- [4] <http://www.craxing.com>.
- [5] <https://productforums.google.com/forum/#!topic/maps/WNqgWUVLJdQ>.
- [6] W. T. Ho, H. W. Lim, and Y. H. Tay, "Two-stage license plate detection using gentle adaboost and SIFT-SVM," in *Proceedings of the 2009 1st Asian Conference on Intelligent Information and Database Systems, ACIIDS 2009*, pp. 109–114, IEEE, Dong Hoi, Vietnam, April 2009.
- [7] M. Sarfraz, M. J. Ahmed, and S. A. Ghazi, "Saudi Arabian license plate recognition system," in *Proceedings of the 2003 International Conference on Geometric Modeling and Graphics*, pp. 36–41, IEEE, London, UK, July 2003.
- [8] D. Zheng, Y. Zhao, and J. Wang, "An efficient method of license plate location," *Pattern Recognition Letters*, vol. 26, no. 15, pp. 2431–2438, 2005.
- [9] V. Kamat and S. Ganesan, "An Efficient Implementation of the Hough Transform for Detecting Vehicle License Plates using DSPs," in *Proceedings of Real-Time Tech. Applicat*, pp. 58–59, 1995.
- [10] T. D. Duan, D. A. Duc, and T. L. H. Du, "Combining Hough transform and Contour Algorithm for Detecting Vehicles' License-Plates," in *Proceedings of the Int. Symp. Intell. Multimedia Video Speech Process*, pp. 747–750, 2004.
- [11] M. Chacon and S. Zimmerman, "License Plate Location based on a Dynamic PCNN Scheme," in *Proceedings of Int. Joint Conf*, pp. 1195–1200, 2003.
- [12] K. Miyamoto, K. Nagano, M. Tamagawa, I. Fujita, and M. Yamamoto, "Vehicle License-Plate Recognition by Image Analysis," in *Proceedings of the Int. Conf. Ind. Electron*, pp. 1734–1738, 1991.
- [13] R. Zunino and S. Rovetta, "Vector quantization for license-plate location and image coding," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 1, pp. 159–167, 2000.
- [14] H. Caner, H. S. Gecim, and A. Z. Alkar, "Efficient embedded neural-network-based license plate recognition system," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 2675–2683, 2008.
- [15] C. Hsieh, Y. Juan, and K. Hung, "Multiple License Plate Detection for Complex Background," in *Proceedings of the Int. Conf. Adv. Inform. Netw. Applicat.*, vol. 2, pp. 389–392, 2005.
- [16] H. Zhang, W. Jia, X. He, and Q. Wu, "Learning-based license plate detection using global and local features," in *Proceedings of the 18th International Conference on Pattern Recognition*, pp. 1102–1105, IEEE, Hong Kong, China, August 2006.
- [17] J. Yoon and D. Kim, "Car License Plate Detector Robust to Shape Variation for Privacy Protection in Road View Service," in *Proceedings of Fall Conference of Korean Institute of Information Scientists and Engineers*, vol. 38, pp. 426–430, 2011.
- [18] X. Shi, W. Zhao, and Y. Shen, "Automatic License Plate Recognition System based on Color Image Processing," *Lecture Notes in Computer Science*, vol. 3483, pp. 1159–1168, 2005.
- [19] E. R. Lee, P. K. Kim, and H. J. Kim, "Automatic recognition of a car license plate using color image processing," in *Proceedings of the 1st IEEE International Conference on Image Processing (ICIP '94)*, vol. 2, pp. 301–305, Austin, Tex, USA, 1994.
- [20] Y. Yang, J. Tian, and N. Liu, "A Vehicle License Plate Recognition System based on Fixed Color Collocation," in *Proceedings of the International Conference on Machine Learning and Cybernetics*, vol. 9, pp. 5394–5397, IEEE, Guangzhou, China, 2005.
- [21] K. Deb and K.-H. Jo, "A vehicle license plate detection method for intelligent transportation system applications," *Cybernetics and Systems*, vol. 40, no. 8, pp. 689–705, 2009.
- [22] J. Matas and K. Zimmermann, "Unconstrained licence plate and text localization and recognition," in *Proceedings of the 8th International IEEE Conference on Intelligent Transportation Systems*, pp. 572–577, IEEE, Vienna, Austria, September 2005.

- [23] S. Draghici, "A neural network based artificial vision system for license plate recognition," *International Journal of Neural Systems*, vol. 8, no. 1, pp. 113–126, 1997.
- [24] F. Alegria and P. S. Girão, "Vehicle plate recognition for wireless traffic control and law enforcement system," in *Proceedings of the 2006 IEEE International Conference on Industrial Technology, ICIT*, pp. 1800–1804, IEEE, Mumbai, India, December 2006.
- [25] J. Kim, "Distortion invariant vehicle license plate extraction and recognition algorithm," *Journal of Korean Contents Association*, vol. 11, no. 3, pp. 1–8, 2011.
- [26] A. Frome, G. Cheung, A. Abdulkader et al., "Large-scale privacy protection in Google Street View," in *Proceedings of the 12th International Conference on Computer Vision*, pp. 2373–2380, IEEE, Kyoto, Japan, October 2009.
- [27] <https://bitbucket.org/kyungkoo/koreanplate>.

