ICIC Express Letters
Volume **14**, Number **8**, August **2020**

ICIC International ©2020 ISSN 1881-803X
pp. 751–759

REAL-TIME LICENSE PLATE DETECTION AND RECOGNITION FOR ODD-EVEN PLATE RATIONING SYSTEM

Carvin Edlin[1], Deny Kiantono[1], Steven Martin[1], Williem[1,2]
Rini Wongso[1] and Meiliana[1]

[1]Computer Science Department
School of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
{ carvin.edlin; deny.kiantono; steven.martin; williem003 }@binus.ac.id
{ rwongso; meiliana }@binus.edu

[2]Verihubs
Noble House 30-103 Jl. Dr.Ide Anak Agung Gde Kav. E 4.2 No. 2, Kuningan
Jakarta 12950, Indonesia
williem@verihubs.com

Received November 2019; accepted February 2020

Abstract. *This paper presents a real-time license plate and recognition to monitor the odd-even plate rationing system. Automatic vehicle detection and license plate recognition are employed to detect the traffic violation. To automatically detect the vehicle, Tiny-DSOD object detection is utilized, which has been proven for real-time performance. Then, Tiny-DSOD and WPOD-NET are combined to detect the unconstrained license plate with better performance and accuracy. Finally, a novel license plate recognition network is introduced to detect the characters in the license plate. Those methods are integrated into a web-based system to detect traffic violations in real time automatically. Experimental results show the performance of each module in the proposed system. The proposed DSWPOD-NET is confirmed to have less parameter and higher accuracy than the state-of-the-art WPOD-NET due to integration between Tiny-DSOD backend and WPOD license plate detector.*

Keywords: Vehicle detection, License plate detection, Optical character recognition, Real-time system

1. **Introduction.** Traffic jam has become a daily problem in fast-developing countries, such as Indonesia, which can slow down economic growth. There have been various approaches to tackle the traffic jam, such as a 3-in-1 car rationing system and an odd-even plate rationing system. However, those approaches become less effective due to the difficulty in detecting the violation. There are needs for the police to capture the drivers who violate the regulation manually. Notwithstanding, it is difficult to monitor the violation due to limited resources. This paper focuses on solving the problem in detecting odd-plate rationing system violations using state-of-the-art vehicle and license plate recognition methods with real-time performance.

Automatic vehicle and license plate recognition have become well-developed research topics in the past few years. The idea of an automatic vehicle and license plate recognition comes from object detection methods. Various lightweight object detection methods have been introduced with high accuracy and real-time performance [1, 2, 3, 4]. Similar methods have been introduced to detect the license plate [5, 6]. However, those methods were failed in the unconstrained condition. Silva and Jung have introduced a warped planar object detection network (WPOD-NET) to detect and rectify the unconstrained license

DOI: 10.24507/icicel.14.08.751

plate. In the early days, license plate recognition utilizes a template matching technique to find a similar pattern for each letter. However, modern object detection methods have changed the trend and become the state-of-the-art methods to automatically recognize characters in license plate [5, 6, 7].

In this paper, we propose a novel system to monitor the odd-even rationing system. It does not use CCTV or cameras mounted on the edge of the highway, but we utilize dashboard cameras mounted on the car due to limited public datasets. The images captured from the dashboard cameras have a nearer view so that the license plate is sufficient to be used as the input. On the other hand, images captured from CCTV have a distant view, which leads to the low resolution of the license plate. At first, we detect the vehicle using Tiny-DSOD (deeply supervised object detector) [2]. Then, we propose a novel deeply supervised warped planar object detection network (DSWPOD-NET), which combines the Tiny-DSOD and WPOD-NET to detect and rectify the license plate. Finally, we develop a novel text recognition network to recognize the characters in the license plate. Experimental results show that the proposed system has better accuracy and real-time performance.

The contributions of this paper are summarized as follows:

- Automatic and real-time system to detect odd-even plate rationing violation.
- A novel real-time algorithm to detect an unconstrained license plate.
- A novel license plate character recognition method to recognize the license plate.

This paper is organized as follows. We introduce the related works in Section 2. Section 3 describes the detail of the proposed method. The analysis of the evaluation results is presented in Section 4. Finally, we give a conclusive remark in Section 5.

2. **Related Works.** In this paper, we focus on developing a novel license plate detection and recognition algorithm. In general, the task can be further broken down into three subtasks, which are license plate detection, character segmentation, and character recognition. However, many approaches do vehicle detection before license plate detection in order to reduce processing time and the number of false positives. In this section, we review several works related to each automatic license plate recognition subtask, both using conventional image processing techniques and the state-of-the-art deep learning approaches.

2.1. **License plate detection.** To locate license plate, edge information was often used [8, 9, 10, 11]. Because of its rectangular shape and fixed aspect ratio, a license plate could be extracted from all possible rectangles in the image. In [12, 13, 14], color was used as feature to locate license plate. In [12], the authors used a color edge detector that was sensitive to only three kinds of edges (i.e., black-white, red-white, green-white). In the experiment, 1088 images taken from various scenes and under different conditions were employed, and it achieved a 97.9% success location rate. However, the implementation was country-specific. Many recent works, both for academic and commercial purposes, had been leaning towards deep learning approaches. In [15], three separately trained deep convolutional neural networks (CNNs) were used. Each model performed separate tasks, which were license plate detection, character detection, and character recognition. However, the paper did not provide a detailed description of the network architecture because it was a commercial method.

A network that was considered state-of-the-art for real-time object detection was Y-OLO [1]. Many works targeted for real-time license plate detection employed YOLO network [5, 6, 7]. In [7], they used YOLOv2 as a black box without any change or refinement. In [5], the network is built based on Fast-YOLO with slight modification. The modification was made so that the network output two classes (car frontal views and plates) instead of 20 classes. Silva and Jung proposed a novel license plate detection that performed well under unconstrained scenarios [7]. Their method could detect and rectify the license plate into the frontal view.

2.2. **Character segmentation.** Character segmentation task was usually performed by turning the detected license plate into a binary image and then applying connected component analysis to extracting the bounding boxes [12, 16, 17]. Binarization was performed to highlight characters and suppress the background. Choosing an appropriate threshold was a crucial task because important information might be lost or results in joined characters. Many techniques have been proposed for choosing the right threshold. In [18], image quality was enhanced before choosing the threshold. In [12], the variable thresholding technique, as proposed in [19], was employed.

In [5], the authors proposed a YOLO-based convolution neural networks used both for character segmentation and recognition. It achieved the incredible result, able to segment 99% of the characters, and correctly recognize 93% of them. In [6, 7], CNN proposed in [5] was employed for segmentation and recognition tasks.

2.3. **Character recognition.** Template matching is a simple technique used for recognition. The extracted character was resized into the same size before template matching is performed [20, 21, 22]. In [20], Hamming distance was used as a similarity measuring technique. In [14], Jaccard value was used instead of Hamming distance. Template matching was prune to fail when the extracted character was different from the template due to any font change, rotation, or noise [23]. Therefore, template matching was not suitable for distorted but readable images.

In [24], the authors proposed random CNNs to extract features and train linear support vector machines (SVMs) using the resulting features. It achieved a recognition rate of over 98% for digits and 96% for letters, significantly better than using image pixels or CNNs trained with backpropagation. In [25], the authors used a one-against-all SVM classifier using a histogram of oriented gradient (HOG) features, leading to a total of 36 SVMs. Deep learning approaches are used in [5, 15, 25]. In [25], the authors used character-sequence encoding CNN with 16 layers for string recognition, avoiding the character segmentation task.

3. **Proposed Method.**

3.1. **System architecture.** In this paper, we introduce a web-based application system to monitor the odd-even rationing policy. The input of this system is the images from cameras mounted on roads that impose odd-even rationing policy. Each frame is then delivered to the backend server to be processed. The backend system is implemented using the Django framework, which is connected to the database. All automated processes are done in the backend system, such as vehicle detection, license plate detection, and license plate recognition modules. The results are then shown in the React-based front end web. Figure 1 shows the overview of the system architecture.

The backend system receives video stream input from the camera, and then processes the video frame from the camera and forwards it to the vehicle detection module. The vehicle detection module is responsible for detecting vehicles contained in the video frame. After the vehicle is detected, this module will crop the image of the vehicle and save the vehicle coordinates. Then the results (cropped images and coordinates) are delivered to the license plate detection module.

From the given vehicle image, the license plate detection module detects the license plate. Then, the image is cropped and rectified to represent the frontal view. The processed license plates are sent to the last module, which is the license plate recognition module. This module is responsible for recognizing the characters that appeared on the

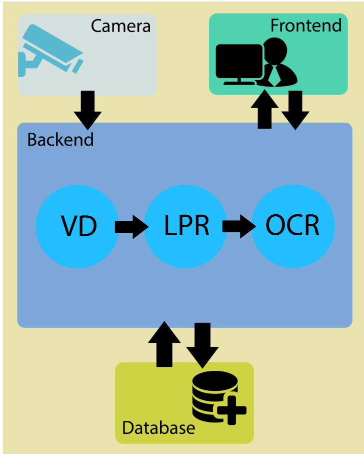

FIGURE 1. The overview of the system architecture

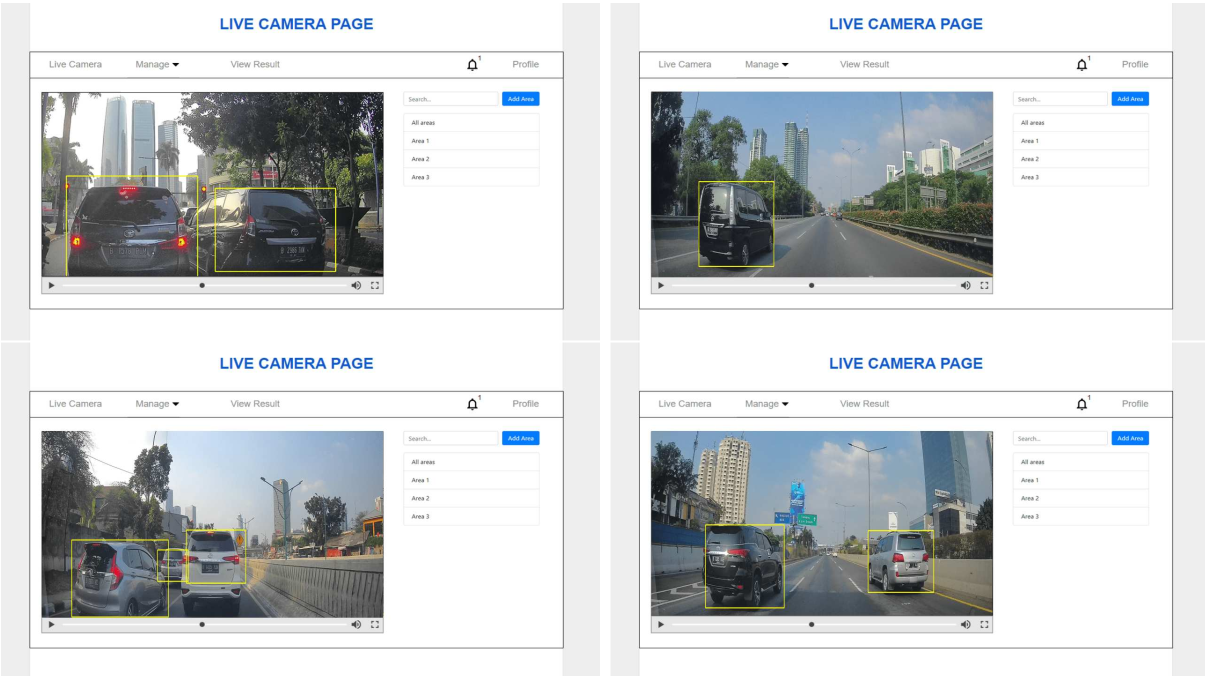

FIGURE 2. The screenshot of front-end web system

plate. Furthermore, it sends the characters back to the backend for further processing. Figure 2 shows the screenshots of the front-end web application.

3.2. **Vehicle detection.** To automatically detect the vehicle, we compare the state-of-the-art lightweight object detection methods that have been trained on accessible object detection datasets, such as ImageNet and PASCAL-VOC. Based on the evaluation, Tiny-DSOD [2] is acknowledged to have the best trade-off between accuracy and computation speed.

3.3. **License plate detection.** The positive detections obtained from the previous module are cropped and fed to this module. One common problem with images acquired from more relaxed scenarios, such as a dashboard camera, is oblique views of the license plate. These distorted license plate images can significantly reduce the accuracy of the license plate recognition module. This issue is addressed in [7] and solved using the proposed warped planar object detection network (WPOD-NET). WPOD-NET regresses affine transformation parameters, which then are used to rectify the license plate area to resemble the frontal view.

Inspired by Tiny-DSOD and WPOD-NET, we develop DSWPOD-NET (deeply supervised warped planar object detection network), a network that is both lightweight and can be used in more relaxed scenarios. The network is created by combining the backbone architecture from Tiny-DSOD with the detection block from WPOD-NET. Figure 3 shows the detail of each module in the DSWPOD-NET. DSWPOD-NET consists of three blocks, which are stem block, feature extractor block, and detection block. The stem block is inspired from DenseNet [26] in which each layer becomes the input of several following layers. It is useful to reduce the image resolution without any loss information. The feature extractor block learns to extract the important information with high depth. Both stem and feature extractor blocks are inspired from Tiny-DSOD method. Finally, the detection block is inspired from WPOD-NET to predict the location of license plate and the rectification matrix. We significantly reduce the number of parameters of WPOD-NET, which is initially 1.6M to only 0.73M. The rectified license plate images are then fed to the final license plate recognition module.

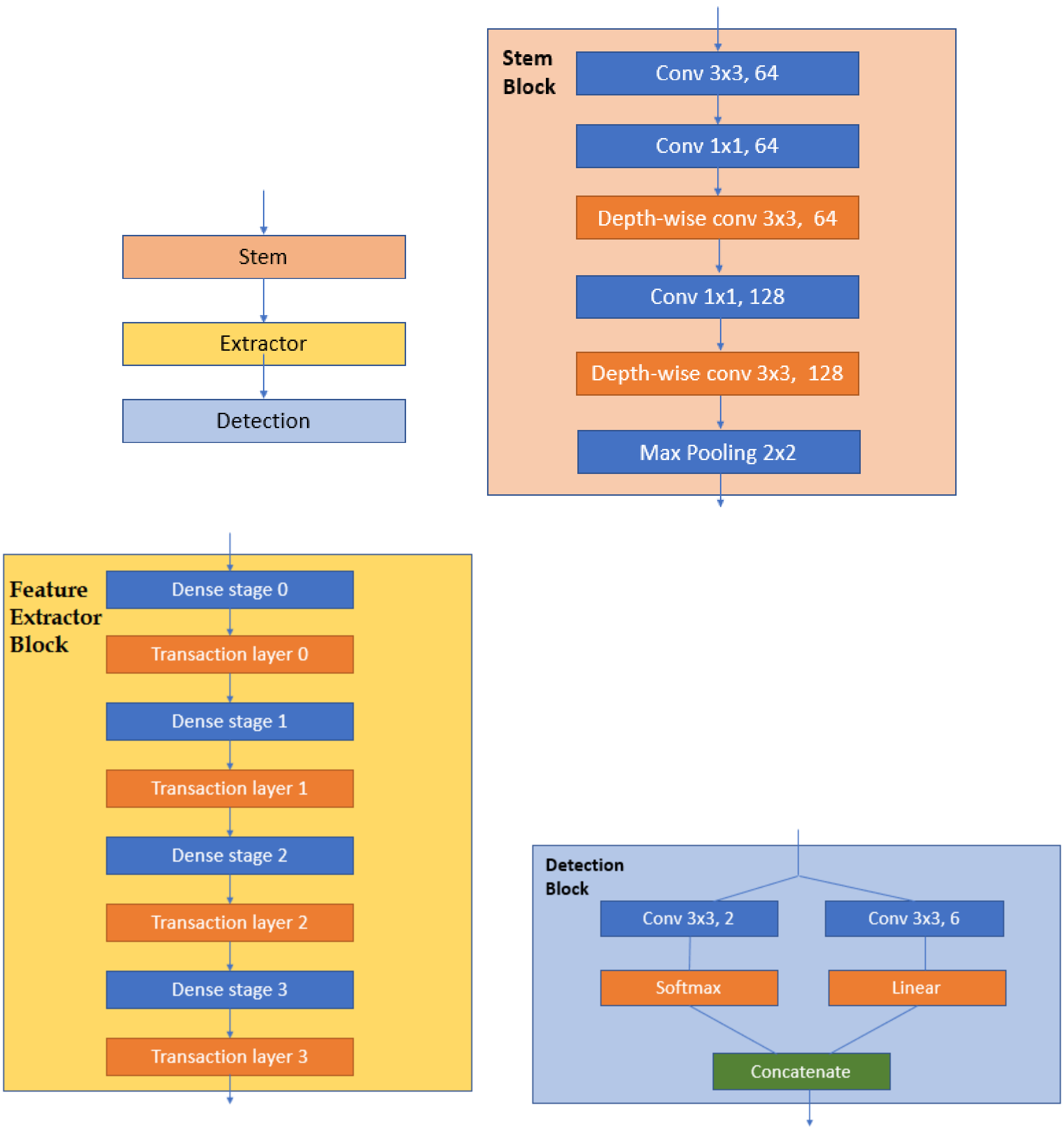

Figure 3. The network architecture of DSWPOD-NET

To train the DSWPOD-NET, we created a dataset consisting of 400 images taken from the dashboard camera. The selected images contain Indonesian license plates, where each image can contain more than one license plate. We manually annotated the four corners of each license plate in the images. Figure 4 shows the examples of annotated Indonesian license plate dataset. Training is done with 43k iterations using mini-batches of size 32. The total iteration is obtained using the early-stopping method. Adam Optimizer is used with a learning rate set to 0.001 which follows the training procedure in [7].

FIGURE 4. Examples of annotated license plate dataset

3.4. **License plate recognition.** For the license plate recognition module, we propose a novel license plate recognition method. On the contrary to previous work, we do not utilize object detection based text recognition but classification based recognition. The backbone network of the license plate recognition is SqueezeNet [27] with a spatial transformer network [28] as the pre-network. Each feature from SqueezeNet is then fed into two bidirectional gated recurrent units (GRUs) [29]. The cost function is the connectionist temporal classification [30]. The network treats the license plate characters as a string and performs recognition on the whole string instead of segmenting each character and performs the recognition separately.

4. **Experimental Results.** Because we targeted for real-time scenarios with the restricted resource, we decided to run the test on CPU machine with the following specification:

- Processor: Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz.
- RAM: 8GB DDR4.
- Operating System: Ubuntu 18.04.3 LTS.

We evaluate each module of the smart transportation system independently. For vehicle detection, the metric being focused is the execution time. The performance between various state-of-the-art object detection networks is compared [1, 2, 3, 4]. For license plate detection, we compare the performance between the proposed DSWPOD-NET with the WPOD-NET by examining the accuracy of the license plate recognition module.

For the testing purpose, we created a dataset consisting of 100 images also taken from the dashboard camera. Images are fed to the vehicle detection networks, and the execution time is computed. Computed execution time is the sum of times needed to load the image, detect vehicle(s), draw the bounding boxes, and save the processed image. For each image, we run the vehicle detection procedure ten times and calculate the average time. The resulting time is the execution time for that specific image.

As can be observed in Figure 5, Tiny-DSOD outperforms other networks in terms of execution time. When tested on the dataset, Tiny-DSOD can perform the vehicle detection in 0.53s on average, better than Grid R-CNN and DSOD-Smallest, which can do the same task on 0.64s. Tiny-YOLO network performed worst and needed 1.72s to do the task on average.

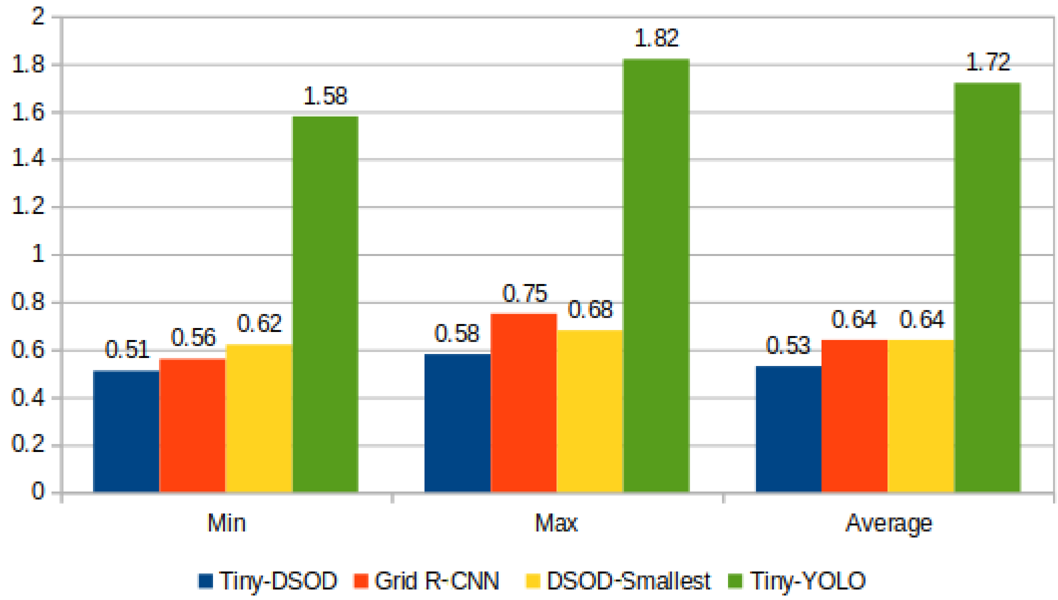

Figure 5. Comparison of computational time (in seconds)

The detection result generated from Tiny-DSOD is being fed to the license plate detection module. There are 175 vehicle images in total that are cropped and contain the license plate. The cropped license plates are then fed to the final license plate recognition module to calculate the accuracy. The accuracy calculation is only taking consideration of license plates in which the characters are still readable. For cases where the vehicle is too far, causing the license plate characters to be not readable is not included. Accuracy is calculated by measuring the number of correctly classified characters.

From the experiment shown in Table 1, it can be concluded that the proposed DSWPOD-NET performed slightly better than WPOD-NET, even with much fewer parameters. DSWPOD-NET correctly classified 428 out of 469 characters, outperforming WPOD-NET with 421 characters classified. It is important to note that both networks have been trained with the identical dataset. Figure 6 shows the illustration of the full-automatic license plate recognition (ALPR) system.

Table 1. Character recognition accuracy

	# Parameters	Accuracy
WPOD-NET	1.6M	86.09%
DSWPOD-NET	0.73M	87.53%

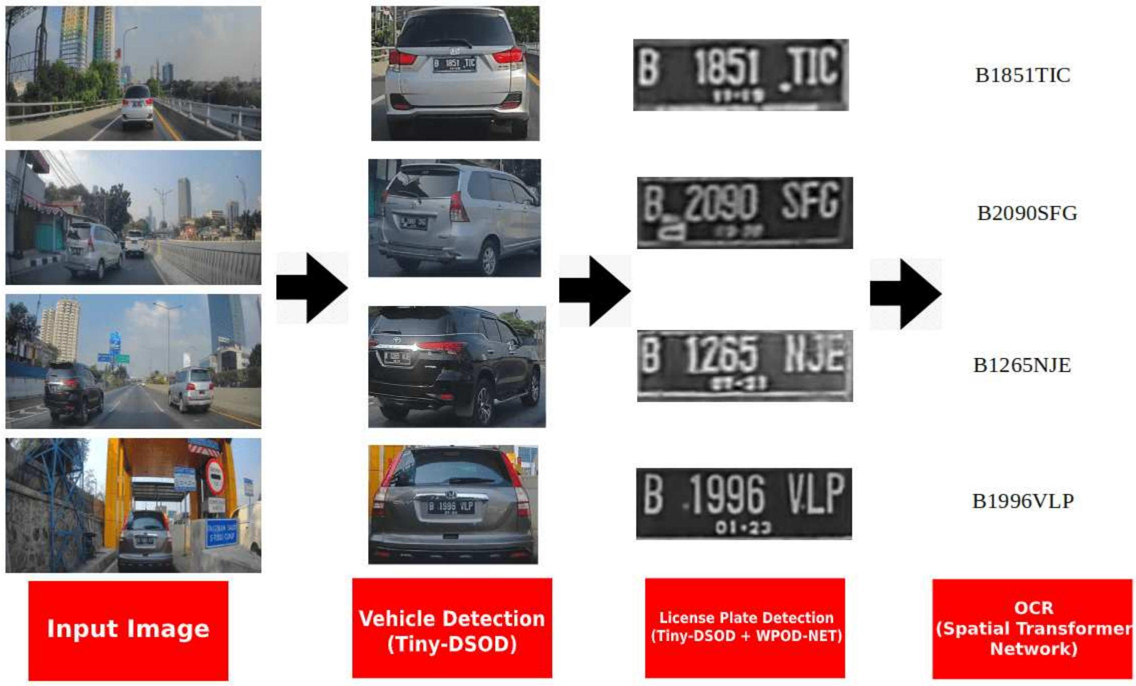

Figure 6. Illustration of the complete ALPR system

5. **Conclusions.** In this paper, we proposed a novel real-time license plate detection and recognition to monitor odd-even plate rationing policy. Tiny-DSOD was chosen to perform vehicle detection due to its fast and accurate performance. Then, DSWPOD-NET was proposed to detect and rectify the license plate simultaneously. Finally, Squeeze-Net based license plate recognition was introduced. All methods were integrated into a web-based application system so that users could monitor when the violation occurred. Experimental results confirmed that the proposed methods achieved better accuracy and faster performance. The limitation of the proposed web-based system is the high latency of data transmission. In the future, we are going to develop a system that requires less data transmission by using embedded hardware. In addition, an end-to-end lightweight license plate recognition method is considered as our future work.

REFERENCES

[1] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, You only look once: Unified, real-time object detection, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.779-788, 2016.
[2] Y. Li, J. Li, J. Li and W. Lin, Tiny-DSOD: Lightweight object detection for resource restricted usage, *Proc. of the British Machine Vision Conference*, 2018.
[3] X. Lu, B. Li, Y. Yue, Q. Li and J. Yan, Grid R-CNN, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.7363-7372, 2019.
[4] Z. Shen, Z. Liu, J. Li, Y. G. Jiang, Y. Chen and X. Xue, DSOD: Learning deeply supervised object detectors from scratch, *Proc. of the IEEE International Conference on Computer Vision*, pp.1937-1945, 2017.
[5] S. M. Silva and C. R. Jung, Real-time Brazilian license plate detection and recognition using deep convolutional neural networks, *Proc. of the SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pp.55-62, 2017.
[6] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, W. R. S. G. R. Goncalves and D. Menotti, A robust real-time automatic license plate recognition based on the YOLO detector, *Proc. of the International Joint Conference on Neural Networks*, 2018.
[7] S. M. Silva and C. R. Jung, License plate detection and recognition in unconstrained scenarios, *Proc. of the European Conference on Computer Vision*, pp.580-596, 2018.

[8] H. Bai and C. Liu, A hybrid license plate extraction method based on edge statistics and morphology, *Proc. of the International Conference on Pattern Recognition*, pp.831-834, 2004.
[9] 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.
[10] S.-Z. Wang and H.-J. Lee, Detection and recognition of license plate characters with different appearances, *Proc. of the IEEE International Conference on Intelligent Transportation Systems*, pp.979-984, 2003.
[11] F. Faradji, A. H. Rezaie and M. Ziaratban, A morphological-based license plate location, *Proc. of the IEEE International Conference on Image Processing*, pp.57-60, 2007.
[12] S.-L. Chang, L.-S. Chen, Y.-C. Chung and S.-W. Chen, Automatic license plate recognition, *IEEE Trans. Intelligent Transportation Systems*, vol.5, no.1, pp.42-53, 2004.
[13] A. H. Ashtari, M. J. Nordin and M. Fathy, An Iranian license plate recognition system based on color features, *IEEE Trans. Intelligent Transportation Systems*, vol.15, no.4, pp.1690-1705, 2014.
[14] E. R. Lee, P. K. Kim and H. J. Kim, Automatic recognition of a car license plate using color image processing, *Proc. of the IEEE International Conference on Image Processing*, 1994.
[15] S. Z. Masood, G. Shu, A. Dehghan and E. G. Ortiz, License plate detection and recognition using deeply learned convolutional neural networks, *arXiv Preprint, arXiv:1703.07330*, 2017.
[16] D. Llorens, A. Marzal, V. Palazon and J. M. Vilar, Car license plates extraction and recognition based on connected components analysis and HMM decoding, *Proc. of the Iberian Conference on Pattern Recognition and Image Analysis*, pp.571-578, 2005.
[17] B.-F. Wu, S.-P. Lin and C.-C. Chiu, Extracting characters from real vehicle licence plates out-of-doors, *IET Computer Vision*, vol.1, no.1, pp.2-10, 2007.
[18] P. Comelli, P. Ferragina, M. N. Granieri and F. Stabile, Optical recognition of motor vehicle license plates, *IEEE Trans. Vehicular Technology*, vol.44, no.4, pp.790-799, 1995.
[19] Y. Nakagawa and A. Rosenfeld, Some experiments on variable thresholding, *Pattern Recognition*, vol.11, no.3, pp.191-204, 1979.
[20] M. Sarfraz, M. J. Ahmed and S. A. Ghazi, Saudi Arabian license plate recognition system, *Proc. of the International Conference on Geometric Modeling and Graphics*, 2003.
[21] K. Kanayama, Y. Fujikawa, K. Fujimoto and M. Horino, Development of vehicle-license number recognition system using real-time image processing and its application to travel-time measurement, *Proc. of the IEEE Vehicular Technology Conference*, 1991.
[22] K. Miyamoto, K. Nagano, M. Tamagawa, I. Fujita and M. Yamamoto, Vehicle license-plate recognition by image analysis, *Proc. of the International Conference on Industrial Electronics, Control and Instrumentation*, 1991.
[23] M.-S. Pan, J.-B. Yan and Z.-H. Xiao, Vehicle license plate character segmentation, *International Journal of Automation and Computing*, vol.5, no.4, pp.425-432, 2008.
[24] D. Menotti, G. Chiachia, A. X. Falcao and V. J. O. Neto, Vehicle license plate recognition with random convolutional networks, *Proc. of the SIBGRAPI Conference on Graphics, Patterns and Images*, 2014.
[25] G. R. Gonçalves, D. Menotti and W. R. Schwartz, License plate recognition based on temporal redundancy, *Proc. of the IEEE International Conference on Intelligent Transportation Systems*, 2016.
[26] G. Huang, Z. Liu, L. V. D. Maaten and K. Q. Weinberger, Densely connected convolutional network, *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.2261-2269, 2017.
[27] N. F. Iandola, M. W. Moskewicz, L. Ashraf, S. Han, W. H. Dally and K. Keutzer, SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and $<$ 1MB model size, *CoRR*, Abs/1602.07360, 2016.
[28] M. Jaderberg, K. Simonyan, A. Zisserman and K. Kavukcuoglu, Spatial transformer network, *Proc. of the Conference on Neural Information Processing Systems*, pp.2017-2025, 2015.
[29] K. Cho, B. van Merriënboer, C. Gulcehre, F. Bougares, H. Schwenk and Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, *Proc. of the EMNLP*, 2014.
[30] A. Graves, S. Fernandez and F. Gomze, Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks, *Proc. of the International Conference on Machine Learning*, pp.369-376, 2006.