# Automatic License Plate Recognition for Parking System using Convolutional Neural Networks

Joshua
*Faculty of Information Technology*
*Universitas Tarumangara*
Jakarta, Indonesia
jsshua11@gmail.com

Janson Hendryli
*Faculty of Information Technology*
*Universitas Tarumangara*
Jakarta, Indonesia
jansonh@fti.untar.ac.id

Dyah E. Herwindiati
*Faculty of Information Technology*
*Universitas Tarumangara*
Jakarta, Indonesia
dyahh@fti.untar.ac.id

***Abstract*—Traditionally, a parking staff enters the license plate of the outgoing vehicle in the system to generate the parking tickets. To improve the efficiency of the system, we propose an automatic vehicle license plate numbers recognition to automatically detect and record the plate numbers to the system eliminating the manual entry. The system consists of a YOLO model to automatically detect the license plate from the image of a vehicle, some preprocessing and image segmentation process to detect the digits in the license plate, and finally, a ResNet model to classify the plate numbers. The license plates used in this research are from Indonesia. From the experiments, we find that the YOLO can accurately detect the license plate with a high degree of confidence. Yet, the ResNet model achieves around 80% accuracy from the validation data. Despite the high accuracy, the model can sometimes wrongly classify plate numbers due to noises from the segmentation process, non-standard or damaged plates, and similarly looking digits.**

***Keywords—convolutional neural networks, license plate recognition, YOLO***

## I. INTRODUCTION

The traditional parking system commonly used in Indonesia is where the parking attendant enters the license plate of the outgoing vehicle manually to the system. It aims to record the transaction and to ensure that the outgoing vehicle is the correct vehicle that is recorded in the parking ticket for the security reason. Manually entering the license plate number is a tedious task and prone to mistake, since it depends on the concentration of the staff who handles a large number of transactions. Furthermore, entering the plate numbers takes time and relatively inefficient.

With the current development of computer vision technology, we propose a license plate recognition system to automatically recognize plate numbers of incoming and outgoing vehicles from a static camera positioned in the parking area entrance. The recognition system is divided into two main parts. The first part is responsible for detecting the license plate from an image of a vehicle. The second part of the system recognizes the numbers of the detected plate and record it into the parking system.

Past research employs a combination of Hough transform, contour algorithm, and hidden Markov model for recognition of Vietnamese license plate [1]. Their system, although performs well on various types of plates, even on scratched plates, still cannot deal with bad quality plates. In [2], the color and shape information of the license plate is used to segment the plate numbers and a minimum Euclidean distance-based template matching is used to detect the plate numbers. An extensive review of the automatic license plate recognition can be found in [3].

While many past research have study machine learning models for license plate recognition, this research aims to develop specifically a parking system that utilized an automatic license plate recognition to save time and minimize typing error by the parking attendant. The methods used in the recognition system are based on deep learning approaches, namely the YOLO algorithm and residual network architecture, due to their speed and performance in object detection and recognition tasks.

## II. METHODS

Our system works initially by employing a license plate detector using the YOLO algorithm. Once the license plate is detected and cropped, several image processing methods, such as grayscaling, binarization, and segmentation, are applied to get each license plate digits. Finally, a ResNet model is used to classify each digit.

**Convolutional neural networks**, usually abbreviated as CNN or convnet, is a class of deep neural networks [4] which typically consists of three types of layers: the convolutional, pooling, and fully-connected layer [5]. The convolutional layer acts as a feature extractor that automatically finds the best feature to represent the image. Meanwhile, the pooling layer has an effect of downsampling the feature maps. Finally, the fully-connected layer is responsible to classify the image.

**You only look once** or YOLO is a state-of-the-art real-time object detection system that utilizes a single neural network and fully-convolutional model [6]. Past research have used YOLO for vehicle detection [7], security checks [8], and also in a multi-class multi-object tracker [9]. For this research, we employ YOLOv2 for license plate detection in an image. The implementation of YOLOv2 can be found in https://pjreddie.com/darknet/yolov2/. It has several advantages compared to the earlier version of YOLO or other object detectors, namely faster detection and better performance with fewer localization errors. Basically, YOLOv2 implements similar architecture to CNN, though it only uses the convolution and pooling layer. To improve the convergence rate, it also applies batch normalization on all of the convolutional layers and removes dropout from the model.

978-1-7281-7071-8/20/$31.00 ©2020 IEEE 13-14 August 2020

Authorized licensed use limited to: Polytechnic University of Bucharest. Downloaded on March 17,2025 at 19:15:50 UTC from IEEE Xplore. Restrictions apply.

**Image segmentation** process aims to crop each digit from the detected license plate. Initially, the cropped license plate image is converted to grayscale. The next step is to convert the pixel image into a binary image. This step is known as image thresholding or binarization. After that, we find the contour and apply a condition that for contours with heights in the range of 26% to 50% of the image and with the total area of 500 to 20,000 pixels are detected as the license plate digits.

**Residual networks** or ResNet [10] is a CNN model architecture used to classify the digits once the license plate digits are cropped. The ResNet is an improvement of CNN which introduces the residual learning to address the difficulty of training very deep neural networks. Let $(\mathbf{x})$ denotes an underlying mapping to be fit in a neural network layers with $\mathbf{x}$ denotes the inputs to the layers, the residual function approximated by the layers is $(\mathbf{x}) := (\mathbf{x})\ \mathbf{x}$ [10]. This connection is also sometimes called a shortcut connection. The output of the ResNet is the classification of the license plate digits.

## III. EXPERIMENTS

### *A. Data*

We arbitrarily collect 500 license plate images of cars and motorcycles in a parking area in Jakarta, Indonesia with various, but not exhaustive, lighting conditions. It should be noted that the data do not cover all types of additional license plate accessories, e.g. acrylic case, and all possible camera orientation, e.g. tilting and angles. Fig. 1 shows the example of the license plate images in the data set. The standard license plate in Indonesia starts with one or two alphabet characters denoting the area where the vehicle is registered, followed by one to four numerical digits and one to three alphabet characters. For example, in Fig. 1 above, both plates are registered in the Jakarta area since the starting character is ”B”. The plates are followed by four (3009) and three (100) digits, respectively, and ends with four characters (”USC” and ”JAP”). Unfortunately, our data set only covers privately owned vehicle and there is no scenario where the plate does not have a last letter.

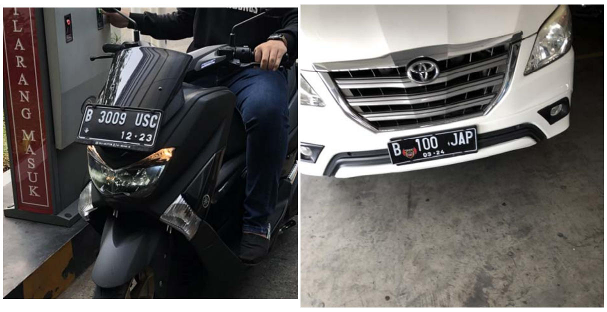
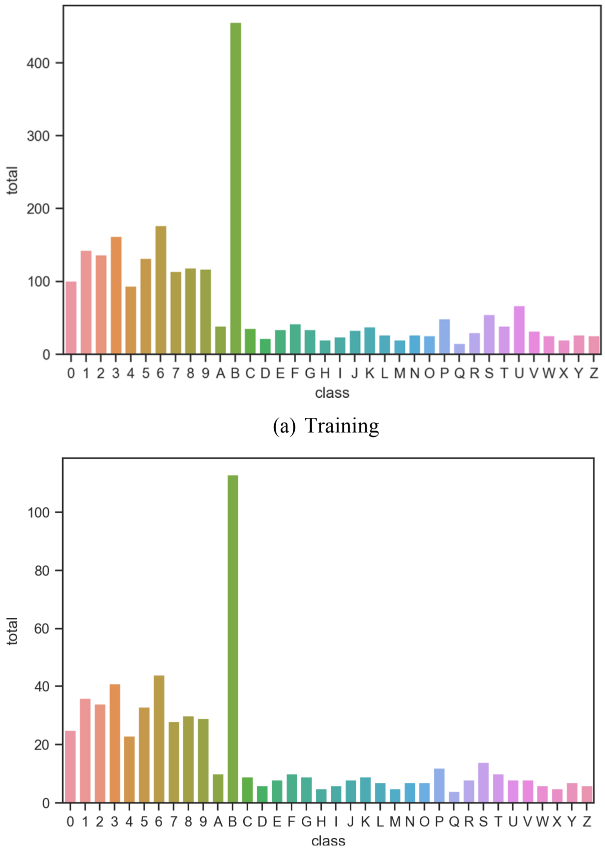


Fig. 1. Examples of the license plate in the data set.

For testing the license detection and the overall recognition system, we use 50 license plate images from the data set. Moreover, Fig. 2 shows the proportion of the training and validation data for the ResNet model used in the license plate digits classification. The data set contains more examples for the ”B” class since the data are collected in the Jakarta, Indonesia area. Despite that, our data set still contain several license plate originally from another area of Indonesia, e.g. Banten (”A”), Sumatera (”BE”, ”BG”, ”BN”, and ”BP”), West Java (”D”, ”F”, and ”T”), Central Java (”G”, ”H”, and ”R”), Kalimantan (”KB” and ”KH”), and so on.



(a) Training

(b) Validation

Fig. 2. Total number of data per class for the training and validation of the ResNet model.

### *B. Experimental Settings*

For the YOLOv2, we use the implementation, default configuration, and weights (transfer learning) from https://github.com/AlexeyAB/darknet. The YOLO model is trained in 100,000 epochs and default 0.2 for the threshold. The ResNet50 from the Keras library is used for the residual network. The model is pre-trained using the ImageNet data set and we add 3 fully-connected layers for the classification of the license plate digits. Furthermore, we also employ a dropout of 0.2 and image augmentations. The model is trained using Adam optimizer with a learning rate of 1e-5 in 50 epochs. It should also be noted that all of the input images are resized to 175 x 175 pixels. Additionally, the OpenCV library is used for the image preprocessing and segmentation. Both models is trained in 64GB RAM and Intel i7-7700 system with two GPUs, the Titan X and Titan Xp. We also implement a simple

Authorized licensed use limited to: Polytechnic University of Bucharest. Downloaded on March 17,2025 at 19:15:50 UTC from IEEE Xplore. Restrictions apply.

parking web application to demonstrate the automatic license plate recognition system.

## IV. RESULTS AND DISCUSSIONS

Fig. 3 shows the result of license plate detection using YOLOv3. The YOLO algorithm returns the upper-left coordinates, length, and width of the detected object, in this case, the license plate, with some degree of confidence. Overall, the result of the detection is very satisfactory with the average degree of confidence around 0.949 for all test images.

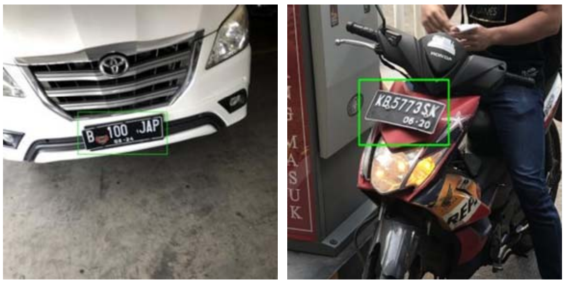

(a) (b)

Fig. 3. Examples of detected license plate from YOLOv2 surrounded by the green bounding boxes.

The next step is to crop the detected license plate and employ the image segmentation to retrieve the digits in the plate as in Fig. 4. The individual digits are then cropped and feed into the ResNet for the digit classification.

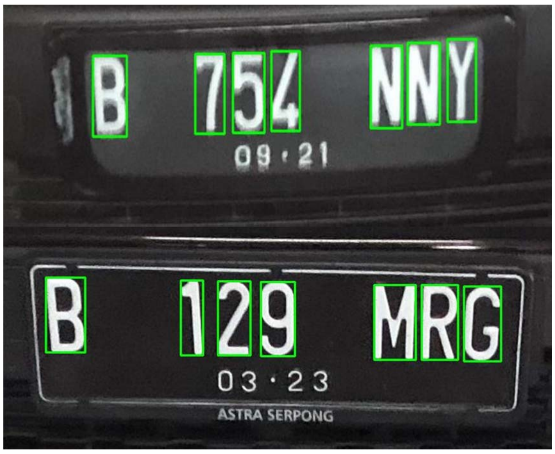


Fig. 4. Segmentation results of the detected license plate

Fig. 5 shows the graphics of the loss and accuracy of the model. During the training, we save the weights which returns the best validation accuracy and used it for the license plate recognition system. The model has 0.80159 validation accuracy.

We develop a web-based parking application that uses the trained models to automatically detect and recognize the license plate number from a web-cam. Fig 6 shows the interface of the application when the vehicle is checking out. The license plate is automatically recognized by the system and compared with the image when the vehicle is checking in to the parking area.

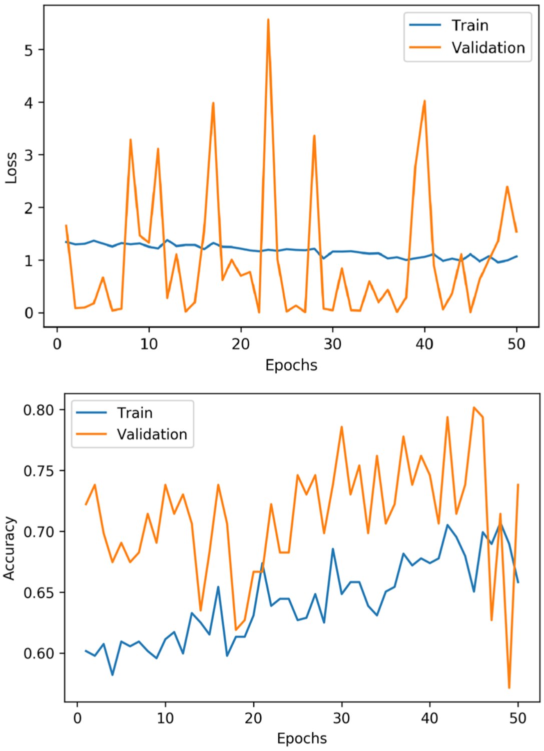
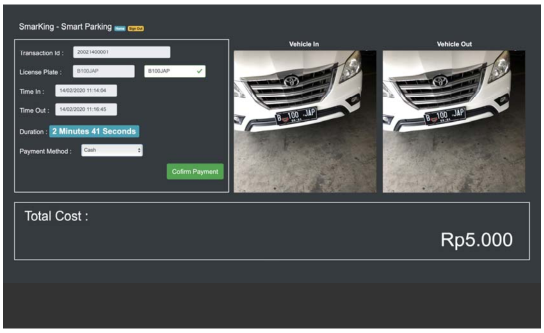


Fig. 5. The graphics for the loss and accuracy of the ResNet model



Transaction Id : 20021400001

License Plate : B100JAP B100JAP ✓

Time In : 14/02/2020 11:14:04

Time Out : 14/02/2020 11:16:45

Duration : 2 Minutes 41 Seconds

Payment Method : Cash

Total Cost : Rp5.000

Fig. 6. The interface of the parking application

The application is tested functionally by observing the traffic of incoming and outgoing vehicles in a parking area and capturing the image of the vehicle (license plate). We find several scenarios where our system fails to recognize the license plate numbers. Overall, the system can detect the plate on the vehicle with the YOLO algorithm, but the segmentation process sometimes fails to extract the plate numbers. For example, as in Fig. 7, the segmentation algorithm is affected by noises in the image and wrongly detect the license plate numbers.

We also find that the ResNet model incorrectly classify several images, such as in the Fig. 7, the ”SIC” plate is classified as ”SIG” and the ”BOS” plate is classified as ”B0S”

978-1-7281-7071-8/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: Polytechnic University of Bucharest. Downloaded on March 17,2025 at 19:15:50 UTC from IEEE Xplore. Restrictions apply.

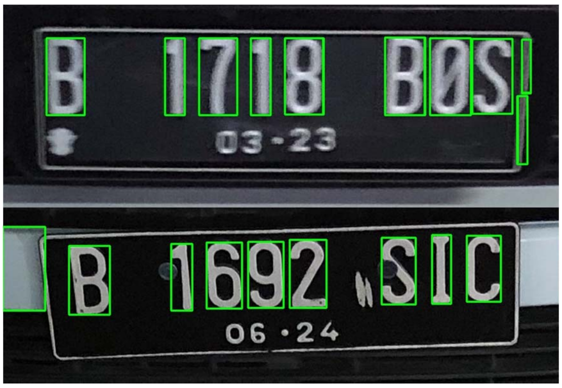


Fig. 7. Example of cases when the system encounters failure in the license plate numbers segmentation

(with the alphabet ”O” is categorized as the number zero ”0”). The latter is partly due to the non-standard license plate. We also find that the model cannot accurately classify the license plate numbers from damaged plates.

## V. CONCLUSIONS

In order to increase the efficiency of the parking system, we explore an automatic system to detect the license plate numbers in vehicles. Our system employs the YOLO algorithm and ResNet model to detect and classify the license plate numbers. The performance of the YOLO algorithm is deemed satisfactory where it can detect the license plate in several testing scenarios. The ResNet model achieves around 80% accuracy from the validation data, but when tested in the system, it can sometimes inaccurately classify the license plate numbers, largely due to noises affecting the segmentation process, and also because of some non-standard or damaged plates.

For future works, directly using YOLO for detecting plate numbers can be considered to eliminate the image segmentation process. Moreover, collecting more license plate data can also be done to get more balance data for all digits and alphabets.

## ACKNOWLEDGMENT

This research is supported by the donation of the Titan X and Titan Xp GPU from NVIDIA Corporation.

## REFERENCES

[1] D. D. Tran, L. H. D. Tran, V. P. Tran, and V. H. Nguyen. ”Building an automatic vehicle license plate recognition system,” in Proc. Int. Conf. Computer Science RIVF, 2005, vol. 1, pp. 59-63.

[2] X. Shi, W. Zhao, and Y. Shen. ”Automatic license plate recognition system based on color image processing,” in Proc. Int. Conf. on Computational Science and its Applications, 2005, pp. 1159-1168.

[3] S. Du, M. Ibrahim, M. Shehata, and 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, 2012.

[4] B. A. Ashqar and S. S. Abu-Naser. ”Identifying images of invasive hydrangea using pre-trained deep convolutional neural networks,” Int. J. Academic Engineering Research, vol. 3, no. 3, pp. 28-36, 2019.

[5] R. Yamashita, M. Nishio, R. K. G. Do, et al. ”Convolutional neural networks: an overview and application in radiology,” Insights Imaging, vol. 9, pp. 611–629, 2018.

[6] J. Redmon and A. Farhadi. ”YOLO9000: better, faster, stronger,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2017, pp. 7263-7271.

[7] W. Yang, J. Zhang, H. Wang, and Z. Zhang ”A vehicle real-time detection algorithm based on YOLOv2 framework,” in Proc. Real-Time Image and Video Processing, May. 2018.

[8] Z. Liu, J. Li, Y. Shu, and D. Zhang, ”Detection and recognition of security detection object based on YOLO9000,” in Proc. Int. Conf. Systems and Informatics, Nanjing, 2018, pp. 278-282.

[9] K. Jo, J. Im, J. Kim, and D. Kim, ”A real-time multi-class multi-object tracker using YOLOv2,” in Proc. IEEE Int. Conf. on Signal and Image Processing Applications, Kuching, 2017, pp. 507-511.

[10] K. He, X. Zhang, S. Ren, and J. Sun. ”Deep residual learning for image recognition,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition, 2016, pp. 770-778.

Authorized licensed use limited to: Polytechnic University of Bucharest. Downloaded on March 17,2025 at 19:15:50 UTC from IEEE Xplore. Restrictions apply.