

# CHARACTER RECOGNITION FOR INDONESIAN LICENSE PLATE BY USING IMAGE ENHANCEMENT AND CONVOLUTIONAL NEURAL NETWORK

## PENGENALAN KARAKTER UNTUK PLAT NOMOR KENDARAAN DENGAN MENGGUNAKAN IMAGE ENHANCEMENT DAN CONVOLUTIONAL NEURAL NETWORK

Sahid Bismantoko<sup>a</sup>, M. Rosyidi<sup>a</sup>, Umi Chasanah<sup>a</sup>, Adityo Suksmono<sup>a</sup>, Tri Widodo<sup>a</sup>

<sup>a</sup> Center of Technology for System and Infrastructure of Transportation  
Agency for the Assessment and Application of Technology, Tangerang Selatan, Indonesia  
e-mail: sahid.bismantoko@bppt.go.id, m.rosyidi@bppt.go.id, umi.chasanah@bppt.go.id,  
adityo.suksmono@bppt.go.id, tri.widodo@bppt.go.id

### Abstract

Many Intelligent Transport System technology have been applied in real world problems such as traffic monitoring, parking management, toll collection, law enforcement. ALPR system is one of the ITS technologies that is widely applied, however this ALPR system can not produce faultless recognition yet, especially for Indonesia license plate. In this research, image enhancement and Convolution Neural Network are proposed to the character recognition. The dataset used in this research are Indonesia license plate. The first step is train dataset to recognize character and evaluate the model with recall, precision, and f-1 score from test dataset. The model achieves accuracy and loss just over 0.99 and just below 0.01 on validation dataset respectively.

Key Words : ALPR; ITS; Recall; Precision; F-1 Score; Accuracy; Loss.

### Abstrak

*Intelligent Transport System sudah diterapkan dalam berbagai aplikasi, diantaranya untuk memantau kondisi lalu lintas, pengelolaan lahan parkir, penerapan pada jalan toll, penegakkan hukum. Sistem ALPR (Pengenalan Tanda Nomor Kendaraan Bermotor) adalah merupakan salah satu teknologi yang banyak diterapkan pada bidang ITS, namun demikian sistem ALPR masih belum dapat mendeteksi dan mengenali plat kendaraan secara baik tanpa kesalahan, terutama untuk karakteristik plat kendaraan di Indonesia. Pada riset ini dilakukan pengenalan karakter dengan menggunakan teknik image enhancement dan convolutional neural network. Dataset yang digunakan pada penelitian ini merupakan plat kendaraan bermotor Indonesia. Langkah pertama adalah melakukan pelatihan dengan menggunakan dataset yang sudah ditentukan dan melakukan pengenalan karakter. Selanjutnya melakukan evaluasi model yang dihasilkan dari training dengan mengambil nilai precision, recal dan f1-score dari Test dataset. Model ini dapat mencapai nilai akurasi lebih dari 0,99 dan mendapatkan nilai loss dibawah 0,01 pada validation set.*

*Kata kunci : ALPR; ITS; Recall; Precision; F-1 Score; Akurasi; Loss*

Received : 09 July 2019, Revised : 03 August 2020, Accepted : 07 August 2020

## INTRODUCTION

The license plate on a vehicle is a unique identity on the vehicle that can be an identification on a vehicle. The detection of a plate number using computer vision approach is an important component in the application of ITS (Intelligent Transport System) to build effective and efficient on transportation system. ALPR (Automatic License Plate Recognition) is a system that can be applied to traffic monitoring, parking management, toll collection, law enforcement, and others. The characteristic of Indonesia license plate system is rather different from other country. It consist of several types, including of private vehicles, consisting of old vehicles and new vehicles, public vehicles, government vehicles, and military or police vehicles. Commonly, The characters that used are a combination of alphabet and number, also it have special features to distinguish similar characters. Character recognition is a vital component because in the real world scene there are a lot of disturbances from environmental conditions, weather, and conditions of license plate on vehicles that cause image quality to be reduced, there by reducing the system ability to recognize a character.

Several studies have shown that deep learning is the optimal technique to read characters on a license plate. With a deep learning approach the computer can recognize characters based on a training dataset that has been processed so that the recognition becomes more accurate. A systematic study of the application of deep learning for number plate recognition has been carried out by Weihong et al<sup>1)</sup>, in this study the existing algorithm was tested for the detection and recognition of license plate characters, based on the quality of the image tilt dataset, noisy image, blurring image, dataset from several available public datasets. The character recognition algorithm divided into two categories, employ character segmentation and misemploy character segmentation. In addition there are several other algorithms used to recognize characters, including SVM (Support Vector Machine)<sup>2-4)</sup>, OCR(Optical Character Recognition)<sup>5,6)</sup> and Neural Network<sup>7-11)</sup>.

In this research, we propose the character recognition with image enhancement (Histogram Equalization) and Convolution Neural Network employ Indonesian number plate character as a dataset. The discussion of further studies can be formulated as follows. In the Section

2 elaborating previous studies on character recognition on vehicle license plate, in the Section 3 introducing the method used in this study and in the Section 4 explaining the experimental results and the conclusions founding the Section 5.

## RELATED WORKS

In this section, we mention several studies for character recognition into 4 parts, OCR method, SVM method, Neural Network method, and Convolutional Neural Network method.

### A. OCR

Zheng et al.<sup>5)</sup> proposed system is robust under poor illumination conditions and for moving vehicles using modified open source Optical Character Recognition. Wang et al.<sup>6)</sup> recognize Chinese characters using modified template matching techniques to produce speed and accuracy from previous template matching techniques.

### B. SVM

Wen et al.<sup>2)</sup> recognize chinese and kana characters, by searching for features using the global direction contributivity density (G-DCD) and local direction contributivity density (L-DCD) methods, the next step is to carry out a classification using the Support Vector Machine method. Panahi et al.<sup>3)</sup> utilize SVM to classify characters with data sample conditions in various weather and environmental conditions such as rain, dust, night, day, fog moreover in any other obstacles.

### C. CNN

Henry et al.<sup>7)</sup> Using improved YOLO Network for character recognition used in the ALPR system for various types of license plates throughout the world. YOLO is used after bounding boxes are found for each character on the vehicle license plate. Pustokhina et al.<sup>8)</sup>. This research employ CNN to recognize the characters on license plate, in this study used several dataset including FZU cars dataset, Stanford cars dataset, and HumAIIn 2019 dataset. Wang et al.<sup>9)</sup> applies the CRNN (Convolutional Recurrent Neural Network) and CTC (Connectionist Temporal Classification) structure, which recognizes the number plate characters without segmenting the characters on license plate. Yang et al.<sup>11)</sup> combines CNN with ELM (Extreme Learning Machine), CNN is used to extract features, then kernel-based ELM is applied to character classification.

**METHODS**

The goal of the present research is to recognize character on Indonesia license plate. For this purpose, we will conduct several stages. Firstly we preparing dataset. Secondly we performed pre-processing dataset, Thirdly we set up CNN model.

**A. Dataset Generation**

The character dataset for Indonesia License Plate is not openly available, therefore we prepare a dataset by taking pictures of existing vehicles and then start cropping the characters in the license plate that we have collected manually, then the augmentation data is carried out to multiply variants and amounts image.

1) Initial Data: initial data is the result of manually cropping characters from a vehicle image. The initial data consists of 36 class 10 digits, and 26 alphabets, collected in each folder and labeled according to its character. Example cropping character is show on figure 1.



Figure 1.  
Example of Character on Dataset

2) Augmented Data: There are several combinations of methods implemented for data augmentation, the firstly is pre-processing function to adjust brightness, next we applying, width shift, height shift, zoom range, shear range, rotation range, channel shift, and fill mode, the values of those methods are shown in the table I.

Table 1.

Augmentation Methods Combination		
Methods	Values	Description
Pre-processing	Brightness adjustment	Function that will be applied on each input
Width shift	0.1	Fraction of total width

Height shift	0.1	Fraction of total height
Zoom	0.02	Range for random zoom
Shear	0.1	Shear intensity (shear angle in counter clockwise direction in degrees)
Rotation	10	Range within which to randomly rotate pictures
Channel shift	4	Adds/subtracts the same intensity from all channels in the image
Fill Mode	Nearest	Points outside the boundaries of the input are filled according to the given mode



Figure 2. Initial Data Example

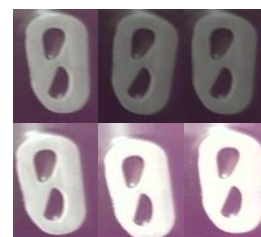


Figure 3.  
Augmentation Result Example

Figure 2 shows the initial data example and we proceed with specified augmentation data parameter, several result image of augmentation data can be seen in figure 3.

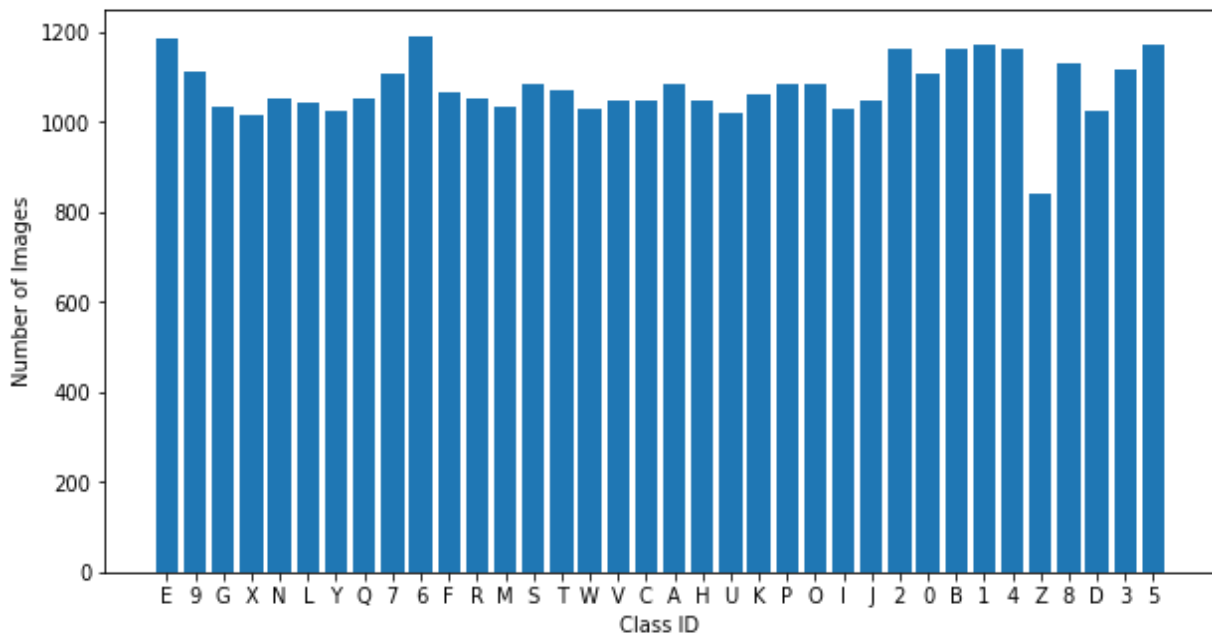


Figure 4. Dataset Distribution

Total data image is 38783 on data augmentation afterward. The distribution amount of data against class can be seen in Figure 4.

**B. Image Pre-processing**

1) Grayscale Image: RGB (Red Green Blue) images are acquired in the acquisition task, next process is converting these images to grayscale. Grayscale image is utilize as the input of this algorithm, the purpose grayscale image is simplify images, and minimizing process to images. The formula of converting RGB to grayscale image as follow equation :

$$gray(x,y) = 0.2989 \times f(x,y,R) + 0.5870 \times f(x,y,G) + 0.1140 \times f(x,y,B) \tag{1}$$

Image Enhancement: Histogram is a representation of the intensity distribution of an image. Histogram equalization is method to contrast adjustment using histogram. The purpose this method is for image enhancement, and this method can provide better quality of images without losing any information.  $H'$  is defined as equalized histogram, where  $H'$  is cumulative distribution as follow equation :

$$equalized(x,y) = H'(src(x,y)) \tag{2}$$

$$H'(i) = \sum_{0 \leq j \leq i} H(j) \tag{3}$$

**C. Character Recognition**

1) Encode Mapping : Mapping is technique to convert categorical value in class ID to numerical value of the entry in the dataset. The mapping of categorical dataset can be seen in Table II.

Table 2. Character Encode Map

Class	Encode	Class	Encode
0	0	I	18
1	1	J	19
2	2	K	20
3	3	L	21
4	4	M	22
5	5	N	23
6	6	O	24
7	7	P	25
8	8	Q	26
9	9	R	27
A	10	S	28
B	11	T	29
C	12	U	30
D	13	V	31
E	14	W	32
F	15	X	33
G	16	Y	34
H	17	Z	35

2) Model Design : A convolutional neural network consist of an input and an output layer, and multiple hidden layer. The input layer is image with dimension 100 x 75 x 1 (h x w x d) grayscale image. Next stages of this model is using several layer,

combination of convolution layer, and pooling for feature learning, next stage is for classification using flatten drop out to classify result into 36 class. The model architecture can be seen in figure 6.

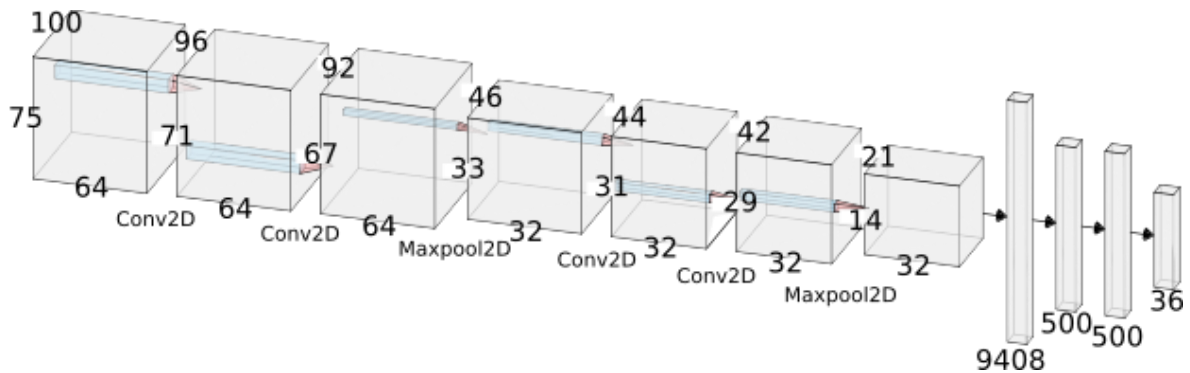


Figure 6.

CNN Model Architecture The Detail Configuration of Convolutional Neural Network Architecture Layer Can be Seen on The Table 3.

Table 3.  
Networks Parameter

Layer Type	Value
Input	100 x 75 x 1
Conv2D + Relu	Filter = 64 ; kernel = 5 x 5
Conv2D + Relu	Filter = 64 ; Kernel = 5 x 5
MaxPool	Poolsize = 2 x 2
Conv2D + Relu	Filter = 32 ; Kernel = 3 x 3
Conv2D + Relu	Filter = 32 ; Kernel = 3 x 3
MaxPool	Poolsize = 2 x 2
Dropout	0.5
Dense	500 nodes
Dropout	0.5
Dense	36 Class

3) Hyper parameter training: Hyper parameters are the variables which determines the network structure and how the network is trained. Hyper parameter are set before model training process. There are several hyper parameter set. Specifically, Adam optimizer Learning rate set on 0.001, batch size number 2000, and number of epochs 50 respectively.

## EXPERIMENTAL RESULT

### A. Dataset

In this section, we splitting dataset into 3 parts, training dataset, test dataset, and validation dataset.

Total training dataset is 24820 images, for testing dataset is 7757 images, and 6206 images for validation dataset as table IV.

Table 4.  
Dataset Ratio

Dataset	Amount
Training	24820
Test	7757
Validation	6206
<b>Total</b>	<b>38783</b>

The training dataset data for each class category consist approximately between 600 images and 700 images, the distribution of training dataset can be seen on figure 7.

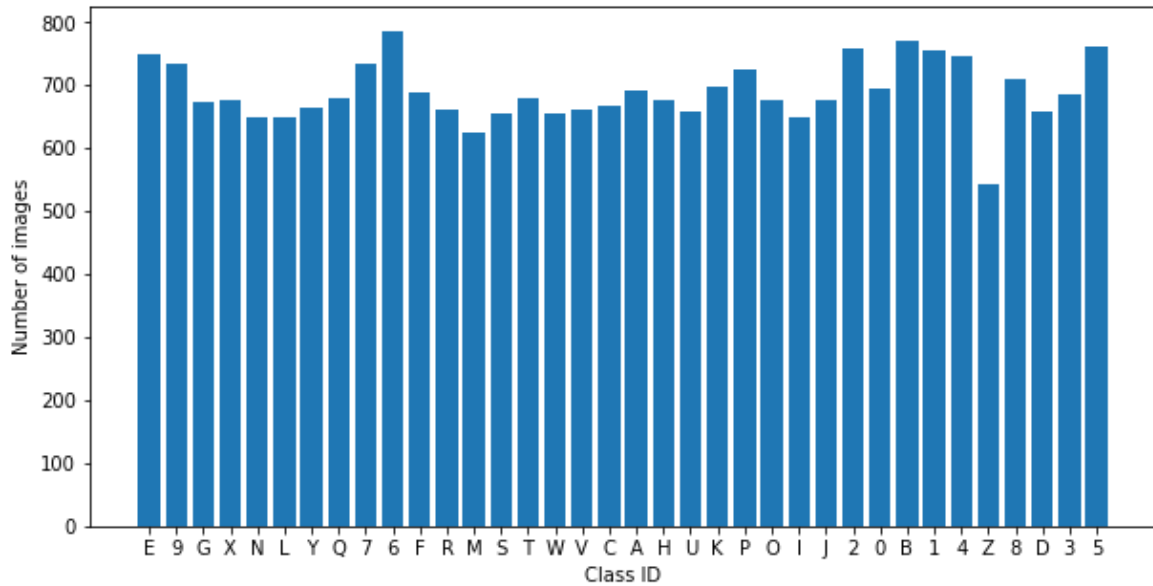


Figure 7. Training Dataset Distribution

B. Evaluation model

In this stage training model showed accuracy and loss on training dataset and validation dataset. The accuracy of this model for the training dataset 0.9876 and 0.9976 for the validation dataset respectively. The loss value of this model for training dataset and validation dataset are 0.0407 and 0.0117 respectively.

From third epoch training dataset continued to increase until last epoch and reaching 0.9876.

Identical to training dataset, validation dataset accuracy had increased from its initial value 0.9879 and 0.9932 in second epoch. Accuracy of validation dataset remains stable and reaching 0.9976 at last epoch.

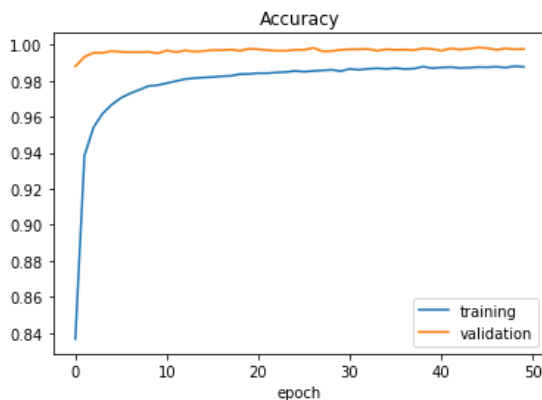


Figure 8. Model Accuracy

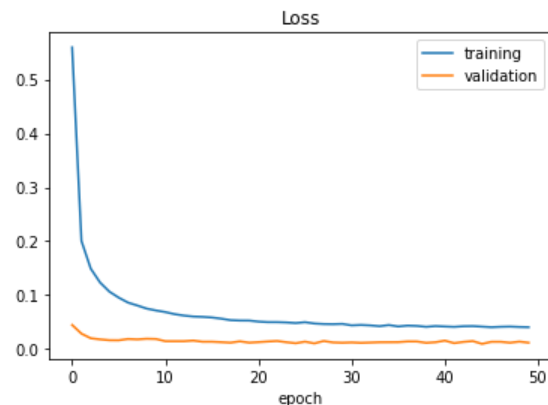


Figure 9. Model Loss

The line graph shown by figure 8 compares model accuracy between training dataset and validation dataset from epoch 0 to 50. Overall both dataset had increased accuracy from their initial value. Looking in more detail training dataset increase significantly for 0.9385 in second epoch from its initial value for 0.8366.

In figure 9, show Model loss from training dataset, and validation dataset. Overall both dataset show decreasing loss from their initial value until the last epoch. In addition training dataset had high initial value 0.5592 at first epoch compare to validation dataset value for 0.0446. Training dataset had decreased progressively, ended up at 0.0407 in last epoch. Equal to training dataset, validation dataset had decreased

from first epoch, and remain stable just below 0.01. On the last epoch validation dataset reach 0.0117.

Table 5.  
Test Dataset Digits Recognition

Class	Precision	Recall	f-1 score	Support
0	0.99	0.99	0.99	200
1	1.00	0.99	0.99	244
2	1.00	1.00	1.00	225
3	1.00	1.00	1.00	239
4	0.99	1.00	1.00	223
5	1.00	1.00	1.00	243
6	1.00	1.00	1.00	237
7	1.00	1.00	1.00	235
8	1.00	0.99	1.00	223
9	1.00	1.00	1.00	205

The table V shows the value of precision, recall, f1 score from each digit class with amount of data in column support. For precision value class 0, and class 4 had a 0.99, the rest class had 1.00. class 0, class 1, and class 8, get 0.99 in Recall value, leftover from digits class get 1.00. f-1 score with 0.99 is only in class 0, and class 1, the rest of it get 1.00.

Table 6.  
Test Alphabets Digits Recognition

Class	Precision	Recall	f-1 score	Support
A	1.00	1.00	1.00	221
B	0.99	1.00	1.00	219
C	0.99	1.00	0.99	215
D	1.00	0.99	1.00	195
E	0.99	1.00	1.00	238
F	1.00	1.00	1.00	208
G	1.00	0.99	0.99	205
H	1.00	1.00	1.00	214
I	1.00	0.99	1.00	206
J	1.00	1.00	1.00	230
K	1.00	1.00	1.00	204
L	0.99	0.99	0.99	200
M	1.00	1.00	1.00	217
N	1.00	1.00	1.00	217
O	1.00	1.00	1.00	233
P	1.00	1.00	1.00	216
Q	0.99	0.99	0.99	187
R	1.00	1.00	1.00	227
S	1.00	1.00	1.00	218
T	1.00	1.00	1.00	204
U	1.00	1.00	1.00	205
V	1.00	1.00	1.00	223
W	0.99	1.00	1.00	198
X	0.99	1.00	1.00	194
Y	1.00	1.00	1.00	222
Z	1.00	0.99	1.00	167

The table VI shows the value of precision, recall, and f-1 score from alphabet class. The largely of this class get 1.00 in precision except class B, class C, class D, class L, class Q, class W, and class X. In Recall value only 6 class get 0.99 there are class D, class G, class I, class L, class Q, and class Z, the rest of it get 1.00. Class C, class G, class L, and class Q get 0.99 in f-1 score, and 1.00 for the rest.

## CONCLUSION

In this research we propose the character recognition using the image enhancement technique using the Histogram Equalization and Convolutional Neural Network using specific Indonesian license plate character dataset. This model reach accuracy 0.9876 and loss 0.0407 on Training dataset, furthermore in the validation dataset this model can reach accuracy 0.9976 and loss 0.0117.

## ACKNOWLEDGEMENT

Thanks to Center of Technology For System and Infrastructure of Transportation, Agency for the Assessment and Application of Technology, that fully support this research related to the technical support and research funding.

## REFERENCES

1. W. Weihong and T. Jiaoyang, "Research on License Plate Recognition Algorithms Based on Deep Learning in Complex Environment," *IEEE Access*, vol. 8, pp. 91661–91675, 2020.
2. Y. Wen, Y. Lu, J. Yan, Z. Zhou, K. M. Von Deneen, and P. Shi, "An algorithm for license plate recognition applied to intelligent transportation system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 830–845, 2011.
3. R. Panahi and I. Gholampour, "Accurate Detection and Recognition of Dirty Vehicle Plate Numbers for High-Speed Applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 767–779, 2017.
4. A. H. Ashtari, M. J. Nordin, and M. Fathy, "An Iranian license plate recognition system based on color features," *IEEE Trans. Intell. Transp.*

- Syst., vol. 15, no. 4, pp. 1690–1705, 2014.
5. L. Zheng, X. He, B. Samali, and L. T. Yang, "An algorithm for accuracy enhancement of license plate recognition," *J. Comput. Syst. Sci.*, vol. 79, no. 2, pp. 245–255, 2013.
  6. J. X. Wang, W. Z. Zhou, J. F. Xue, and X. X. Liu, "The research and realization of vehicle license plate character segmentation and recognition technology," 2010 International Conference on Wavelet Analysis and Pattern Recognition, ICWAPR 2010, no. July, pp. 101–104, 2010.
  7. C. Henry, S. Y. Ahn, and S. W. Lee, "Multinational License Plate Recognition Using Generalized Character Sequence Detection," *IEEE Access*, vol. 8, pp. 35185–35199, 2020.
  8. I. V. Pustokhina, D. A. Pustokhin, J. J. Rodrigues, D. Gupta, A. Khanna, K. Shankar, C. Seo, and G. P. Joshi, "Automatic Vehicle License Plate Recognition Using Optimal K-Means with Convolutional Neural Network for Intelligent Transportation Systems," *IEEE Access*, vol. 8, pp. 92907–92917, 2020.
  9. W. Wang, J. Yang, M. Chen, and P. Wang, "A Light CNN for End-to-End Car License Plates Detection and Recognition," *IEEE Access*, vol. 7, pp. 173875–173883, 2019.
  10. H. Li, P. Wang, and C. Shen, "Toward End-to-End Car License Plate Detection and Recognition with Deep Neural Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1126–1136, 2019.
  11. Y. Yang, D. Li, and Z. Duan, "Chinese vehicle license plate recognition using kernel-based extreme learning machine with deep convolutional features," *IET Intelligent Transport Systems*, vol. 12, no. 3, pp. 213–219, 2018.