Journal of
Information Systems Engineering
and Business Intelligence

Vol.8, No.1, April 2022
Available online at: http://e-journal.unair.ac.id/index.php/JISEBI

License Plate Character Recognition using Convolutional Neural Network

Firman Maulana Adhari[1)], **Taufik Fuadi Abidin**[2)*], **Ridha Ferdhiana**[3)]

[1)2)] *Department of Informatics, Faculty of Mathematics and Natural Sciences, Universitas Syiah Kuala, Indonesia*
Jl. Syech Abdurrauf No. 3, Kopelma Darussalam, Banda Aceh
[1)]firman@mhs.unsyiah.ac.id, [2)]taufik.abidin@unsyiah.ac.id

[3)] *Department of Statistics, Faculty of Mathematics and Natural Sciences, Universitas Syiah Kuala, Indonesia*
Jl. Syech Abdurrauf No. 3, Kopelma Darussalam, Banda Aceh
[3)]ridha.ferdhiana@unsyiah.ac.id

Abstract

Background: In the last decade, the number of registered vehicles has grown exponentially. With more vehicles on the road, traffic jams, accidents, and violations also increase. A license plate plays a key role in solving such problems because it stores a vehicle's historical information. Therefore, automated license-plate character recognition is needed.
Objective: This study proposes a recognition system that uses convolutional neural network (CNN) architectures to recognize characters from a license plate's images. We called it a modified LeNet-5 architecture.
Methods: We used four different CNN architectures to recognize license plate characters: AlexNet, LeNet-5, modified LeNet-5, and ResNet-50 architectures. We evaluated the performance based on their accuracy and computation time. We compared the deep learning methods with the Freeman chain code (FCC) extraction with support vector machine (SVM). We also evaluated the Otsu and the threshold binarization performances when applied in the FCC extraction method.
Results: The ResNet-50 and modified LeNet-5 produces the best accuracy during the training at 0.97. The precision and recall scores of the ResNet-50 are both 0.97, while the modified LeNet-5's values are 0.98 and 0.96, respectively. The modified LeNet-5 shows a slightly higher precision score but a lower recall score. The modified LeNet-5 shows a slightly lower accuracy during the testing than ResNet-50. Meanwhile, the Otsu binarization's FCC extraction is better than the threshold binarization. Overall, the FCC extraction technique performs less effectively than CNN. The modified LeNet-5 computes the fastest at 7 mins and 57 secs, while ResNet-50 needs 42 mins and 11 secs.
Conclusion: We discovered that CNN is better than the FCC extraction method with SVM. Both ResNet-50 and the modified LeNet-5 perform best during the training, with F measure scoring 0.97. However, ResNet-50 outperforms the modified LeNet-5 during the testing, with F-measure at 0.97 and 1.00, respectively. In addition, the FCC extraction using the Otsu binarization is better than the threshold binarization. Otsu binarization reached 0.91, higher than the static threshold binarization at 127. In addition, Otsu binarization produces a dynamic threshold value depending on the images' light intensity.

Keywords: *Convolutional Neural Network, Freeman Chain Code, License Plate Character Recognition, Support Vector Machine*

Article history: Received 2 January 2022, decision after peer review 26 January 2022, accepted 4 March 2022, available online 28 April 2022

I. INTRODUCTION

In the last decade, the number of registered vehicles has grown significantly [1], which triggers more transportation problems, such as traffic congestions, accidents and violations. Traffic signs provide warnings, prohibitions, orders, or instructions to road users, but these are often violated. An example of a traffic sign violation is when a motorist runs a red light, posing a danger to other drivers and road users. The vehicle and the driver can be tracked from the license plate.

A license plate is a unique code consisting of digits and letters representing the vehicle's identity. It distinguishes one motorised vehicle unit from another and conveys information about the owner, as well as the violation history. Manually recognizing a unique code on a license plate is difficult due to the large volumes. It is also prone to human errors. A better solution is using an automated license plate recognition system that extracts information from images.

[*] Corresponding author

doi: http://dx.doi.org/10.20473/jisebi.8.1.51-60

An automated recognition system, which can be developed using machine learning [2], will help improve security, traffic control, and parking payment systems [3]. However, detecting and extracting each character from the license plate are challenging. Any incorrectness can directly affect the system's accuracy [4].

Machine learning is a branch of artificial intelligence that studies pattern recognition from an existing dataset and predicts new data [5]. A computer can learn and recognize license plate characters correctly from images with machine learning. One technique with good computer vision results is the deep neural network (DNN) [6], which is developed from an information processing method inspired by a network of interconnected neurons in the human brain called the artificial neural network (ANN) [5]. ANN consists of information processing elements (neurons) connected at different layers. The first and last layers are called the input and output layers, and the layer between them is called the hidden layer [7].

The convolutional neural network (CNN) is a DNN method with good image classification and object recognition performance [6]. CNN has incomparable advantages in image feature representation and, therefore, is a suitable method for license-plate character recognition [8]. CNN consists of layers that can efficiently learn features from many labelled data [9]. It has both convolutional and pooling layers needed for extracting features. A fully connected layer is effective for training and classification [10]. Therefore, CNN is extensively used for image processing, where each pixel value in an image becomes an input for the system [11]. It has different layer architectures with a different number of layer-stacks comprising convolutional, pooling, and fully connected layers [11]. The layer architectures in CNN include LeNet-5, AlexNet, and ResNet-50. Yann LeCun designed a LeNet-5 architecture in 1998 to recognize handwritten digits [12]. The AlexNet architecture, designed by Alex et al., is similar to LeNet-5 but is more extensive and in-depth. AlexNet won the ImageNet Large-Scale Visual Recognition Competition (ILSVRC) in 2012 by reducing the top five error values from 26% to 15.3% [6]. In addition, ResNet-50 won the ILSVRC competition in 2015. Microsoft Research Asia [13] team developed the architecture.

Studies have been conducted to improve license plate character recognition. Reference [2] recognizes license plate characters by classifying the 36 characters, with each category representing a letter or a character, using a support vector machine (SVM) [14]. The character's image is divided into four, six, and eight zones. Then, the Freeman chain code method extracts the numerical features from each zone. The results show that the feature extraction using the Freeman chain code method in the eight zones outperformed other zones with an accuracy of 0.87 [2].

Another work in license plate character recognition, as in [15], uses a Canny filter to retrieve a character from digital image data. The image is resized to 12×7 and converted into a binary form as data features. The converted data are classified using an ANN and trained using a backpropagation method at a learning rate of 0.3 and a momentum of 0.9. The accuracy yield is 0.94 [15]. Research on license plate detection and character recognition using the Multi-Task Convolutional Neural Network for License Plate Detection and Recognition (MTLPR), conducted by [16], claims a high accuracy and low computational process.

The MTLPR method consists of two stages, i.e., detection and recognition. In the detection stage, the MTLPR method adopts the Multi-task Convolutional Neural Network (MTCNN)—which was originally used to determine the face area in the face detection problem—and uses it to detect the license plate area effectively. MTLPR has three layers, i.e., P-Net, R-Net, and O-Net. In the recognition stage, there are three layers: the convolutional, recurrent, and transcription (CTC) layers. The results show that the augmented MTLPR performs better at the detection stage (precision: 0.988) than YOLO-V3 and RPNet, with an average precision of 0.977.

Another study [17] uses deep learning by implementing optimal k-means (OKM) clustering for segmentation and CNN for recognition, referred to as the OKM-CNN model. It consists of three stages: detecting a license plate with an improved Bernsen algorithm and connected component analysis (CCA) model, segmenting the characters using the OKM clustering and recognizing the characters using CNN. The study was evaluated using Stanford Cars, FZU Cars, and the HumAIn 2019 challenge datasets. The OKM-CNN method was also evaluated using the VGG-16, ResNet-50, and ResNet-10 architectures. The results showed that OKM-CNN has an overall accuracy of 0.98, slightly better than the ResNet-50 with an overall accuracy of 0.976.

This study analyses and compares CNN using LeNet-5, modified LeNet-5, AlexNet, and ResNet-50 architectures for license-plate character recognition. The contributions are three-fold:

1. Determining the best CNN model to recognize license-plate characters written in white letters on a black background.
2. Discovering whether the CNN methods outperform the Freeman chain code extraction with SVM.
3. Determining the best mechanism to select the region of interest of the characters from the license plate and deciding whether Otsu binarization is better.

II. METHODS

In this study, the trained CNN models are used to recognize the license plate characters. The stages are data collection, segmentation and labelling, CNN modelling, and classification, as shown in Fig. 1.

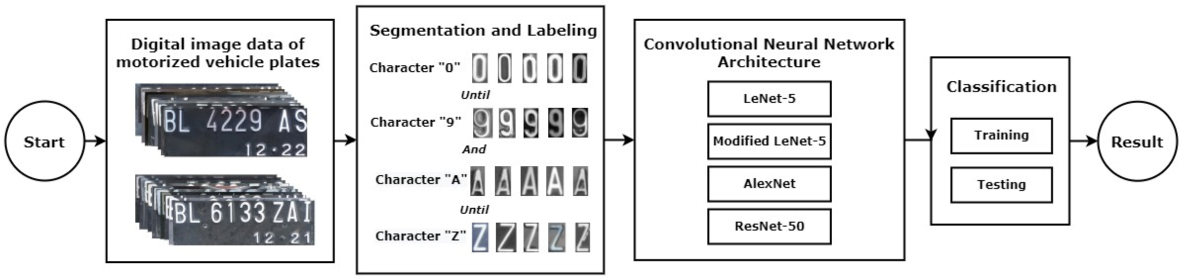


Fig. 1. Research methodology

A. Data Collection

The license plate images were collected manually by taking photos of the plates at daytime using a front camera positioned perpendicular to the plates to produce sharply delineated and non-slanted photos. The license plates' characters—written in white on a black background—did not undergo any alteration and had no defects in their nature. We obtained 618 images. The focus of our study is to determine the letters and numbers on license plates and is not to identify the license plates' regions. Hence, the study starts with license plate images directly.

B. Segmentation and Labelling

The segmentation stage aims to produce pieces of character from the license-plate images. The pieces comprise only the characters of the plates. The processes performed are grayscaling, binarization, finding the coordinates of the corner points for character intersections, and intersecting the original image data. Each set of characters is located in a folder. Fig. 2 shows the segmentation and labelling phases.

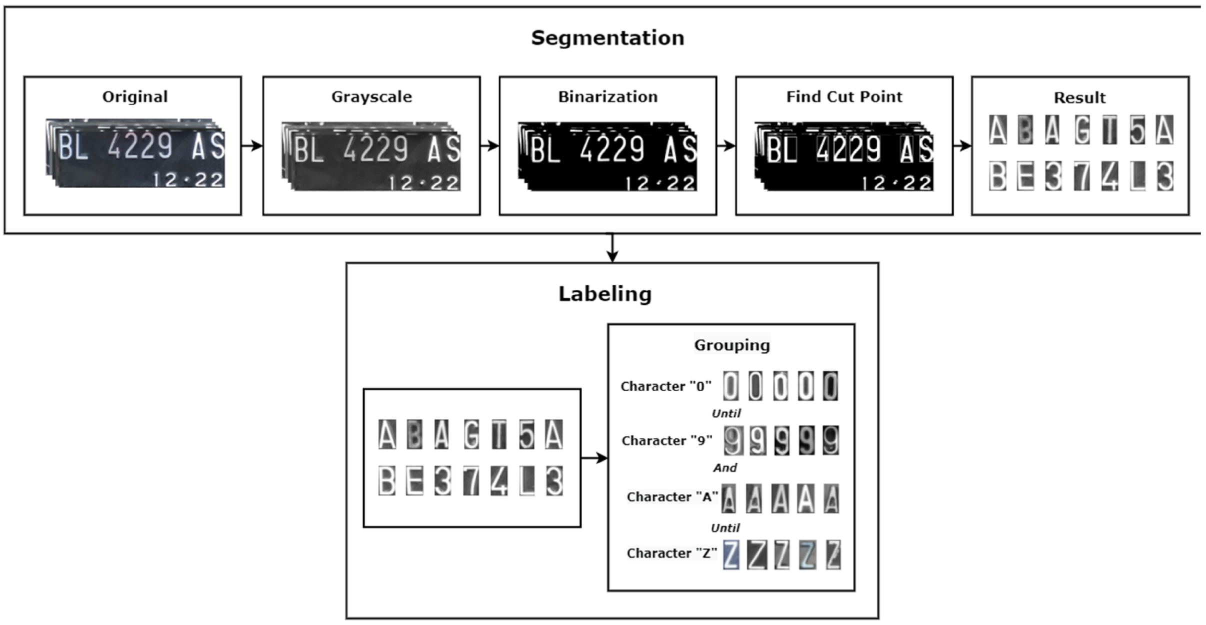


Fig. 2. The segmentation and labelling phases

Each pixel has three layers in a colour digital image with different pixel values in each layer. These layers are red, green, and blue (RGB). Combining each pixel value of the different layers produced various colours that turned the image into RGB. The grayscale converted the colour image into gray. It converts the image with the three RGB layers into a gray layer. Each pixel in the grayscale image has a value between 0 and 255. A pixel value approaching 0 indicates a black colour, while a pixel approaching 255 indicates a white colour. The grayscale was obtained using a binarization process. The grayscale image was converted into black and white by changing the value of each pixel of the grayscale image to 0 or 255 using a threshold. If the pixel value is smaller than the threshold, the value is 0. Otherwise, the value is changed to 255.

Next, we found the coordinates of the corner points for intersecting characters by determining the different colour points between black and white and making a rectangular border on the white part of the character. To get the coordinates of the intersection points and make the boundary lines on these characters accurately, the results of the binarization process must be exact because finding the coordinates of these corner points depends on the different colour points between black and white.

The threshold value and Otsu binarization process were evaluated using 618 digital images. The binarization method was evaluated by calculating the percentage of characters with the correct cut-off points on each image and the average percentage value of the entire image. The threshold value is 127. If a pixel value in grayscale is smaller than 127, the value is changed to 0. Otherwise, the value is changed to 255. The Otsu binarization uses dynamic threshold values based on each image's pixel statistic value [18]. The Otsu binarization can provide a threshold value by the light intensity on the license-plate image. The threshold is obtained from the histogram that allows each grayscale image to have a different threshold value [19].

The number of classes is 36, consisting of 10 numeric characters from 0 to 9 and 26 letter characters, i.e., 'A–Z.' Each class is saved in a folder.

C. Convolutional Neural Network (CNN)

Inspired by a visual cortex biological system in the brain, CNN is an algorithm that can be used to recognize image patterns [11]. It encompasses feature extraction and classification, as shown in Fig. 3. The feature extraction consists of a convolutional layer, a pooling layer, and an activation function. Meanwhile, the classification part consists of fully connected layers [10]. CNN uses the value of an image's pixel representation as a feature in the input layer, resulting in many features used in the input layer. The convolutional and pooling layers overcome this problem [11].

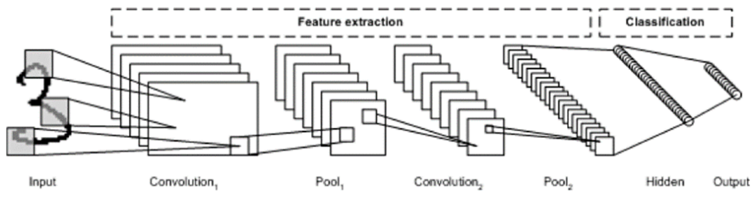


Fig. 3. The convolutional neural network [20]

The convolutional layer plays a key role in CNN [11]. It consists of kernels or filters to extract local features from the input and compute a feature map—the calculation output between the input and filter by shifting the filter above and performing dot operations [21]. The CNN filter is a weight determined by the system and is reset in the backward propagation process [11].

The pooling layer occurs after the convolutional layer. It contains no learning parameters and reduces the dimensions of the feature map, speeding up the computation process. It has only hyper-parameters, which are set before the learning process begins. The two operations in the pooling layer are the maximum and the average pooling. The maximum pooling takes the highest value of the fields in the input, which is the most commonly used pooling layer nowadays. Meanwhile, the average pooling calculates the average of the cell fields in the input [22]. The hyper-parameters include filter size, stride, padding, and pooling operation.

After the convolutional and pooling layers produce the final feature map, the two-dimensional feature map is changed into one-dimensional (vector) features. The objective is to use the feature map value as input for the fully connected layer, consisting of several layers and neurons similar to DNN. The last layer of the fully connected layer has neurons representing data labels to be classified. In this output layer, the probability of each label is calculated using the SoftMax function [22]. CNN has a different layer architecture based on the number of convolutional, pooling, and fully connected layers [11]. Our study used the LeNet-5, modified LeNet-5, AlexNet, and ResNet-50 architectures.

1) LeNet-5 Architecture

In 1998, Yann LeCun designed the LeNet-5 architecture for gradient-based learning applied to document recognition. He implemented the architecture for optical character recognition (OCR) and document character recognition [12]. The LeNet-5 architecture was used in this work because it is suitable for character recognition, especially license-plate character recognition. Fig. 4 shows the LeNet-5 architecture.

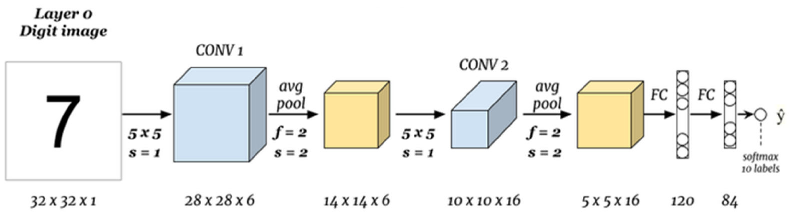


Fig. 4. LeNet-5 architecture [19]

2) Modified LeNet-5 Architecture

Reference [24] designed a modified LeNet-5 architecture that can improve the layer composition. Max pooling was used in the pooling layer, ReLU was used for the activation function, and the same type of padding was used. This padding produces an output size equal to the input size, with several neurons in the fully connected layer reaching 500 neurons [24]. Fig. 5 depicts the modified LeNet-5 architecture.

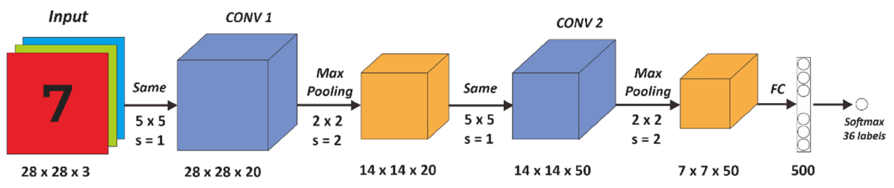


Fig. 5 Modified LeNet-5 architecture [24]

3) AlexNet Architecture

Alex et al. designed the AlexNet architecture [6] by implementing CNN to classify 1.2 million image data with 1,000 classes. They obtained a top-five error test result of 15.3% [6]. The AlexNet architecture is similar to the LeNet-5 architecture but more extensive and profound. The study also compares AlexNet with the other CNN architectures. Fig. 6 illustrates the AlexNet architecture.

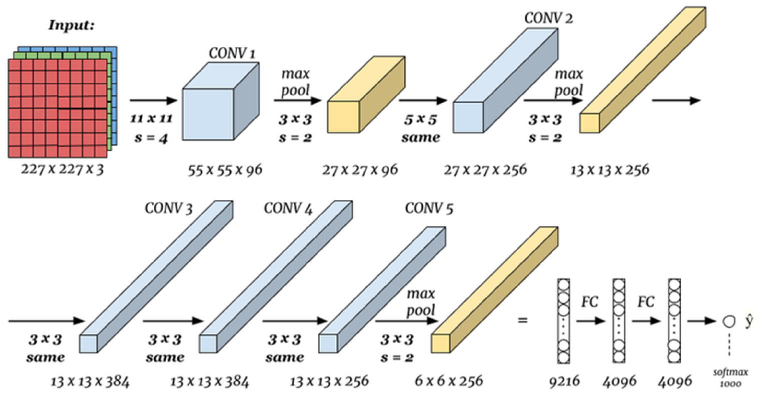


Fig. 6 AlexNet architecture [19]

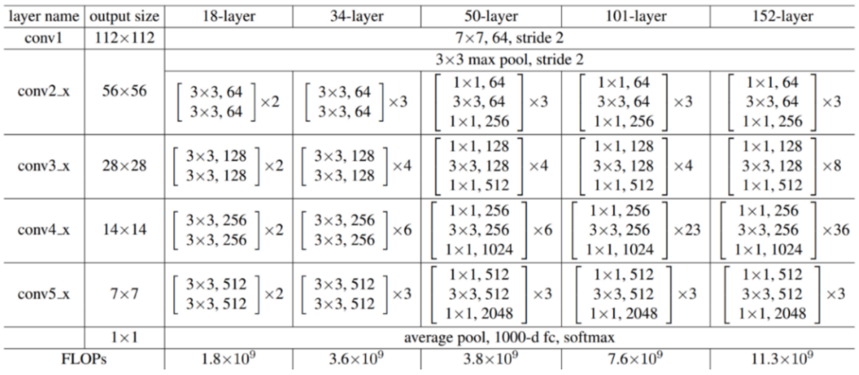
4) ResNet-50 Architecture

ResNet consists of several architectural model structures, ranging from 18 to 152 layers [13]. We use ResNet architecture with 50 layers to reduce the complexity and expedite the computational process. The number of channels in the fully connected layer is set according to the total number of categories. ResNet total parameters are 26 million. We use ResNet-50 architecture in this study. Fig. 7 shows the ResNet architecture.

D. Classification

The classification was performed on each character in the images using LeNet-5, modified LeNet-5, AlexNet, and ResNet-50. The Freeman chain code (FCC) extraction [2] was also examined using the same images for comparative analysis. A value of 140 was set as a threshold binarization of the FCC [2]. The Otsu binarization method was also implemented in the FCC extraction for comparison.

The classification consists of training and testing stages. The training is a learning stage that produces a classification model, while the testing is a model evaluation stage. Each stage uses a different set of data. Five images for each class were used for the testing, and the rest were used for the training. In general, 75% of the data from each class were used for the training, and 25% were used for the testing. Epoch, batch size, and the learning rate is the CNN methods' hyper-parameter set before the training. The epoch and batch size values are 100, respectively, at a learning rate of 0.001.



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 7. The ResNet architecture [13]

III. RESULTS

We compared the Otsu and threshold binarization methods. Fig. 8 shows the comparison of the accuracy of the two methods.

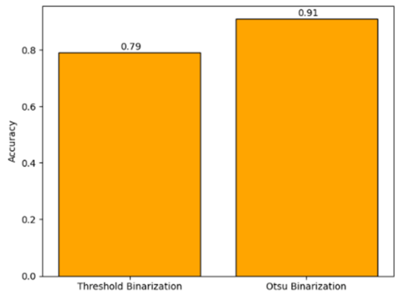


Fig. 8 The comparison of Otsu and threshold binarization

The segmentation using the Otsu binarization method on 618 images produced 3,048 characters. The labelling was performed by grouping 3,048 characters based on their classes—36 classes consisting of 10 numeric characters (0–9) and 26 letters (A–Z). We measured the data size for each class and found that the class was imbalanced. The 'L' character has 291 images, and the 'Y' character has only 18 images. Fig. 9 depicts the segmentation using the Otsu and threshold binarizations.

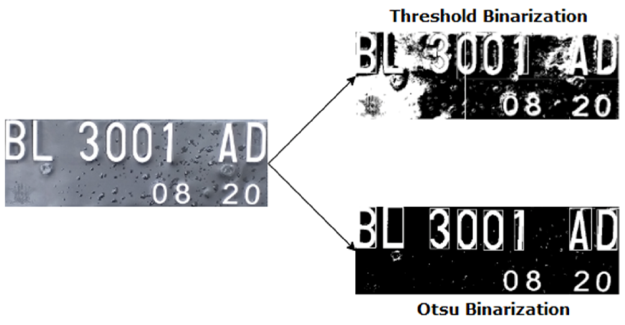


Fig. 9. Comparison of the threshold and Otsu binarization methods

Accuracy and time are the two performance indicators in the current study. Accuracy was measured using each character's average F-measure. In addition, precision and recall scores were also measured. Fig. 10 and Fig. 11 show the classification results for each method during the training and testing stages. Tables 1 and 2 summarize the measurement values.

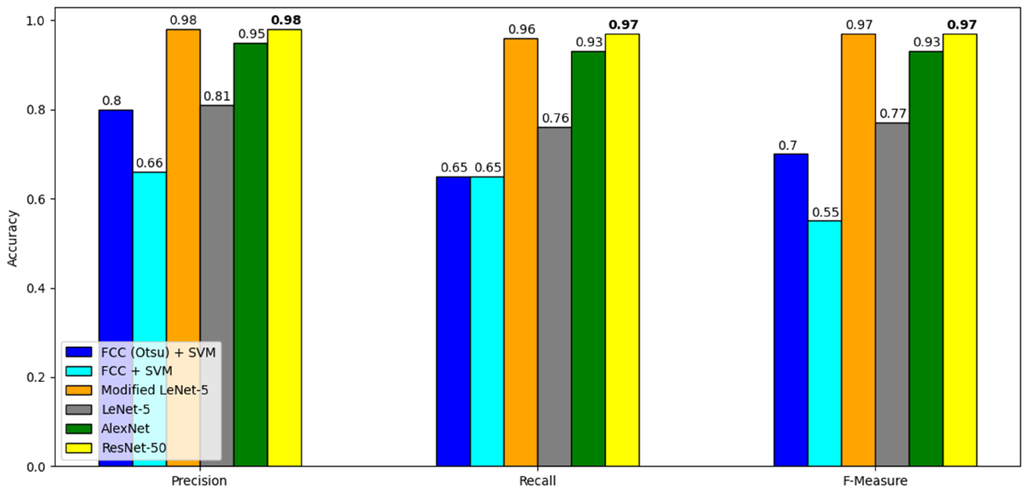


Fig. 10. Comparison of the classification methods in the training stage

TABLE 1
RESULTS OF CLASSIFICATION METHODS AT THE TRAINING STAGE

Model	Epoch	Precision	Recall	F-measure
LeNet-5	100	0.81	0.76	0.77
Modified LeNet-5	**100**	**0.98**	**0.96**	**0.97**
AlexNet	100	0.95	0.93	0.93
ResNet-50	**100**	**0.98**	**0.97**	**0.97**
FCC + SVM	-	0.66	0.65	0.55
FCC (Otsu) + SVM	-	0.80	0.65	0.70

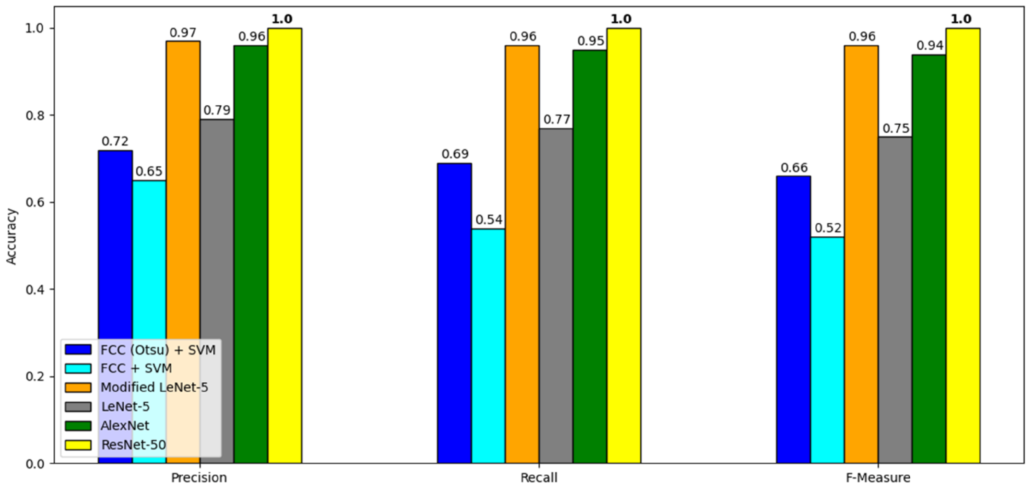


Fig. 11 Comparison of the classification methods in the testing stage

TABLE 2
RESULTS OF CLASSIFICATION METHODS AT THE TESTING STAGE

Model	Epoch	Precision	Recall	F-measure
LeNet-5	100	0.79	0.77	0.75
Modified LeNet-5	100	0.97	0.96	0.96
AlexNet	100	0.96	0.95	0.95
ResNet-50	**100**	**1.00**	**1.00**	**1.00**
FCC + SVM	-	0.65	0.54	0.52
FCC (Otsu) + SVM	-	0.72	0.69	0.66

In addition to the accuracy, the methods' performances were also evaluated from the computation time. The computation time of the CNN was calculated from the classification process of the segmented images, whereas the computation time of the FCC with the SVM extraction method was calculated from feature generation and classification, as summarized in Table 3.

TABLE 3
COMPUTATION TIME CLASSIFICATION METHODS

Model	Time
AlexNet	37 minutes and 35 seconds
LeNet-5	8 minutes and 15 seconds
Modified LeNet-5	**7 minutes and 57 seconds**
ResNet-50	42 minutes and 11 seconds
FCC	11 minutes and 21 seconds
FCC (Otsu)	**7 minutes and 48 seconds**

IV. DISCUSSION

The 618 license plate images evaluation shows that the Otsu binarization method [18] performs better than the threshold binarization, with an average accuracy of 0.91. The threshold binarization produces an average accuracy of 0.79. Fig. 10 and 11 summarize the classification results of each method during the training and testing stages. The ResNet-50 and the modified LeNet-5 produced the best accuracy during the training at 0.97. The recall and precision values for the ResNet-50 were both 0.97, whereas the modified LeNet-5 values are 0.98 and 0.96, respectively. LeNet-5 was slightly more precise, but the recall was lower. The modified LeNet-5 has lower accuracy during the testing than ResNet-50, which returns 1.00 accuracy. The FCC extraction method [2] using the Otsu binarization is better than the threshold binarization. Nevertheless, the FCC extraction technique performs less effectively than the CNN methods.

Table 3 shows that among the CNN methods, the modified LeNet-5 results in the fastest computation time at 7 mins and 57 secs, whereas ResNet-50 takes the longest to compute at 42 mins and 11 secs. ResNet-50 takes much time to finish because of its many architectural layers. As for the FCC extraction, the Otsu binarization is faster than the threshold binarization. It requires 7 mins and 48 secs to complete, while the threshold binarization takes 11 mins and 21 secs.

V. CONCLUSIONS

The comparison analysis indicates that the CNN methods outperform the FCC with the SVM method. Both ResNet-50 and the modified LeNet-5 perform best during the training, scoring 0.97 in F-measure. However, ResNet-50 outperforms modified LeNet-5 during the testing, with F-measure 0.97 and 1.00. In addition, the FCC extraction using the Otsu binarization is better than the thresholding binarization. The Otsu binarization reached 0.91, higher than the static threshold binarization at 127. In addition, the Otsu binarization provides a dynamic threshold value depending on the light intensity of the images.

Author Contributions: *Firman Maulana Adhari*: Conceptualization, Methodology, Software, Investigation, Data Curation, Writing - Original Draft. *Taufik Fuadi Abidin*: Conceptualization, Methodology, Writing - Review and Editing, Supervision. *Ridha Ferdhiana*: Writing - Review and Editing.

Funding: This research received no specific grant from any funding agency.

Acknowledgements: We would like to thank Abbas Adam AzZuhri for sharing the license plate images for the Freeman chain code extraction method to compare with this work.

Conflicts of Interest: The authors declare no conflict of interest.

REFERENCES

[1] ASEANStatsDataPortal, Retrieved from https://data.aseanstats.org/indicator/ASE.TRP.ROD.B.005, on February 14, 2022.

[2] T.F. Abidin, A.A. AzZuhri, and F. Arnia, "Pengenalan karakter plat nomor kendaraan bermotor menggunakan zoning dan fitur Freeman Chain Code," Jurnal Rekayasa Elektrika, vol. 14, pp. 19-25, April 2018.

[3] C. Anagnostopoulos, I. Anagnostopoulos, V. Loumos, and E. Kayafas, "A license plate recognition algorithm for intelligent transportation system applications," IEEE Transactions on Intelligent Transportation Systems, pp. 1-16, 2006.

[4] H.K. Sulehria, Y. Zhang, and D. Irfan, "Mathematical morphology methodology for extraction of vehicle number plates," International Journal of Computers, pp. 69-73, 2007.

[5] A. Simon, M.S. Deo, S. Venkatesan, and D.R. Babu, "An overview of machine learning and its applications," International Journal of Electrical Sciences & Engineering, pp. 22-24, 2015.

[6] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet classification with deep convolutional," Advances in Neural Information Processing Systems, pp. 1-9, 2012.

[7] D. Svozil, V. Kvasnicka, and J. Pospichal, "Introduction to multi-layer feed-forward neural networks," Chemometrics and Intelligent Laboratory Systems, pp. 43-62, 1997.

[8] M. Xin, and Y. Wang, "Research on image classification model based on deep convolution neural networks," EURASIP Journal on Image and Video Processing, 40, 2019.

[9] H. Li and C. Shen, "Reading car license plates using deep convolutional neural networks and LSTMs," arXiv Preprint arXiv:1601.05610, 2016.

[10] M. Khoshdeli, R. Cong, and B. Parvin, "Detection of nuclei in H&E stained sections using convolutional neural networks," In Proceedings of the IEEE EMBS International Conference on Biomedical & Health Informatics, pp. 105-108, 2017.

[11] A. Geron, "Hands-on machine learning with scikit-learn, keras, and tensorflow," Sebastopol, California: O'Reilly Media, 2019.

[12] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," In Proceedings of the IEEE, 1-46, 1998.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770-778, 2016.

[14] V. Vapnik, "The nature of statistical learning theory," New York: Springer-Verlag Publisher, 1995.

[15] A. Haryoko, S.H. Pramono, "Pengenalan karakter plat kendaraan bermotor berbasis citra dengan menggunakan metode Canny dan algoritma backpropagation," Journal of Mechanical Engineering and Mechatronics, pp. 93-105, 2016.

[16] W. Wang, J. Yang, M. Chen, and P. Wang, "A light CNN for end-to-end car license plates detection and recognition," IEEE Access, 7, pp. 173875-173883, 2019.

[17] I.V. Pustokhina, D.A. Pustokhin, J.J.P.C. 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, Special Section on Artificial Intelligence (AI)-Empowered Intelligent Transportation Systems, vol. 8, pp. 92907-92917, May 2020.

[18] J. Yousefi, "Image binarization using Otsu thresholding algorithm," Ontario, Canada: University of Guelph, 2011.

[19] B. Prijono, "Student notes: convolutional neural networks (CNN) introduction - belajar pembelajaran mesin Indonesia," Retrieved from Belajar Pembelajaran Mesin Indonesia - IndoML: https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction, 2018.

[20] M.T. Jones, "Deep learning architectures - IBM developer," Retrieved from IBM Developer: https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures, 2017.

[21] S. Albelwi and A. Mahmood, "A framework for designing the architectures of deep convolutional neural networks," Entropy, 19(6), 242, 2017.

[22] R. Yamashita, M. Nishio, R.K. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," Insights into Imaging 9.4, pp. 611-629, 2018.

[23] Puneet and N. Garg, "Binarization techniques used for grey scale images," International Journal of Computer Apps., 71(1), pp. 8-11, 2013.

[24] A. Rosebrock, "Image classification with keras and deep learning," Retrieved from PyImageSearch: https://www. pyimagesearch.com /2017/12/11/image-classification-with-keras-and-deep-learning, 2017.