Journal of Theoretical and Applied Information Technology
31st August 2023. Vol.101. No 16

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IMPLEMENTATION OF AUTOMATIC NUMBER PLATE RECOGNITION TO DETECT BAD DEBT VEHICLES

LUSIUS EVODIUS TANZANIA[1], ANDRY CHOWANDA[2]

[1]Computer Science Department, BINUS Graduate Program, Master of Computer Science

[2]Computer Science Department, School of Computer Science

Bina Nusantara University, Jakarta, 11480, Indonesia

Email: [1]lusiusevotan@gmail.com, [2]achowanda@binus.edu

ABSTRACT

Automatic Number Plate Recognition is very useful in various institutions. Researchers continue to develop and improve vehicle number plate detection in various countries. Each vehicle number plate has a different pattern in each country. There are various stages in the introduction process so that the vehicle number plate can be recognized. In recognizing each character, various algorithms can be used. Convolutional Neural Network (CNN) is a model that can find objects. Many researchers previously performed image classification using CNN. Many previous studies conducted research on handwriting in various types of languages. The research to develop a suitable CNN model to recognize the characters on vehicle license plates in Indonesia. The existing dataset is divided into 36 classes with 0 - 9, A - Z, and each class is approximately 1016 images in the grayscale form. In training the CNN model, the Python programming language and Keras library were used to make it easier to create layers on the CNN model. The evaluation and validation process will use a confusion matrix to find out the results of the predictions for each class. In this training, we will use LeNet and VGG16 architectures. Based on these two architectures, the best results will be selected to be used in the character recognition process on vehicle number plates. The LeNet architecture yielded 98% with 50 epochs during the training process, while VGG16 was 84% with 20 epochs.

Keywords: *CNN, ANPR, Indonesian Plate Number, Indonesian Plate Number Recognition, Image Pre-Processing*

1. INTRODUCTION

As time goes by, the need for transportation is increasing. In big cities or remote areas, we can see many people use motorized vehicles. Each motorized vehicle has its own identity, such as license plates, engine numbers, and engine frames. Due to the increasing need for transportation / motorized vehicles, they are causing residents to have their vehicles. With this, financial service companies provide credit facilities in payment for the vehicle to be purchased. Unfortunately, a credit payment system causes naughty buyers who do not fulfil their obligations to pay creditors.

Research in image recognition has been studied in the last few decades. This is because so many systems can be made from this object recognition, one of which is the recognition of letters/numbers. This is not only done as an introduction but can be developed into various products that can facilitate human beings in the future.

With this, research can be carried out to check the vehicle plate number through images taken through the camera. The image taken can be checked whether the number plate is problematic in credit payments. This is one thing that research can do in image recognition. CNN has been widely used in research to perform object recognition such as facial recognition, cat and dog recognition, handwriting recognition, including character recognition on license plates. In doing character recognition, a model that has been trained is needed so that each character can be recognized[1]. In number plate character recognition, accuracy becomes a very influential aspect so that each character in the image is recognized correctly.

As can be seen in [4][9][11][12][13][26], character recognition can be made into a variety of products that can help humans. Character recognition is a process that converts text in the form of images into a format that can be understood electronically, thus enabling computers to recognize text in the form of images [21]. Various methods can be used in image recognition, one of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

which is using the Convolutional Neural Network (CNN). As seen in [4] [9] [11] [12] [13], CNN is widely used in image processing, especially character recognition. This is because CNN has a convolution layer and a pooling layer that allows learning features hierarchically from the data. This research contributes to the development of Automatic Number Plate Recognition of Indonesian Vehicle using deep learning algorithm.

2. LITERATURE RIVIEW

In conducting this research, a literature study was conducted to understand how the ANPR works and what components are used. To understand this, we have summarized several previous studies in order to be able to conduct research on ANPR, which can be seen in table 2.1.

Table 1: The results of previous studies

No	Author	Country	Method	Result
1.	[1]	China	CNN	97%
2.	[2]	China	CNN	90,07%
3.	[3]	Malaysia	CNN	94,6%
4.	[4]	Colombia	CNN	98%
5.	[5]	Indonesia	CNN	99,35%
6.	[6]	India	CNN	84%
7.	[7]	Indonesia	CNN	99,762%

Based on research that has been done previously, as shown in table 2.1, ANPR has been made in various countries which have their format[4] so that research can be carried out to determine the best model to produce high enough accuracy for the dataset that has been collected. However, it must be admitted that finding the location of the plate to the character segmentation process is quite a difficult challenge, so this process also needs to be considered so that the extracted characters can be recognized properly.

.jpeg format is needed that can be used for training. The data source used is the English font Characters, where each character has approximately 1000 images.

The dataset used for the training was taken from a black and white image measuring 128x128. The existing dataset is divided into 36 classes including 0-9, A-Z. Because Indonesian number plates use capital letters, so that the letter class is only taken A-Z, not taking class a-z.

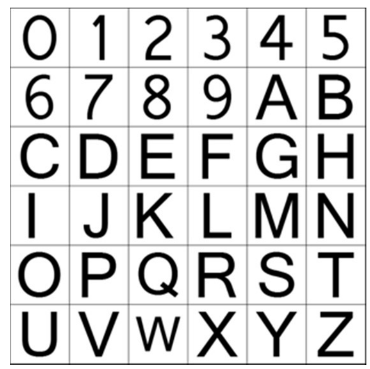

Figure 2: Sample of dataset

Figure 1 is a sample of the dataset that is used to conduct training on the model that has been created. Each of the images consists of approximately 1016 images of letters/numbers. Each class is then grouped into training, validation, and testing where the composition of each division is 70% training, 20% validation, 10% testing.

3.2. Building Convolutional Neural Network

In this sub-chapter, a CNN model will be created which will be trained. Seeing the many different architectures on CNN, but in fact their basic components are very similar[8]. The layers on CNN consist of one or more convolutional layers followed by one or more fully connected layers[6]. In this study, the LeNet architecture will be used which can be seen in Figure 3 and the VGG16 architecture in Figure 2.

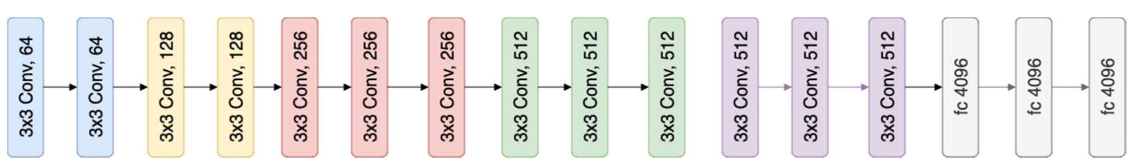

Figure 1: Architecture of VGG16

3. PROPOSED METHOD

3.1. Datasets

To recognize each character on the vehicle number plate requires a training for each letter and number, therefore a dataset in the form of images in

The convolutional layer receives image input and then performs convolution operations.

The convolution operation performs a linear combination filter operation on the local area. The convolution operation will be carried out using the kernel in the image field by shifting the convolution kernel pixel by pixel. Then the result of the convolution operation will be stored in a new matrix. The following is a function to get the output size of the convolution operation.

$$H_{out} = \frac{H_{in} - K_{height} + 2P}{S} + 1 \quad (1)$$

$$W_{out} = \frac{W_{in} - K_{width} + 2P}{S} + 1 \quad (2)$$

In the training process on CNN which aims to get high accuracy from the classification carried out, there are 2 stages in this process. The stages of this process are feedforward and backpropagation. This feedforward process works where the vector image will go through a convolution process using a number of filters or hidden layers. The backpropagation process is a stage of the training process after the feedforward is completed. The task of this process is to correct the previous weights from the output layer to the first layer. In the backpropagation process, the fully-connected layer pays attention to the error value and its derivatives in improving the weight of each hidden layer node. The output layer can calculate the error using cross-entropy.

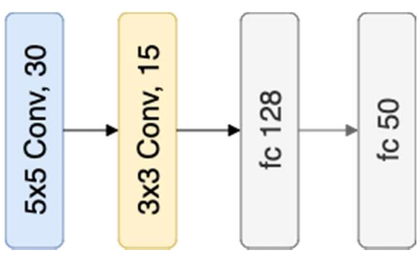

Figure 3: Architecture of LeNet

3.3. Evaluation Model

After the training process is carried out, then the testing process is carried out on the prepared images. There are a total of 16 images for each class from the dataset that has been collected. After the testing process for the model that has been trained, an evaluation of the testing that has been carried out is then carried out. The evaluation of this research uses Recall, Precision, Accuracy, and F1-Score. Recall and Precision are generally used in evaluating the effectiveness of machine learning algorithms[9].

$$Precision = \frac{TP}{(TP+FP)} \quad (3)$$

$$Recall = \frac{TP}{(TP+FN)} \quad (4)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

$$F1\ Score = \frac{2\ x\ Recall\ x\ Precision}{Recall + Precision} \quad (6)$$

True Positive (TP) is a correctly predicted positive value which means the actual class value is “YES” and the predicted class value is also “YES”. True Negative (TN) is a negative value that is correctly predicted which means the actual class value is "NO" and the predicted class value is also "NO". False Positive (FP) is when the actual class is "NO" but the predicted class is "YES", as well as False Negative (FN) where the actual class is "YES" but the predicted class is "NO".

3.4. ANPR

ANPR is an algorithm or technique that is carried out in the process of recognizing vehicle number plates without any human intervention[10] in general, ANPR consists of several stages, namely image input, image pre-processing, plate finder, character segmentation, and also character recognition.

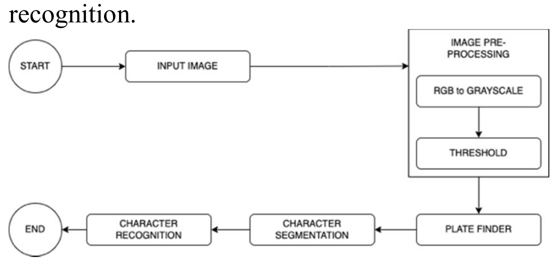

Figure 4: ANPR Algorithm

Based on Figure 4, each stage will be explained further in the following sub-chapters:

3.4.1. Input Image

This process is carried out so that we can recognize the plate from the image we take. The image format to be processed has a .JPEG format. At this stage is very influential on the image taken. The clearer the photo from good lighting will determine the results of the number plate search process.

3.4.2. Image Pre-processing

at this stage it is intended that the image that has been taken will be converted from RGB to grayscale, then a threshold will be carried out or often change the image that converts a grey image into a black image[5] so that the plate can be done properly.

3.4.3. Plate Finder

At this stage, the Python library, OpenCV, will be used to detect the vehicle number plate which will look for an object that is rectangular in shape 4 in the image.

3.4.4. Character Segmentation

This stage is a process that cuts every character on the plate. The purpose of this process is to be able to separate each letter on the vehicle plate. At this stage will also be separated between the vehicle number plate and also the validity period on the plate. It also aims to make it easier to recognize the letters on the vehicle number plate.

3.4.5. Character Recognition

At this stage, character recognition is a process of converting an image containing written words or characters into an ASCII sentence/letter that is recognized by the computer. Images that contain sentences/letters can be in the form of scanned documents, or photos. In order for the letters/numbers in the image to be recognized, a model that has been trained using the dataset provided is needed.

4. EXPERIMENTAL AND DISCUSSION

4.1. Convolutional Neural Network

Based on Figure 4, to perform character recognition on vehicle plates, the Convolutional Neural Network (CNN) algorithm is used. Based on the literature study that has been done, the CNN algorithm is quite often used in the character recognition process with good accuracy. This is the reason in this study to look for a good CNN model in carrying out the character recognition process.

In finding a good model for character recognition, several CNN architecture trainings will be conducted with the existing datasets. In making the model will be developed using python, Tensorflow, & Keras. For the training process, use a device with Intel Core I5 specifications, 8GB of RAM.

In this sub-chapter, we will discuss the models that have been made and have been trained to recognize the characters on the vehicle number plates. Two architectural models have been created on CNN itself, such as LeNet and VGG16. The training results from the LeNet architecture that have been created can be seen in table 4.1, while the VGG16 architecture that has been created can be seen in table 4.2.

Table 2: Summary of LeNet architecture training

Layer	*Output Shape*	Param #
Conv2D_1	(None, 124, 124, 30)	2280
Max_pooling2D_1	(None, 62, 62, 30)	0
Conv2D_2	(None, 60, 60, 15)	4065
Max_pooling2D_2	(None, 30, 30, 15)	0
Dropout_1	(None, 30, 30, 15)	0
Flatten_1	(None, 13500)	0
Dense_1	(None, 128)	1728128
Dense_2	(None, 50)	6450
Dense_3	(None, 36)	1836

Table 3: Summary of VGG16 architecture training

Layer	*Output Shape*	Param #
Input	(None, 224, 224, 3)	0
Conv2D	(None, 224, 224, 64)	1792
Conv2D	(None, 224, 224, 64)	36928
Max_pooling2D	(None, 112, 112, 64)	0
Conv2D	(None, 112, 112, 128)	73856
Conv2D	(None, 112, 112, 128)	147584
Max_pooling2D	(None, 56, 56, 128)	0
Conv2D	(None, 56, 56, 256)	295168
Conv2D	(None, 56, 56, 256)	590080
Conv2D	(None, 56, 56, 256)	590080
Max_pooling2D	(None, 28, 28, 256)	0
Conv2D	(None, 28, 28, 512)	1180160
Conv2D	(None, 28, 28, 512)	2359808
Conv2D	(None, 28, 28, 512)	2359808
Max_pooling2D	(None, 14, 14, 512)	0
Conv2D	(None, 14, 14, 512)	2359808
Conv2D	(None, 14, 14, 512)	2359808
Conv2D	(None, 14, 14, 512)	2359808
Max_pooling2D	(None, 7, 7, 512)	0
Flatten	(None, 25088)	2359808
Dense	(None, 4096)	2359808
Dense	(None, 4096)	2359808

In table 2 is a summary model of the LeNet architecture that has been trained with a total of 50 epochs. After training and validation, the accuracy obtained from the training results is 98% with the accuracy of the validation results is 89.83%. Each training process is recapitulated into 2 diagrams consisting of accuracy which can be seen in Figure 5 and loss which can be seen in Figure 6.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3 is a summary model of the VGG16 architecture that has been trained with a total of 20 epochs. After training and validation, the accuracy obtained during training is 84.21%, while the accuracy obtained from the validation results is 76.65%. During the training process on the VGG16 architecture, this is recapitulated into a graph which can be seen in Figures 7 and 8.

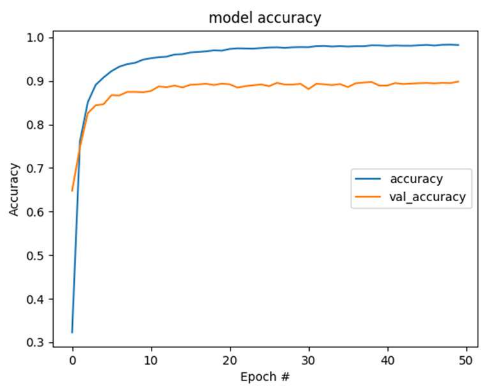

Figure 5: Accuracy graph of LeNet architecture training

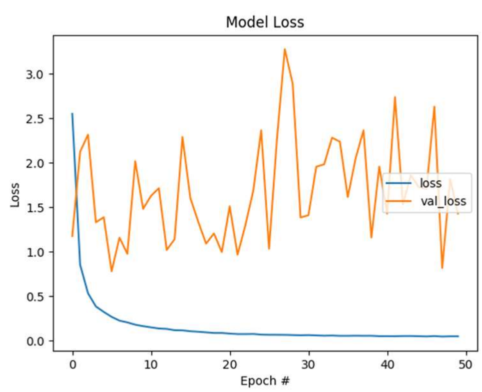

Figure 6: Loss graph of LeNet architecture training

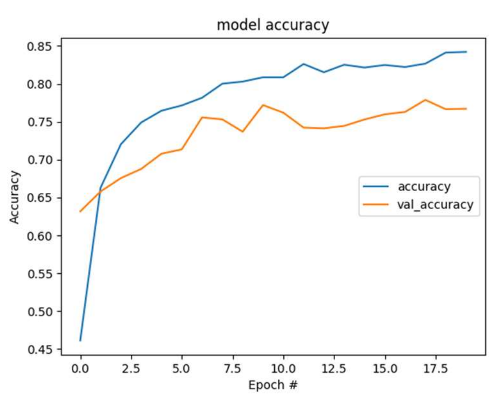

Figure 7: Accuracy graph of VGG16 architecture training

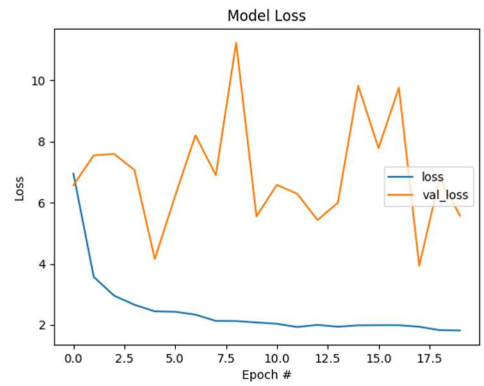

Figure 8: Loss graph of LeNet architecture training

After training on the LeNet architecture, tests were carried out on 16 images that have been separated from each class, and visualized into a confusion matrix which can be seen in Table 4.

Table 4: LeNet architecture training evaluation

class	precision	recall	f1-score	support
0	1.00	0.50	0.67	16
1	1.00	0.75	0.86	16
2	1.00	0.75	0.86	16
3	1.00	0.75	0.86	16
4	0.94	1.00	0.97	16
5	1.00	0.94	0.97	16
6	0.94	1.00	0.97	16
7	1.00	1.00	1.00	16
8	0.83	0.94	0.88	16
9	1.00	0.88	0.93	16
A	0.80	0.75	0.77	16
B	0.71	0.75	0.73	16
C	1.00	0.75	0.86	16
D	0.92	0.75	0.83	16
E	1.00	0.75	0.86	16
F	0.93	0.81	0.87	16
G	0.89	1.00	0.94	16
H	0.36	1.00	0.53	16
I	1.00	0.88	0.93	16
J	0.86	0.75	0.80	16
K	1.00	0.75	0.86	16
L	0.86	0.75	0.80	16
M	0.80	1.00	0.89	16

class	precision	recall	f1-score	support
N	1.00	0.81	0.90	16
O	0.55	1.00	0.71	17
P	1.00	0.75	0.86	16
Q	1.00	0.88	0.93	16
R	0.86	0.75	0.80	16
S	0.87	0.81	0.84	16
T	1.00	0.94	0.97	16
U	1.00	0.75	0.86	16
V	1.00	0.81	0.90	16
W	0.63	0.75	0.69	16
X	0.75	0.94	0.83	16
Y	0.76	1.00	0.86	16
Z	0.78	0.88	0.82	16
micro avg	0.84	0.84	0.84	577
macro avg	0.89	0.84	0.85	577
weighted avg	0.89	0.84	0.85	577

Table 5: VGG16 architecture training evaluation

class	precision	recall	f1-score	support
0	0.61	0.88	0.72	16
1	0.84	1.00	0.91	16
2	0.76	0.81	0.79	16
3	1.00	0.88	0.93	16
4	0.50	1.00	0.67	16
5	0.93	0.81	0.87	16
6	0.00	0.00	0.00	16
7	1.00	0.88	0.93	16
8	1.00	0.81	0.90	16
9	0.35	0.81	0.49	16
A	1.00	0.62	0.77	16
B	0.86	0.75	0.80	16
C	1.00	0.81	0.90	16
D	0.87	0.81	0.84	16
E	0.81	0.81	0.81	16
F	1.00	0.62	0.77	16
G	0.84	1.00	0.91	16
H	1.00	0.81	0.90	16
I	0.90	0.56	0.69	16

class	precision	recall	f1-score	support
J	1.00	0.88	0.93	16
K	1.00	0.75	0.86	16
L	0.67	0.75	0.71	16
M	0.75	0.75	0.75	16
N	1.00	0.75	0.86	16
O	0.80	0.71	0.75	17
P	0.88	0.88	0.88	16
Q	0.94	0.94	0.94	16
R	0.80	0.75	0.77	16
S	1.00	0.75	0.86	16
T	0.57	1.00	0.73	16
U	1.00	0.81	0.90	16
V	0.86	0.75	0.80	16
W	0.67	0.88	0.76	16
X	0.59	0.81	0.68	16
Y	0.86	0.75	0.80	16
Z	0.80	0.75	0.77	16
micro avg	0.79	0.79	0.79	577
macro avg	0.82	0.79	0.79	577
weighted avg	0.82	0.79	0.79	577

Based on tables 4 and 5, it can be seen that the results of the recall on the LeNet architecture are 0.84 or 84%, while VGG16 gets 0.79 or 79% results where recall means the ratio of positive observations that are correctly predicted to all observations in the actual class.

4.2. Automatic Number Plate Recognition

ANPR is an algorithm or technique that is carried out in the process of recognizing vehicle number plates without any human intervention[10] In general, ANPR consists of several stages, namely image input, image pre-processing, plate finder, character segmentation, and also character recognition. ANPR process flow can be seen in Figure 9.

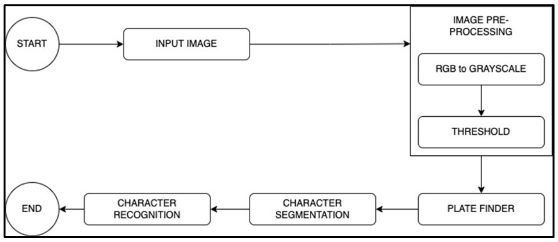

Figure 9: ANPR process flow

Based on Figure 9, it can be seen that ANPR has several stages to be carried out. The ANPR flow process will be shown in the following explanation:

4.2.1. Input Image

In this process, RGB image input is carried out on a motor vehicle in the Jakarta area as shown in Figure 10. In the picture there is 1 type of vehicle with an example of a vehicle number plate in accordance with existing regulations.

Figure 10: photo of the vehicle that will be carried out the recognition process

It can be seen in Figure 10, where the vehicle number plate can still be said to be in good condition, because the vehicle plate is still in good condition and it is hoped that plate recognition can be carried out properly.

4.2.2. Image Pre-Processing

At this stage, Figure 10 will be processed further so that a position search can be carried out on the vehicle plate. At the source of the image that has been taken, the image will be converted to grayscale as shown in Figure 11, then invert the color as shown in Figure 12.

Figure 11: Image source is converted to grayscale

Figure 12: Threshold Image

4.2.3. Plate Finder

Images that have been processed can be continued by checking the contours of the image. If there is a square 4, it will be said that the image is a vehicle number plate and then it will be cut as shown in Figure 13.

Figure 13: Vehicle license plates that have been detected and have been cropped

4.2.4. Character Segmentation

Figure 14 is a result of character segmentation on vehicle license plates that have been cropped. In the picture it can be seen that each vehicle number plate will be given a box as a marker that what is in the box is a letter / number on the vehicle number plate. At this stage, find a contour in an image using the functions in OpenCV to detect contours in an image.

Figure 14: Result of character segmentation on number plate

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.2.5. Character Recognition

After going through the character segmentation process, the last stage is the introduction of each segmented character. Each character will be predicted using the results of the model that has been trained using the CNN algorithm. The CNN model used is based on the best test results in terms of accuracy. Based on Figure 14 each segmented character will be carried out a character recognition process.

Prior to the introduction process, the image will be converted into binary to match the data set that has been trained. In this process, it will be displayed according to Figure 15 where the rows of plates that have gone through the threshold process will be displayed.

Figure 15: Draw the characters to be predicted using the trained CNN model

Based on Figure 4.7, it will then be read one by one on the image containing the characters, this is because each character has gone through a segmentation process so that each character has been separated.

5. CONCLUSION

This study concludes that each process in ANPR is very influential because the processes are sequential so that the process from the beginning of taking the image, which is then the image through image pre-processing, is then searched for the number plate by looking for contours that are square or square, then segmentation is carried out. Each character that has the plate must run well. Therefore, taking pictures is very decisive for the next process. In the training process, it is also decisive in making predictions on each character that has been segmented. It can be seen in sub-chapter 4.1 where the training results on the LeNet architecture are better, namely with 98% training accuracy results with 50 epochs, while VGG16 gets 84.21% accuracy results in training with 20 epochs.

REFERENCES:

[1] N.-A.- Alam, M. Ahsan, M. A. Based, and J. Haider, “Intelligent System for Vehicles Number Plate Detection and Recognition Using Convolutional Neural Networks,” *Technologies*, vol. 9, no. 1, p. 9, 2021, doi: 10.3390/technologies9010009.

[2] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, Mar. 2018, vol. 2018-January, pp. 1–6, doi: 10.1109/ICEngTechnol.2017.8308186.

[3] M. Arora, U. Kanjilal, and D. Varshney, “Evaluation of information retrieval: precision and recall,” *Int. J. Indian Cult. Bus. Manag.*, vol. 12, no. 2, p. 224, 2016, doi: 10.1504/ijicbm.2016.074482.

[4] D. E. Arroyo-Pérez, O. I. Alvarez-Canchila, A. Patĩo-Saucedo, H. Rostro González, and A. Patĩo-Vanegas, “Automatic recognition of Colombian car license plates using convolutional neural networks and Chars74k database,” *J. Phys. Conf. Ser.*, vol. 1547, no. 1, 2020, doi: 10.1088/1742-6596/1547/1/012024.

[5] S. Bismantoko, M. Rosyidi, U. Chasanah, A. Haryono, and T. Widodo, “a Comparison Pre-Trained Models for Automatic Indonesian License Plate Recognition,” *Maj. Ilm. Pengkaj. Ind.*, vol. 15, no. 1, 2021, doi: 10.29122/mipi.v15i1.4738.

[6] O. Bulan, V. Kozitsky, P. Ramesh, and M. Shreve, “Segmentation- and Annotation-Free License Plate Recognition with Deep Localization and Failure Identification,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 9, pp. 2351–2363, 2017, doi: 10.1109/TITS.2016.2639020.

[7] C. Cheng, L. Mei, and J. Zhang, “License Plate Recognition via Deep Convolutional Neural Network,” *IOP Conf. Ser. Earth Environ. Sci.*, vol. 189, no. 6, 2018, doi: 10.1088/1755-1315/189/6/062030.

[8] S. Das and J. Mukherjee, “Automatic License Plate Recognition Technique using Convolutional Neural Network,” *Int. J. Comput. Appl.*, vol. 169, no. 4, pp. 32–36, 2017, doi: 10.5120/ijca2017914723.

[9] D. M. Escrivá, P. Joshi, V. Mendonça, and R. Shilkrot, *Building Computer Vision Projects with OpenCV 4 and C++: Implement complex computer vision algorithms and explore deep learning and face detection*. Packt Publishing, 2019.

[10] A. S. F. Gani *et al.*, “A live-video automatic number plate recognition (anpr) system using convolutional neural network (CNN) with data labelling on an android smartphone,” *Int. J. Emerg. Technol. Adv.*

Eng., vol. 11, no. 10, pp. 88–95, 2021, doi: 10.46338/ijetae1021_11.

[11] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, 2018, doi: 10.1016/j.patcog.2017.10.013.

[12] S. Harvey and R. Harvey, "An introduction to artificial intelligence," *Appita J.*, vol. 51, no. 1, 1998, doi: 10.2514/6.1994-294.

[13] H. H. Kim, J. K. Park, J. H. Oh, and D. J. Kang, "Multi-task convolutional neural network system for license plate recognition," *Int. J. Control. Autom. Syst.*, vol. 15, no. 6, pp. 2942–2949, 2017, doi: 10.1007/s12555-016-0332-z.

[14] J. Liu, X. Li, H. Zhang, C. Liu, L. Dou, and L. Ju, "An Implementation of Number Plate Recognition without Segmentation Using Convolutional Neural Network," *Proc. - 2017 IEEE 19th Intl Conf. High Perform. Comput. Commun. HPCC 2017, 2017 IEEE 15th Intl Conf. Smart City, SmartCity 2017 2017 IEEE 3rd Intl Conf. Data Sci. Syst. DSS 2017*, vol. 2018-January, pp. 246–253, 2018, doi: 10.1109/HPCC-SmartCity-DSS.2017.32.

[15] P. Marzuki, A. R. Syafeeza, Y. C. Wong, N. A. Hamid, A. Nur Alisa, and M. M. Ibrahim, "A design of license plate recognition system using convolutional neural network," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 3, pp. 2196–2204, 2019, doi: 10.11591/ijece.v9i3.pp2196-2204.

[16] A. Narayan and R. Muthalagu, "Image Character Recognition using Convolutional Neural Networks," *Proc. 2021 IEEE 7th Int. Conf. Bio Signals, Images Instrumentation, ICBSII 2021*, no. April, 2021, doi: 10.1109/ICBSII51839.2021.9445136.

[17] H. R. Palampatla, "Automatic Number Plate Recognition Using Image Processing," *Int. J. Res. Appl. Sci. Eng. Technol.*, vol. 9, no. VII, pp. 2394–2400, 2021, doi: 10.22214/ijraset.2021.36889.

[18] S. Parvin, L. J. Rozario, and M. E. Islam, "Vehicle Number Plate Detection and Recognition Techniques: A Review," *Adv. Sci. Technol. Eng. Syst. J.*, vol. 6, no. 2, pp. 423–438, 2021, doi: 10.25046/aj060249.

[19] I. J. Rukmana, E. C. Djamal, F. Kasyidi, and A. T. Bon, "Identification of vehicle plate number using convolutional neural networks," *Proc. Int. Conf. Ind. Eng. Oper. Manag.*, no. August, pp. 1955–1965, 2020.

[20] N. Saif *et al.*, "Automatic License Plate Recognition System for Bangla License Plates using Convolutional Neural Network," *IEEE Reg. 10 Annu. Int. Conf. Proceedings/TENCON*, vol. 2019-October, pp. 925–930, 2019, doi: 10.1109/TENCON.2019.8929280.

[21] K. Sarbjit, "An Efficient Approach for Automatic Number Plate Recognition System under Image Processing," *Int. J. Adv. Res. Comput. Sci.*, vol. 5, no. (6), pp. 43–50, 2014.

[22] S. Shanavas, S. Raju, S. Sreeganesh, S. B. Nair, and A. Paul, "Modified Anpr using Neural Networks," *Int. J. Eng. Adv. Technol.*, vol. 9, no. 4s, pp. 22–26, 2020, doi: 10.35940/ijeat.a1006.0594s20.

[23] TEE KAI FENG, "License Plate Recognition Using Convolutionalrecurrent Neural Network," 2019. [Online]. Available: www.iosrjournals.org.

[24] M. Yousef, K. F. Hussain, and U. S. Mohammed, "Accurate, data-efficient, unconstrained text recognition with convolutional neural networks," *Pattern Recognit.*, vol. 108, p. 107482, 2020, doi: 10.1016/j.patcog.2020.107482.

[25] X. Zhai, F. Bensaali, and K. McDonald-Maier, "Automatic number plate recognition on FPGA," *Proc. IEEE Int. Conf. Electron. Circuits, Syst.*, no. January, pp. 325–328, 2013, doi: 10.1109/ICECS.2013.6815420.

[26] Robby, G. Abdul, et al. "Implementation of optical character recognition using tesseract with the javanese script target in android application." Procedia Computer Science 157 (2019): 499-505.