Journal of Artificial Intelligence and Software Engineering
Vol. 5, No. 2, Juni 2025, pp. 492~499
E-ISSN: 2777-001X, DOI: 10.30811/jaise.v5i2.6570

Edge Implementation of Vehicle Plate Identification using Haar Classifier and Convolutional Neural Networks

Risky Ari Wibowo[1], Fadil Muhammad[1], Ceri Ahendyarti[1], Alimuddin[1], Imamul Muttakin [1*]
[1] Department of Electrical Engineering, Universitas Sultan Ageng Tirtayasa, Cilegon, 42435, Indonesia

Informasi Artikel

Diterima	:	31 Maret 2025
Revisi	:	21 Mei 2025
Publikasi	:	20 Juni 2025

Kata Kunci:

Plate Identification
Haar Classifier
CNN
OCR
Edge Device

ABSTRAK

Meningkatnya kepemilikan kendaraan bermotor setiap tahunnya menyebabkan kurangnya monitoring informasi pada setiap kendaraan. Sebagai salah satu metode yang digunakan untuk mengetahui informasi kendaraan, pengenalan setiap plat nomor menjadi salah satu solusi untuk identifikasi kendaraan. Pemanfaatan teknik deteksi objek menggunakan *computer vision* nomor kendaraan dapat mempermudah proses pengenalan plat nomor kendaraan. Proses identifikasi dan pengklasifikasian karakter pada plat dilakukan secara bersamaan dengan implementasi yang sederhana yang menjadi keunggulan penggunaan *computer vision* dalam mengenali plat kendaraan. Penggunaan algoritma Haar cascade classifier pada penelitian ini mengatasi permasalahan deteksi plat yang dikombinasikan dengan Convolutional Neural Networks (CNN) untuk melakukan Optical Character Recognition (OCR) pada plat kendaraan. Hasil percobaan pengenalan plat kendaraan secara *in-situ* pada empat kali pengujian secara *real-time* diperoleh nilai akurasi rata-rata sebesar 42,67%.

ABSTRACT (10 PT).

The increase in vehicle ownership every year causes a lack of information monitoring on each vehicle. As one of the methods used to find vehicle information, recognizing each number plate is a solution for identifying vehicles. Utilizing object detection techniques using computer vision in recognizing vehicle number plates can simplify the plate recognition process. The process of identifying and classifying the characters on the plates is conducted simultaneously with a simple implementation which is a benefit of using computer vision in recognizing vehicle plates. The use of the Haar cascade classifier algorithm in this research overcomes the problem of plate detection combined with the Convolutional Neural Networks (CNN) to conduct Optical Character Recognition (OCR) on vehicle plates. The results of vehicle plate recognition in-situ experiments in four real-time tests obtained an average accuracy value of 42.67%.

This is an open-access article under the CC BY-SA license

***Penulis Koresponden**

Email: imamul@untirta.ac.id

Cara sitasi IEEE::

R. A. Wibowo, F. Muhammad, C. Ahendyarti, Alimuddin, & I. Muttakin, “Edge Implementation of Vehicle Plate Identification using Haar Classifier and Convolutional Neural Networks,” *Journal of Artificial Intelligence and Software Engineering (J-AISE),* vol. 5, no. 2, pp. 485-491, Juni 2025. doi: 10.30811/jaise.v5i2.6570

1. INTRODUCTION

Vehicles are a means of transportation that is widely used by humans to facilitate the activity of moving from one place to another in a very short period [1]. A vehicle license plate, also known as a vehicle police number, is an identifier owned by each vehicle as proof that the police legally identify the vehicle. Each vehicle has a different license plate containing a specific combination of numbers and letters. Vehicle license plates can store information such as the validity period of the vehicle license plate and the domicile where the vehicle is registered [2]. As a result, developing a license plate recognition system is important to know the information of each vehicle. License plate recognition is a research topic of great interest in the field of Computer Vision (CV) and Deep Learning (DL) using digital image processing [3]. CV has recently achieved remarkable results in the availability of datasets and graphics processing units (GPUs) capable of running computations on enormous amounts of data [4]. One of the applications of license plate recognition is electronic traffic law enforcement which is a utilization of license plate reading algorithms using closed circuit television (CCTV) cameras as an analysis in the event of a traffic violation [5].

The problem faced by license plate recognition is the optimality of the system in overcoming the influence of surrounding environmental conditions such as varying vehicle positions, uncertain light intensity, variations in the shape and character of license plates, and the speed of passing vehicles [6]. License plate recognition is used to identify an object so that the object requires good image quality so that the segmentation process can be conducted [7]. The solution to the problem that can be solved to overcome lighting conditions is to use a character-based object classification approach. The use of this character classification considers areas that allow the presence of characters in the vehicle license plate area [8][9]. Another problem that becomes an obstacle in the recognition of vehicle license plates is the variation in the position of the detected object. The difference in the position of the object to be detected sometimes changes into a problem in capturing the image, so it imposes the need for input data of recognition results to be compared with the dataset that has been provided [10]. A study said that to solve the problem of variation in the position of the detected object, prior knowledge of horizontal and vertical segmentation algorithms can be used [8]. By utilizing the relationship between the position of the taillight and the license plate, a simple method is obtained so that the object becomes easier to detect [11].

One of the primary challenges in license plate recognition using a webcam camera is the low quality of the captured images. To address this, the Haar cascade classifier can be applied in optical character recognition (OCR) [12]. Research on object recognition in vehicle license plates highlights the necessity of adequate hardware specifications. With its advanced capabilities, the NVIDIA Jetson Nano can handle higher computational loads for character recognition, achieving an accuracy increase of up to 87% [13]. Further studies aimed at enhancing the accuracy of Tesseract OCR and Easy OCR indicate that Easy OCR, leveraging a deep learning approach, achieves a 95% accuracy rate on the NVIDIA Jetson Nano and Raspberry Pi 3B platforms [14]. Additionally, research on multi-stage classifiers applied to Haar cascades demonstrates improved classification accuracy, reaching 95.72% when using high-quality datasets [15]. This investigation addresses challenges related to edge device implementation by integrating multiple algorithms. A webcam camera, combined with the OCR method and Haar cascade classifier, will be employed to recognize images and accurately identify characters on vehicle license plates.

2. METHOD

In this research, Haar Cascade is used as a detector to detect vehicle plates and Convolutional Neural Networks (CNN) are used to detect and classify alphanumeric characters on vehicle plates. This system algorithm integration is conducted to formulate the design structure of identifying the characters contained in the vehicle license plate. The proposed vehicle plate detection algorithm architecture in this research is shown in Figure. 1.

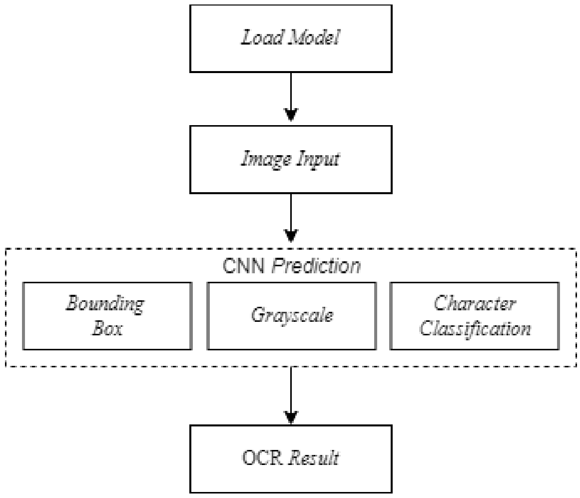

Figure 1. Proposed Algorithm Pipeline Architecture

The input data used is in the form of vehicle plate images obtained through real-time streams of vehicle traffic entering the university area using a webcam. The results of real-time plate images detected will be stored in a special folder that holds images of the region of interest of the plate and screenshot images that have a timestamp. The region of interest image of the plate will be converted into a grayscale image and scanned to see the character segmentation. If there are characters detected, then the segmentation bounding box will be given. The reading result will be classified by the OCR algorithm.

2.1. Proposed Experimental Design

In this experiment, there are details of the design during the direct simulation of the vehicle license plate detection with the model that has been prepared at the specified location (see Table 1). The tests were conducted at the university entrance gate. The determination of the research location is based on consideration of the entry point of vehicles from the main road into the university.

Table 1. Experimental Design

Parameter	Description
Coordinate	5°59’46.1” S 106°01’55.5” E
Name of lane	Jenderal Sudirman
Time of experiment	At noon
Camera position	Two meters above ground
Camera angle	160°

2.2. Interface Design

Vehicle license plate recognition is displayed online along with the plate detection process using the Haar cascade approach. On the interface, the vehicle license plate recognition system displays information in the form of date, month, and real-time, and there is a bounding box when the license plate is detected. The prepared algorithm can generate a file of vehicle plate detection in ‘.csv’ format along with an image of the plate and a screenshot of the webcam (Logitech C270) monitoring window during the detection process. The detection result file contains details related to the time the plate was detected, the reading of the vehicle plate, and the time of the final segmentation reading process. Detection results are automatically saved. Figure 2 depicts the live detection setup and interface display.

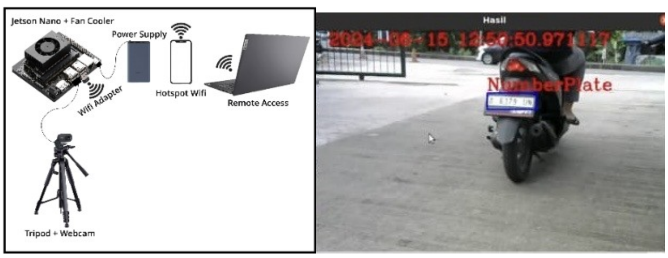

Figure 2. Live Detection Setup and Display

2.3. Performance and Evaluation

Performance testing and evaluation are conducted to determine the reliability of the system in conducting the process of detecting and reading vehicle license plates. In evaluating the accuracy of the plate reading results, there are several parameters used, namely precision, recall, accuracy, and f1-score. The use of confusion matrix in this study is to measure the performance of license plate reading based on true positive (TP), which is the number of objects in this context characters on the plate that are detected correctly; false positive (FP), which is the number of characters on the plate that are incorrectly detected; and false negative (FN), which is the number of characters that are not detected for various classes of characters read.

Precision (1) is a parameter that evaluates the performance of the model in the process of classifying characters on detected vehicle license plates based on the type of character class. In precision, the suitability of readings is the result of the comparison between true positive readings and overall positive readings.

$$Precision=\frac{TP}{TP+FP} \tag{1}$$

Recall (2) measures actual positive case number that can be identified by the model. In recall, the percentage can be calculated by comparing the detected plate readings with the actual number of vehicle plates.

$$Recall=\frac{TP}{TP+FN} \tag{2}$$

Accuracy (3) is the measure of the model in reading OCR on vehicle plates. Accuracy can be calculated using the concepts of precision and recall with true negative.

$$Accuracy=\frac{TP+TN}{TP+TN+FP+FN} \tag{3}$$

F1-score (4) is a combination of precision and recall providing a more comprehensive picture of the model's performance. It is especially important to balance between the two if there is an imbalance between positive and negative classes.

$$F1-score=2x\frac{PrecisionxRecall}{Precision+Recall} \tag{4}$$

2.4. Edge Computing Devie Configuration

In this research, the edge computing device used is NVIDIA Jetson Nano. It is not as fast as the computing process of a laptop or PC because it is only limited to a microprocessor. Therefore, to maximize the performance of the device used, it is important to do some configuration to get good test result performance. Memory partitioning is necessary because the memory allocation that is read on the internal is not complete. The partitioning of the remaining memory is done to add to the main storage. Furthermore, remote desktop connection using RDP can facilitate the testing and monitoring process and adjust the conditions at the data collection location. This configuration is done so that there is no need to use hardware devices such as monitors, mice, and keyboards directly.

3. RESULTS AND DISCUSSION

The images that had been collected and adjusted into a certain format were passed into several pre-processing procedures such as resizing and converting the image to grayscale. Next, after pre-processing was done, the collected dataset was used in the training and validation process. This stage was necessary to obtain the accuracy of the model in the classification of the various character classes that would be scanned by the system. Images that had been pre-processed were stored in '.JPG' format. The 'train' folder was dedicated for use as training data and the 'val' folder was dedicated for use as validation data. The separation into different folders was done so that the process of managing and using data became more organized. On the 'train' and 'val' folders, there were thirty-six other folders for each character class. Each class folder contains datasets of 241 to 243 data with a total of 8732 data from the 'train' and 'val' folders which would be processed into a model. From those data, the composition of the dataset that would be used in the OCR model training process is determined. In this study, the composition of the dataset used can be seen in Table 2.

Table 2. Dataset Composition

Source Dataset	Total Images	Train	Validation
Kaggle	5000	90%	10%
Roboflow	3000	90%	10%
Independent	732	90%	10%

In addition, data augmentation by characters' rotation was also conducted. This procedure is important in OCR mechanism for various shape and orientation of vehicle plate number.

Figure 3. Example of Data Augmentation

Figure 3 shows an example of augmentation of character '6' being rotated. The left-tilted character is anticipated in the case of viewing angle causes a non-straight image capture when the vehicle is approaching/leaving from/to certain trajectories.

3.1. Character Segmentation

The character segmentation process on vehicle license plates was carried out to provide a bounding box for each identified character which would be divided into several character parts on the license plate. Character segmentation on vehicle license plates was an important step in the OCR process, which was used to recognize individual characters on vehicle license plates. This process involved several stages, from image pre-processing to character extraction.

In the segmentation stage, it was important to note that the vehicle plate to be segmented must already be in a grayscale image. The main purpose of converting color images into grayscale images was to simplify the segmentation process. The next step is the detection of the character's region of interest to find the contour surrounding the character. Having found the contours, a bounding box would be created around each character. Figure 4 illustrates the result of aforementioned procedures.

Figure 4. Vehicle Number Plate Segmentation Visualization

This segmentation process could be affected by the angle of capture of the vehicle license plate, so the segmentation process became inaccurate. This segmentation process was done by utilizing the implementation of the OpenCV library in Python so that the characters contained on the plate could be identified by the model. Furthermore, the predetermined area of the plate image would be read by the OCR system and recognized for each character based on classes that have similarities with the character. So that from this process, the output is obtained in the form of reading the letter and number characters from the vehicle license plate.

3.2. Training Model

The model training stage was carried out using transfer learning with ImageNet as weights for the model to be trained. The OCR model to be trained was based on the VGG16 and VGG19 architecture models with the input data size for this model being 28x28. These two architectural models were chosen because they had an accuracy rate of 90.1% for VGG16 and 90.0% for VGG19 and these architectural models tend to be widely used in object detection. Before training the model, configuring the training path in the program script that will be run needs to be done. The configuration for the output path model was the same as configuring the dataset path used as training data. The difference is from the folder used to store the model.

The dataset training process in this research was carried out with a training script that had been prepared separately from the main script for plate detection and reading. The dataset that had been collected previously would be adapted to become a model with the ".h5" file format. Excluding top fully-connected layers and adding 32 layers for ouput stage, the total parameters are 14,733,156 (18,468 trainable; 14,714,688 non-trainable). The resulting model occupies 57,775KB of disk space.

3.3. Training Results

The dataset training process was carried out for 100 epochs which takes approximately 6 hours for each architecture on VGG16 and VGG19. The results of the training process would be stored in a special folder containing the model in the form of a file format that had been determined on the internal storage. Table 3 details the training model results.

Table 3. Training Model Results

Model	Accuracy	Loss	Val_Accuracy	Val_Loss
VGG16	97.05%	10.4%	94.96%	16.95%
VGG19	97.38%	8.79%	95.18%	16.07%

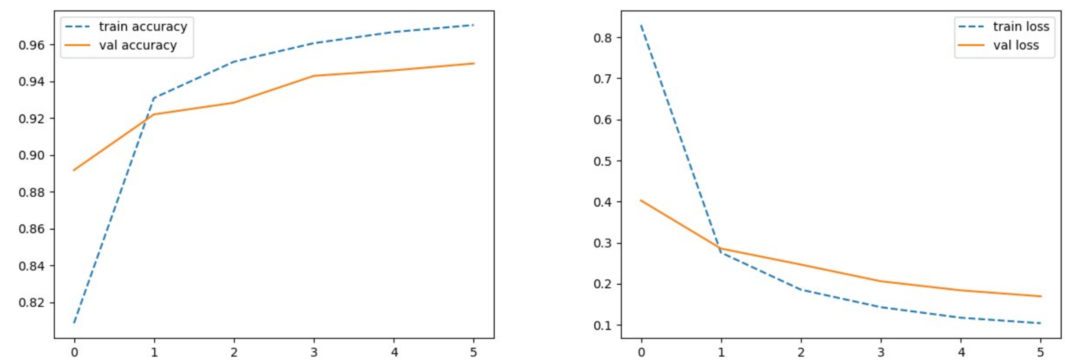

Figure 5. Accuracy and Loss for VGG16 Model

Figure 5 shows training and validation accuracy and loss for VGG16 Model. Higher training accuracy was obtained compared to validation. On the other hand, training loss is lower than validation loss. Avoiding overfitting, this model is deemed suitable for implementation.

3.4. Testing

In this research, the model results that had been exported are evaluated to determine the performance in detecting and reading vehicle license plates in real time. The assessment of vehicle license plate identification in this study was an in-situ scenario. Subsequently, the computational performance of the Jetson Nano in its implementation as a device that performs the main computational process for detecting and identifying vehicle license plates using Haar cascade and CNN is also evaluated.

Vehicle plate identification testing was conducted four times directly during the day. Vehicles passing through the university entrance gate are captured. The power supply resources used in the form of power banks have limited capacity and there was a slight decrease in FPS caused by an increase in temperature on Jetson Nano. Therefore, data collection was done in several repetitions or not continuously with a duration of 30 to 45 minutes in each test to avoid failure during data collection.

Lighting does not vary due to testing conditions during work hours. The plate angle is fixed by setting the vehicle path and camera position during testing. The vehicle speed when crossing is set at a typical level. The camera resolution is limited by the compatibility of the device used. From the elimination of these controlled test conditions, the factor that can be identified is the performance of the edge device.

Table 4. Testing

	1			2			3			4		
	Motor	Car	Total Characters	Motor	Car	Total Characters	Motor	Car	Total Characters	Motor	Car	Total Characters
Vehicle Enter	28	0	28	35	0	35	37	0	37	55	0	55

Vehicle Exit	0	0	0	21	14	35	26	0	26	64	0	64
Total	28	0	28	56	14	70	63	0	63	119	0	119
TP	9	0	9	18	5	23	29	0	29	71	0	71
FN	12	0	12	25	7	32	23	0	23	40	0	40
FP	7	0	7	13	2	15	11	0	11	8	0	8

In the first test, the total number of vehicles passing through the entrance gate was twenty vehicles, with details of twelve motorbikes with black background plates, four motorbikes with white background plates, and four cars with black background plates. From the first test results, as seen in Table 4, vehicle plates with a white background are identified by the detection system with the measurement value obtained for an accuracy of 32.14%. This accuracy can be said to be classified as exceptionally low due to the speed of vehicles passing fast so the system cannot capture it accurately.

The second test observes a total of twenty-five vehicles passing through the entrance gate, with details of eleven motorbikes with black background plates, eight motorbikes with white background plates, two cars with white background plates, and four cars with black background plates. The results detailed in Table 4 show that only vehicle plates with a white background can be identified with the measurement value obtained for the overall accuracy of 32.85%. This accuracy could be classified as exceptionally low due to the speed of vehicles passing extremely fast so the system cannot capture it accurately.

In the third test, the total number of vehicles passing through the entrance gate was fourteen vehicles, with details of five motorbikes with black plate backgrounds, and nine motorbikes with white plate backgrounds. Table 4 shows the results where only vehicle plates with a white background can be identified with the measurement value obtained for an accuracy of 46.03%. This accuracy could be classified as exceptionally low due to the speed of vehicles passing extremely fast so the system cannot capture it accurately.

The fourth test observes a total of twenty-three vehicles passing through the entrance gate, with details of four motorbikes with black background plates, seventeen motorbikes with white background plates, and two cars with black background plates. From the fourth test results in Table 4, vehicle plates with a white background could be identified with an accuracy of 59.66%. This accuracy could be classified as exceptionally low due to the speed of vehicles passing fast so the system cannot capture it accurately.

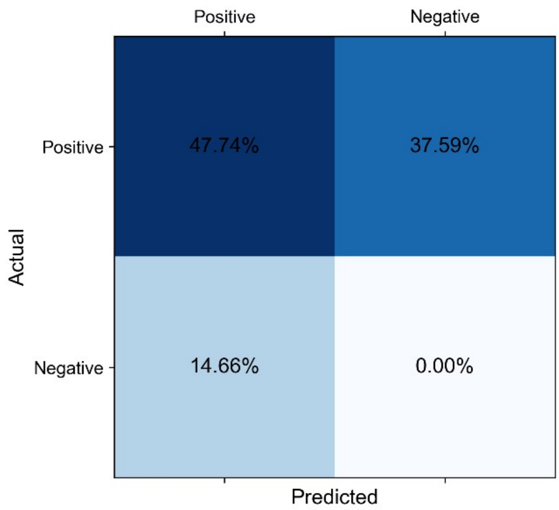

Figure 6. Confusion Matrix

Figure 6 shows confusion matrix for four test results. It provides a breakdown of a classification model's performance by comparing actual outcomes with predicted results. A high false positive rate suggests that the model frequently misidentifies negative cases as positive, which might indicate an issue with precision. Similarly, the false negative rate shows how often actual positives are incorrectly classified as negatives, affecting recall. The conformity percentage between actual positive and predicted positive indicates an acceptable result.

4. CONCLUSION

A vehicle license plate identification system for OCR has been developed and implemented on an edge device. An identification algorithm can be built utilizing CV through a combination of the Haar cascade classifier and CNN. Model results at high performance with VGG19 yielded an accuracy of 97.38%, precision of 75.7%, recall of 69.2%, and f1-score of 72.3%. In-situ testing for real-time schemes and non-real-time schemes produces an OCR average reading accuracy of 42.67%. A crucial factor that affects the performance during testing is the computational capability of the device. This research uses NVIDIA Jetson Nano as an edge device that performs computations directly through a webcam interface consuming RAM usage of 93.2%. In terms of computational processing speed for one frame, it could be classified as very slow. This processing speed is related to the time required for the system to detect and read the vehicle license plate contained in the frame. Nevertheless, this research includes original datasets obtained from in-situ experiments, which provide valuable real-world data for further research on similar themes.

ACKNOWLEDGEMENTS

The authors would like to thank members of the Applied Electromagnetics Laboratory FT UNTIRTA for their help in experiments.

REFERENCE

[1] I. H. Al Amin and A. Aprilino, "IMPLEMENTASI ALGORITMA YOLO DAN TESSERACT OCR PADA SISTEM DETEKSI PLAT NOMOR OTOMATIS," *JTI*, vol. 16, no. 1, p. 54, Jan. 2022, doi: 10.33365/jti.v16i1.1522.

[2] W. Setiawan and N. H. Farhan, "Deteksi Objek Plat Nomor Kendaraan dengan Metode CNN," *JCB*, vol. 16, no. 1, pp. 46–50, Jun. 2022, doi: 10.56447/jcb.v16i1.49.

[3] A. Anggraeni and A. P. Widyassari, "Deteksi Plat Nomor Menggunakan Connected Components dan Pra-Pengolahan untuk Mempercepat Proses Deteksi," *generation*, vol. 6, no. 2, pp. 37–44, Aug. 2022, doi: 10.29407/gj.v6i2.15785.

[4] R. Laroca *et al.*, "A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector," in *2018 International Joint Conference on Neural Networks (IJCNN)*, Rio de Janeiro, Brazil: IEEE, Jul. 2018, pp. 1–10. doi: 10.1109/IJCNN.2018.8489629.

[5] A. Bagasatwika, "Electronic Traffic Law Enforcement: Is it Able to Reduce Traffic Violations?," *Unnes L.J.*, vol. 6, no. 1, pp. 73–96, Apr. 2020, doi: 10.15294/ulj.v5i1.28642.

[6] J.-Y. Sung, S.-B. Yu, and S. P. Korea, "Real-time Automatic License Plate Recognition System using YOLOv4," in *2020 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, Seoul, Korea (South): IEEE, Nov. 2020, pp. 1–3. doi: 10.1109/ICCE-Asia49877.2020.9277050.

[7] V. Khare *et al.*, "A novel character segmentation-reconstruction approach for license plate recognition," *Expert Systems with Applications*, vol. 131, pp. 219–239, Oct. 2019, doi: 10.1016/j.eswa.2019.04.030.

[8] J. Shashirangana, H. Padmasiri, D. Meedeniya, and C. Perera, "Automated License Plate Recognition: A Survey on Methods and Techniques," *IEEE Access*, vol. 9, pp. 11203–11225, 2021, doi: 10.1109/ACCESS.2020.3047929.

[9] P. Dhar, Md. Z. Abedin, R. Karim, Fatema-Tuj-Johora, and M. S. Hossain, "Bangladeshi License Plate Recognition Using Adaboost Classifier," in *2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR)*, Spokane, WA, USA: IEEE, May 2019, pp. 342–347. doi: 10.1109/ICIEV.2019.8858580.

[10] Md. N. I. Suvon, R. Khan, and M. Ferdous, "Real Time Bangla Number Plate Recognition using Computer Vision and Convolutional Neural Network," in *2020 IEEE 2nd International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, Kota Kinabalu, Malaysia: IEEE, Sep. 2020, pp. 1–6. doi: 10.1109/IICAIET49801.2020.9257843.

[11] 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, doi: 10.1109/ACCESS.2020.2994287.

[12] A. Ashrafee, A. M. Khan, M. S. Irbaz, and M. A. A. Nasim, "Real-time Bangla License Plate Recognition System for Low Resource Video-based Applications," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, Waikoloa, HI, USA: IEEE, Jan. 2022, pp. 479–488. doi: 10.1109/WACVW54805.2022.00054.

[13] S. Suhartono, S. Gunawan Zain, and S. Sugiawan, "SISTEM OBJECT RECOGNITION PLAT NOMOR KENDARAAN UNTUK SISTEM PARKIR BANDARA," *JESSI*, vol. 3, no. 2, p. 127, Nov. 2022, doi: 10.26858/jessi.v3i2.38458.

[14] N. Awalgaonkar, P. Bartakke, and R. Chaugule, "Automatic License Plate Recognition System Using SSD," in *2021 International Symposium of Asian Control Association on Intelligent Robotics and Industrial Automation (IRIA)*, Goa, India: IEEE, Sep. 2021, pp. 394–399. doi: 10.1109/IRIA53009.2021.9588707.

[15] C. M. G. Sabóia and P. P. R. Filho, "Brazilian Mercosur License Plate Detection and Recognition Using Haar Cascade and Tesseract OCR on Synthetic Imagery," in *Intelligent Systems Design and Applications*, vol. 418, A. Abraham, N. Gandhi, T. Hanne, T.-P. Hong, T. Nogueira Rios, and W. Ding, Eds., Cham: Springer International Publishing, 2022, pp. 849–858. doi: 10.1007/978-3-030-96308-8_79.