Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 22, No. 3, July 2023, pp. 469∼480
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v22i3.2818

Detecting Vehicle Numbers Using Google Lens-Based ESP32CAM to Read Number Characters

Tukino Paryono[1], Ahmad Fauzi[1], Rizki Aulia Ananda[1], Saeful Apriyanto[2], Muhammad Khaerudin[3]
[1]Universitas Buana Perjuangan Karawang, Indonesia
[1]Universitas Buana Perjuangan Karawang, Indonesia
[2]Universitas Islam Negeri Syarif Hidayatullah, Jakarta, Indonesia
[3]Universitas Bhayangkara Jakarta Raya, Jakarta, Indonesia

Article Info

Article history:

Received March 07, 2023
Revised May 25, 2023
Accepted Juny 16, 2023

Keywords:

ESP32CAM
Google Lens
Internet Of Things
Vehicle Number Detector

ABSTRACT

Cameras as vehicle plate detection sensors have been widely applied; similar research to detect vehicle plates continues to increase. This research aimed to detect vehicle license plates using ESP32CAM and utilize photo text reading using Google Lens, which can be used online to retrieve numeric characters. The method approach was to connect Wifi connectivity to the ESP32CAM, which had been programmed to detect vehicle plates. Vehicle plates that have been detected and recognized were inputted into Google Lens to capture the resulting text from the ESP32CAM camera recording. The results of this study were that for 70 seconds, ten plate samples were obtained, which were 100% perfect in reading license plates on Google Lens, namely six plates and two plates read 90%, one plate read 60%, and one plate read 0%. The research conclusions obtained were ten samples, six samples with perfect readings, and one error sample because of the white plate color. Thus, the main objective was to obtain the results of the vehicle plate detection and retrieve the text from the recording results.

Corresponding Author:

Tukino Paryono, +6282260250247,
Faculty of Computer Science and Information Systems,
Universitas Buana Perjuangan Karawang, Karawang, Indonesia,
Email: tukino@ubpkarawang.ac.id

How to Cite:
T. Paryono, A. Fauzi, R. Nanda, S. Aripiyanto, and M. Khaerudin, ”Detecting Vehicle Numbers Using Google Lens-Based ESP32CAM to Read Number Characters”, MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer, vol. 22, no. 3, pp. 469-480, Jul. 2023.

1. INTRODUCTION

Traffic violations always occur, both under the supervision and outside the supervision of the police, where these violations lead to unwanted vehicle accidents. The police have given e-tickets to mark a violation, but traffic violators continue to increase [1]. As a marker for violating vehicles, the police need a vehicle number plate recorder to facilitate electronic ticketing. Through these license plates, the police can give warnings to vehicle owners. The number plates we know are Indonesian number plates. The State of Indonesia follows the format of three serial numbers. The front shows where the number is registered. The letter code behind itself usually indicates the location of the STNK (Vehicle Number Certificate) [2, 3]. In Figure 1, one can see an example of translating vehicle numbers. How to quickly and accurately read text from a license plate? Next, how can we use wifi to access the license plate detector remotely? Using the ESP32 CAM as a camera, the camera detection results are converted into a Google Lens recorder to retrieve text, one of which is researching vehicle plate detection. The only tools used in this study for recording vehicle license plate detection were the ESP32 CAM camera and Google Lens.

Figure 1. Vehicle Plate Detection

With these problems, we need an automatic recording device that can record and read license plates, but there have been many related studies to discuss this problem. The new problem formulation is how to get the text of the vehicle license plate online in a short time when the ESP32CAM detects the license plate. Therefore, using the ESP32CAM camcorder, recorded results in photos and videos can be processed as segmented images [4]. Thus this research needs to be investigated to reduce this problem. ESP32 can detect color segmentation received through the recognition process [5]. The ESP32CAMera is equipped with wifi to store recorded and detected data in a web browser [6]. However, the transfer speed of the ESP32CAM camera also needs to be analyzed to identify color deficiencies in images due to slower data transfers [6, 7]. The ESP32 CAM camera aims to capture face detection, but the ESP32 CAM is a type of camera that is very easy to use in changing programming algorithms and is combined with Open CV so that all types of detectors can be used. So that makes this reason to use the ESP32 CAM camera. The modified detector algorithm detects the shape of the vehicle plate. The combination of using a vehicle plate detector using ESP32 CAM and Google Lens makes it easier to compile an algorithm to retrieve text characters that have been detected, so there is no need to add the Open CV algorithm in taking vehicle text. In other words, using Google Lens makes it easier to use in retrieving text on vehicle plates.

Previous research on using ESP32CAM to detect vehicle plates has been tested in the research of Fahri Novaldi and colleagues. This study describes detecting violations at road crossings using a single shot detector on ESP32 [8, 9]. Yasinta and friends' research also explains using image segmentation from a camera to detect vehicle plates [2]. Nur Wakhidah's research explains the use of cameras in determining the character of the numbers when detecting vehicle license plates so that they can determine the numbers on the plates [10]. Research by Mohammad M. Abdellatif and friends describes similar research using ESP32CAM to detect vehicle numbers in Saudi Arabia with Arabic numeral characteristics. The results of this research are vehicle number data stored in a database using RPi hosting [11]. Ilham and friends' research explained research on utilizing the ESP32CAM camera, which can simultaneously recognize faces and vehicle numbers by storing data on a personal computer [6]. Wickramanayake and friends' research also describes similar research, namely using the ESP32CAM camera to detect vehicles and facial recognition of drivers at parking entrances and exits to indicate that the parking lot has been used or is still available [12]. This study aims to detect vehicle license plates using the ESP32CAMera and read photo text using Google Lens, so how accurately the license plates are reading using the ESP32CAM camera recorder. The novelty of this research is that the detection results using the CAM ESP32CAMera will be sent to Google Lens to detect the numbers on the vehicle's license plate specifically. The difference between this research and previous studies is the use of Google Lens. Google Lens is an image recognition technology developed by Google designed to provide relevant information about identified objects through visual analysis based on neural networks [13]. Image processing data that has been sent via a web browser (IOT) [14] will be stored in the database and will be fully processed on Google Lens. The IOT network makes it easy to transfer data on processed image processing [15]. By combining the two methods, this research is expected to provide the latest research in developing vehicle plate detection devices quickly and accurately.

2. RESEARCH METHOD

This research method uses an auto plate detection algorithm and collects motor plate detection data. The data collected through ESP32CAM camera recording will be stored in a database via a web browser and analyzed how accurately it detects vehicle plates using the ESp32CAM combined with Google Lens text [16]. This data can be sent using an IOT-based network [17]. Implementing a system IOT on a stable network will provide good and accurate data reception results in storing good recording results [18]. Figure 2 shows the research flow to obtain vehicle license plate detection data.

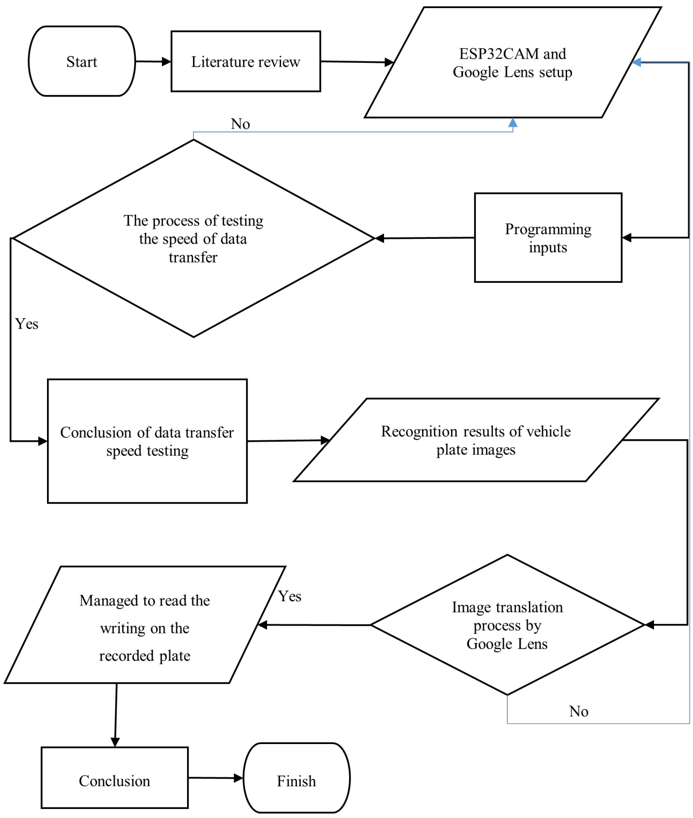


Figure 2. Research Flow

The preparations made were setting up the ESP32CAM camera along with a series of electrical systems to produce images that matched the camera. The design of an electrical system to make vehicle license plate reading products having a current of 5V DC is designed using Fritzing based on the IOT System [14, 15]. Each electrical system has an electric current and resistance to optimize the performance of the microcontroller system in processing Input and Output [19]. Figure 3 shows the ESP32CAM circuit system. Table 1 displays the names and functions of each component that has been assembled in this tool.

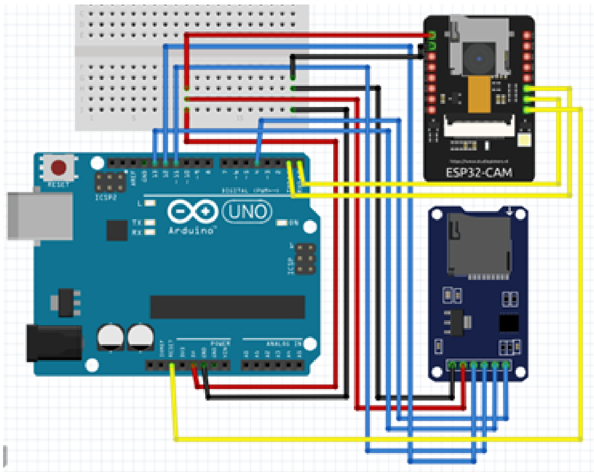


Figure 3. The Series of Components

Table 1. Vehicle Plate Research Components

Name	Function
Arduino	Components of Processing Data
ESP32CAM	Image capture face detection and recognition
MicroSD	Save of data

Table 1 shows that the Arduino and ESP32CAM functions play a role in interconnection to send commands and receive data so that data can be stored properly through an IOT-based network resulting in a monitoring system [20]. After making all the circuits in Figure 3, the vehicle plate detector programming process is carried out. The programming algorithm can be explained as follows:

1. Programming the C language is used to detect vehicle plates combined with Open CV on the PC.
2. Programming the C language on Arduino functions to store data that detects vehicle plates.
3. In the Open CV Programming, the data plate detection will be detected on the Google lens to retrieve the text of the vehicle number.

Next, prepare Google Lens to read the text on the image recognition results from the ESP32CAM camera. Figure 4 is a display on Google Lens.

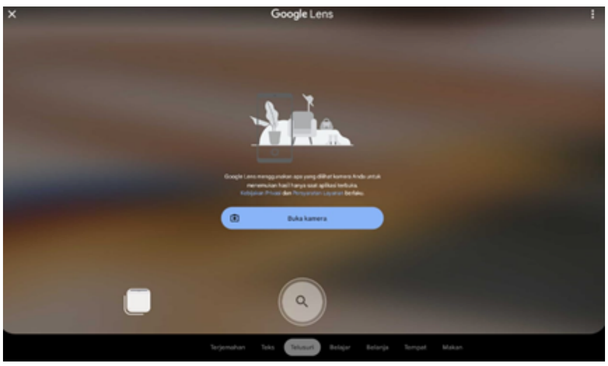


Figure 4. Google Lens

After all the preparations were made, the recording test was carried out from the ESP32CAM camera following color image recognition and processing, and the results of color image recognition and processing became data to be collected. From the processes and methods that have been described, it can be concluded that a new algorithm is programming the C language on the ESP32CAM to record and open data on the web browser and integrated into the OpenCV programming to detect the shape of the license plate on the vehicle so that the detection results can read text functions using Google Lens, reading results text can be used and stored in the database. The number detection process has several stages, namely:

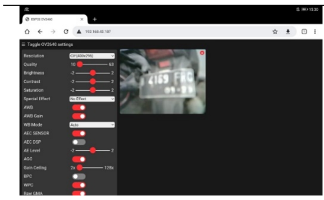
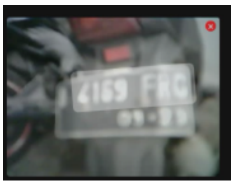
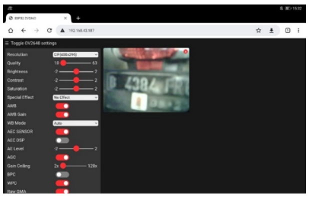
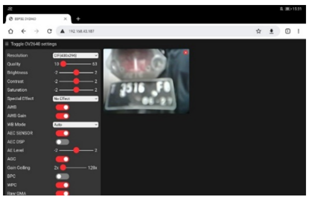
1. Measure the signal strength of the ESP32CAM against available network connectivity.
2. After performing connectivity, the ESP32 CAM provides IP and connects the IP to the PC (personal computer). The ESP32 CAM IP and Computer IP must be connected to the same wifi network.
3. After the ESP32 CAM and PC are connected, the IOT recording process is carried out on the vehicle plate.
4. The detected plate will be automatically saved on the PC.
5. The recording results will be processed on Google Lens to retrieve the text on the recording results. This process takes a long time because sending data is long.
6. The results of text retrieval by Google Lens will be inputted into Excel. Then the vehicle number stored in Excel will be used accordingly because the limitations of this research are only up-to-text retrieval via GoogleLens.

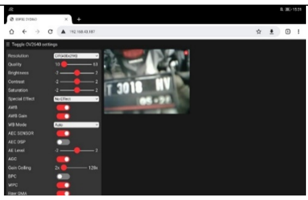
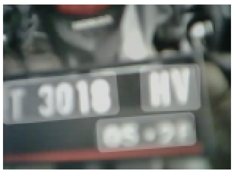
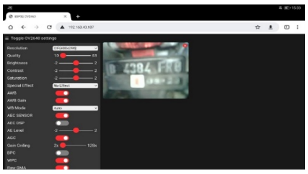
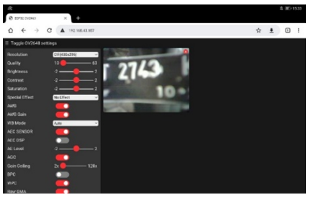
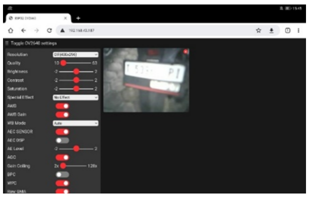
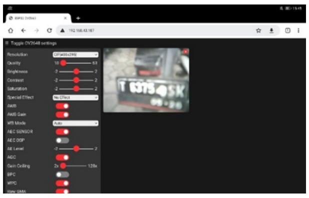
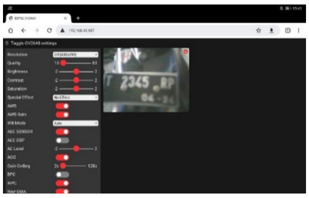
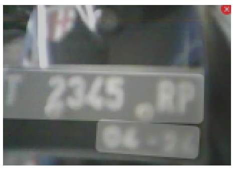
The research method and the method of conducting this research is a mixed method between quantitative methods and visual data processing algorithm methods on Open CV. Where the results of visual processing are presented in the form of numbers indicating the level of success in this study, according to or not in the data collection process according to the purpose of this study, the data that has been collected goes into Google Lens. The initial test was conducted to test the speed of reading data from plate detection to entering the IOT database. This test aimed to see how much influence the speed of data transmission has on the results of recording color images. The reading results will be inputted into Google Lens when the image is successfully recorded. The Google Lens reading process analyzes [21] how accurate the data reading results from the ESP32CAM recording are. The data source was tested on the Karawang UBP campus parking lot, taking only ten detection data samples. After conducting the test, the presentation equation for the level of accuracy can be seen in the following formula.

$$Detect = \frac{Max\ Real\ Number -\ Undetectable\ Number}{Full\ Range} \times 100 \tag{1}$$

From this equation, the calculation is carried out on the total numbers on the plate against the number of numbers that are not detected, divided by the total numbers on the plate, and multiplied by 100%, which results in a presentation of clarity on the numbers that have been detected. From testing the transfer speed and testing the reading of the vehicle license plate text, if it fails, the process will return to ESP32 preparation to check the camera lens, wifi speed, and influence on the electrical system.

Table 2. Result of Detection

Camera detection with ESP32CAM	Detection text with Google Lens	Detect Number	Percent to Data Base
		4169 FRC	90%
		4384 FRU	90%
		T 3516 FO	100%

Camera detection with ESP32CAM	Detection text with Google Lens	Detect Number	Percent to Data Base
		T 3018 HV	100%
		T 2743 KF	100%
		B 4384 FRO	100%
		2743	60%
		-	0%
		T 63754SK	100%
		T 2345 RP	100%

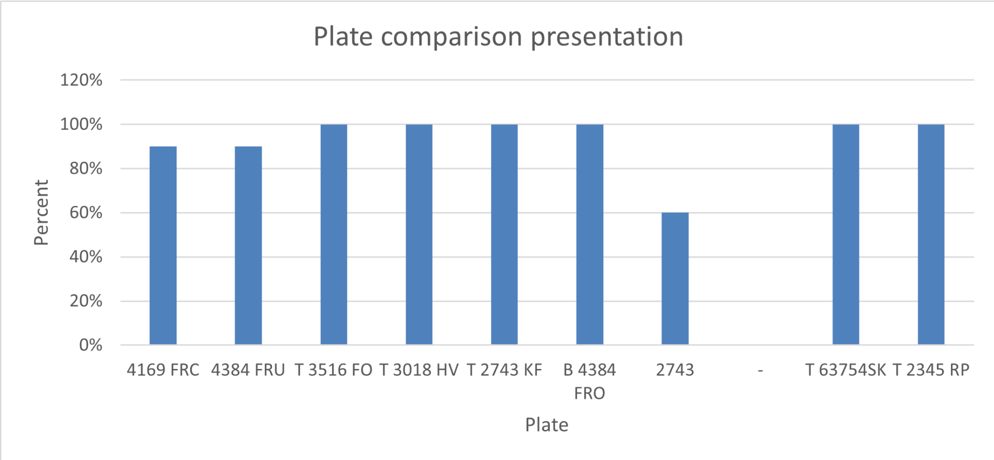


Figure 5. Chart Percentage Comparison

From some of the limitations of this study resulted in some recent findings as well, which can be improved in further research:

1. Utilization of the C programming language and Open CV in Python can make collaboration in detecting the shape of the vehicle plate on the Arduino board.
2. Using Google Lens, take the number characters on the vehicle license plate into visual text.
3. Consolidation in storing data from ESP32CAM to microSD using the Arduino Uno Board.
4. The disadvantages of the ESP32CAM camera lens can be improved by replacing the camera lens, which is better at detecting and recording vehicle license plates.

The police can swiftly and easily retrieve the owner's information from the motorized vehicle's license plate using the license plate detector on the vehicle. This research concludes that certain parties can continue to use it to quickly identify vehicle license plates and gather information on offenses committed by motorcycle owners. Moreover, as a continuation of this research, the ESP32 CAM will be used to further develop the plate detection parameter function with various data transfer rates, camera lenses, and synchronization applications. This research will be developed with this follow-up to maximize the outcomes from all facets of the current methodologies.

3. RESULT AND ANALYSIS

Following the method applied, the first test was carried out for the IOT connectivity of the ESP32CAM camera. Cameras paired and connected to a wifi router produce a visual transfer rate or frame rate (fps) and data transfer rate (b/s). The transfer speed can be seen in Figure 6.

3.1. Transfer Speed Analysis

From Figure 6, it can be concluded that the test for collecting vehicle plate data is focused on motorcycle plates. The test was carried out for 70 seconds to produce ten vehicle samples. Of the ten samples, it has a visual and data transfer speed; from the first second to the 70th second, there is an increase in the visual transfer of 5.2 fps with a data transfer rate of 1400 b/s. In other words, the display recognition on the ESP32CAMera must wait for 10 seconds to enable the sending and transferring of image processing by IOT to work properly. From 0-10 seconds, the visual transfer speed is at 4.5 fps with 1100 b/s data transfer. The delay in sending IOT data is caused by the concentration of the wifi signal on the ESP32CAM not being properly connected. After measuring how fast the data transfer and visual transfer are, the resulting recording and detection of the motor plate, the connected motor plate will be input to Google Lens.

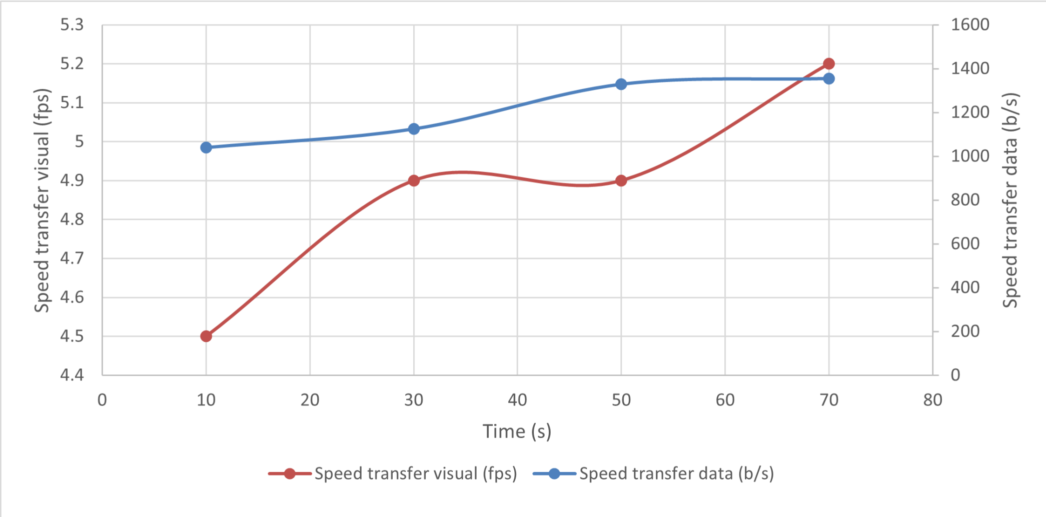


Figure 6. Graph of Transfer Speed

3.2. Google Lens Text Detection Analysis

The test was carried out for 70 seconds to produce ten detected plate samples, and then the detection results were inputted into Google Lens. Table 2 is the result of verifying the ESP32CAM recognition for input on Google Lens. Readings on Google Lens will detect the number of motorbike plates and how accurate the reading of the plate number data is. From Table 2, it can be explained that data is received using the ESP32CAM camera using the C language programming algorithm and OpenCV, the camera that has been detected will auto save on the IOT Web Browser, the stored vehicle license plate photo will be inputted to Google Lens to retrieve data in text form. From here, the test was carried out to see how accurate the ESP32CAM camera lens is in detecting vehicle plates. During the test, ten plates or samples had 100% perfect plate reading results on Google Lens, namely six plates and two plates read 90%, one read 60%, and one read 0%. The plate reads 0% or cannot detect the numbers due to the high lighting on the white plate, so it does not detect numbers on Google Lens. In other words, the numbers on a white plate are unreadable. However, the plate reads 90% and 60% of the data due to the position of the ESP32CAMera, which does not detect the plate accurately. The presentation obtained from this test is to calculate the total number entered in the detection using Google Lens. The numbers received by Google Lens will be converted into text that can be downloaded. If these numbers do not match the results recorded by the ESP32CAM camera, the presentation will show how much was read in total. The presentation is by calculating the total number on the plate from the ESP32CAM camera against the Google Lens text reading results and testing data on license plate readings on vehicles using Google Lens with the aim that the detected numbers can send data to the Indonesian Traffic Police Data Base. Because each vehicle plate has data storage in the POLANTAS Data Base according to the vehicle owner used, Figure 5 shows the presentation graph of the test of 10 plate samples during the 70-second test.

Figure 5 shows that the T 3516 FO, T 3018 HV, T 2743 KF, B 4384 FRO, T 63754 SK, and T2345 RP plates are 100% detection plate data. Of the ten plate samples and six were perfect. The presentation had a 60% success rate for plate detection and went flawlessly when detected using Google Lens. The results of this study explain the level of success of previous research in detecting vehicle plates, namely Yasinta's research and other studies that discuss vehicle plate detection [2]. This research follows previous research discussing color segmentation in detecting vehicle numbers. The differences and developments from this research are expected to provide an increase in developing vehicle plate detectors. So the main finding of this research is utilizing Google Lens to retrieve the text of the vehicle number that the ESP32CAM camera has detected. This text can store a database of vehicle numbers that commit violations on the highway. This research appears causal in previous studies using ESP32CAM only using number and vehicle number detectors without taking the text automatically. From this research, some limitations occur, among others:

1. The camera lens is still limited, so it becomes a CCTV detector which is still limited.
2. The accuracy of the lens is still limited, so that Google Lens detects numbers that are still not accurate.
3. The data transfer is slow when testing the ESP32CAM camera without attaching an additional antenna.

The police can swiftly and easily retrieve the owner's information from the motorized vehicle's license plate using the license plate detector on the vehicle. This research concludes that certain parties can continue to use it to quickly identify vehicle license plates and gather information on offenses committed by motorcycle owners. Moreover, as a continuation of this research, the ESP32 CAM will be used to further develop the plate detection parameter function with various data transfer rates, camera lenses, and synchronization applications. This research will be developed with this follow-up to maximize the outcomes from all facets of the current methodologies.

4. CONCLUSION

The conclusion of this study is to solve problems due to traffic violations that continue to increase and must be given a ticket. However, the ticket process cannot run effectively due to a shortage of police personnel. With this problem, a vehicle plate detector is needed that can find the vehicle number in the ticket process. Using the ESP32CAM as a vehicle detector and using Google Lens to determine the number of characters on the license plate are expected to provide information about the vehicle number. This finding was found because previous studies have discussed using ESP32CAM in detecting vehicle plates but only recognizing numeric characters through Open CV programming and not converting them into text form. An update from this research is to improve the trial of reading vehicle license plates and combining it with a license plate detection tool using ESP32CAM, using Google Lens to retrieve text from the results of vehicle number detection. It is expected that license plate numbers read by motorized vehicles can send data to the required database. The results of this study are in the form of a vehicle license plate detector. When testing the data transfer speed in the first second to 70 seconds experienced an increase in the visual transfer of 5.2 fps with a data transfer rate of 1400 b/s. In other words, display recognition on the ESP32CAMera must wait for 10 seconds so that the transmission and transfer of IOT image processing results can run properly, resulting in 10 detected plate samples. The detection results will be input to Google Lens. During the test, ten plates or samples using Google Lens had 100% perfect plate reading results; namely, 6 plates and two plates read 90%, one plate read 60%, and one plate read 0%. The plate reads 0% or cannot detect the numbers due to the high lighting on the white plate, so it does not detect numbers on Google Lens. In other words, the numbers on a white plate are unreadable. However, the plate reads 90% and 60% of the data due to the position of the ESP32CAMera, which does not detect the plate accuratelytesting data on license plate readings on vehicles using Google Lens with the aim that the detected number can send data to the Data Base. Suggestions for further research so that this research continues to be developed, create a database that can directly store translation data for vehicle license plates that ESP32CAM and Google Lens have detected. Furthermore, it can improve the image quality of the lens used by the ESP32CAM camera.

5. ACKNOWLEDGEMENTS

Collate acknowledgments in a separate section at the end of the article before the references and do not, therefore, include them on the title page as a footnote to the title or otherwise. List here those individuals who provided help during the research (e.g., providing language help, writing assistance, proofreading the article, and so on.

6. DECLARATIONS

AUTHOR CONTIBUTION

The author's contribution to this study was to provide the material tools and several components, then conduct research, receive data, and process research data. The results of the processing of research data are used in the writing of scientific articles. The main authors and researchers are Tukino, Rizki Aulia Nanda, and Ahmad Fauzi. Mr. Saepul Aripiyanto and Muhammad Khaerudin provided input and assisted in preparing the required components and tools.

FUNDING STATEMENT

The LPPM Buana Perjuangan University provided this funding.

COMPETING INTEREST

There is no interest in competing in any way.

REFERENCES

[1] N. Hanum Harani, C. Prianto, and M. Hasanah, "Deteksi Objek dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan Metode Convolutional Neural Network (CNN) Berbasis Python," *Jurnal Teknik Informatika*, vol. 11, no. 3, pp.

47–53, 2019.

[2] Y. O. L. Rema, “Deteksi Plat Nomor Kendaraan Bermotor dengan Segmentasi Gambar,” *Jurnal Saintek Lahan Kering*, vol. 2, no. 1, pp. 20–23, jun 2019.

[3] R. K. Varma P, S. Ganta, H. K. B, and P. Svsrk, “A Novel Method for Indian Vehicle Registration Number Plate Detection and Recognition Using Image Processing Techniques,” *Procedia Computer Science*, vol. 167, no. 2019, pp. 2623–2633, 2020.

[4] Y. Rahmawati, I. U. V. Simanjuntak, and R. B. Simorangkir, “Rancang Bangun Purwarupa Sistem Peringatan Pengendara Pelanggar Zebra Cross Berbasis Mikrokontroller ESP-32 CAM,” *Jambura Journal of Electrical and Electronics Engineering*, vol. 4, no. 2, pp. 189–195, jul 2022.

[5] M. V*, V. M. Vinod, and M. M, “Face Recognition Based Attendance System,” *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 12, pp. 520–525, oct 2019.

[6] I. F. Ashari, M. D. Satria, and M. Idris, “Parking System Optimization Based on IoT using Face and Vehicle Plat Recognition via Amazon Web Service and ESP-32 CAM (Case Study: Institut Teknologi Sumatera),” *Computer Engineering and Applications*, vol. 11, no. 2, pp. 137–153, 2022.

[7] R. Hassan, F. Qamar, M. K. Hasan, A. Hafizah, M. Aman, and A. S. Ahmed, “Internet of Things and Its Applications: A Comprehensive Survey Rosilah,” *Symmetry*, vol. 12, no. 1647, pp. 1–29, 2020.

[8] M. C. T. M. Fahri Novaldi, Iqbal Amrulloh, I Wayan Wiprayoga Wisesa, “Pendeteksian Pelanggaran Pada penyebrangan Jalan Menggunakan Single-Shot Detector Pada ESP32,” *Jurnal Teknologi Informasi Komunikasi*, vol. 9, no. 2, pp. 6–10, 2023.

[9] N. Omar, A. Sengur, and S. G. S. Al-Ali, “Cascaded Deep Learning-Based Efficient Approach for License Plate Detection and Recognition,” *Expert Systems with Applications*, vol. 149, p. 113280, jul 2020.

[10] N. Wakhidah, “Deteksi Plat Nomor Kendaraan Bermotor Berdasarkan Area pada Image Segmentation (Licence Vehicles Detection With Area Based on Image Segmentation),” *Journal Transformatika*, vol. 9, no. 2, pp. 55–63, 2012.

[11] M. M. Abdellatif, N. H. Elshabasy, A. E. Elashmawy, and M. AbdelRaheem, “A Low Cost IoT-Based Arabic License Plate Recognition Model for Smart Parking Systems,” *Ain Shams Engineering Journal*, vol. 14, no. 6, p. 102178, jun 2023.

[12] W. Wickramanayake, P. Kumararathhe, U. Jayashan, and M. Maduranga, “ParkMate: An Image Processing Based Automated Car Parking System,” in *2022 International Conference on Communication, Computing and Internet of Things (IC3IoT)*. IEEE, mar 2022, pp. 1–5.

[13] Z. I. Bilyk, Y. B. Shapovalov, V. B. Shapovalov, A. P. Megalinska, F. Andruszkiewicz, and A. Dołhańczuk-Śródka, “Assessment of Mobile Phone Applications Feasibility on Plant Recognition: Comparison with Google Lens AR-app,” *CEUR Workshop Proceedings*, vol. 2731, no. January, pp. 61–78, 2020.

[14] M. J. A. Baig, M. T. Iqbal, M. Jamil, and J. Khan, “Design and Implementation of an Open-Source Iot and Blockchain-Based Peer-To-Peer Energy Trading Platform Using ESP32-S2, Node-Red and, MQTT protocol,” *Energy Reports*, vol. 7, pp. 5733–5746, nov 2021.

[15] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, “Pre-Trained Image Processing Transformer,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2021, pp. 12 294–12 305.

[16] E. Winarno, I. Husni Al Amin, H. Februariyanti, P. W. Adi, W. Hadikurniawati, and M. T. Anwar, “Attendance System Based on Face Recognition System Using CNN-PCA Method and Real-time Camera,” in *2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*. IEEE, dec 2019, pp. 301–304.

[17] M. F. Rozi, E. S. Pramukantoro, and K. Amron, “Analisis Performansi dan Skalabilitas pada Event-Based IoT Middleware,” *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 1, no. 7, pp. 593–601, 2017.

[18] J. Xiong, L. Chen, M. Z. A. Bhuiyan, C. Cao, M. Wang, E. Luo, and X. Liu, "A Secure Data Deletion Scheme for IoT Devices Through Key Derivation Encryption and Data Analysis," *Future Generation Computer Systems*, vol. 111, pp. 741–753, oct 2020.

[19] R. A. Nanda, A. Arhami, and R. Kurniawan, "Perancangan dan Pengujian Model Mobil Robot Penanam Bibit Kangkung," *Rona Teknik Pertanian*, vol. 13, no. 2, pp. 14–28, oct 2020.

[20] K. Xing, X. Liu, Z. Liu, J. Mayer, and S. Achiche, "Low-Cost Precision Monitoring System of Machine Tools for SMEs," *Procedia CIRP*, vol. 96, pp. 347–352, 2021.

[21] M. L. Paramita, K. Orphanou, E. Christoforou, J. Otterbacher, and F. Hopfgartner, "Do You See What I See? Images of the COVID-19 Pandemic Through the Lens of Google," *Information Processing & Management*, vol. 58, no. 5, p. 102654, sep 2021.

[This page intentionally left blank.]