Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 227 (2023) 932–938

Procedia Computer Science

www.elsevier.com/locate/procedia

8th International Conference on Computer Science and Computational Intelligence (ICCSCI 2023)

Extracting Information from Vehicle Registration Plate using OCR Tesseract

Agung Yuwono Sugiyono[a], Kendricko Adrio[a], Kevin Tanuwijaya[a], Kristien Margi Suryaningrum[a]*

[a]*Computer Science Departement,School of Computer Science, Bina Nusantara University, Jakarta, Indonesia 11480*

Abstract

The increase in population and vehicle ownership causes high levels of traffic density. The very high mobility of the population ultimately has an impact on routine congestion which is felt to be getting worse over time. There are several factors that cause traffic jams to get worse. One of them is due to the increasing number of motorized vehicles. The number of vehicles that continues to increase is not in accordance with the road capacity that can accommodate these cars. For this reason, a system is needed that can categorize cars that can run on certain days according to the number plate (eg odd or even) to reduce congestion. This built system can extract the license plates of vehicles passing on the highway using image processing and character recognition methods. This research paper proposed an Automatic Number Plate Recognition ANPR system is an image processing and character recognition system that is used to recognize a car's license plate using Optical Character Recognition (OCR). The inputted license plate is automatically localized, segmented, and recognized using the OCR algorithm provided in the Tesseract library. The experiment shows 83.3% accuracy due to the difference in license plate format, background, and fonts.

Peer-review under responsibility of the scientific committee of the 8th International Conference on Computer Science and Computational Intelligence 2023

Keywords: Automatic Number Plate Recognition, License Plate, Localization, OCR – Optical Character Recognition, Segmentation, Tesseract, Vehicle Detection

*Corresponding Author.
E-mail address: kristien.s@binus.edu

Peer-review under responsibility of the scientific committee of the 8th International Conference on Computer Science and Computational Intelligence 2023
10.1016/j.procs.2023.10.600

1. Introduction

Inrix, a traffic analytics company conducted annual research in 2020 about the world's most traffic congestion city. The Inrix company observed the root and the effects of congestion against human life in 1000 cities that spread in 50 countries. On the top of the list, Bogota city in Colombia had the heaviest traffic in the world followed by Bucharest, New York, Moscow and Philadelphia as top five most congested cities in the global by impact rank. This happened due to their large populations and heavy mobility during the rush hours despite the fact that many public transportation options are provided by the government [1].

Nowadays in big cities, vehicles have become human primary needs. Only with a certain period of time, humans can travel far easily and effortlessly. However, the number of roads provided within a city is not sufficient to accommodate a huge number of those vehicles. Therefore, it has resulted in an increased amount of traffic jams. One of the solutions that the governor took is to regulate the odd-even vehicle regulation for example in Jakarta, Indonesia. But the implementation of this regulation is not effective because of the limited amount of police officers, which hampers the enforcement of said regulation.

Modern technology has become more advanced, it can outperform human weaknesses and support human activities. Moreover, Artificial Intelligence on a computer can work like a human being. Based on training from the programmer, an AI model can predict the best output from hundreds or even thousands of conditions. Computer Vision is one example of an AI model which focuses on how a computer recognizes an object from a given image or video input. Currently, Computer vision is widely used in industries to help them recognize an object.

For this application, researchers will focus on creating an ANPR systems that can categorize vehicle after its license plate is successfully read. This research implements Computer Vision especially OCR to read letters and numbers in the license plate. The most challenging part is to locate the vehicle license plate within an image before it is read with OCR. If the license plate is not being located, the reading result will be wrong. After all information within a license plate is gather, systems will categorize the vehicle into odd or even based on the last digit of numbers.

2. Literature Review

A. License Plate (LP)

The license plate is a rectangular sheet of iron containing numbers, letters and words in it and must be attached to every motorized vehicle as the identity of the vehicle. A license plate is usually written in English and has the same design in each city in a country. However, there are several countries that have local language letters such as in the State of Iraq with Arabic [2], in the State of Mongolia with the Mongol language [3], and several other countries in the world. Each letter and number in a license plate represent a meaning and divided into three sections. The first section represents a region code, the second section represents registered number, and the last section represents random letters which give more data variation. In a literature published by Tarigan, J., Diedan, R., & Suryana, Y. [4], researchers used a dataset of vehicle arounds the world [4]. In one literature, published in the IEEE by Kilic, I., & Aydin, G. [5], also only uses the dataset in Turkey so that it has a different format from other countries [5]. In addition, there is literature published by Usmankhujaev, S., Lee, S., & Kwon, J. [6], which discusses Korean license plates, where these Korean number plates have *Hanggul* writing[6].

B. Automatic Number Plate Recognition (ANPR)

Automatic Number Plate Recognition (ANPR) is a system that uses Optical Character Recognition (OCR) to read image input that contains a vehicle number plate automatically. OCR is an algorithm that can read the characters in an image and translate them into a computer, making it easier to process the information. Usually, ANPR is made on the PC platform but there is also an ANPR algorithm that is designed to work on the android platform [7].

The emergence of ANPR can be a solution in overcoming motorized vehicle management problems such as parking problems [8], difficulty obtaining vehicle data [9], and security problems [10]. Currently, ALPR technology has been widely used in countries with adequate infrastructure such as Smart Parking System (SPS) [8] [11],

sophisticated security doors or Barrier Access Control [10], and traffic violation enforcement [12].

The problems faced in using ANPR are the different formats of vehicle plates in each country, especially for vehicle plates that contain local characters [2] [3] [13]. Safaa Omran and Jumana A. Jarallah [2] examined the ANPR of Iraqi state vehicle plates, Sanja Bold and Batchimeg Sosorbaram [3] examined the ANPR of Mongolian state vehicle plates, Raluca Marina Sferlie and Elisa Valentina Moisis [13] examined ANPR license plates of countries in Europe. These research algorithms cannot be combined or used because they have different datasets.

However, in some cases, there are many obstacles that can be faced in researching this ANPR topic, one of which is the low camera quality. Therefore, it is necessary to have a more thorough examination of this topic so that all these obstacles can be resolved. One thing that can be done is to apply an algorithm to fill in the missing pixels using a deep learning approach [12]. There are several tools that can be used to conduct ANPR research, such as using the OpenCV library to detect images and the open-source OCR engine as character detection [15]. OpenCV library is used to change the color of the image into a grayscale format. This process is conducted to reduce the processing time, difference lightning conditions and to make a program to detect edges using Canny edge detector [10]. After all the edges gathered, a combination of edges that form a rectangular object can be a candidate of a license plate and will be process further by the system.

C. Optical Character Recognition (OCR)

Optical Character Recognition is an algorithm in recognizing handwriting and digital writing from an inserted image [16] or with the help of an optical instrument [9]. The algorithm works like a human reading. The eye, which is the optical instrument of a human being, captures the image per second which is then sent and processed in the brain. As with humans, OCR is also influenced by variables or factors in the success of recognizing writing. An OCR algorithm can recognize text, letters, numbers, symbols and punctuation.

The very first step of OCR is to do image preprocessing where the image will be processed first to simplify the next steps. This step usually consists of converting the image to greyscale [17] / HSV [18]. Then the image will be blurred to reduce the existing noise by using filtering, one of the algorithms that is often used is the bilateral filter [19]

The development of this OCR has been growing rapidly. One of the developments that should be explored more deeply is creating a character recognition with a real-time approach. This real time can allow us to directly detect the number plate in the video stream with a deep learning approach [20]. One of the challenges in making Real-Time Automatic License Plate Recognition is the hardware required. This hardware becomes a problem because by running a real-time system,the required hardware must have high specifications. Therefore there is literature, made by Castro-Zunti, RD, Yépez, J., & Ko, SB, who built an RT-ALPR system using low-end hardware, namely using the Rasberry Pi 3 and developed with the help of the OpenVINO library. as a library in its embedded systems [21].

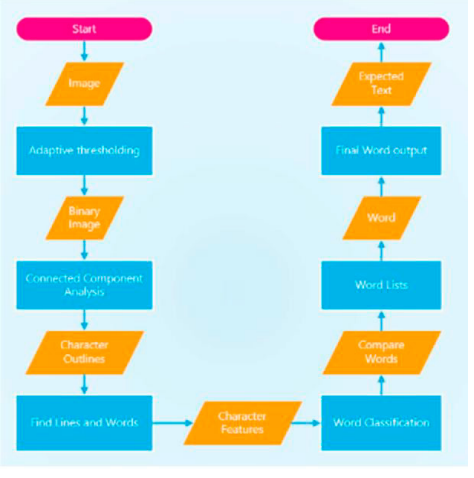

Fig 1. Program Flowchart

3. Methodology

Tesseract's architecture is shown in Figure 3.1. The first step is adaptive thresholding. In this step the input image will be converted into a binary image that only contain black and white color, this will be done using a thresholding algorithm. Thresholding is a process in which a threshold is applied in a particular scale of value, to filter in some way an image. One of the uses of thresholding in converting an image in color/greyscale into a black and white image, so applied to a histogram of an image, a value will be chosen and all value bellow that number will be converted into white (0) and all value that exceed said number will be converted to black (255). But choosing a number in random to be the threshold will result in the resulted image to be non-optimal, meaning that the produced image will be unrecognizable. To prevent this happening Tesseract uses Otsu's binarization algorithm to do the image thresholding, Otsu's algorithm will allow the program to quickly and automatically obtain the correct threshold value so that the thresholding can be executed in an optimal manner.

The next step is page layout analysis, in which the program will analyze each component that connect to said component to find region in the image that contain a shape that resemble text. After all of the image has been analyzed, regions in the image that doesn't contain shape that resemble word will be discarded and region that contain said shape will be kept for further analysis. The algorithm will split multi-column text into columns. In the next stage, the baselines of each line are detected, and the text is divided into words using definite spaces and fuzzy spaces.

In the next step, recognition of text is then started as two-pass process. In the first pass, word recognition is done using the static classifier. Each word passed satisfactory is passed to an adaptive classifier as training data. A second pass is run over the page, using the newly learned adaptive classifier in which words that were not recognized well enough are recognized again.

4. Result and Discussion

A. Experiment Result

Researchers tested 30 different kinds of car registration plate images and performed a registration test using OCR. All the inputted images sourced from the internet because researchers want to test the built system in various type of license plate. All tested images come from jpg, jpeg, webp, and jfif format, but the majority was in jpg format. All the picture that the researcher gathered was transformed into a grayscale image with the help off the inbuild method in tesseract. This experiment was carried on computer with CPU Intel i7 1065G7 and 8 GB RAM. Table 1 shows the experiment results. An overview of the experiment came with 83.3% Accuracy, which means that the program could differentiate between odd and even registration plate.

Table 1. Experiment's Result

Image	Image Type	Program Result	Expected Result	Detected Number	Detected Odd/Even	Expected Odd/Even
1	Grayscale	BSA%335CO]	SA 335CO	335	Odd	Odd
2	Grayscale	CZ20FSE	CZ20FSE	20	Even	Even
3	Grayscale	MH12DE1433	MH12DE1433	1433	Odd	Odd
4	Grayscale	M7 ERC	M7 ERC	7	Odd	Odd
5	Grayscale	749 ADO	749 ADO	749	Odd	Odd
6	Grayscale	"LR33 TEE	LR33 TEE	33	Odd	Odd
7	Grayscale	"MH 04 JM 8765	MH 04 JM 8765	8765	Odd	Odd
8	Grayscale	884 VD 69	884 VD 69	69	Odd	Odd
9	Grayscale	AVE 068:	AVE 068	68	Even	Even
10	Grayscale	FBR	MII 888	-	-	Even
11	Grayscale	no G JO7DAIISS	G JO7DA9988	-	-	Even
12	Grayscale	MH.02.BY.3123	MH.02.BY.3123	3123	Odd	Odd
13	Grayscale	WH 20 EJ 0365)	WH 20 EJ 0365	365	Odd	Odd
14	Grayscale	PZ65 BYY	PZ65 BYV	65	Odd	Odd
15	Grayscale	NULL	GJ05 JD9759	-	-	Odd
16	Grayscale	A-24(4	**A-2474**	24	Even	Even
17	Grayscale	187550	D 87550	187550	Even	Even
18	Grayscale	Null	7MAG222	-	-	Even
19	Grayscale	BKTP: 665	BKTP 665	665	Odd	Odd
20	Grayscale	CNFG*/85	CMFG 785	85	Odd	Odd
21	Grayscale	/30 Wd	S 1470 WJ	30	Even	Even
22	Grayscale	Null	G 1023 XX	-	-	Even
23	Grayscale	TN-326-G	TN-326-G	326	Even	Even
24	Grayscale	42-33	42-93	4233	Odd	Odd
25	Grayscale	AD4 18	AD4 18U	4 18	Even	Even
26	Grayscale	884 VD 69	884 VD 69	69	Odd	Odd
27	Grayscale	PZ62 FD	PZ62 FDX	62	Even	Even
28	Grayscale	SN66 XMZ	SN66 XMZ	66	Even	Even
29	Grayscale	ASD 29238	ASD 9238	9238	Even	Even
30	Grayscale	FZJ 557	FZJ 55T	557	Odd	Odd
			Accuracy (in percent)		83.33%	

B. Disscusion

As seen on table 1 the program has been run on some picture with the result in said table. Some images were detected without any error or extra character, but some of the image have extra characters that was detected. For example, take Figure 2 the expected result is “SA 335CO” but the result produced is “BSA%335CO]”. As we could see there are 2 extra character and 1 wrong character. This is happened during the word classification state where tesseract compares the segmented word with its dataset. Since the tesseract dataset only contains regular symbol [22], the result was not completely the same.

Fig 2. Sample Image 1

Take another example in Figure 4.2 the expected result is “LR33 TEE” but the program output was ““LR33 TEE” which contains an extra character in the front. Researcher believe this error happened as the result of the poor quality of the image has since researchers picked the sample image from internet randomly. The image quality is 292px x 173px. In fact, to get a good result from a license plate, the recommended pixel is between 85x35 to 300x120 pixel according to Jianjun Gao Research [23]. The program successfully found the license plate and the tesseract framework is successfully executed but because of the low quality image lead to the interpretation failure

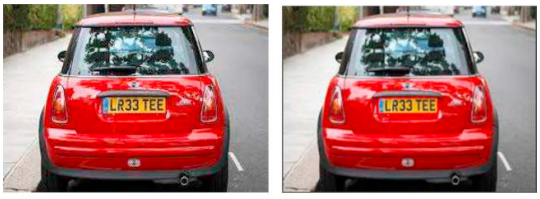

Fig 3. Sample Image 6

Another example in Figure 4. The expected result is “MII 888” however the output was “FBR” which means that none from the output matches the any characters in the picture. This happened due to size of the license plate in the image, so tesseract OCR interpreted the target license plate in incorrect way. This sample images related with research conducted by Jianjun Gao that mention the pixel of a license plate before reading [23]

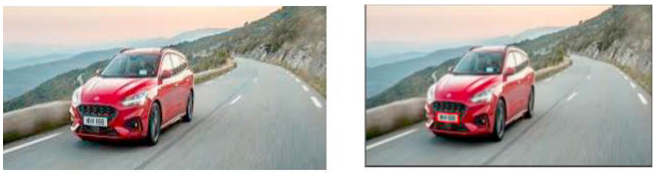

Fig 4. Sample Image 10

Another case that researchers found out during the experiment was the image number 14 that shown on Figure 5. There is one mismatch characters at the end of the result, the expected character is “V” however the program interpretation is “Y”. The error happened during the word classification state where tesseract reads the letter. Since the image was taken in tilted way, the reading process get an issue by the reason of the system does not correct normalize the angle before the reading process. Compares to previous research [2][10][11][23], a license plate was taken perpendicularly or 180° with the camera for the best result.

Fig 5. Sample Image 14

One interesting case is on Figure 6 where the program detected the registration plate, but the output is null (empty). This is usually happened during the comparison process where no character matches in tesseract dataset. According to Omran, S. S. research [2], A model must be trained with the related dataset, the model can determine the target character outside the default dataset provided by OCR. First, researcher does not aware of the difference font might affect the reading result.

Fig 6. Sample Image 15

5. Conclusion and Suggestion

A. Conclusion

The preprocessed image shows the crop image of the plate itself. Most of the time, openCV library could detect the plate number and then crop it so tesseract could read the image and convert it to text. But sometimes, openCV could not detect the border of plate. This happen because the OpenCV library search for the rectangle border, but the input plate is not always in rectangle shape because it depends on the angle of photo itself. The experiment's result show that tesseract could read the plate number in the photo. Most of the time, tesseract could detect the text from preprocessed image. But there are some problems when the tesseract could not detect the font of the plate. If the tesseract could not detect, it would be affected the output. After the tesseract detect the text, then the output will be preprocessed again to clean the extra trailing space in the text and eliminate the non-numeric character. Then the program will determine whether the plate is odd or even.

B. Suggestion

1. As the result of our code, the code must be enhanced with new model. To read special characters, a model must be trained with letters and numbers dataset so the prediction from tesseract library will be more precise.
2. To reduce the preprocessing image problem, the model also must be enhanced with new image filtering process such as Sobel mask operator so that openCV library could detect the edge of the plate number precisely.

References

[1] Pishue, B. (2021). *2020 Inrix Global Traffic Scorecard.* Kirkland: Inrix.

[2] Omran, S. S., & Jarallah, J. A. (2017, March). Iraqi car license plate recognition using OCR. In *2017 annual conference on new trends in information & communications technology applications (NTICT)* (pp. 298-303). IEEE.

[3] Bold, S., & Sosorbaram, B. (2017). Smart license plate recognition using optical character recognition based on the multicopter. *International Journal on Recent and Innovation Trends in Computing and Communication*, *5*(9), 92-96.

[4] Tarigan, J., Diedan, R., & Suryana, Y. (2017). Plate recognition using backpropagation neural network and genetic algorithm. Procedia Computer Science, 116, 365-372.

[5] Kilic, I., & Aydin, G. (2018, September). Turkish vehicle license plate recognition using deep learning. In 2018 International Conference on Artificial Intelligence and Data Processing (IDAP) (pp. 1-5). IEEE.

[6] Usmankhujaev, S., Lee, S., & Kwon, J. (2019, June). Korean license plate recognition system using combined neural networks. In International Symposium on Distributed Computing and Artificial Intelligence (pp. 10-17). Springer, Cham.

[7] Gunawan, T. S., Mutholib, A., & Kartiwi, M. (2017). Design of automatic number plate recognition on Android smartphone platform. *Indonesian Journal of Electrical Engineering and Computer Science*, *5*(1), 99-108.

[8] Farag, M. S., El Din, M. M., & El Shenbary, H. A. (2019). Parking entrance control using license plate detection and recognition. *Indonesian Journal of Electrical Engineering and Computer Science*, *15*(1), 476-483.

[9] Salimah, U., Maharani, V., & Nursyanti, R. (2021, March). Automatic License Plate Recognition Using Optical Character Recognition. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1115, No. 1, p. 012023). IOP Publishing.

[10] Ullah, F., Anwar, H., Shahzadi, I., Ur Rehman, A., Mehmood, S., Niaz, S., ... & Kwak, D. (2019). Barrier Access Control Using Sensors Platform and Vehicle License Plate Characters Recognition. *Sensors*, *19*(13), 3015.

[11] Dalarmelina, N. D. V., Teixeira, M. A., & Meneguette, R. I. (2020). A real-time automatic plate recognition system based on optical character recognition and wireless sensor networks for ITS. *Sensors*, *20*(1), 55.

[12] Sahu, P., & Pareyani, S. (2019). Features Extraction OCR Algorithm in Indian License Plates.

[13] Sferle, R. M., & Moisi, E. V. (2019, June). Automatic Number Plate Recognition for a Smart Service Auto. In *2019 15th International Conference on Engineering of Modern Electric Systems (EMES)* (pp. 57-60). IEEE.

[14] Seibel, H., Goldenstein, S., & Rocha, A. (2017). Eyes on the target: Super-resolution and license-plate recognition in low-quality surveillance videos. IEEE access, 5, 20020-20035.

[15] Agbemenu, A. S., Yankey, J., & Addo, E. O. (2018). An automatic number plate recognition system using opencv and tesseract ocr engine. International Journal of Computer Applications, 180(43), 1-5.

[16] Nayak, V., Holla, S. P., AkshayaKumar, K. M., & Gururaj, C. (2020). Automatic number plate recognition. *International Journal*, *9*(3).

[17] Kaur, E. K., & Banga, V. K. (2013). Number plate recognition using OCR technique. *International Journal of Research in Engineering and Technology*, *2*(09), 286290.

[18] Kakani, B. V., Gandhi, D., & Jani, S. (2017, July). Improved OCR based automatic vehicle number plate recognition using features trained neural network. In *2017 8th international conference on computing, communication and networking technologies (ICCCNT)* (pp. 1-6). IEEE.

[19] Vedika Kamble , Chinmayi Gurav , Sharmishtha Mohite , Rupali Gurav, Prof. Neha S. Sakhalkar, 2019, A Review Paper on Vehicle Number Plate Recognition, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 08, Issue 04 (April – 2019).

[20] Hashmi, S. N., Kumar, K., Khandelwal, S., Lochan, D., & Mittal, S. (2019). Real Time License Plate Recognition from Video Streams using Deep Learning. *International Journal of Information Retrieval Research (IJIRR)*, *9*(1), 65-87.

[21] Castro-Zunti, R. D., Yépez, J., & Ko, S. B. (2020). License plate segmentation and recognition system using deep learning and OpenVINO. IET Intelligent Transport Systems, 14(2), 119-126.

[22] Liang, Jisheng & Chalana, Vikram & Phillips, Ihsin & Haralick, Robert. (2000). A Methodology for Special Symbol Recognitions.. 4. 4011-4014. 10.1109/ICPR.2000.902854.

[23] Gao, Jianjun & Blasch, Erik & Pham, Khanh & Chen, Genshe & Shen, Dan & Wang, Zhonghai. (2013). Automatic Vehicle License Plate Recognition with Color Component Texture Detection and Template Matching. Proceedings of SPIE - The International Society for Optical Engineering. 8739. 87390Z. 10.1117/12.2014595.