

PAPER 86 – DEVELOPMENT OF AN AUTOMATED PLATE NUMBER RECOGNITION AND IDENTIFICATION SYSTEM

Z. Haruna, U. Abdulmalik*, N. S. Usman, A. Umar, Z. M. Abubakar

Department of Computer Engineering, Faculty of Engineering, Ahmadu Bello University, Zaria-Nigeria.

*Email: fabulouskaura@gmail.com

ABSTRACT

In this paper, an automated plate number recognition and identification system was developed to explore problems associated with many successful solutions in the field of computer vision. However, these solutions are typically tuned towards a particular environment due to the variations in the features of plate numbers across the universe. As such, a universal solution would be difficult to realize as the image analysis techniques used in developing this algorithm cannot generate a result with a hundred percent accuracy. Thus, this paper focuses on the development of an automated plate number recognition and identification system optimized for Nigerian plate numbers. The system comprises algorithms written in Python with the OpenCV library, canny edge detection, and feature detection techniques combined with the Tesseract optical character recognition engine to identify the characters on the plate. A Microsoft visual basic was used to develop a user interface for data presentation and MySQL server was used as a database. When a local image was tested using the developed recognition algorithm, the simulation result presented an accuracy of 90% at the localization stage and 70% for character extraction. This shows that the algorithm effectively localizes and extracts the characters on the plate number.

KEYWORDS: Plate number, OpenCV, canny edge detection, tesseract optical character recognition engine, MySQL.

1. INTRODUCTION

Modern intelligent transportation systems (ITS) rely heavily on vehicles. Plate characters offer a uniform method of identifying any vehicle. Vehicle License Plate Recognition (VLPR) has drawn a lot of interest from scientists. It can be utilized by government organizations to locate or identify stolen cars and to gather information for traffic management and planning. VLPR requires generalization and high accuracy in real-world applications due to its intimate ties to public security. Due to their diverse applications, such as electronic toll collection, highway surveillance, unattended parking lots and security monitoring of restricted areas, urban logistics, and traffic law enforcement, intelligent transport systems (ITS) significantly contribute to the advancement of smart cities (Ugwu *et al.*, 2022). Since each car's license plate is unique, automatic plate detection will have a wide range of uses. For instance, while employing intelligent parking, the entrance gate will only be opened and closed for allowed vehicles. Additionally, in vehicles, speed control operations and management of vehicles through tolls will be implemented (Akbarzadeh *et al.*, 2022). Different models were developed for various hardware configurations based on computational power and low cost using a novel set of neural networks. The suggested model precision, energy efficiency, communication latency, and computing delay are verified using various license plate datasets both during the day and at night as well as in real-time presented (Padmasiri *et al.*, 2022). Implemented automatic number plate recognition technology in police patrol cars to automate access control and security, such as identifying fast vehicles on highways and locating stolen cars in real time (Etomi & Onyishi, 2021). Using an automated license plate recognition (ALPR) system in resource-constrained contexts can help with the construction and management of smart cities as demonstrated by Padmasiri *et al.* (2022). This stage is heavily reliant on the number plate's specifications/characteristics. These characteristics include symmetry, spatial frequency, shape, color, height, and width. The conditions (lighting, visibility, image skew, and camera quality) under which the image was taken, as well as the characteristics of the image itself, determine the algorithm's capacity to recognize the plate. The majority of ANPR algorithms also include a pre-processing stage that gets the image ready using methods like grey scaling, thresholding, and noise removal. The most popular way to scale color pixels to grey is to simply average their RGB values (Etomi & Onyishi, 2021). It is impossible to overstate the importance of an automatic licensed plate detection system (ALPDS) in the modern world. This is because it is crucial to the development of smart cities' intelligent transportation systems. In both the academic and commercial sectors, there is growing interest in the field of license plate number detection (Ugwu *et al.*, 2022). Thus, this paper aimed at developing automated plate number recognition and identification system that explores different successful solutions in the field of computer vision. The system comprises stages: the input stage (camera for image acquisition), the processing stage (conversion to greyscale, filtering, contour detection, extraction, and recognition), plate number identification, and finally, the ticketing stage from a visual basic database.

2. THEORETICAL ANALYSIS

2.1 The overview of major keywords in this area of research.

Li and Wang (2022) presented Python as a powerful dynamic programming language that is free, open-source, and cross-platform. It features a vast number of potent built-in objects, standard libraries, and extension libraries, and it supports a variety of programming paradigms. By directly calling the built-in functions or standard libraries, powerful programming functionalities can be implemented. Students can expand their comprehension of the algorithm and principle by studying the source code, which will help them develop their programming skills and code quality (Li & Wang, 2022). Python is a reasonably simple language to learn. Dome and Sathe (2021) presented OpenCV as a library for many programming tasks with the main goal of offering real-time computer vision. Users can import image, does image segmentation, and extracts the image (Li & Wang, 2022). Intel's research led to the creation of the OpenCV computer vision library's source code. The majority of image-processing software programs are developed in C/C++, and OpenCV is no exception. This is done to account for computing speed. It is a library built on the IPL that has more than 300 C functions and C++ classes (Image Processing Library). Thakare *et al.* (2018) presented an optical character recognition (OCR) tool for Python software called Python Tesseract. In other words, it will recognize and read any text that is embedded deeply within an image. Tesseract-OCR is implemented by the Python-Tesseract module in Python to transform the image into text. Utilizing a module to call the specified method, image to a string makes the implementation easier. Dome and Sathe (2021) use optical character recognition (OCR) to identify text contained in a digital image. It is mostly used to transform printed, handwritten, or transcribed text into editable and reusable text data. Alfath *et al.* (2022) referred to Visual Basic as a programming language, Visual Basic is also a tool for creating Windows-based application programs. A database is a collection of connected data that makes it simpler for activities to access information Wang *et al.* (2018). Setting the administrator permission enhanced the system's security. It was crucial to develop the database management system since it offered crucial technical support.

2.2 This section discusses related works done in this area of research

Mujawar and Vadi (2022) proposed a Raspberry Pi-based system for automatically identifying license plates that use image processing. In this system, a Raspberry Pi is coupled to a camera and an LCD circuit. The tool continuously looks for signs of license plates in the incoming camera footage. When it notices a number plate in front of the camera, it analyses the camera input and extracts the number plate component from the image. The extracted image is processed using OCR to get the license plate number. The program then displays the extracted number on an LCD. Mujawar and Vadi (2022) presented a system in which when it notices a number plate in front of the camera, it analyses the camera input and extracts the number plate component from the image. The extracted image is processed using OCR to get the license plate number. Etomi and Onyishi (2021) work on a Raspberry Pi, an automated system that makes use of the OpenCV library and py-tesseract-OCR demonstrated. After the camera takes a picture, the Raspberry Pi converts it to grayscale, applies filters, segments it, and recognizes characters before displaying the number plate on the console and then storing it in a database. Making sure the code is effective and that the computational complexity is reduced while improving flexibility is one of the trickiest parts of creating a portable program. The performance time is also shown, and it shows that there is a notable 3-second delay in the final output throughout the entire procedure. AG (2019) presented an Optical Character Recognition (OCR) on images and OpenCV techniques are used by the current Automatic Number Plate Recognition and Identification system to read and identify vehicle license plates. This work concentrated on license plate localization and extraction from images for use in various applications. It does this by utilizing several Python-developed OpenCV library techniques for image segmentation, from which some image segments were checked for characters. AG (2019) identifies the most likely license plate by using the characters on the plate's associated picture's height, width, aspect ratio, and other characteristics. Using these pixel properties, it was able to remove undesired contour lines and curves that stood out as noise while identifying the actual area of the image that contained the plate number. Following the acquisition of the region, OCR was used to extract the license plate text format using a trained template for various character styles. Akbarzadeh *et al.* (2022) developed an improved algorithm to carry out the plan. First, we go through many difficult steps. The placement of the plate is determined as the initial stage. In the second phase, an initial improvement to reduce the expected noise using a suitable filter, which is provided by the Gaussian function. The remaining stages of the program are: locating the edge of photographs, enhancing updated photos, and choosing the precise location of the plate. The study's ultimate step eventually entails several stages, such as using a neural network to automatically extract the characters from the plate. Padmasiri *et al.* (2022) provided an Automated License Plate Recognition (ALPR) solution for edge computing in situations with limited resources, which could enable the creation and administration of smart cities. The accompanying environmental restrictions, such as changing weather, plate variances among areas, vehicle speed, distorted characters, filthy plates, shadow, and reflection, make research

on ALPR though even though it is a well-established area in the field of image processing. Ugwu *et al.* (2022) deliver a clever technique that uses computer vision to recognize car license numbers. Images of car license plates were used as training data for the system. Following the identification procedure, the trained images were pre-processed using the OpenCV library for image conversion and masking. The training images were initially annotated using the Visual Graphic Generator (VGG) annotation tool. Then, Tesseract-OCR was applied to isolate the texts from the photos. BC *et al.* (2022) presented a surveillance system that records images of moving targets and reads their license plates, which become a significant area of study this time. The dataset for Indian license plates was used in the study to identify license plates. Kaur *et al.* (2022) give an example of a successful ALPR system that recognizes characters with a CNN. The system efficiency was improved by applying a combination of pre-processing and morphological techniques to improve input image quality. The system can recognize license plates with several lines, slanted text, and multiple fonts, among other qualities.

3. MATERIALS AND METHODS

In this section, the procedure follow for the development of an automated plate number recognition and identification system was presented, which comprises some of the following interfaces as materials for both the hardware and software such as the programming language using Python, Pycharm IDE, Tesseract-OCR, USB camera, Microsoft visual basic, and SQL server management studio as the database in actualizing the research procedure. The methods used in achieving the objectives of this research are presented in this section as follows:

3.1 Implementation of plate number recognition and identification algorithm

Plate number recognition and identification algorithm was implemented on the pre-processed images using OpenCV and pytesseract. OpenCV was used for image processing tasks such as image filtering, edge detection, and extraction of contour image while pytesseract was used to extract the characters on the contour image. These tasks were programmed using Python with the aid of Pycharm IDE.

a. Local plate number images

In this paper, 115 local plate numbers were acquired by capturing images of vehicles with plate numbers using an Android mobile phone with the following specification: 1GBRAM, 2.0Hz*4 processor, 8MP camera, 2432x2432 image size. Some of the captured images are shown in Plate I as follows:



a. Front View

b. Rear View

Plate I: Sample Local Captured Images

The acquired images were pre-processed by reducing the image size to 500x360 and cropping the image to view the front or rear area of the vehicle as shown in Plate II.



a. Front View

b. Rear View

Plate II: Resizing and Cropping of Images

The next stage of image pre-processing is image conversion, i.e., converting an image from blue ray to grayscale to reduce the high resolution of the image. This is then followed by the final stage which is the image filtering, i.e., the stage at which noise is removed from the image with the aid of a bilateral filter (Blurring).

a. Image localization

In this step, the following processes were followed:

- i. Canny edge detection technique was used to acquire the edges of the vehicle in the image and then used contours to ignore the unwanted contours in the image, such as the windscreen, spotlight, etc.
- ii. Loop over contours was used to find the rectangle of the plate number.
- iii. After localizing the plate number, the remaining background is negligible, then we proceed with removing the image except for the plate number contour.

b. Extract of character from a plate number

The following are the steps implemented for the extraction of characters from the localized plate number

- i. Character segmentation cropped the image and save it with a new name.
- ii. The last step is character recognition which reads the plate number character in the image and used a command to print the extracted character in the terminal.

The flowchart for the automatic plate number detection and recognition system for the plate number extraction from a captured vehicle image is presented in Figure 3.1 comprising of pre-processing of the captured image; conversion, contours detection, and also the plate number recognition:

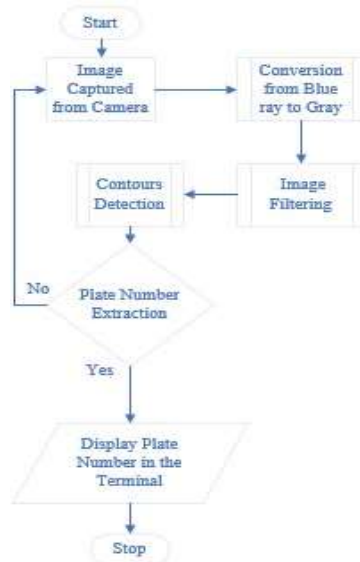


Figure 3.1: Vehicle Capture and Plate Number Recognition Algorithm

3.2 Development of desktop application and database

The design of a desktop software application using Microsoft visual basic and the development of a database using Microsoft SQL Server Management Studio is used in this research, the interface for both the front and back end are shown in the appendix that includes the authentication, vehicle user interface, daily access report, and database report for all entries.

4. RESULTS AND DISCUSSION

4.1 Detection Result

The following shows the result obtained when actualizing the plate number detection after a vehicle image is captured during the entrance. The processes below describe the steps for character recognition by capturing the vehicle image and then converting it to greyscale to reduce the image quality, dictating the vehicle edge for contour detection. After the detection of contour, tesseract-ocr was used to detect the character on the vehicle for daily entrance report by the developed program.

i. The original image captured from the vehicle



Plate III: Original Image

ii. Converting the original image from the actual image grayscale to help reduce the noise



Plate IV: Grayscale Image

iii. The image shows edge detection for the vehicle



Plate V: Canny Edge Image

iv. Contour edge detection

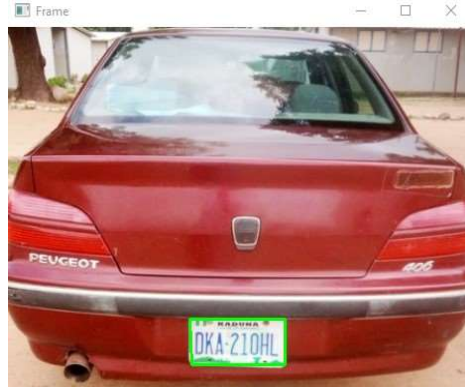


Plate VI: Contour Detection Image

v. Final result of extracted plate number

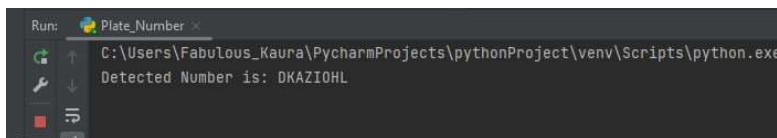


Plate VII: Result from Image

From the result in Plate VI, it is evident that the system recorded a good accuracy in plate number localization, while the result in Plate VII shows that the system can recognise 6 characters correctly out of the eight characters on the plate. This translates to 75%-character recognition accuracy

4.1.1 Analysis Table

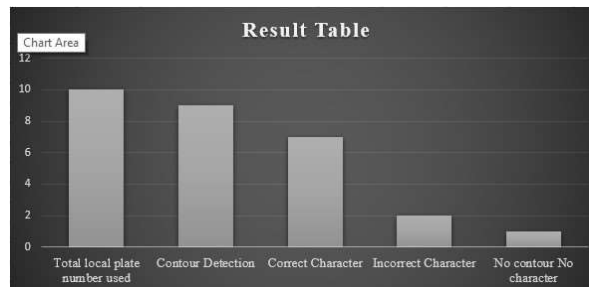


Plate VIII: Result Analysis

The above plate shows the local plate number sample tested on the algorithm. The total number of plates used is ten, only seven give correct characters, two of them incorrect characters, and only one does not have character or contour. Also, nine out of ten detect contour which gives us 90% contour detection and 70%-character recognition as the overall system performance.

4.2 Result Obtained from the Software Developed for Record Based

Below is the interface for the daily entrance report and the database entire report for daily entrance. The program is designed to accept information related to vehicle owners before gaining access to the daily entrance which captures the plate number, time, gate passing by, either staff, student, or visitor, and the security attending you during the entrance.

i. Daily entrance verification report



Figure 4.1: Daily Entrance Interface

ii. Database results obtained from daily report

	Plate_Numbr	Date_entmce	Time_entmce	ID_number	pass_gate	attdby	Pass_in_out	ID_Number2
1	seyhts	2021-07-29	8:42 PM	ghyjsdz	sdu	Label1	Entrance	hgjs
2	FGHJKH	2021-08-10	3:06 PM	qwety2376892	Main Gate	Label1	PASSING-IN	
3	FGHJKH	2021-08-10	3:06 PM	qwety2376892	Main Gate	Label1	PASSING-IN	
4	FGHJKH	2021-08-10	3:09 PM	qwety2376892	Main Gate	Label1		
5	JSJ332JKD	2021-08-10	3:27 PM	eshj8499	Main Gate	UMAR RIDWAN	PASSING-IN	
6	RSH-185TT	2021-09-29	1:34 PM	8475ABCD	North Gate	ABDULMALIK UMAR	PASSING-IN	
7	RSH-185TT	2021-09-29	1:50 PM	8475ABCD	Main Gate	ABDULMALIK UMAR	PASSING-IN	

Figure 4.2: Daily Entrance Report

5. CONCLUSION

The algorithm developed in this work for the extraction of Nigeria plate numbers proved quite effective and efficient. It had a 90% accuracy for license plate localization but a reading accuracy of 70%. MYSQL server database was used in this study which allows large storage of driver's particulars and vehicle data. With the output tied to a database, this system can be used as an automated process for managing a large number of vehicle user profiles as well as access control. Further work can be conducted to improve the accuracy of this system, using deep learning (tensor flow). Performance comparison for the captured plate number and the record stored in the database can be automated to make sure that each vehicle must be registered before accessing premises.

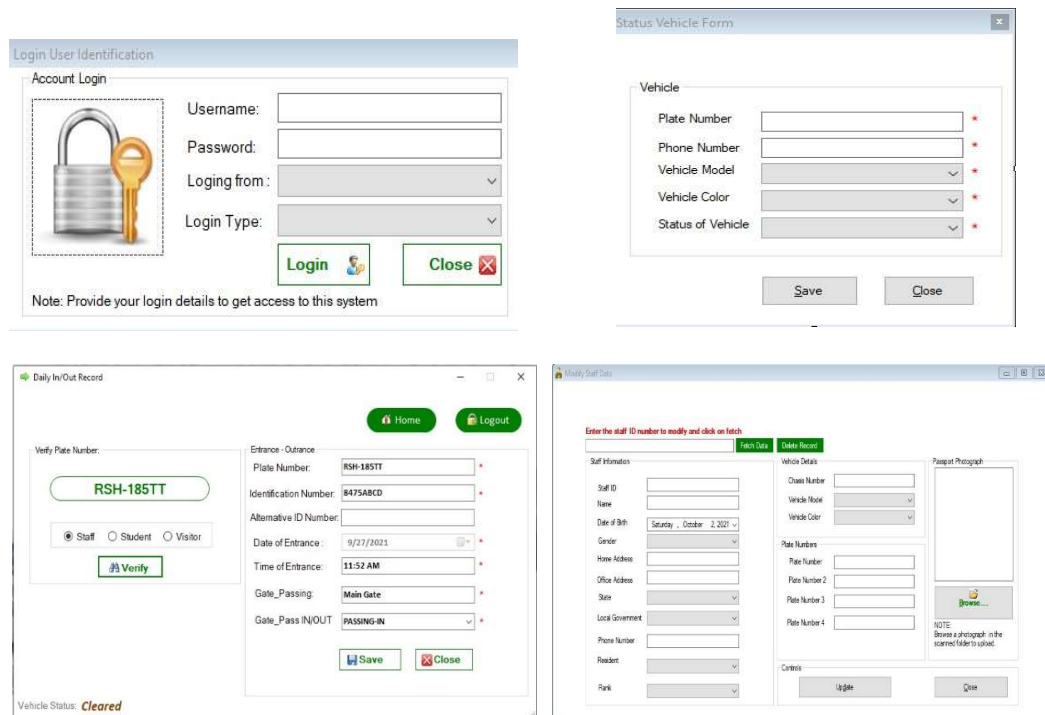
ACKNOWLEDGEMENT

I wish to express my profound gratitude to Almighty Allah; the Creator, Fashioner, without His mercies and favour I would certainly not exist. I would also like to express my profound gratitude to my supervisor Dr. Zaharuddeen Haruna for giving me the opportunity, for his patience, guidance, and careful instruction throughout the research, and a lot of others whom I cannot mention here whose hard work and dedication during this study assisted me with advice and encouragement. I have nothing to offer them but for Allah to reward them abundantly and enable them to succeed in all their endeavours. This acknowledgment would not be complete without expressing my sincere gratitude to my parents, brothers, and sisters for their love, patience, encouragement, and understanding which are the source of my motivation and inspiration throughout my work. May Almighty Allah in his abundance reward you all greatly, and then some more. I must in the final analysis not fail to appreciate all those special individuals, colleagues, family members, and friends whom I cannot mention here that have contributed in one way or another, in the course of my struggle and improved on my qualification. Thank you all.

REFERENCES

- AG, S. F. (2019). Development of portable automatic number plate recognition (ANPR) system on Raspberry Pi. *International Journal of Electrical and Computer Engineering*, 9(3), 1805.
- Akbarzadeh, O., Khosravi, M. R., & Alex, L. T. (2022). Design and Matlab simulation of Persian license plate recognition using neural network and image filtering for intelligent transportation systems. *ASP Transactions on Pattern Recognition and Intelligent Systems*, 2(1), 1-14.
- Alfath, A. N., Taningngsih, I., & Suharto, E. (2022). Designed an Inactive Archives Management Information System Using Visual Basic 2010 at The Department of Community Empowerment and Village West Java Province. *Journal of Applied Engineering and Technological Science (JAETS)*, 3(2), 105-115.
- BC, M. H., Pukhrabam, S., Gokakakar, R. R., & Kaveri, B. (2022). YOLOV4 object detection for license plate detection.
- Dome, S., & Sathe, A. P. (2021). *Optical character recognition using Tesseract and classification*. Paper presented at the 2021 International Conference on Emerging Smart Computing and Informatics (ESCI).
- Etomi, E. E., & Onyishi, D. U. (2021). Automated number plate recognition system. *Tropical Journal of Science and Technology*, 2(1), 38-48.
- Kaur, P., Kumar, Y., Ahmed, S., Alhumam, A., Singla, R., & Ijaz, M. F. (2022). Automatic license plate recognition system for vehicles using a CNN. *CMC-Computers, Materials & Continua*, 71(1), 35-50.
- Li, F., & Wang, L. (2022). Application of Case-based Teaching in Computer Vision with OpenCV and Python. *environment*, 5, 8.
- Mujawar, S. S., & Vadi, S. (2022). Raspberry Pi and OpenCV for license plate recognition in real-time.
- Padmasiri, H., Shashirangana, J., Meedeniya, D., Rana, O., & Perera, C. (2022). Automated license plate recognition for resource-constrained environments. *Sensors*, 22(4), 1434.
- Thakare, S., Kamble, A., Thengne, V., & Kamble, U. (2018). *Document Segmentation and Language Translation Using Tesseract-OCR*. Paper presented at the 2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS).
- Ugwu, E. M., Taylor, O. E., & Nwiabu, N. D. (2022). An Improved Visual Attention Model for Automated Vehicle License Plate Number Recognition Using Computer Vision. *European Journal of Artificial Intelligence and Machine Learning*, 1(3), 15-21.
- Wang, Y., Liu, J., He, X., & Wang, B. (2018). Design and realization of rock salt gas storage database management system based on SQL Server. *Petroleum*, 4(4), 466-472.

APPENDICES





The screenshot shows a 'Staff Registration' window with the title 'Staff's Data Form'. It is divided into three main sections: 'Staff Information', 'Vehicle Details', and 'Passport Photograph'.
Staff Information: Includes fields for Staff ID, Name, Date of Birth (set to Saturday, October 2, 2021), Gender, Home Address, Office Address, State, Local Government, Phone Number, Resident, and Rank.
Vehicle Details: Includes fields for Chassis Number, Vehicle Model, Vehicle Color, and a 'Plate Numbers' section with four input fields (Plate Number 1-4).
Passport Photograph: Features a 'Browse...' button and a 'NOTE: Browse a photograph in the scanned folder to upload.'
At the bottom, there are 'Save' and 'Close' buttons.