# Automatic License Plate Recognition System

ROSALINE A. EKE[1], STELLA I. ORAKWUE[2]
[1, 2] *Electrical/Electronic Engineering Department, University of Port Harcourt, Nigeria*

***Abstract-*** ***A fast, accurate and reliable vehicle plate recognition system is a useful tool that will help stem the rising tide of vehicle theft and traffic violations perpetuated in society and improve the overall safety of our roads.Automatic License Plate Recognition (ALPR) system is a real-time embedded system that recognizes the license plate of vehicles.The model proposed in this work for implementing the ALPR system uses an open-source C/C++ library called Open ALPR, based on OpenCV, Leptonica and Tesseract-OCR. The goal of the system is to use image processing to identify number plates on images of vehicles and also in a video stream. The hardware is implemented with Raspberry Pi zero, and a Pi camera module. The operating system which would run on the Raspberry pi is known as the Raspbian Jessie Pixel, a Linux distribution built for the Raspberry Pi which is more or less a combination of Raspberry and Debian. Results showed that the model has an accuracy of 97.6% for image extraction, 96% for character segmentation and 98.8% for character recognition. The overall systemperformance whichis a product of all unit's accuracy rates (Extraction of plate region, segmentation of characters and recognition of characters) shows an improvement from existing models.***

***Indexed Terms-*** ***Automatic License Plate Recognition, Character segmentation, Optical Character Recognition, OpeCV, Leptonica***

## I. INTRODUCTION

The increasing number of vehicles on our roads has made manual control and monitoring of traffic not only herculean but an enormous task for our law enforcement agents. Besides being straining, time-consuming and costly as a result of the needed manpower, manual traffic control and monitoring is also plagued with so much inaccuracy. Hence, the need for an automatic license vehicle plate recognition system to reduce the dependency on human labour. The license plate number uniquely distinguishes each vehicle [1].Fast and accurate recognition of license plates is one of the most important challenges in the field of license plate recognition systems.The vehicle plate is the unique identification (ID) of vehicles, which comes in different sizes, shapes, colors, and most times, even texture making it difficult to create a reference template, and this in turn requires complex procedural analysis and computation. Automatic License Plate Recognition (ALPR) system is a real-time embedded system which automatically recognizes the license plate of vehicles.It is a process of identifying information in license-plate images[2]. Automatic parking access using a real-time embedded system for Automatic License Plate Recognition (ALPR) and access control is common these days in urban commercial spaces, office buildings and other similar public spaces. Automatic license plate recognition (ALPR) methods and software are used in toll collection, traffic monitoring and other areas of the road transport industry. Automatic license plate recognition (ALPR) systems have many traffic applications, such as traffic control, traffic analysis, parking automation,electronic tollgate management, and speed control [1, 3, 4, 5 and 6] as well as border control and the tracking of stolen cars, [7]. ALPR system is an image processing system which is used to recognize vehicles by identifying the number plate.

License plate detection and recognition system consists of three important stages, namely plate detection, character segmentation and character recognition [2, 3, 4, 6]. The most critical and challenging of these stages is plate detection, as it plays an important role in the other stages. Wrongly detected plates, negatively impacts the other stages and thus give an entirely wrong output. According to [4]these three processes may have problems when plate size variations, viewing angle variations, low contrast plates, high-speed vehicles, time-consuming recognition algorithms and dirty/filthy vehicle plates

subsists. Automatic license-plate recognition is executed first by extracting the license plate, followed by character segmentation on the extracted license plate to separate characters. And finally, character recognition is performed, in which the isolated characters are detected to identify the letters and numbers on the license plate.

## II. RELATED WORK

The work in [8] proposed a novel deep learning-based automatic license plate recognition model based on bounding rectangle-based (BR) segmentation and convolutional neural network-based (CNN) recognition called BR-CNN model to enhance ALPR accuracy. The proposed BRCNN model works on three primary stages to identify and recognize license plate numbers, namely license plate detection, bounding rectangle segmentation, and recognition of license plate characters with CNN. The work proposed by the authors in [6] presented an end-to-end method for efficiently detecting and recognizing number plates where the vehicles are first detected using a single-shot detector- (SSD-) based deep learning model in the video frames and the input images, the location of the plate is then identified using the proposed architecture based on convolutional networks and finally, with a deep convolutional network and long short-term memory (LSTM), the characters related to the plate are recognized. Their proposed method is capable of processing 30 frames per second without losing any data and is shown to be viable for real-time applications based on its response time. An algorithm to detect license plate region and edge processing both vertically and horizontally was proposed by [4] to improve the performance of the systems for high-speed applications. The proposed method which is based on the bounding box method detected and recognized the original images and filtered them both vertically and horizontally. They gave the overall accuracy rate of success recognition as 93% in sunlight conditions, 72% in cloudy, 71% in shaded weather conditions.

Fuzzy ARTMAP neural network was used by [1] for recognizing vehicle license plate numbers. The Fuzzy ARTMAP neural network is a new architecture of the neural network family with capability to learn incrementally unlike conventional backpropagation (BP) algorithm. The image of the vehicles are captured by a monochrome charged-coupled device (CCD) camera and processed digitally through a frame grabber. A program is written using Visual C++ and the Matrox Image Library to process the captured image of the vehicle. The image processing technique includes noise reduction and image enhancement, which then finds the region of interest which is the license plate of the vehicle. Once the ROI is identified, each character or digit in the license plate is isolated and sequentially recognized by the trained neural network. The techniques designed are able to detect and recognize normal vehicle license plates in Malaysia. It however has some drawbacks, from failure of image acquisition to locate the position of the license plate, character overlapping, misclassification of characters and inability to deal with license plates with rare font styles.According to [5] two deep learning techniques to improve license plate visibility towards more accurate license plate recognition. A one-stage object detector known as YOLO (You Only Look Once) was implemented for locating license plates under challenging situations. (YOLO) is a framework for identifying objects within an image in a single pass using a specialized convolutional neural network (CNN) architecture. Super-resolution generative adversarial networks were considered for image up-scaling and reconstruction to improve the clarity of low-quality input. It focused on training these systems on datasheets which include difficult-to-detect license plates, enabling better performance in unfavourable conditions and environments.

The authors in [9] designed and implemented an efficient License Plate Recognition (LPR) method for Indian license Plates. They presented a robust method of license plate location, segmentation and reorganization of the characters present in the located plate. Images of different vehicles are acquired manually and converted in to gray-scale images. Wiener2 filter was then used to remove noise present in the plates. The segmentation of gray scale image generated by finding edges using Sobel filter for smoothing image was used to reduce the number of connected component and then label was used to calculate the connected component. Finally, single character is detected. The results showed an achieved

accuracy of 98% by optimizing various parameters with higher recognition rate than the traditional methods. According to [2] a modified license-plate recognition pipeline, with license-plate template matching replacing character segmentation and recognition can be used for identifying vehicles with similar license plates. Edge detection and license plate ratio was used to extract license-plates. The extracted license-plate templates are then compared for license-plate matching. The results showed that the method performed well in different circumstances, and is computationally cost-effective, also that license-plate template matching is a reliable method of identifying similar vehicles, and has a lower computational cost when compared with character segmentation and recognition.

## III. METHODOLOGY

The design of the Automatic License Plate Recognition system (ALPR), using the Raspberry Pi Single board computer SBC is divided into two parts: the first part locates the license plate, and the second part reads the characters from the extracted plate. The Raspberry Pi and the Pi camera module serve as the input/output for our system. The camera module gets the images or video stream and stores it in a specified folder. The OpenALPR engine, when called through the command line interface of the Raspberry Pi then locates the license plate in the image and returns the characters of the license plate to the command line interface (CLI).

### *A. Software Specifications*

The software used to achieve the ALPR on the Raspberry Pi are briefly discussed:

1. The Raspbian Jessie Pixel: This is a Linux distribution which is more of a combination of Debian and Raspberry. It is built specifically to run on the Raspberry Pi. It has a desktop view and also a command line interface.
2. The SD Card Formatter: This is a Windows-based software which is used to format the microSD card in which the Raspbian image is going to be mounted on.
3. Etcher: This is the software used to write the Raspbian image to the Raspberry Pi microSD card
4. Putty was used to connect to the command line interface (CLI) of the raspberry pi through a protocol known as SSH (Secure Shell). The raspberry pi and the computer that request access to the raspberry are required to be connected on the same Local Area Network (LAN). Also, the username and password of the raspberry pi to be accessed is needed when trying to login to the raspberry through SSH, using Putty. The VNC Viewer was used to provide a server for viewing the X windows of the raspberry pi. In order to get a license plate recognition engine working on the Raspberry Pi, the OpenALPR library, an open-source library, was employed. This library cannot be used on its own as it requires several other libraries and modules (dependencies) in other to serve its purpose. The dependencies used alongside the OpenALPR library to achieve license plate recognition on the raspberry pi are OpenCV, Leptonica and Tesseract

### *B. Hardware Specifications*

The hardware components used in this work include Raspberry Pi Zero, a Wi-Fi router, Raspberry Pi Camera Module, Wi-Fi Dongle, USB Mouse and keyboard, an HDMI cable, a power adaptor and a micro USB hub.The raspberry pi zero has a RAM of 512MB, a micro HDMI port, a micro USB port, and a port to attach the camera module. It also has a microSD card slot. The Wi-Fi router is needed to set up the raspberry pi zero for the first time if the pi is being run in headless mode (i.e. without a monitor, mouse and keyboard). Also, the router provides internet access to raspberry for updating and upgrading its firmware to the latest release. The raspberry pi camera module is used as the input device for getting the video stream in order to perform license plate recognition. In order to use the camera module, the default configuration settings of the raspberry pi was changed, as the camera was set to be disabled by default. This can be re-enabled using the CLI or by using the desktop. The USB Wi-Fi dongle was attaching to the raspberry pi in order to give it internet connectivity and also to be able to wirelessly logon to it via SSH or VNC. The USB mouse was used to navigate around the desktop of the raspberry pi, when the raspberry pi is connected to a monitor. (i.e. when the raspberry pi is not running in headless mode). The USB keyboard was used as a

text input to the raspberry pi when the raspberry pi is connected to a monitor. The mini HDMI cable was used to connect the raspberry pi zero to a monitor in order to view the desktop of the operating system running on the raspberry pi. A power adaptor with a specification of 5V, 1.5A was used to provide power to the raspberry pi. The USB port can also serve as the power source for the raspberry pi, but the power it provides is not sufficient for the Pi. Micro USB Hub was used to enable more than one USB device to be attached to the raspberry pi Zero, since it has only one micro USB port.

*C. Software Implementation of License Plate Recognition on the Raspberry Pi*

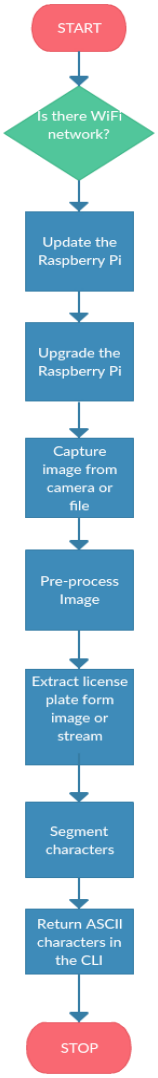


Figure 1: Flowchart of License Plate Recognition System Using Raspberry Pi

This system is divided into two parts: the first part locates the license plate, and the second part reads the characters from the extracted plate. The Raspberry Pi and the Pi camera module serves as the input/output for our system. The camera module gets the images or video stream and stores it in a specified folder. The OpenALPR engine, when called through the command line interface of the Raspberry Pi then locates the license plate in the image and returns the characters of the license plate to the command line interface (CLI). This process is outlined in the following flowchart shown in figure 1.

Automatic number plate recognition has three major parts:

1. Vehicle number plate extraction
2. Character segmentation and
3. Optical Character Recognition (OCR).

- Vehicle number plate extraction

Number plate extraction is that stage where the vehicle number plate is detected. The detected number plate is pre-processed to remove the noise and then the result is passed to the segmentation part to segment the individual characters from the extracted number plate. The segmented characters are normalized and passed to an OCR algorithm. At last, the optical character information will be converted into encoded text. The characters are recognized using Template matching. The final output must be in the form of string of characters. The OpenALPR library is used alongside OpenCV, Leptonica, and Tesseract to achieve the automated license plate recognition. OpenALPR operates as a pipeline. The input is an image, various processing occurs in stages, and the output is the possible plate numbers in the image. The pipeline stages occur in the following order:

1. Detection: The detection is done once for each input image. It uses the LBP algorithm (generally used for face detection) to find possible license plate regions (x,y, width, height). Each of these regions is sent to the later pipeline phases for further processing. The detection phase is usually the most process-intensive phase. It can be GPU accelerated to improve performance.
2. Binarization: This phase (and all subsequent phases) occurs multiple times for each possible license plate region. The binarization phase

creates multiple binary images for each plate region. The reason multiple binary images are used is to give us the best possible chance of finding all the characters. A single binarized image may miss characters if the image is too dark or too light for example. Binarization uses the Wolf-Jolien method as well as the Sauovola method with various parameters. Each of the binary images is processed in subsequent phases.

3. Character Analysis: Character analysis attempts to find character-sized regions in the plate region. It does this by first finding all connected blobs in the license plate region. Then it looks for blobs that are roughly the width and height of a license plate character and have tops/bottoms that are in a straight line with other blobs of similar width/height. This analysis is done multiple times in the region. It starts by looking for sall characters, then gradually looks for larger characters. If nothing is found in the region, then the region is thrown out and no further processing takes place. If it finds some potential characters, then the character region is saved and further processing takes place.
4. Plate Edges: The next phase is to find the edges of the license plate. Keep in mind that the detection phase is only responsible for identifying a possible region where a license plate may exist. It often is going to provide a region that is a little larger or smaller than the actual plate. The plate edges try to find the precise top/bottom/left/right edges of the license plate. The first step is to find all of the rough lines for the license plate region. platelines. cpp processes the plate image and computes a list of horizontal and vertical lines. platecorners uses this list as well as the character height (computed in Character Analysis) to find the likeliest plate line edges. It uses a number of configurable weights to determine which edge makes the most sense. It will try using a default edge (based on the ideal width/height of the plate) to see if that makes a good match.
5. Deskew: Given the plate edges, the deskew stage remaps the plate region to a standard size and orientation. Ideally, this will give us a correctly oriented plate image (no rotation or skew).

- Character Segmentation:

The character segmentation phase tries to isolate all the characters that make up the plate image. It uses a vertical histogram to find gaps in the plate characters. This phase also cleans up the character boxes by removing small, disconnected speckles and disqualifying character regions that are not tall enough. It also tries to remove "edge" regions so that the edge of the license plate doesn't inappropriately get classified as a '1' or an 'I'

- Optical Character Recognition (OCR)

The OCR phase analyzes each character independently. For each character image, it computes all possible characters and their confidences.

- Post-Processing

Given a list of all possible OCR characters and confidences, post-processing determines the best possible plate letter combinations. It is organized as a top N list. Post-processing disqualifies all characters below a particular threshold. It also has a “soft” threshold -- characters that are below this threshold will still be added to the possible list, but they also add a possible blank character -- since it's possible that the low-confidence character is not really part of the plate. The post-processing also handles region validation if requested.

### *D. Training of License Plate Detector for Nigerian License Plate Numbers*

A repository containing all the python scripts that help train a license plate detector for any region was downloaded from GitHub. The trained region detector was then used in OpenALPR. The license plate region detector uses the Local Binary Pattern (LBP) algorithm. In order to train the detector, many positive and negative images are needed. This repository already contains a collection of negative images. Positive images of Nigerian plate numbers have been added. To get started, many cropped plate images containing positive license plate matches were collected. After which the training script is configured to use the correct dimensions. The prep.py script is edited to change the WIDTH, HEIGHT, and COUNTRY variables to match the country that is being trained, in this case, Nigeria. The width and height should be proportional to the plate size

(slightly larger is OK). A total pixel area of around 650 worked best. Also, adjust the path to your OpenCV libraries, if that needs to be changed. The out/cascade.xml file is copied out to the OpenALPR runtime directory (runtime_data/region/[countrycode].xml). The region (Nigeria) can now be used for plate detection.

## IV. RESULTS AND DISCUSSION

The proposed system was tested to measure its degree of accuracy. The images for the input to the system are colored images with the size of 1200x1600. The test images were taken under various illumination conditions.

The various conditions by which the images were tested include:

i. Images with black borders around the license plate
ii. Images without black borders around the license plate
iii. License plates overexposed or underexposed to light
iv. License plates with proper lighting conditions
v. Images with skewed license plates

The results obtained are as follows:

A = Images with black borders around the license plate

B = Images without black borders around license plate

C = License plates over exposed or underexposed to light

D = License plates with proper lighting conditions

E = Images with skewed license plates

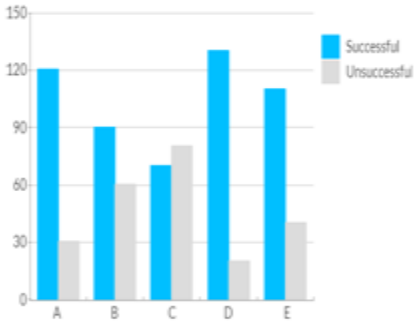


Figure 2: Chart of the Module Localization of the License Plate

The figure 2 represents the results of how well the module could perform localization of the license plate from 150 images containing license plates under different conditions. Figure 2 shows the success of the localization of license plate from the images in percentage.

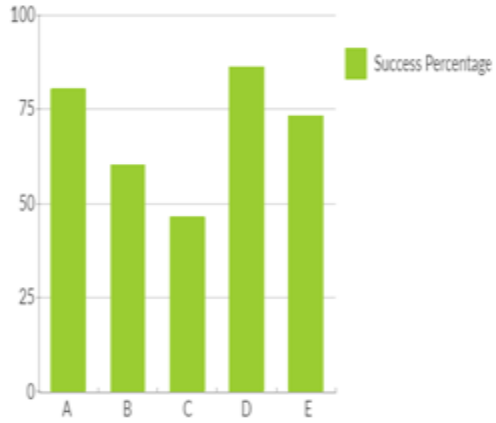


Figure 3: Success Percentage of the Module Localization of License Plates

From the license plates that were successfully localized from the image, further tests were carried out to segment the localized license plate. The segmentation step has been carried out to determine the actual characters of the license plate region which was localized by the module as shown in figure 3

The representation of the success of the segmentation of the localized license plates from the images in percentage is shown in figure 4.

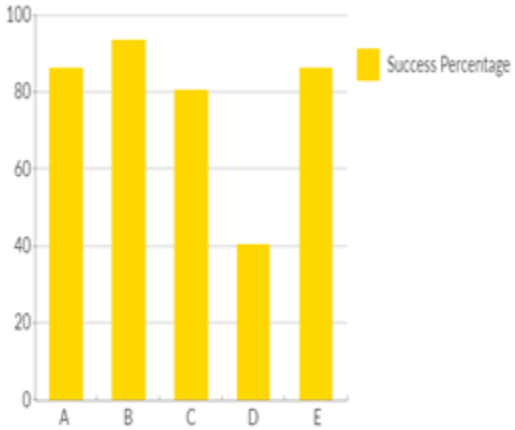


Figure 4: Success Percentage of Segmentation of Localized License Plates

It is shown that accuracy for the extraction of plate region is 97.6%, 96% for the segmentation of the characters and 98.8% is the percentage of accuracy of the recognition unit. The overall system performance can be defined as the product of all unit's accuracy rates as shown in table 1 (Extraction of plate region, segmentation of characters and recognition of characters).

For vehicle detection in, a database of pictures with their locations and class of objects were manually determined.

Table 1: Results of the Test

| S/No. | Units of ALPR System | Number of Accuracy | Percentage of Accuracy |
|---|---|---|---|
| 1 | Extraction of Plate Region | 146/150 | 97.33% |
| 2 | Segmentation | 143/150 | 95.33% |
| 3 | Recognition of Characters | 148/150 | 98.67% |

## CONCLUSION

The automatic license plate recognition system designed in this work offers remote monitoring of the video stream of the vehicles with the license plates being recognized. The stream can be viewed if logged on to the same wireless network as the Raspberry Pi and if the user has access to the username and the password of the Raspberry Pi. The system also offers recognition of Nigerian license plate numbers as a unique cascade classifier is trained for this purpose. The designed system is able to detect license plates both in a still image and in a video stream provided by the raspberry pi camera module attached to the Raspberry Pi. The license plate recognition is command line based i.e. after all the libraries are installed and the sources are built, the software is called using the command line interface.

## REFERENCES

[1] Siah, Y. K Haur, T. Y. Khalid, M. and Ahmad, T. (2012). Vehicle Licence Plate Recognition by Fuzzy Artmap Neural Network, obtained from www.researchgate.com, on February, 16 20203.

[2] Manana, M. Tu, C. and Owolawi, P. A.(2021). Edge-Based Licence-Plate Template Matching for Identifying Similar Vehicles, Vehicles 2021, 3, 646–660. https://doi.org/10.3390/vehicles3040039.

[3] Wu, B.-F., Lin S.-P. and Chiu, C.-C.(2007). Extracting characters from real vehicle licence plates out-of-doors, IET Computer Vision · April 2007. DOI: 10.1049/iet-cvi:20050132. Source: IEEE Xplore.

[4] Baranidharan V., Varadharajan, K. Sudhakar K., and Lokesh, N.V. (2018). Bounding Box Method Based Accurate Vehicle Number Detection and Recognition for High Speed Applications, ICTACT JOURNAL ON IMAGE AND VIDEO PROCESSING, NOVEMBER 2018, VOLUME: 09, ISSUE: 02, ISSN: 0976-9102 (ONLINE), DOI: 10.21917/ijivp.2018.0264.

[5] Boby, A. and Brown, D. (2022). Improving Licence Plate Detection using Generative Adversarial Networks. Pattern Recognition and Image Analysis · April 2022 DOI: 10.1007/978-3-031-04881-4_47.

[6] Pirgazi, J. Kallehbasti , M. M. P. andSorkhi, A. G.(2022). An End-to-End Deep Learning Approach for Plate Recognition in Intelligent Transportation Systems. Wireless Communications and Mobile Computing Volume 2022, Article ID 3364921, 13 pages. https://doi.org/10.1155/2022/3364921.

[7] Tiwari, B. Sharma, A. Singh, M. G. and Rathi, B. (2022). Automatic Vehicle Number Plate Recognition System using Matlab. IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 11, Issue 4, Ver. II (Jul.-Aug .2016), PP 10-16. www.iosrjournals.org.

[8] Pattanaik, A. andBalabantaray, R. C. (2022). Licence Plate Recognition System for Intelligence Transportation Using BR-CNN. Chapter · March 2022 DOI: 10.1007/978-981-16-8403-6_60

[9] Lazrus, A. Choubey, S. and Sinha G.R. (2011). An Efficient Method of Vehicle Number Plate Detection and Recognition. International

Journal of Machine Intelligence. ISSN: 0975–2927 & E-ISSN: 0975–9166, Volume 3, Issue 3, 2011, pp-134-137. Available online at http://www.bioinfo.in/contents.php?id=31