Nigerian Research Journal of Engineering and Environmental Sciences
Journal homepage: www.rjees.com

Original Research Article

Nigerian License Plate Number Recognition System

***[1]Obayuwana, A., [2]Ero, O.E. and [2]Idemudia, E.**
[1]Department of Computer Engineering, University of Benin, Benin City, Nigeria.
[2]Department of Electrical/Electronics Engineering, University of Benin, Benin City, Nigeria.
*augustine.obayuwana@uniben.edu; osazee.ero@uniben.edu; efosaidemudia@outlook.com

ARTICLE INFORMATION

Article history:
Received 25 February, 2019
Revised 25 April, 2019
Accepted 25 April, 2019
Available online 30 June, 2019

Keywords:
Image detection
Image processing
Character recognition
Artificial neural network
Character segmentation

ABSTRACT

Number plate recognition system usually varies for different countries. Thus, getting a single algorithm/method to accurately detect and recognize all number plates may be rather difficult. This paper presents a methodology to detect, localize, and recognize the presence of a Nigerian vehicle number plate in a picture frame. The procedures used in the image preprocessing stage involved image resizing, converting the RGB image to gray scale image, dilation and erosion, noise reduction, image sharpening, median filtering, edge detection, thresholding and contour analysis. Character segmentation was carried out with the aid of connected component analysis while blob extraction technique was used to extract the characters from the plate using some heuristics. Character recognition was achieved using Artificial neural network with back-propagation learning algorithm. A security access control system was developed using Microsoft Visual C# and the .NET Framework 4.0 to test the algorithm. In the application, the recognized plate numbers were compared with a list of known license plates in a database. If the recognized licensed plate is on the list, access is granted to vehicle otherwise access is denied. The algorithm was able to carry out number plate detection and localization under various environmental and lighting conditions with an average processing time of 0.872 s.

1. INTRODUCTION

Computer vision is a field that includes methods for acquiring, processing, analyzing, and understanding images, and in general, high-dimensional data from the real world in order to produce numerical or symbolic information, in the forms of decisions (Shapiro and Stockman, 2001).

License plate recognition (LPR) is an image-processing technology used to identify vehicles by their licensed plates (Kwaśnicka and Wawrzyniak, 2002). LPR systems aim to detect car plates from input image and read the license plate number. License plate detection and recognition has a number of different applications relevant to transportation systems, such as detection of stolen vehicles, parking assistance, access control, tolling, border control, security control in restricted area, and statistical research (NI, 2007).

A license plate recognition system consists of three main parts which are license plate detection, character segmentation and character recognition. The first part is the most crucial part due to the diversity of parameters involved in car images (Kwaśnicka and Wawrzyniak, 2002). Factors which may negatively affect the result of license plate identification from images include; weather conditions, lighting conditions, mechanical plate damage, vehicle movement, license plate placement in picture. Generally, the license plate detection section analyzes the captured image to find plate location or alphanumeric characters. The second and third parts, segments and recognizes the characters by using optical character recognition (OCR) to read the segmented characters that appear on the plate. There are numerous LPR systems available today. These systems are based on different methodologies, and many algorithms have been developed for the purpose of license plate detection, segmentation and recognition. Some of these algorithms have a good accuracy with more complexity than others while some are computationally intensive. However, most of the procedures discussed in literature use more than one approach, hence, it is not possible to do category wise discussion. Bagade et al. (2010) proposed car plate recognition using the template matching method the LPR system consisted of 4 modules, viz; image acquisition, licensed plate extraction, segmentation and recognition of individual characters. However, the template matching method did not need the 'segmentation' process of input image. Another efficient method of vehicle license plate recognition which is based on sliding concentric windows and artificial neural network is proposed by Kaushik (2012) which employed segmentation technique named as sliding concentric windows. The method helped in analyzing road images which often contained vehicles and extracted license plates by finding the vertical and horizontal edges in vehicle region. On this basis, an adaptive image segmentation technique was used for detecting candidate region and verification of color by using HSI color model to verify green/yellow LP using hue and white LP using intensity. Kumar and Subin (2012) proposed the Support Vector Machine (SVM)-based license plate recognition system, which is a supervised learning technique which takes statistical learning theory as its theoretical foundation, and the structural risk minimization as its optimal object to realize the best generalization.

From literature survey it has been observed that there are certain limitations about these algorithms such as poor image resolution, less accuracy, poor lighting and low contrast, higher computational cost, lack of standards of the plate of the vehicles and also, improperly segmented characters which could result in misrecognized characters (Patel et al, 2013). Besides, license plate definition may vary from one country to another due to the different shapes, sizes and colors used. Hence, there is the need to find an alternate definition of number plate that will be comprehensible for Nigerian vehicle plate recognition system.

2. METHODOLOGY

License plate recognition systems are usually divided into four different sections – image acquisition, license plate detection, segmentation and character recognition. The image acquisition section, though mostly overlooked, plays a very important role in the whole process of license plate detection and computer vision in general. After an image has been obtained, various methods of processing can be applied to the image to perform the many different vision tasks.

This section presents a system algorithm that was developed to detect and recognize the Nigerian vehicle license plate and a computer application for access control was also developed to test the algorithm. The system algorithm is basically divided into three stages as depicted in Figure 1.

A. Obayuwana et al. / Nigerian Research Journal of Engineering and Environmental Sciences
4(1) 2019 pp. 431-446

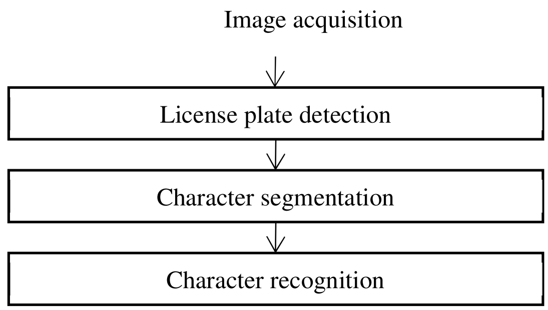


Figure 1: The flow of the license plate recognition system

2.1. License Plate Detection Stage

This first stage of the image processing which comes after the acquisition of the car digital image is the most crucial part due to the diversity of parameters involved in car images (Kwaśnicka and Wawrzyniak, 2002). Factors which may negatively affect the result of license plate identification from images include; weather conditions, lighting conditions, mechanical plate damage, vehicle movement, license plate placement and other captions in the capture image (Roberts and Casanova, 2012). Generally, the license plate detection stage analyzes the captured image to find plate location or region. Many algorithms have been developed for this purpose. Some algorithms are based on finding the license plates by using image features such as shape, color, or height-to-width ratio (Olivier, 2015).

The digital images can be obtained from different sources (such as digital cameras, scanners, etc.) which may introduce some noise interference into the images. These noises, which are usually a variation of brightness and color information in images, when passed through edge detection algorithms are detected as edges and thus affect the detection of the required edges of the license plate (Muhammad and Zahra, 2011). For noise reduction, each image was passed through a series of image processing filters. Figure 2 shows a flow chart of all the various preprocessing steps the image went through.

The sizes of the acquired images varied from 734 x 552 pixels to 2592 x 1944 pixels. These variations in image size means that the total number of pixels to be processed by the algorithms will vary between images and thus the processing rate (i.e. the total time taken to process all the pixels in each image) will be different for each image. In order to ensure that the total number of pixels to be processed by the algorithm was constant, all the images were resized to a dimension of 640 x 480 pixels. This reduction in size also ensured that the amount of time taken by the algorithms to process an image was reduced.

All standard cameras capture images in the RGB (Red Green Blue) color mode which is unsuitable for most image processing filters, as most algorithms require the images to be in grayscale (Kanan and Cottrel, 2012). Thus, the images needed to be converted to grayscale. There are several algorithms that can be used to convert an image from RGB to grayscale. In many cases, the images are transformed from RGB into YUV color model, where the Y is the luminance part (gray-scale part) and U, V are the chrominance parts (Öztürk and Özen, 2012). To get the Y part from the RGB model, Equation 1 was used.

$$Y=0.2126R+0.7152G+0.0722B \tag{1}$$

The grayscale image was then passed through a series of image processing filters, the first of which is a morphological dilation filter. The image was then taken through a morphological erosion filter. Both morphological filters were carried out for three iterations with a 3x3 structuring element (SE). More noises were removed from the image by down-sampling the image using the Gaussian Pyramid filter and then up-sampling the resultant image with the Laplacian Pyramid filter. Afterwards, the image was sharpened by applying a convolution filter with a kernel Mx. This made the edge features of the image pronounce (Kolour and Shahbahrami, 2011).

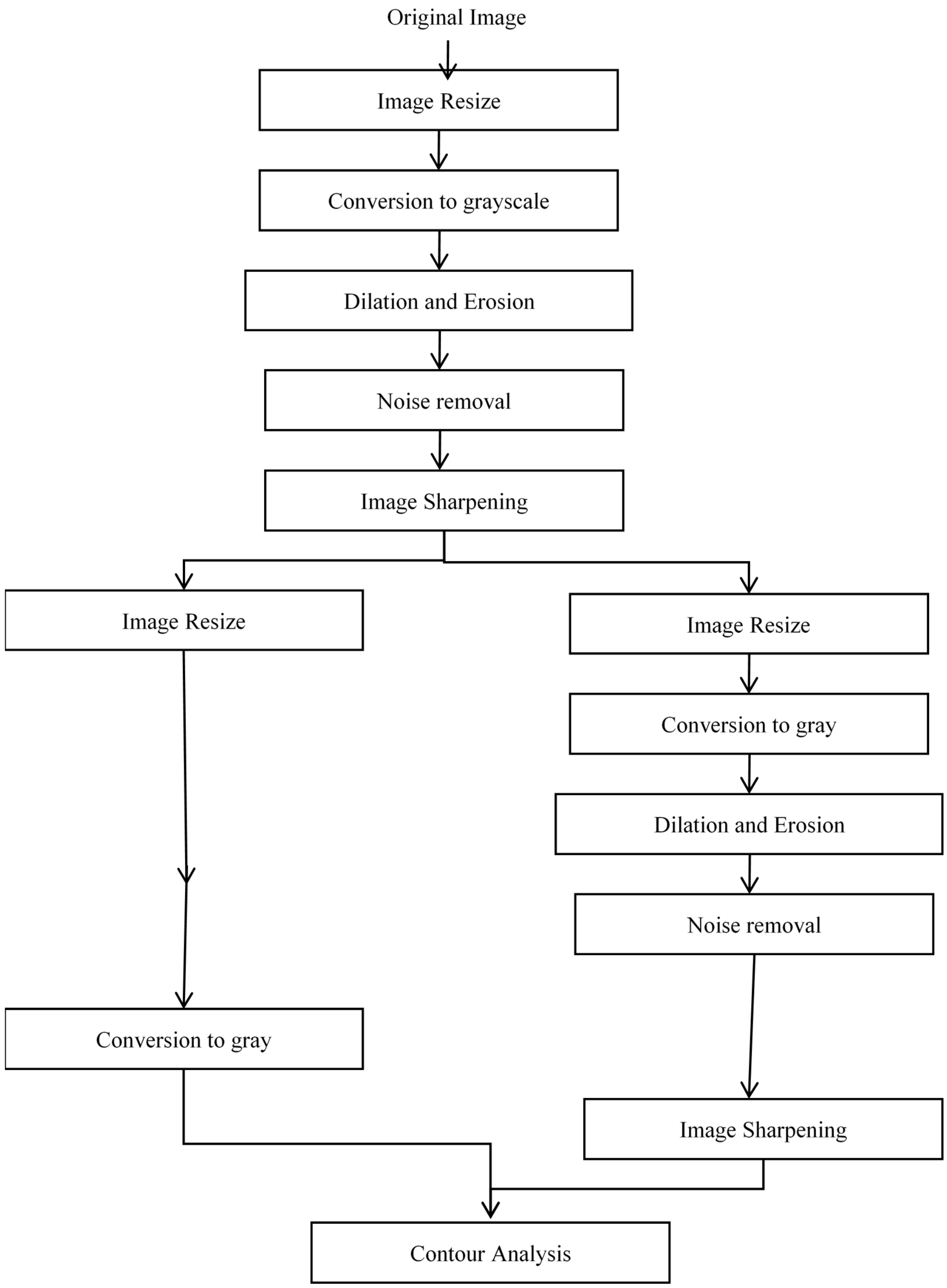


Figure 2: Image processing stages

As may have been noticed from Figure 2, the image processing steps were further broken down into two independent paths. This was necessary because, it was observed that some images needed a more processing than others before the edges would be pronounced enough for detection. The image was first taken through the first path (i.e. the stage with less processing) after which contour analysis was carried out on it with an attempt to find possible number plate regions. If no possible number plate regions were detected, the image was then taken through the second path for further processing after which contour analysis was carried out for possible number plate regions detection.

Thus, a heuristic was developed to test each rectangular region to verify if they are a license plate. The heuristic was based on the fact that license plates usually have dark-colored characters on a white background and that the license number of the number plate is usually centralized vertically on the plate (Pandya and Singh 2011). The heuristics was applied to each rectangular region and any region found to contain less than three characters were rejected. Any region containing more than three characters was passed on to the character segmentation stage. If no license plate region was found at the end of this first image processing stage, the image was then taken through the second stage of image processing.

Since the Nigerian number plate is a fixed rectangular metallic plate with characteristic dark colored characters on a light background, a morphological based edge detection technique was used (Pandya and Singh, 2011).

However, besides the noise, there may be other structures or objects in the image which have sharp and pronounced edges. In some cases, these edges were rectangular in shape and even have the same aspect ratio with a license plate. This may lead to the false detection of these edges as license plates. Noise reduction filters cannot be used to eliminate or reduce the distraction caused by these objects. In this work, what was done was to use the fact that license plates usually have dark-colored characters on a white background to eliminate rectangular regions that are not likely to be license plates. After a license plate region has been found, the coordinates were taken and mapped on the original image from which the license plate is extracted and passed on to the segmentation stage.

2.2. Image Segmentation Stage

The main purpose of the segmentation process is to divide or split the license plate image into several parts so that the required part which contains the characters of the license plate number can be extracted. Before character segmentation was carried out on the number plate image, the image was first passed through a series of image pre-processing steps to ensure successful segmentation. Depending on the angle of the camera relative to the car, the extracted number plate can be skewed. Thus, the next stage of the image processing was de-skewing of the image. The de-skewed image was color-inverted and then binarized. The binarization of the image was done using thresholding. Due to factors such as poor lighting conditions, illumination, plate deformation etc., it was difficult to get a perfect threshold value that can be used to binarize all the images for successful character segmentation. The simple image statistic (SIS) thresholding method was used for this purpose (Cika et al., 2011). This approach uses a threshold value that is based on the image statistics. Thus, different threshold values are used for different images depending on the image statistics. In this approach, the process started from a threshold value of 10 to a threshold value of 240, with an increment of 10. At each iteration, the license plate characters were extracted from the images. The threshold value at which the highest numbers of characters were extracted was chosen as the threshold value for the binarization of that image.

In the second stage of image processing, the sharpened image was smoothened with a median filter. The Sobel edge operator was then applied on the image followed by morphological dilation. The resultant image then went through further noise reduction by successive down-sampling and up-sampling using the Gaussian and Laplacian Pyramid filters respectively. Adaptive thresholding was used to binarize the image. The image was further smoothened with median filter before contour analysis with polygon approximation was carried out. Each rectangular region found was then filtered just like in the first stage and the developed heuristics was applied. Then the connected components analysis and blobs extraction technique was applied to achieve character segmentation (Kwaśnicka and Wawrzyniak, 2002). Connected component analysis was used to find and label connected regions in the license plate image and blob extraction technique was then used to filter out the license plate characters (Halgaonkar, 2011) based on the following conditions:

- A number plate character should have a width between 15 and 60
- A number plate character should have a minimum height of 0.25 times the height of the number plate
- The top edge of the character should lie between 0.15 to 0.5 times the height of the number plate
- The bottom of the character should lie between 0.9 to 0.5 times the height of the number plate.

When the above conditions were applied, the license plate number characters were extracted and passed on to the character recognition stage.

2.3. Character Recognition Stage

In the recognition stage, an optical character recognition technique was used to read out the segmented characters. Character recognition was achieved using artificial neural network. A three-layer feed forward supervised neural network was developed. This network uses the back-propagation learning algorithm and it is trained to do character recognition (Gershenson, 2003) In the training process, different characters (alphabetic and numeric) were used as the inputs into the network. These characters were normalized to a dimension of 30 x 30.

After the training had been completed, the network was then ready for character recognition. The segmented license plate characters were then normalized to the same dimension as the training character set and passed on to the network for recognition (Sharma et al., 2012). The network then compared each segmented character with the character sets used to train the network and returned the character (from the training set) that closely matches the segmented character. It was observed that some alphabetic characters were wrongly recognized as numeric characters and vice versa. Examples of such cases include ‘A’ being recognized as ‘4’, ‘B’ as ‘8’, ‘D’ as ‘0’, ‘O’ as ‘0’, ‘5’ as ‘S’, ‘6’ as ‘G’, etc. To resolve this issue, the syntactic analysis of the Nigerian vehicle license plate was carried out. Currently, the Nigerian vehicle license plate is of two types (i.e. the old and the new number plates) (Owamoyo 2013). After syntactic analysis was carried out on the plates, the following assumptions where made. The Nigerian number plate consists of either 8 or 7 characters. The plate number is divided into three sections of alphabetic and numeric characters. The arrangement of these sections on the plate depends on the type of the plate. The syntactic arrangement of the new plate is reverse that of the old plate. For the old plates, the syntactic arrangement is: XXYYY-XXX; while for the new characters, the syntactic arrangement is: XXX-YYYXX. Where the Xs are alphabetic characters and the Ys numeric characters.

Based on these assumptions, two separate neural networks were trained. One was trained to recognize only numeric characters while the other was trained to recognize only alphabetic characters. A heuristic was designed to detect if the number plate is an old number plate or a new one. If the system detects the number plate as an old number plate, it passed the first two and the last three sets of characters to the neural network that has been trained to recognize only alphabetic characters. The remaining characters in the middle (i.e. the numeric characters) were passed to the neural network trained to recognize only numeric characters. For a new number plate, the first three and the last two characters were passed to the alphabetic neural network while the remaining middle characters were passed to the numeric neural network. Thus, with this heuristic, wrong detection of alphabetic characters as numbers and vice versa was prevented, which improved the recognition rate. After recognition is done, the recognized plate number is used for further processes such as detection of stolen vehicles, parking assistance, access control, and security control in restricted area, tolling, and border control.

2.4. Development of Security Access Control Application

In this work, a simple security application was created to show how the license plate recognition system is used for access control into a restricted area. The application consists mainly of two parts namely; the database engine and the application interface. One of the applications of license plate recognition systems is in security access control. As part of the objectives of this work, a simple computer application was created to show how the license plate recognition system can be used for access control. In the application, the recognized plate numbers were compared with a list of known license plates in a database. If the recognized licensed plate is on the list, access is granted to vehicle, otherwise access is denied. In order to test how the proposed algorithm will perform under different image qualities, three different cameras with resolutions; 1.3 MP, 2 MP and 5 MP were used to acquire the Nigerian number plate images.

2.4.1. Database design

The database used for the application was the Microsoft Access Database. Figure 3 shows the design view of the database.

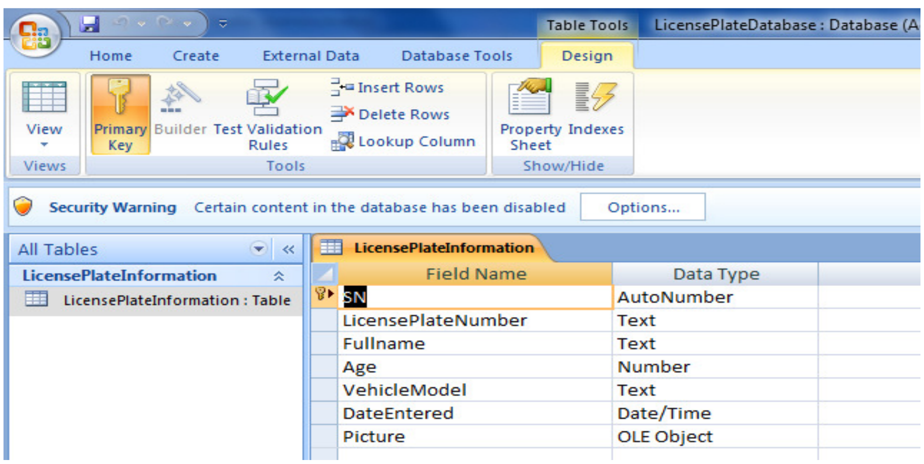


Figure 3: Design view of the access database used by the LPR system

2.4.2. The application interface

The user interface was designed to be simple to use and to ensure that, little effort will be required for a new user to operate the software. It was designed using the Microsoft Visual C# and the .NET Framework 4.0. The application was compiled on both a 32-bit and a 64-bit system. The main interface of the program is shown in Figure 4. The 'Load Image File...' is used to load a picture that contains an image of a car for recognition. The 'Detect Number Plate' button is used to detect and recognize the number plate. If the recognized plate number exists in the database, access is granted to the car owner into a secured area. Otherwise, access is denied. The application also includes a speech engine that says if the user is granted access or not.

A. Obayuwana et al. / Nigerian Research Journal of Engineering and Environmental Sciences
4(1) 2019 pp. 431-446

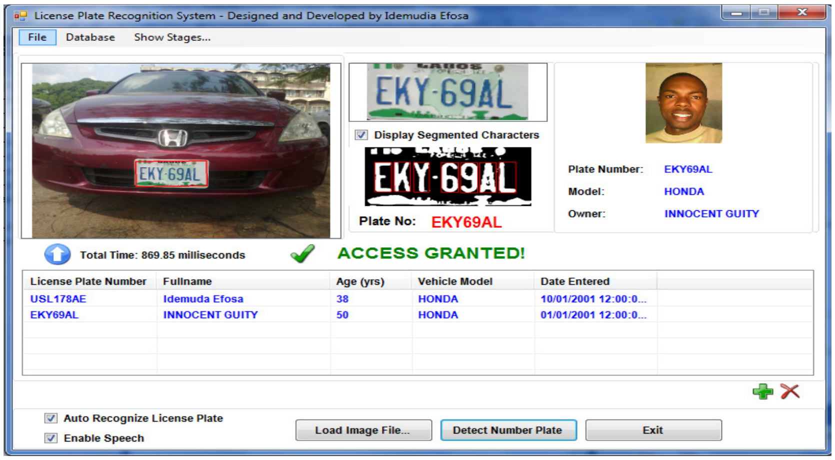


Figure 4: The application interface

3. RESULTS AND DISCUSSION

3.1. Result Analysis for the License Plate Detection Stage

As earlier stated, the images used to test the algorithm were of different dimensions and were each resized to a dimension of 640 x 480 pixels. Figure 6 shows the resized original image used for the testing and Figures 7 shows the converted to greyscale using Equation 1.

Figure 5: Original image

Figure 6: Grayscale image


(a) Image after 3x3 SE dilation (b) Image after 3x3 SE erosion

Figure 7: Effect of morphological dilation and erosion

The image was then passed through the Gaussian and Laplacian pyramid filters which down-sampled and up-sampled the image respectively and thus noise removal was achieved. Figure 8 shows the result of the pyramidal operation. The resultant image was then sharpened as show in Figure 9.

A. Obayuwana et al. / Nigerian Research Journal of Engineering and Environmental Sciences
4(1) 2019 pp. 431-446

Figure 8: Result of pyramidal operation

Figure 9: Sharpened image

The image at this point was then passed through different stage depending on the image quality. All images passed through stage one, and if at the end of the stage no license plate region is found, the images were then passed through the second stage. If a license plate region is found at the end of stage one, there is no need taking the image through stage two. Figures 10 to 13 show the effects of passing the image through stage one.

Figure 10: Applying the Sobel edge operator

Figure 11: Image dilation after Sobel edge detection

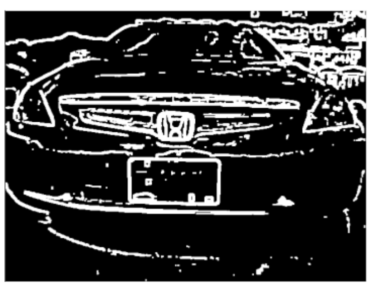


Figure 12: The result of SIS thresholding

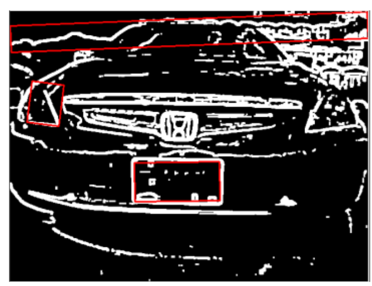


Figure 13: Result of contour analysis

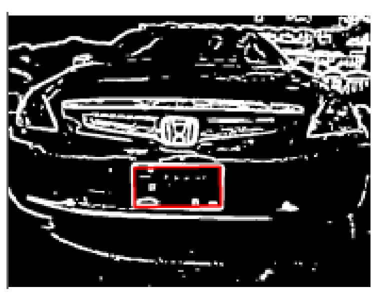
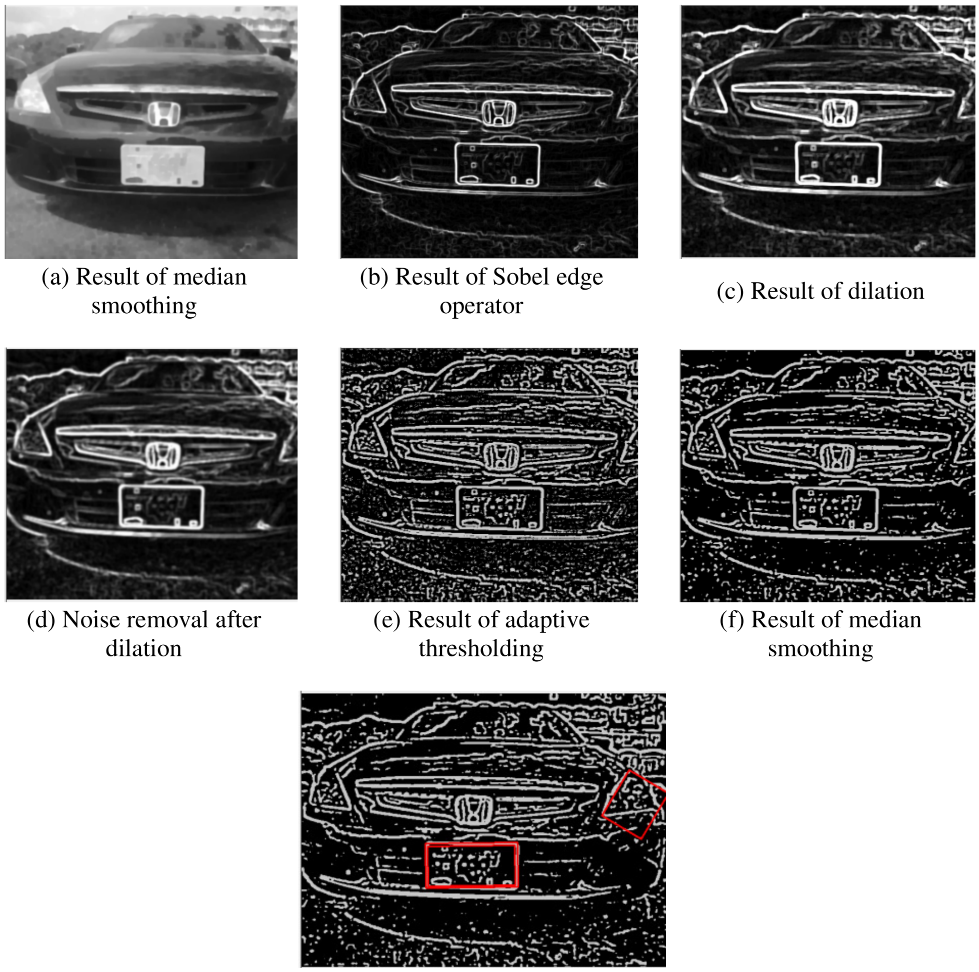


Figure 14: The detected rectangular region

As can been seen from Figure 10, the edges in the image were made visible after the application of the Sobel edge detector. After dilating the resultant image, it can be seen from Figure 11 that the edges became brighter and thicker. Figure 13 shows the result after contour analysis with polygon approximation was carried out on the binarized image and the resultant contours filtered based on their area. Figure 14 shows the result with the rectangular region (in red) that met all the conditions and thus qualifies it as a license plate region. If no license plate region was found at the end of the first stage, the image was taken through the second stage where it underwent additional image processing as illustrated pictorially in Figure 15.



(a) Result of median smoothing
(b) Result of Sobel edge operator
(c) Result of dilation
(d) Noise removal after dilation
(e) Result of adaptive thresholding
(f) Result of median smoothing
(g) The result of contour analysis

Figure 15: Pictorial illustration of stage two of the image processing step

If no license plate region was found at the end of the first and/or second stage, the algorithm was terminated and returned nothing. Otherwise, the coordinates of the detected license plate region were taken and mapped onto the original image and the license plate was extracted as shown in Figure 16.

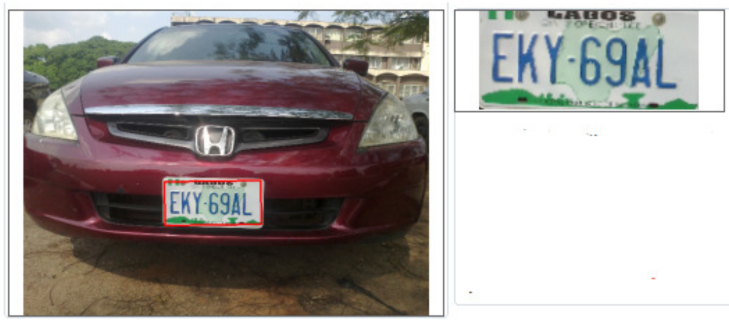


Figure 16: The detected and extracted license plate

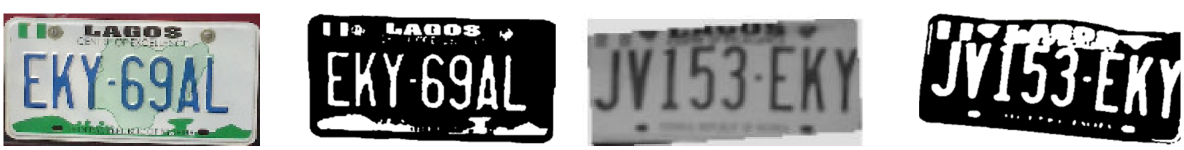
3.2. Result Analysis for the Character Segmentation Stage

The extracted license plate was then passed on to the next stage for character segmentation. The extracted license plate was first taken through some image processing filters before the actual segmentation begins as explained in section 2.2. A median filter was first applied to the image to remove noise while retaining the edges and the image was then converted to grayscale. The image was de-skewed and color inverted as illustrated in Figures 17 and 18. The image was then binarized. The global thresholding technique was not used because of its deficiency as illustrated in Figure 19. A global threshold value of 128 was chosen.

(a) A skewed number plate (b) De-skewed number plate

Figure 17: Result of de-skewing

Figure 18: The inverted image



(a) Original image and binary image (b) Original image and binary image

(c) Original image and binary image (d) Original image and binary image

Figure 19: Result of applying a single global threshold value on different images

As can be seen from Figure 19, a global threshold value of 128 applied to four images with different qualities produced different results. While the result obtained in Figure 19(a) was satisfactory and allowed for proper segmentation to be done, the same cannot be said of the others. Thus, an iterative approach as explained was used to achieve binarization. Figures 20 and 21 show the result of using the iterative approach, which gave a better result. After binarization, connected component analysis was carried out on the image and the connected regions were filtered using blob extraction technique according to the heuristics described in section 2.2. Figure 22 shows the result of this process on two binarized images.

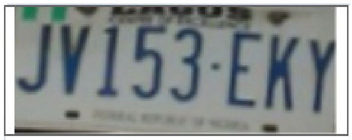


Figure 20: Iterative thresholding at a value of 165

Figure 21: Iterative thresholding at a value of 110

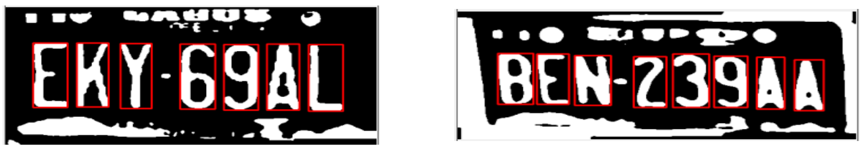


Figure 22: The result of applying the developed heuristics

The characters within the red rectangle are the characters filtered out by the heuristics as the number plate characters. Though the heuristics was successful in segmenting out the license plate characters, some failures were observed. The failures were basically due to fact that the threshold value used was insufficient to binarize the image for proper segmentation. This can be attributed to the factors such as poor lighting conditions, plate deformation, fading of the characters on the plate etc. Figures 23 and 24 show some results where the heuristics failed to accurately filter out the characters of the license number.

Figure 23: The heuristics failed to detect all characters because of the threshold value used

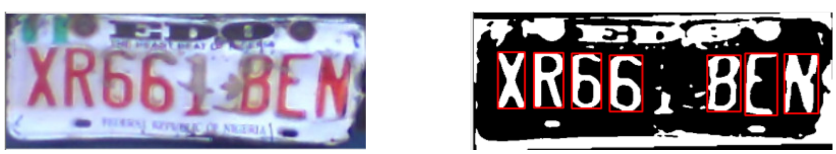


Figure 24: The heuristics failed due to the plate deformation

3.3. Results Analysis for the Character Recognition Stage

The extracted license plate characters were then passed on to the neural networks for character recognition. Some failures were observed in character recognition. Some of these failures are attributed to the font type used on the Nigerian license plate. The Nigerian number plate uses a fixed-width font-type and the design made it difficult for the algorithm to differentiate between characters such as ‘A’ and ‘4’, ‘B’ and ‘8’, ‘0’ and ‘D’, etc. While training the neural networks, attempts were made to get the exact font type used so as to use it for the training. Figure 25 shows the old plate number and the new plate number while Figure 25 shows the closest font type found that was similar to the one used on the Nigerian number plate.

(a) The old number plate (b) The new number plate

Figure 25: Images of the Nigerian number plates

Figure 26: The font type used to train the network

Poorly segmented characters also contributed to failed recognition results. For example, Figure 27 shows where the character was wrongly recognized as ‘J’ instead of ‘U’, while Figure 28 shows two cases where recognition was successful. After the design and implementation, the system was put to test and some of the results are shown in Figures 29 to 31.

Figure 27: Poorly segmented character

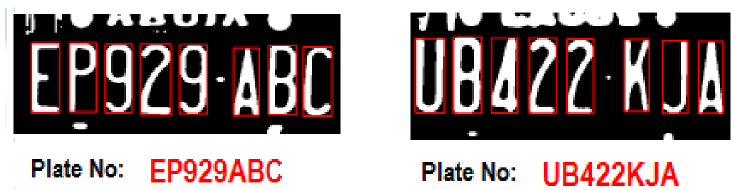


Figure 28: Successfully segmented characters

Figure 29: Number plate detected and recognized successfully

Figure 30: Number plate detected and recognized successfully

Figure 31: Number plate detected successfully but character recognition failed

3.4. Evaluation and Performance of the System

The proposed approach was tested on a dataset of 73 images taken with three different cameras, with resolutions 1.3, 2.0 and 5.0 mega pixels. The system was developed and tested on an Intel(R) Pentium(R) Dual-Core Processor CPU T2390 (@ 1.8GHz), with 2.00 GB Memory. The performance of the system was analyzed and Table 1 shows a summary of the results obtained, while Table 2 shows the average time for each process.

Table 1: Summary of the result

	Stage	Number of samples tested	Success	Failure	Success Rate
1	Number plate detection	73	62	11	84.9%
2	Character segmentation	62	39	23	62.9%
3	Character recognition	39	28	11	71.7%

Table 2: Execution time for various stages

	Stage	Average time (seconds)
1	Number plate detection	0.468
2	Character Segmentation	0.275
3	Character recognition	0.129
	Total average	0.872

Out of the 73 images in the dataset used for testing, the system was able to detect and localize the number plates in 60 of the images. Plate number detection failed in 13 images. These failures are as a result of poor lighting condition, illuminations and some other ‘distractions’. It was also observed that the system experienced more failure in the area of character segmentation. The algorithm used couldn’t properly get the right threshold for proper segmentation. The failure at this section also affected the character recognition rate.

4. CONCLUSION

The method proposed in this work was able to carry out number plate detection, segmentation and recognition of Nigerian vehicle plate number in a picture frame. The Nigerian number plate images were acquired using three different cameras with resolutions; 1.3 MP, 2 MP and 5 MP under various environmental and lighting conditions. Even if the plate were dirty or contained small mechanical damages. However, the final result of effectiveness of proposed license plate localization and recognition system was not very impressive. This was caused by the poor effectiveness of the proposed character segmentation and recognition method. One wrongly segmented or recognized character is enough to reject whole localized, segmented and recognized caption in the syntax analysis process. More improvement still needs to be done, especially in the character segmentation stage, in order to improve the overall detection and recognition process.

5. CONFLICT OF INTEREST

There is no conflict of interest associated with this work.

REFERENCES

Bagade, J.V., Msukanya, K., Kushal, P. and Bhushan, P. (2010). Automatic Number Plate Recognition System: Machine Learning Approach. *IOSR Journal of Computer Engineering*, 6(2), pp. 34-39.

Cika, P., Zukal, M. and Sebela, M. (2011). Vehicle license plate detection and recognition using symbol analysis. In: *34th International Conference on Telecommunications and Signal Processing (TSP)* (pp. 589-592). IEEE.

Gershenson, C. (2003). Artificial neural networks for beginners. *arXiv preprint cs/0308031*.

Halgaonkar, P.S. (2011). Connected component analysis and change detection for images. *International Journal of Computer Trends and Technology*, pp. 128-133.

Kanan, C. and Cottrell, G. W. (2012). Color-to-Grayscale: Does the Method Matter in Image Recognition? Retrieved from http://cseweb.ucsd.edu/~ckanan/publications/Kanan_ Cottrell_PloS_ONE_2012.pdf on 05-03-2018.

Kaushik, D. (2012). An Efficient Method of Vehicle License Plate Recognition Based on Sliding Concentric Windows and Artificial Neural Network. *IOSR Journal of Computer Engineering*, 6(2), pp. 25-35.

Khalil, M. I. (2010). Car Plate Recognition Using the Template Matching Method. *International Journal of Computer Theory and Engineering,* 2(5):27-40.

Kolour, H.S. and Shahbahrami, A. (2011). An evaluation of license plate recognition algorithms. *International Journal of Digital Information and Wireless Communications (IJDIWC)*, *1*(1), pp. 247-253.

Kwaśnicka, H. and Wawrzyniak, B. (2002). License plate localization and recognition in camera pictures. In *3rd Symposium on Methods of Artificial Intelligence*, pp. 243-246.

Kumar, P. and Subin, P. S. (2010). SVM Based License Plate Recognition System. *IEEE International Conference on Computational Intelligence and Computing Research,* 2, pp. 7-11.

Muhammad, H. D. and Zahra, D. B. (2011). A Novel Approach for Vehicle License Plate Localization and Recognition. *International Journal of Computer Applications,* 26(11), pp. 975 – 8887.

Ni, D.X. (2007). Application of neural networks to character recognition. *Proceedings of students/faculty research day, CSIS, Pace University, May 4th*, 2007.

Olivier, D. (2015). Face Detection using Viola and Jones Method and Neural Networks. *International Journal of Information and Communication Technology Research,* 7, pp. 1-19.

Owamoyo, N., Fadele, A.A. and Abudu, A. (2013). Number plate recognition for Nigerian vehicles. *Academic Research International*, *4*(3), p.48.

Öztürk, F. and Özen, F. (2012). A new license plate recognition system based on probabilistic neural networks. *Procedia Technology*, *1*, pp.124-128.

Pandya, P. and Singh, M. (2011). Morphology based approach to recognize number plates in India. *International Journal of Soft Computing and Engineering (IJSCE)*, *1*(3), pp. 107-112.

Patel, C., Shah, D. and Patel, A. (2013). Automatic number plate recognition system (anpr): A survey. *International Journal of Computer Applications*, *69*(9), pp. 21-33.

Roberts, D.J. and Casanova, M. (2012). *Automated license plate recognition systems: Policy and operational guidance for law enforcement* (No. 239604).

Sharma, V., Rai, S. and Dev, A. (2012). A comprehensive study of artificial neural networks. *International Journal of Advanced research in computer science and software engineering*, *2*(10), pp. 278-284.