Research Article

Development of Smart Plate Number Recognition System for Fast Cars with Web Application

Olamilekan Shobayo [ID], Ayobami Olajube, Nathan Ohere, Modupe Odusami, and Obinna Okoyeigbo

Department of Electrical and Information Engineering, Covenant University, Canaan Land, KM 10, Idiroko, Road, P.M.B. 1023, Ota, Ogun State, Nigeria

Correspondence should be addressed to Olamilekan Shobayo; olamilekan.shobayo@covenantuniversity.edu.ng

Traffic law violation has been recognized as a major cause for road accidents in most parts of the world with majority occurring in developing countries. Even with the presence of rules and regulations stipulated against this, violators are still on the increase. This is due to the fact that the rules are not properly enforced by appropriate authorities in those parts of the world. Therefore, a system needs to be designed to assist law enforcement agencies to impose these rules to improve road safety and reduce road accidents. This work uses a Vehicle Plate Number Recognition (VNPR) system which is a real-time embedded system to automatically recognize license plate numbers. It provides an alternative means to VPNR using an open-source library known as openCV. The main aim of the system is to use image processing to identify vehicles violating traffic by their plate numbers. It consists of an IR sensor for detecting the vehicle. During testing, a minimum time was set for the sensor to detect the object which was recorded by the microprocessor. Once it was less than the set time, the camera was triggered to capture the plate number and store the image on the Raspberry Pi. The image captured is processed by the Raspberry Pi to extract the numbers on the image. The numbers on the capture imaged were viewed on a web page via an IP address. The system if implemented can be used to improve road safety and control traffic of emerging smart cities. It will also be used to apply appropriate sanctions for traffic law violators.

1. Introduction

With the increase in the number of automobiles, it has become increasingly challenging to track them and almost impossible to identify the owners of these vehicles in case of violation of any traffic law [1]. This has led to increase in traffic congestion, and with it comes traffic problems. Several cases of kidnapping, hit and run, robbery, smuggling, and on-road fatalities are continuously reported, and this is because these vehicles cannot be easily recognized especially moving at high speed [2]. This has led to the need to develop a system that can tackle these issues. Vehicle plate number recognition system was invented by the British in 1976 at the police scientific development branch in the UK. Prototype systems were already in full flow by 1979, and initial contracts were awarded for mass productions and industrial use.

The earliest systems were deployed on the A1 road and at the Dartford Tunnel. However, it gained more recognition in the last decade due to the advancement of the digital camera and the increase in computational speed [3]. The first written down case of VNPR being used to solve a crime case occurred in November 2005, in Bradford, UK. In the early 1990s, significant advancement in technology led to the redesign of vehicle plate number recognition system from limited, highly expensive, fixed base applications to simple easy to install mobile ones [4]. Vehicle plate number recognition system has broad applications and is not limited to traffic control. It can be used in parking; it recognizes a vehicle when it enters a gate and stores the information on a database and when the vehicle is leaving it matches it up with what has already been stored. It is also used in access control to grant access to only vehicles of authorized personnel.

Although vehicle plate number recognition system has been regularly addressed on paper, it has not been completely actualized in real-life scenarios due to certain constraints [5]. We live in a digital age with new technologies springing forth everyday, and as a result, only minimal human effort is required to get things done and life is generally simplified. Due to the digitization of our everyday life images taken from a camera become very useful as these images can be processed to extract important information from them [6]. Integrating vehicle plate number recognition system into intelligent transport systems makes it possible to automate motorway toll collection, analyze traffic, and improve law enforcement. The need to integrate information and technology into all aspects of human life has led to the demand for processing of vehicles as a valuable resource information communication. This is because an information system without any form of data on it is unrealistic. There has always been a need to identify vehicles by their plate number in a real environment and use the information in real-life applications. Due to this, various techniques and vehicle recognition systems have been developed for various applications such as traffic control, automation, and security [7].

2. Related Works

The increasing affluence of modern life has led to an increase in the number of user-road vehicles. This has led to an unprecedented level in civil problems—the problems being vehicle identification and traffic control [8]. Due to the increasing challenge of traffic control, there arises a need to manage traffic better. The recognition of fast-moving vehicles is an important aspect of Intelligent Transportation System (ITS). ITS was developed to implement information and communication technologies in the field of road transport, some of which include vehicle management, traffic management as well as interfacing with other systems of transportation. A lot of researchers have tried to develop ITS systems with different levels of success stories. The authors in [9] designed a system to recognize plate numbers using Python and OpenCV. The design was implemented using a real-time embedded system designed to automatically identify number plates of vehicles. The system made use of the python software and image processing algorithms in the OpenCV library to capture the image. It was also used in preprocessing to resize the image, perform error checking, and convert image to grayscale. Threshold algorithms were used to binarize the image in order to retain the quality of image, and character segmentation was done using image scissoring to scan the image and crop out the white portion by partitioning the image lines into individual characters. The system however was unable to sense moving vehicles and had no presence of database to record information collected. The system is easy to use as it makes use of free-open source software applications that are easy to use and also makes use of computer vision for conversion of still images to information usable by computers. The authors in [10] designed a system that automatically recognizes plate number using Raspberry Pi. The system proposes the implementation of an automated plate number recognition system using Optical Character Recognition (OCR) to interpret information on the images of a vehicle plate number using Raspberry Pi processor. The image captured is processed and verified by the Raspberry Pi processor for authentication purposes. The proposed system then segmented the license plate and identified each particle of the fragment. To get final result, the system extracts the number plate using open ALPR design. This result was then manually compared with the input image. The system used was unable to detect moving object and could not improve quality of image in the system. The system was however able to easily identify the plate number and process the image of the number plate. The authors in [11] implemented a system to detect over-speeding vehicles & inform authorities in case of any violations. The system makes use of LIDAR (Light Detection and Ranging) technology to check the speed of the moving vehicle. LIDAR is a remote sensing method that makes use of light in form of pulsed laser to measure the distance to a target by illuminating the target with pulsed laser light and using a sensor to measure the reflected pulses [12]. The major drawback of the system is its high cost and the amount of time taken to achieve it. The system was however an effective system. The authors in [13] implemented an ALPR for detecting traffic violators for promoting sustainability in modern transport systems. This system made use of a camera to capture the plate number of vehicles. The image was then refined using different techniques such as thinning and convolution; next OCR (Optical Character Recognition) was used to recognize the text on the number plate. The image was then extracted in the form of text; the characters are then matched with a prestored database containing details of the college or campus road users with their cell-phone number and vehicle plate number. When a match is detected, the offender is sent a warning message on his phone number. The major limitation of the system was the difficulty in implementing system in larger areas like cities. The system was highly accurate and cost effective as not many components were required. The authors in [14] designed a system to automatically detect number plates of vehicles in the cases of variations in the number plate. These variations could be the location of the plate, font size on the plate, and various designs and text used on the number plate. The system made use of a preprocessing module to improve the quality of the captured image. Removal of blur and noise was also carried out in this stage. Next, the system used plate localization to create a bounding area around the region of the image. Then the numbers on the number plate were extracted after which they were segmented to identify the region of each character on the number plate. Finally, the numbers were sent to a database with appropriate information on the owner of the vehicle. The downside of the system was the several stages involved which were cumbersome and the several components required which made it expensive. The upside of the system was the high accuracy of 98.75% recorded in number plate detection and the provision of a database to record information collected which was useful for referral purposes. The author in [15] designed a system to identify the plate number of fast-moving vehicles

using snapshots of precaptured images of surveillance cameras. The design made use of an estimation algorithm parameter known as the novel kernel parameter for deblurring the plate number of the fast-moving vehicles. Various algorithms were used to carry out sparse representation, the angle was calculated using the genuine motion angle of the vehicle and the length of the motion kernel was calculated using random transform in the frequency domain. Finally, the image was deblurred using NBID (nonblind image deblurring) algorithm. The downgrade of the system was the extreme difficulty in implementing the system and the lack of a database to record information collected. The positive side of the system was its ability to handle motion blurred number plate which is unrecognizable by the naked eye. The authors in [16] designed a system to recognize dirty and unclear vehicle number plates for real-time applications. The system was implemented to meet real-time conditions, some of which include variations in weather conditions, poor lighting, different traffic situations, and high-speed vehicles all of which affect vehicle detection. The system implemented made use of several hardware platforms along with robust algorithms to detect vehicle plate number. Datasets were used from images captured from crossroads in different conditions, in the day and night and other various conditions. The system made use of plate detection to identify the vehicle plate number which was then segmented into individual characters before character recognition was carried out. The limitation of the system was its inability to detect moving vehicles. The benefits of the system were the high accuracy results of 98.7%, 99.2%, and 97.6% recorded and its numerous real-life applications. The authors in [17] implemented a system to recognize vehicle number plate using IOT (Internet of Things). The system was designed to meet changing urban requirements in the modern-day transportation system. The system proposed four major stages: preprocessing of the obtained image to enhance the quality of the number plate, extraction of the plate number region, character segmentation to segment each character in the number plate, and character recognition. The system also proposed the use of a novel algorithm to extract the plate number region through a unique edge detection algorithm. The system also created a database which contains information about the owner of different vehicles which was done using IOT. However, the system was unable to detect moving vehicles and could not be used in real-life application. The system however had provision of a database to record collected information and was implemented using modern technology—IOT. The authors in [18] developed a system to identify vehicles that go in and out of a barrier gate parking system. The system made use of a real-time embedded system consisting of a Pinoir camera for image capture and a Raspberry Pi for image processing. Once the system detects a vehicle approaching the barrier, it automatically reads the license plate number and compares it with an existing list and opens the barrier if there is a match. The major shortcoming of the system was the inability to capture the license plate number at far distances because of the fixed position of the camera. The plus of the system was that it was automated and made use of a database to compare existing plate numbers.

2.1. The Problem.

Traffic laws and devices are put in place to monitor and control traffic, including pedestrians, motor vehicle drivers, and cyclists. As is well known, there are several traffic law offenders in our country today and this has led to several accident incidents without any repercussion being meted out. These traffic law breakers go about their business freely and confidently knowing that they may never be caught because the appropriate systems have not been put in place. The resultant consequence of breaking traffic laws has led to traffic gird locks, road accidents, and even death when involved in a fatal accident. There is no questioning the need for appropriate systems to check mate traffic law breakers so as to ensure our roads are safer. The current problem involves identifying fast-moving vehicles by their plate number. It can be extremely hard to capture the image of moving vehicles in real time in such a way that the vehicle plate number is detected. The existing systems reviewed above have deployed several means to capture plate number systems. Most of them successfully captured the plate numbers of fast-moving vehicles. The closest to this work, which uses OpenCV with python, was able to detect the plate number but did not incorporate a database system to identify the culprit.

2.2. Proposed Solution.

Our proposed solution will be a novel system that uses the existing OpenCV code, written in Python but will incorporate a web page that will display the result of the saved plate number, which can be used to query a database for the identification of a defaulter. The database will have the plate numbers of all vehicles that will be tested and queried to reveal the defaulter. The system developed will take a very effective and methodological approach in the detection of fast-moving vehicles, using cost-effective techniques and achieving great results. The process flow of the system is shown in Figures 1–5.

2.2.1. Vehicle Detection.

The proposed system identifies the moving vehicle using passive infrared sensors (PIR). Upon sensing the vehicle, the PIR sends an input signal to the Raspberry Pi which in turn triggers the USB camera.

2.2.2. Image Acquisition.

The next step is the image acquisition. The Raspberry Pi which is interfaced with on receiving the input signal triggers the camera to capture the image of the vehicle. This is quite a difficult task as it entails capturing the image of a moving vehicle to extract information in real time. The image should be captured in such a way that it contains either the front or rear view of the vehicle with its plate number [19]. The image to be captured is usually in the RGB (Red, Green, and Blue) colour model. The level of success in this stage can be affected by a number of factors, some of which include system noise, blur, distortion, and excessive relative motion of vehicle.
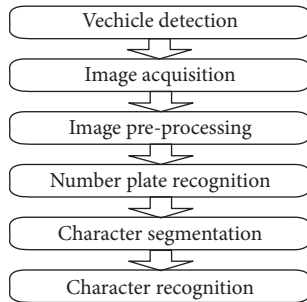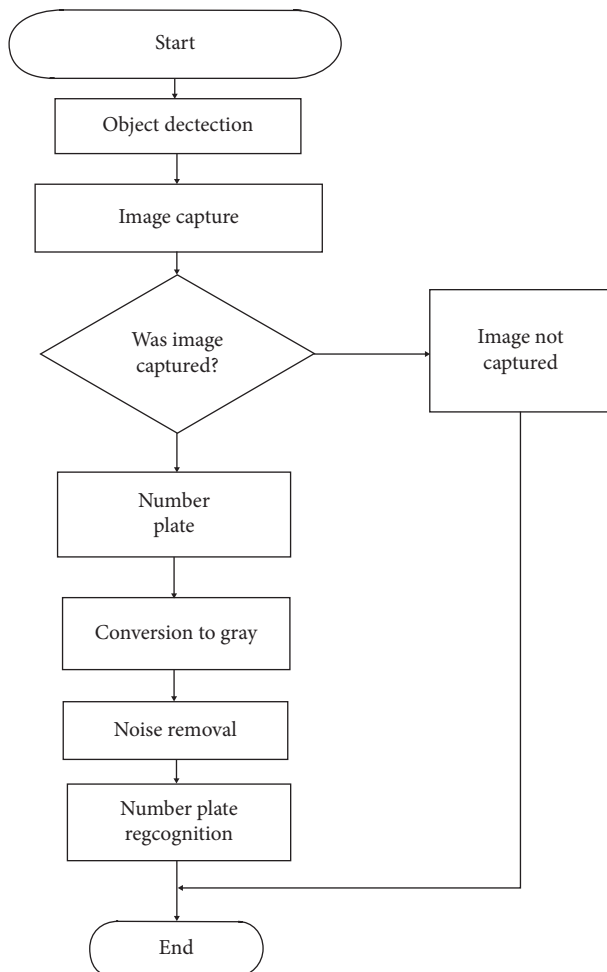
Figure 1: Proposed design process flow.

Figure 2: The system design flowchart.

2.2.3. Image Preprocessing. The image captured by the camera needs to be processed to improve the quality of the image. This involves background subtraction, enhancing contrast, sharpening or smoothing, and noise removal [20]. It serves as a correction mechanism to take care of errors that might have occurred during image acquisition. It entails converting RBG image into gray colour and border enhancement for brightness [21]. To convert images in RGB to gray format, each RGB component requires at least 8 bits for their storage. The preprocessing technique is implemented using preprocessing algorithms. The algorithm converts RGB image in 3-dimensional to 2-dimensional gray format.

2.2.4. Number Plate Recognition. This is a very important stage in the VPN system. Number plate detection searches the image taken in order to locate the specific features that contain the number plate. The system makes use of OpenCV—Open source Computer Vision—to convert data from images on a camera to useful information on the Raspberry Pi. It involves processing of real-time images to provide vision to the computer. Each frame in the image will be processed to find the object in the image [22, 23].

2.2.5. Character Segmentation. This process involves cropping out the required portion of the plate number. It partitions line images into individual characters; it divides an image comprised of characters into subimages of individual symbols. It involves isolating the image from the background in order to properly identify each text.

2.2.6. Character Recognition. This is where optical character recognition and OpenCV are fully utilized. It involves detecting and identifying image frames and converting them to meaningful text in ASCII (American Standard Code for Information Interchange).

3. System Requirements and Design Methods

The proposed system is built on a Raspberry Pi with a USB camera to capture a traffic scene. It makes use of a PIR motion sensor which detects infrared energy radiating from objects (plate number) within its field of vision. The PIR sensor sends an input signal to the Raspberry Pi which in turn triggers the USB camera to capture the image. The Raspberry Pi extracts the number plate part of the image. It then processes the extracted part through a series of image processing techniques to extract the numbers from the number plate and display it on an LCD display. The resulting data are then matched to records on a database set up on the Raspberry Pi. The system requirement is listed below:

(i) Microprocessor: Raspberry Pi model B, 5 V-DC input voltage

(ii) Camera: USB camera, 5 Megapixel

(iii) PIR sensor: 300 W output load

(iv) LCD Display: 16 * 2-character display

(v) Operating system: Linux based Raspbian OS

The flowchart of the designed system is shown in the flowchart below:

The design of the system consists of the hardware component design and the software component design.

The interaction between the two parts is shown in the block diagram below:

The system is implemented in 3 phases, which includes the setting up of the Raspberry Pi, implementing the
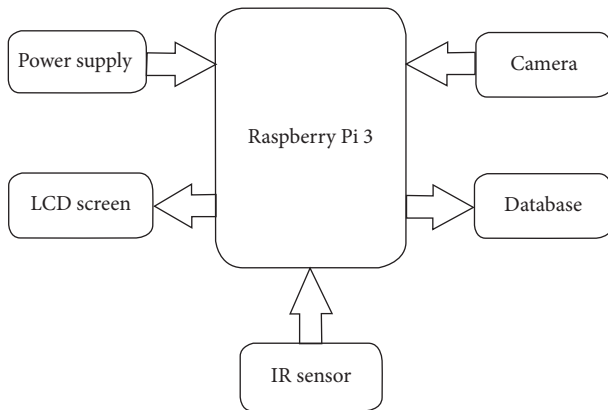
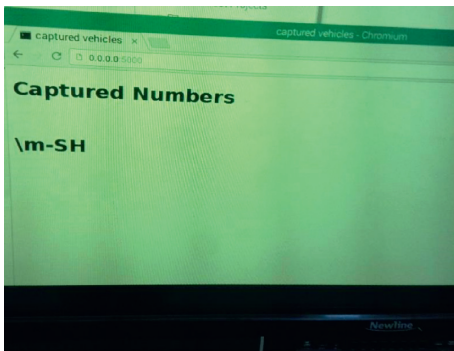Figure 3: Block diagram of the proposed system.

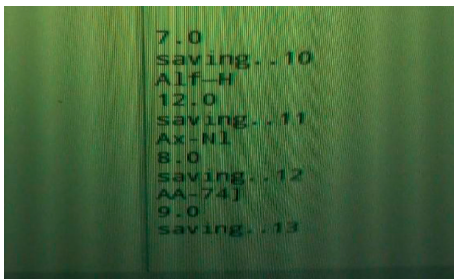Figure 4: Designed web page for the system.

Figure 5: Showing captured plate numbers.

software part, and finally assembling the hardware components of the system. While setting up the Raspberry Pi, which is the most vital part of this work, an SD card was provided for the Raspberry Pi since it does not have a memory of its own. This SD card holds the Operating System of the Raspberry Pi. The OS used is the raspbian-jessie.img OS which is a variant of Debian. The Win32DiskImager is used to copy the OS to the SD card. The size of the raspbian OS is 4 GB; hence a 16 GB SD card was used to provide enough space for other functionalities. Once this is complete, the SD card is inserted into the micro USB slot of the Raspberry Pi. After successfully setting up the SD card, some commands were given to properly set up Raspberry Pi for use. This instruction includes the following:

sudo apt-get *update* (used to download package list from repositories).

sudo apt-get upgrade (this installs newest version of already installed software).

sudo rpi-update (used to update firmware).

For software implementation, some packages and libraries were installed before writing code to properly interface components like the sensor. The code was developed for the Raspberry Pi using the python programming language. The libraries were installed using a line of command "sudo apt-get install" on the python terminal. They include the following:

sudo apt-get install *<python-pip>* (library for LCD screen)

sudo pip install RPLCD (library for LCD screen)

sudo apt-get install <fswebcam> (library for USB camera)

sudo apt-get install < libopencv-dev> (library for image processing)

sudo apt-get install <numpy> (library that supports linear algebra).

sudo apt-get install <pytesseract> (library for optical character recognition)

In programming the Raspberry Pi, all the libraries that were installed were first imported using the import command. Then the pins connecting to the LCD screen were specified using the CharLCD. A potentiometer was used to calibrate the proximity range of the IR sensor so that at a certain range, it sends an input signal to the Raspberrry Pi which triggers the camera to capture the image. Once the image is captured, it stores it in a created folder on the desktop of the Pi. This is where the image processing takes place; this is done through a sequence of operations. First of all, the source path of the image is specified using the img_path command. Next the image was converted from its binary form to grayscale using the COLOR_BGR2GRAY command. This was done to simplify the image for easy analysis. Afterwards a noise removal algorithm introduced using the scr_path + "remove_noise.png command. This was done to smoothen the entire image. The cv2.THRESH_-BINARY,31,2 was used to further simplify the image to a basic form which is either black or white. After thresholding, noise removal was carried out to further smoothen image, after which a series of image processing techniques were carried out to finally extract the actual numbers on the image and display on an LCD screen. Then the number is compared with others on a database. Hardware components were interfaced with the Raspberry Pi. For the hardware implementation, an IR proximity sensor was used to detect the plate number. The IR sensor has 3 pins: one is connected to a 5 v Vcc of the Raspberry Pi, the other one is connected to ground, and the last one which sends the input signal to a GPIO pin on the Raspberry Pi.

Finally, a web page was developed to access the text extracted from the pictures taken by the camera over a
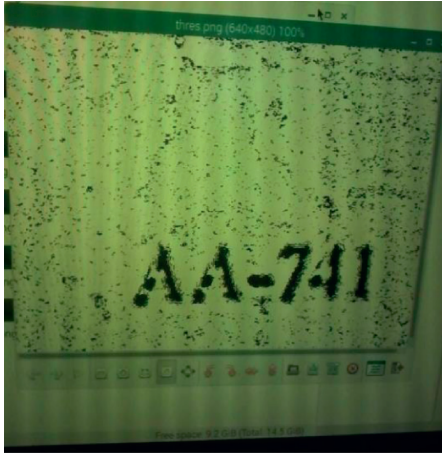
FIGURE 6: Threshold text image.

FIGURE 7: Noiseless text image.

network. Web programming was done using the Hyper-text Markup Language (HTML) on the python IDE. An IP address 0.0.0.0/5000 was used on the network of the web page. The web page can be accessed by anyone on the same network as the Raspberry Pi. The diagram below shows a sample of the text from the designed web page.

4. Testing and Result

After successful assemblage of the hardware parts and the configuration of the software components, the functionality of the system was tested on a built model. The IR sensor was used to detect the moving cars. Once detected, the camera was triggered to capture the image of the plate number. The system was tested on a simulated prototype road, and the calibration was to detect a speed limit of 40 centimeter/hour. Cars were made to pass on this road with varying speeds. Out of the cars that passed, the system was able to detect 3 cars moving with speeds more than the stipulated limit. To detect these fast-moving cars, 3 PIR motion sensors were attached to segments of the road in order to send signal to the Raspberry Pi to trigger the USB Camera. The PIR sensors were placed 40 centimeters apart, i.e., 20 centimeters between each sensor. The sensors were programmed such that when a fast car approaches the first sensor, the PIR cannot trigger the camera in 10 second (this is also scaled by a value of 10 to mark the speed limit), and also the second PIR, then the third PIR will automatically trigger the USB camera to take pictures of the car as it is assumed to be overspeeding. Images of the cars that the first and second PIR sensors triggered the camera for were discarded as it is assumed that these cars were not overspeeding. Only images triggered by the third PIR were processed. The camera was placed in such a way that the rear of the car will be captured so that its plate number can be captured and verified with an existing database. After this was complete, the captured image was made to undergo image processing and optical character recognition. This is made possible by the OpenCV codes written with Python. Once the text was extracted, it was displayed on a web page. The results obtained are as shown below.
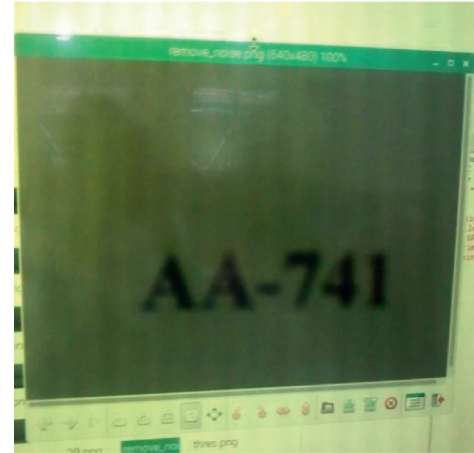
As seen from the above screenshot, the module was able to capture different plate numbers of cars which was deemed to have crossed the set speed threshold, and it was able to tag each one of them with a unique number, i.e., 9.0, 8.0, 12.0. This serves as ways of differentiating each vehicle from the other so it can easily be traced on the database. The saving sequence could also be observed from the screenshots shown below.

As shown in Figure 6, the text received from the camera is in the threshold form, i.e., it is susceptible to noise and other kinds of interference. The Raspberry Pi is configured to perform filtering on the image by removing the noise components as shown in Figure 7. The image in Figure 7 is made available to the web page developed for the system, which is accessible to the database that holds the accurate information of the car owner.

5. Conclusion

In this work, a highly efficient, economical, and VRPN system was developed. This system uses smart IR sensor to detect moving objects, camera to capture the image, extract the text from the image, and save the text on a web page. The major component selected to do the main work in the system is the Raspberry Pi. OpenCV was used together with python programming, for the character segmentation and recognition. This was configured on the Raspberry Pi, including the required library that is required to make the hardware components of the system to work with the Raspberry Pi. As shown in the results, the developed system was able to deliver a clear and readable text on a web page which can be used to query a database. Due to the calibration of the IR sensor, it is able to detect fast-moving cars and also able to log different cars at different times using a tag for each data it receives and having a unique saving sequence. The system is by far one of the cheapest means for the actualization of traffic management in a smart city.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

Acknowledgments

References

[1] D. Mitra and S. Banerjee, "Automatic number plate recognition system: a histogram based approach," *IOSR Journal of Electrical and Electronics Engineering*, vol. 11, pp. 26–32, 2016.

[2] A. Goyal and R. Bhatia, "Various techniques for number plate recognition-a review," *International Journal of Computer Applications*, vol. 143, 2016.

[3] B. Singh, M. Kaur, D. Singh, and G. Singh, "Automatic number plate recognition system by character position method," *International Journal of Computational Vision and Robotics*, vol. 6, no. 1-2, pp. 94–112, 2016.

[4] M. T. Qadri and M. Asif, "Automatic number plate recognition system for vehicle identification using optical character recognition," in *Proceedings of the 2009 International Conference on Education Technology and Computer*, pp. 335–338, Singapore, April 2009.

[5] R. Laroca, E. Severo, L. A. Zanlorensi et al., "A robust real-time automatic license plate recognition based on the YOLO detector," in *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–10, Rio de Janeiro, Brazil, July 2018.

[6] P. Bhogale, A. Save, V. Jain, and S. Parekh, "Vehicle license plate detection and recognition system," *International Journal of Computer Applications*, vol. 137, no. 9, pp. 31–34, 2016.

[7] G. Sharma, "Performance analysis of vehicle number plate recognition system using template matching techniques," *Journal of Information Technology & Software Engineering*, vol. 8, no. 2, 2018.

[8] E. Council, "Directive 2010/31/EU of the European parliament and of the council of 19 may, 2010 on the energy performance of buildings," *Official Journal of the European Union*, vol. 153, pp. 13–35, 2010.

[9] K. Sajjad, *Automatic License Plate Recognition Using Python and Opencv*, Department of Computer Science and Engineering, MES College of Engineering, Kuttippuram, Kerala, 2010.

[10] A. Kumthekar, M. S. Owhal, M. S. Supekar, and M. B. Tupe, "Recognition of Vehicle Number Plate Using Raspberry Pi," *International Research Journal of Engineering and Technology*, vol. 5, no. 4, pp. 391–394, 2018.

[11] J. Muñoz, M. Craft, M. Ahmadian, and S. Wrobel, "Multi-function LIDAR sensors for non-contact speed measurement in rail vehicles: Part I—system installation and implementation," in *Proceedings of the 2013 Joint Rail Conference*, Philadelphia, PA, USA, April 2013.

[12] S.-L. Chang, L.-S. Chen, Y.-C. Chung, and S.-W. Chen, "Automatic license plate recognition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 1, pp. 42–53, 2004.

[13] P. Agarwal, K. Chopra, M. Kashif, and V. Kumari, "Implementing ALPR for detection of traffic violations: a step towards sustainability," *Procedia Computer Science*, vol. 132, pp. 738–743, 2018.

[14] S. Shah, N. Rathod, P. K. Saini, V. Patel, H. Rajput, and P. Sheth, "Automated Indian vehicle number plate detection," in *Soft Computing: Theories and Applications*, pp. 453–461, Springer, Berlin, Germany, 2019.

[15] N. Bolaj and G. Padalkar, "The license plate identification of fast moving vehicles," in *Proceedings of the 2017 International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, pp. 1–5, Pune, India, August 2017.

[16] R. Panahi and I. Gholampour, "Accurate detection and recognition of dirty vehicle plate numbers for high-speed applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 767–779, 2016.

[17] K. Tejas, K. A. Reddy, D. P. Reddy, K. Bharath, R. Karthik, and M. R. Kumar, "Efficient license plate recognition system with smarter Interpretation through IoT," in *Soft Computing for Problem Solving*, pp. 207–220, Springer, Berlin, Germany, 2019.

[18] E. R. Buhus, D. Timis, and A. Apatean, "Automatic parking access using openalpr on raspberry pi3," *Acta Technica Napocensis*, vol. 57, p. 10, 2016.

[19] S. M. Mutua, *An Automatic Number Plate Recognition System for Car Park Management*, Strathmore University, Nairobi, Kenya, 2016.

[20] NOAA, *LIDAR–Light Detection Ranging—is a Remote Sensing Method used to Examine the Surface of the Earth*, NOAA, Silver Spring, MD, USA, 2013.

[21] D. Y. Gaikwad and P. B. Borole, "A review paper on automatic number plate recognition (ANPR) system," *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, vol. 1, 2014.

[22] O. Abayomi-Alli, M. Odusami, D. Ojinaka et al., "Smart-Solar Irrigation System (SMIS) for Sustainable Agriculture," in *Proceedings of the International Conference on Applied Informatics*, pp. 198–212, Medan, Indonesia, September 2018.

[23] K. Okokpujie, E. Noma-Osaghae, S. John, K.-A. Grace, and I. Okokpujie, "A face recognition attendance system with GSM notification," in *Proceedings of the 2017 IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON)*, pp. 239–244, Owerri, Nigeria, November 2017.