International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 11 – No. 3, August 2016 – www.ijais.org

Automated Parking Management System using Image Processing Techniques

Vaidehi P. De
Dept. of Electrical and Computer Engineering
Caledonian College of Engineering
Muscat, Oman

D. Ragavesh
Dept. of Electrical and Computer Engineering
Caledonian College of Engineering
Muscat, Oman

ABSTRACT

Computerized systems being an integral part of the current era, an automated parking system is one of its most commonly used applications. This automated parking management system is designed to give access, exclusively to vehicles registered in the database. It takes the image of the vehicle and locates the license plate and identifies its license number. It then checks with the database to determine its access. This report is a detailed description of the image processing techniques used for Automatic Number Plate Recognition. It deals with computer vision and the various techniques used in image processing. The entire setup is compact enough to improve its portability and efficiency, whilst providing better security. The algorithms, train the machine for better pattern recognition to locate number plates. This system requires the vehicle to be physically present, so that security is not hampered with and access is denied to any vehicle that is not registered. The results were recorded to observe the quality of recognition.

General Terms

Automated Number Plate Recognition (ANPR), Computer Vision, Intelligent Transport Systems

Keywords

Artificial Neural Networking; Image Processing; Beaglebone Black; Edge Detection; Segmentation

1. INTRODUCTION

The transportation and communication sector has adopted automated systems to upgrade its quality. One, such technological component is the Automatic License Plate Reader (ALPR) which is largely used in Intelligent Transport systems (ITS). Car parking systems these days have become automated and no longer use manual systems in most cases. There are many ways of operating a parking gate, e.g. GSM, wireless transmitter, magnetic probe detector, etc. Some of these systems don’t have the option of verifying the cars that enter the parking lot. The proposed method uses image processing to verify the vehicle and open the gate to the parking lot. This technique addresses this issue by taking the image of the vehicle, locating the license plate and checking it with a database of registered license numbers within the system, hence maintaining security of the premises. One of reasons this specific technology (image processing), was employed was because its wide use in various fields of automation. The fact that this device can verify the car before it is allowed into the parking area, increases the security level. One of the more distinctive features of this concept is that the system is compact and dynamic, while being low cost. The system is expected to contain the database of the license numbers authorized to park in the premises. This will also manage the number of cars present. The proposed system captures an image of the vehicle using a camera, so that the license plate can be located within the image. Once it has been located, the characters will be recognized using various image processing algorithms for optical character recognition. After the number is read, the device will decide whether the gate should open or not. The decision is taken by comparing the number with the database of authorized license numbers. Various algorithms and tools were employed in the image processing part of the system such as, image processing, pattern recognition and character recognition. These features helped in license plate detection and recognition. The device has certain restrictions when it comes to damaged license plate or improper lighting and hence were taken into consideration while analyzing the output. The other difficulties faced during the process include: Motion blur – Caused by the moving vehicles; poor file resolution – caused by low quality of camera or when the vehicle was too far away, varying templates of the number plates, etc. The dimensions and template of the Omani license plate was used for the machine training process of this device. The main focus for the results was reading the numbers clearly as opposed to the letters and the Arabic script on the plate. The basic idea of this project is versatile and can be modified to recognize any other license plate template. The concept employed in this device can be used not only in parking areas where there is a need for higher security such as government and certain commercial buildings, but also in toll booths, vehicle registration etc.

2. LITERATURE REVIEW

Fuzzy Logic

The suggested approach by the authors [1] is to use Fuzzy logic system for the recognition of a standard licensed number plate; and template matching system for non-standard licensed plate. It uses Malaysian number plates, which has different forms and hence the segmentation process is distinguished into two sets at the initial process. Standard plates do not have more than three characters. The system it distinguished between two types of plates, i.e. three characters or more than three characters. If it is three characters then the Fuzzy logic system is used for recognition and if more than three, then it recognizes the plate using Template matching theorem. Fuzzy input set helps the system to recognize the elements of the plate irrespective of whether the image is number or a character. Other part of the cognition process which was template matching, matches the pattern of the characters efficiently using single font and fixed sized characters. The author felt that segmentation of standard and non-standard plates helped in reducing the time by applying two separate processes of recognition algorithm.

Artificial Neural Networking

As demonstrated by Patel [2], an automated license plate cognition technique comprised of three essential steps. 1) Recognition of location of the plate mounted on the vehicle, 2) Segmentation of the image of the character and 3) Converting the image of the character into a clean identifiable image. The author suggests Artificial Neural Networks (ANN) for Optical Character Recognition (OCR). ANN is the statistical model of a real world system. It simplifies the code and improves the quality of image cognition process. To train the neutral network characters the author created different sets of images containing digits from 0 to 9 as training block. The network is executed as a library, statistically linked to the project. This isolates the neural network form the rest of the pre-processing and segmentation code. The paper suggests that in case the accuracy was a critical factor then the recognition application should use an artificial neural network having many hidden layers. It also pointed out certain restrictions on parameters like speed of the vehicle, script on the number plate, cleanliness of number plate, quality of captured image, skew in the image which can be removed by enhancing the algorithms further.

According to [3], various factors such as brightness, low light, colour, and abrupt changes within the image, makes image processing difficult. As a result, edge detection in optical character recognition can include Canny as well Sobel detection. The Sobel edge detection algorithm uses a Sobel filter which processes the input image to define only its edges and acute transitions. It also highlights the image's gradient using the intensity function. The canny edge detection algorithm is recognized by most optical edge detectors. There are three key features that stand out in this method. The first one is, it has a lower error rate. The second feature is that the regions with most edges are better defined. The third feature is, each edge has a single response. The Canny edge detector first reduces noise and then finds the gradient of the image to detect regions that are highly derived. Once this is done, it detects the region of with highest possibility of the number plate.

The author [4] points out that in spite the advanced algorithms used in ANPR, the devices result in being inefficient in real-time implementation. The reason stated for it is that, even with the high definition cameras used for the process, the computational cost for image processing is extremely high. To overcome this problem, the author suggests implementing a standard definition ANPR system on a stand-alone FPGA-based processing unit. The algorithms have been developed using MATLAB.

In a recent conference of 2016, the authors [5] proposed an ANPR system for developing countries such as India. It suggests using similar image processing algorithms as above on a Digital Signal Processor TMS320DM6437, which is optimized for video and image processing applications. MATLAB was used for the designing of the algorithms for this device. The process extracts features using sequential minimization for better accuracy in edge detection. The efficiency of this system was accounted for, by the fact that the algorithms used better minimization and segmentation techniques along with the powerful processor designed specifically for image processing.

The software and hardware for this proposed system were chosen based on their compatibility for achieving the expected results. This project requires an advanced open-source hardware and hence the Beaglebone Black has been used. The difference between most microcontrollers and this one is that it is much more superior in terms of its digital signal processing features and is a single-board computer. Beaglebone Black is a low-cost, community-supported development platform, perfect for projects. It has a processing speed of 1GHz and has a 512 MB RAM space, which makes it a very efficient platform. It can accommodate add-ons such as LCD screens, HDMI output, network cards, memory slots, USB ports, Ethernet cables, etc. It operates on the LINUX software and is comparatively low cost (Long & Kridner, 2014). The kit is compatible with operating systems such as Debian, Android, Ubuntu and many others.

Various Emulators, that can duplicate the functions of the kits operating system (OS) onto the user's laptop or desktop can be used in the system, e.g. Tenterm, Putty, WinSCP, etc. These software are open- source software. The software used to develop and use the algorithm for this project is the OpenCV software. All the above mentioned software can be downloaded from the internet and hence are free. Beaglebone Black being an open source board, has the basic functionalities of a computer and hence the entire system is small, making it portable and dynamic. In terms of the software, OpenCV is much lighter in terms of memory space compared to its counter parts used in computer vision, such as MATLAB, VIGRA, SimpleCV, etc.

3. DESIGN

Images possess important details and pictorial information. These details are key parameters, used in various applications such as data storage, remote sensing, medical imaging, human-computer interaction, machine learning, pattern recognition etc. This automatic parking management system locates the number plate and then uses character recognition to read the number and open the gate. In this project, a camera was used to capture an image of the vehicle. The device needs to extract certain features from the image, which provides it with relevant information. This process of feature extraction is known as image processing. Once the image is processed, decoding and location of the license plate number is performed. Image processing is a technique which takes an input in the form of an image and processes it to give specific parameters of the image as an output. It is done using mathematical operation and different algorithms applied over two dimensional layouts. The image processing steps that were used for number plate recognition in this device have been shown in "Fig. 1" flow diagram.

The first step is image acquisition, which means taking a digital image using camera. This image is then preprocessed to eliminate noise (converting colour to grayscale) and finding other parameters that increase the chances of detection of the number plate. After preprocessing, the image is segmented into sections for easier analysis. The segmentation process also reduces the chances of redundancy of the information present within the image.

These sections are then again narrowed down in terms of their features for better recognition. This narrowing down of the image parameters to give specific features is known as feature extraction. The last step includes classification, which assigns each element that was gained from the previous steps a specific class (in this case number plates and non-number plates).

These classes further attribute to the recognition of the license plate number.

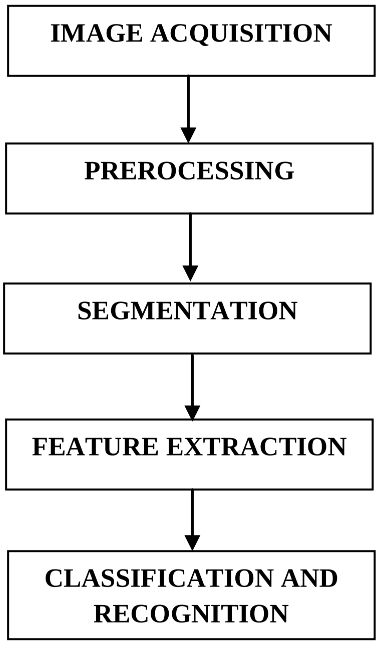


Figure 1 Image Processing Flowchart

The automated parking management system was designed using only a few components. The “fig. 2” shows the block diagram of this system.

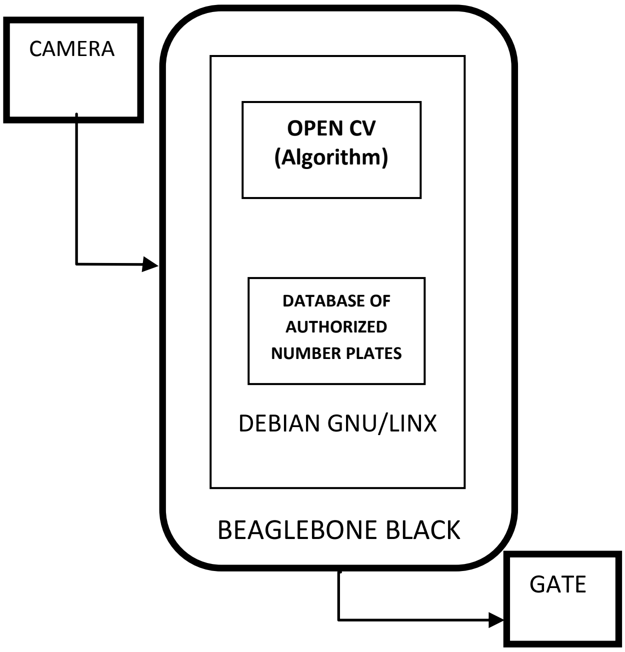


Figure 2 Block Diagram

The hardware selected for this project was the Beaglebone Black. It is a low-power open-source hardware single-board computer with DSP features. Owing to the fact that this board has a community-supported development platform, it is great for developing projects. It has various peripheral connections such HDMI, Ethernet, USB port, mountable SD card slot, input and output pins, etc. and has the basic functionality of a computer, which makes it a very versatile hardware. The operating system (OS) used on the Beaglebone Black is Debian GNU/Linux (It is a Linux based OS) developed by Canonical Ltd. The algorithms used for this project were programmed using the OpenCV software. This software is ideal for computer vision and hence is suitable for this project. It is a library of programming functions dedicated to computer vision. Computer vision deals with analyzing and understanding images. OpenCV was given preference over other software similar to it due to various reasons such as speed, versatility, small size and efficiency.

The camera captures an image of the vehicle waiting for access to the parking lot. The pre-processing steps that the image goes through are: colour to grey conversion, sobel edge detection, threshold value, morphological operation and contouring. Once these processes are carried out, the image is simplified enough for the software to extract the required information. The next step is detecting the number plate from the modified image. Once the license plate is located, the algorithm for optical character recognition identifies numbers and characters and highlights them. The license number read is then compared and verified with the database. If that number is present in the database, an output signal is sent to the gate to open it. The flow diagram shown in “fig. 3” gives a graphical representation of the steps the device carries out in one cycle of license plate recognition.

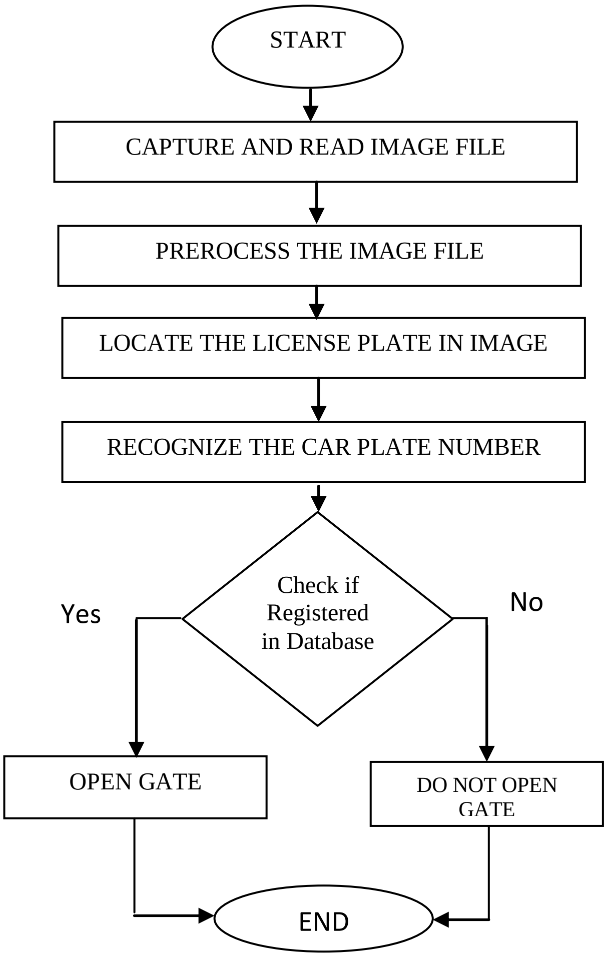

START
CAPTURE AND READ IMAGE FILE
PREROCESS THE IMAGE FILE
LOCATE THE LICENSE PLATE IN IMAGE
RECOGNIZE THE CAR PLATE NUMBER
Check if Registered in Database
Yes
No
OPEN GATE
DO NOT OPEN GATE
END

Fig 3: Process chart

4. METHODOLGY

This system requires machine training for effective pattern recognition and plate detection. To do so, sets of data classes had to be fed to the system to train it to differentiate between these sets of data (images). In this case, the images were divided into two classes; number plates and non-number plates. A total of 70 images were taken for the purpose of machine training. Apart from the images taken to train the machine, 30 more images were taken of various cars for testing the device. The algorithms used for the Support Vector Machine (SVM) training, required the aspect ratio of every image file to be 144 x 33. Hence the number plate from the main picture was cropped out and resized to the specified aspect ratio using Microsoft Paint. These resized number plate images formed one set of data class for the machine. Another set of data class consisting of non-number plates was created by cropping out different regions of the car surrounding the number plate. This was done so that the machine would know the difference between a number plate and a non-number plate. A total of 240 images were used for this set. Once the data classes were defined, they were fed to the algorithm and executed on the g++ compiler. Once the code was compiled and executed, it produced a support vector machine file written in Xml to store all the parameters via which the machine would train itself for pattern recognition.

Just as images were cropped and resized for each number plate, character images were also fed to the machine to be able to read the license number of the car. The aspect ratio chosen for character images was 20 x 20. Digits from 0 to 10 were individually cropped and resized for this step. There were about 10 to 15 digits images of each digit with varying thickness, lighting, distortion, etc. The more images the machine was fed, the better it performed. Once these images were fed to the algorithm, the code was compiled and executed using g++ compiler. The output was an Optical character recognition file written in Xml to store the parameters of the character recognition.

The codes written for this project were based on the largest sized Omani number plate which has the aspect ratio of 520 x 130 = 4. The area of error allowance around the number plate was set to 4. The Arabic script in the plate reads OMAN and is the same in all of them and therefore the Arabic script was not accounted for in this project. A code for capturing the vehicles image was written with the main code. This would implement the process in a sequential manner in one execution cycle. Starting with the capturing of the image of the car, the system would further decode it to locate and read the license number and open the gate in one go. For the final setup of the project, the entire system was put together by making all the peripheral connections. The LCD cape mounted on the Beaglebone black was connected to a keyboard and a mouse using a hub, which aided in implementing the process. The LCD displayed all the directories in the system and the final display result in a window. The camera was also connected to the hub and then further connected to the board. The camera took an image of the vehicle and then ran it through the algorithm to read and highlight the number on the license plate. It then verified it with the database stored in the system and determined whether or not it was authorized. If it was an authorized vehicle it sent a signal to the electronic gate to open.

5. RESULTS AND ANALYSIS

The Sobel detector gave fewer but faster resulting edges from the original image. When taking the images, it was found that a 1 meter distance gave the best results for plate detection. It was also taken into account that damaged number plates or any modification in its usual design returned negative results. When the plate detection was successfully achieved, the OCR algorithms were tested. It was found that even after the number plate was detected and localized, the characters that were segmented and highlighted, were read wrong. In many cases some of the numbers were often jumbled up and misread, e.g. 8 was often read as 6 and vice versa. While in most cases the reason was external, it was realized that the reason was purely a lack of sufficient training. More images of characters were fed to the system with varied images of the same character. The more images the machine was trained with the better it recognized the character.

The images below show the outputs of the ranging from edge detection to final verification and authorization.

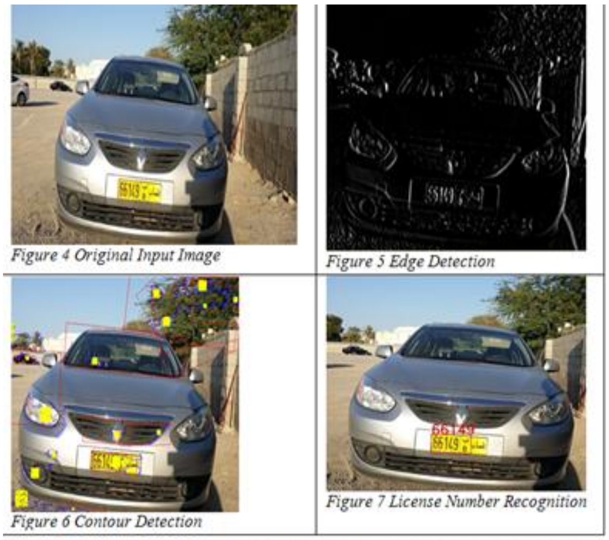
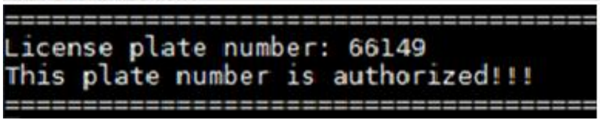


Figure 4 Original Input Image

Figure 5 Edge Detection

Figure 6 Contour Detection

Figure 7 License Number Recognition



```
==========================================
License plate number: 66149
This plate number is authorized!!!
==========================================
```

Fig 4: Example of Number Plate Recognition

The table 1 below depicts the working results of the Automated Car Parking System tested on a sample of 20 cars. The Y represents all number plates that were detected, the N represent number plates not detected due to damaged plates or bad lighting and IN represent plates that were detected with fault in some of the reading of some of the characters.

Table 1. Sample results of 20 cars

No.	Plate Number	Yes, No, Incomplete
1	24179	Y
2	1029	Y
3	90399	N
4	55658	Y

5	82493	Y
6	55658	Y
7	82493	Y
8	67490	Y
9	74486	IN
10	6721	IN
11	15171	Y
12	87295	IN
13	1896	Y
14	92061	Y
15	6868	Y
16	498	Y
17	5546	Y
18	10340	Y
19	71079	N
20	23922	Y

6. CONCLUSION

There are many automated car parking systems already available using technologies such as GSM, wireless transmitter, etc. This project was especially chosen for the purpose of learning more about image processing, as it is one of the most relevant technologies of our times and used in numerous other applications. Apart from its image processing and security feature, the core strength of the device is that it is portable. The entire setup is compact and requires only a few components to operate it. It is small in size and can be directly programmed. This means that whenever there is a change to be made, such as entering new car numbers into the system, it doesn't need to be moved from its fixed location. The LCD cape provides a monitor to make the necessary changes on the spot. The main board used is the Beaglebone Black which is a low-cost and efficient alternative for image processing techniques. OpenCV was used for the algorithms to be executed on the system. Since it is a dedicated software for computer vision, it occupies a much smaller memory space on the board and has a free distribution policy. The combination of this specific hardware and software makes the system fast, compact and portable. In conclusion, this project was developed to be implemented in real-time and to evaluate image processing algorithms. The technique of license plate reading is not just limited to car parking, but can also be used in various other applications wherever there is a need for character recognition, e.g. toll booths, car registration applications, traffic control etc.

7. ACKNOWLEDGEMENT

The authors would like to thank Caledonian College of Engineering and its staff for all the resources provided. For the purpose of this project, images of cars were captured from the Caledonian College of engineering parking lot. These images were only taken for the purpose of testing and will not be publicly distributed or published.

8. REFERENCES

[1] W. Al Faqheri and S. Mashohor, "A Real-Time Malaysian Automatic License Plate Recognition (M-ALPR) using Hybrid Fuzzy", *International Journal of Computer Science and Network Security*, vol. 9, no. 2, pp. 333-340, 2009.

[2] S. G. Patel, "Vehicle License Plate Recognition Using Morphology and Neural Network", *IJCI*, vol. 2, no. 1, pp. 1-7, 2013.

[3] B. Al Taan, "Vehicle Registration Checking System",unpublished.

[4] Zhai, X. and Bensaali, F., 2013, November. Standard Definition ANPR System on FPGA and an Approach to Extend it to HD. In *GCC Conference and Exhibition (GCC), 2013 7th IEEE* (pp. 214-219). IEEE.

[5] Kalaiselvi, K., Selvakani, S., Iyan, P.S. and Ananth, C., 2016. ANPR for Developing Countries

[6] D. Baggio, *Mastering Open CV with practical computer vision projects*. Biringham: Packt Publishing, 2012.R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[7] I. Supriana and A. Nasution, "Arabic Character Recognition System Development", *Procedia Technology*, vol. 11, pp. 334-341, 2013.

[8] S. Saha, S. Basu, M. Nasipuri and D. Basu, "Localization of License Plates from Surveillance Camera Images: A Color Feature Based ANN Approach", *International Journal of Computer Applications*, vol. 1, no. 23, pp. 27-31, 2010.

[9] D. Molloy, "Capturing Images and Video on the Beaglebone | derekmolloy.ie", *derekmolloy.ie*, 2015. [Online]. Available:

[10] http://derekmolloy.ie/beaglebone-images-video-and-opencv/. [Accessed: 14- Nov- 2015].

[11] Coley, G., 2013. Beaglebone black system reference manual. *Texas Instruments, Dallas*.

[12] Bradski, G. and Kaehler, A., 2008. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.".

[13] Bezdek, J.C., Keller, J., Krisnapuram, R. and Pal, N., 2006. *Fuzzy models and algorithms for pattern recognition and image processing* (Vol. 4). Springer Science & Business Media.