Tesseract-OCR Based Automatic NPR System on Raspberry Pi

Sanjay.S.Chopade[1] Sagar P. Gharte[2] Deepak Y. Chauhan[3] Swapnil M. Agale[4] Pranav B. Lute[5]
[1]Professor [2,3,4,5]B.E. Student
[1,2,3,4,5]Department of Electronics & Telecommunication Engineering
[1]Sandip Institute of Technology & Research Centre, Nashik
[2,3,4,5]Savitribai Phule Pune University, Pune, India

Abstract— Automatic number plate recognition (NPR) system plays an important role in numerous applications, such as parking a system, traffic law enforcement, road monitor, electronic-police system, and security systems. NPR involves three main components: Number plate detection, character segmentation and Optical Character Recognition (OCR). For Number Plate Detection and character segmentation, we are going to use OpenCV libraries. And for OCR, Tesseract-OCR. There are numerous NPR systems available today. These systems are based on various methodologies but perhaps it is really tricky job as some of the factors like fast speed of vehicle, inappropriate vehicle number plate, language of motor vehicle number and different light conditions has significant overload on the overall recognition rate. Most of the systems work under these limitations.

Key words: NPR, Open CV, PS3 Camera, Python, Raspberry-Pi, Tesseract

I. Introduction

The scientific world is doing research in intelligent traffic systems which have a significant impact on peoples lives. Number plate recognition system is a computer vision technology to extract the license number of vehicles from images. It is an real time system which has large applications and challenges. Typical ANPR systems are implemented using many technologies and hence are expensive. This closed approach also stops further research and development of the system. With the rise of free and open source technologies the computer world is lifted to new heights. People from variuos communities communicate in enlightening environment to develop solutions for manhood problems. One of the notable contribution of the open source community to the computer world is Python. Intel's researches in Computer Vision bore the fruit called Open Computer Vision (OpenCV) library, which can be used for computer vision development.

In India, basically, there are two types of number plates, black characters in white plate and black characters in yellow plate. The former for indivisual vehicles and latter for public service vehicles. The system tries to address these two categories of plates[1].Conventional NPR algorithms are generally divided in 5 steps:

1) Capture.
2) Preprocess.
3) Licence Plate Extractor.
4) Character Segmentation.
5) Optical Character Recognition.

The image of the number plate is captured using a high resolution pictorial camera shown in Fig.1. A better choice is an Infrared (IR) camera. The camera may be adjusted with respect to the number plates.

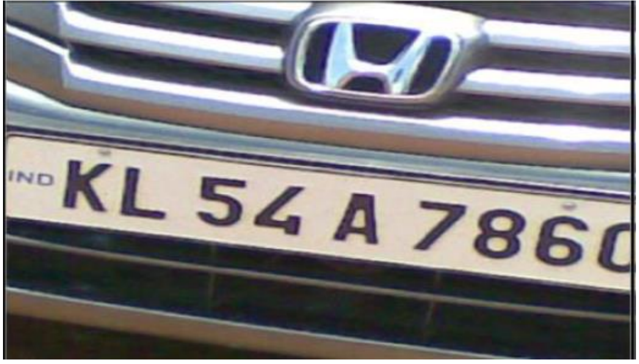


Fig.1: Number Plate with acceptable resolution

Filtering is the set of algorithms applied on the image for better quality. It is an important part of any computer vision system. For the present system filtering consists 2 processes: Resize – The picture size from the camera may be large and can make the system stumpy. It is to be resized to a probable aspect ratio. Convert Color Space – Pictures captured using IR or pictorial cameras will be either in untreated format or prearranged into some multimedia standards. Normally, these images will be in RGB mode, with three components (viz. red, green and blue).

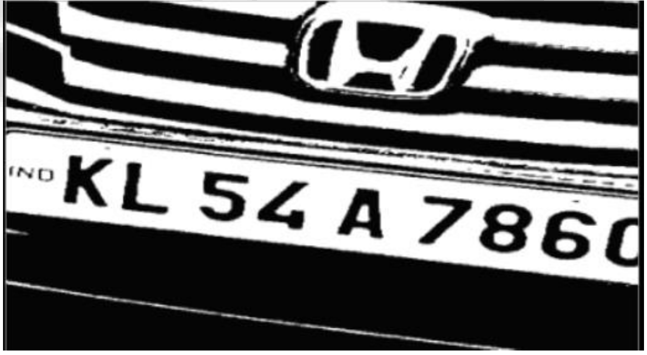


Fig. 2: Image converted in RGB mode

In image handing out techniques, Haar-like features are worn to be familiar with objects from image . If upcoming system is selected to detect only number plates then the Haar-like features are used for this intention and no supplementary processing is done. This technique is aged and backbreaking and more over needs a big database to accumulate the collected samples almost about 10000 images of the plates and characters.Fig.2 shows the filtered image. In the other case, if our projected system has to distinguish number plates, then the binary representation is formed from the image.

In character segmentation image processing is done on extracted number plate to eliminate redundant data[2].Fig. After character segmentation, the extracted number plate has only those characters that fit in number plate. This also accomplished with the size elevation ratios identical with the contours detected on extracted number plate.

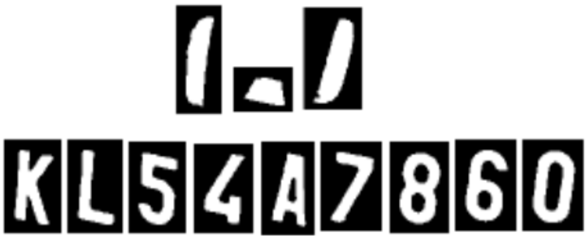


Fig. 3: Character Segmentation

Finally, the chosen bloks are send to a Optical Character Recognition (OCR) Engine, which proceeds the ASCII of the number plate.Currently number plate detection and recognition dispensation time is fewer than 50 ms in several systems. Accurateness of Character Recognition depends on how Number Plate Detection and Character Segmentation are locating vehicle number plate and divide each character. The recognition of a single number plate and the recognition of its font in a reliable way is an pricey task, since it depends on computation concentrated algorithms. Devoted systems have been developed for this reason delivering the necessary computational power. Typical NPR systems are implemented using proprietary technologies. This congested approach also stops extra research and growth of the system. The increase of free (accessible to everyone) and unlocked source technologies the enumerate world is wide open to new heights.

Previous work done on character recognition is shown in section I. There are some issues in Character Recognition and some of these issues are described in section II. Flowchart on Automatic NPR system is described in section III. Algorithm for the same is described in section IV. Section V and VI illustrates methodology and results of Automatic NPR system whereas conclusions are mentioned in last section.

II. System Development

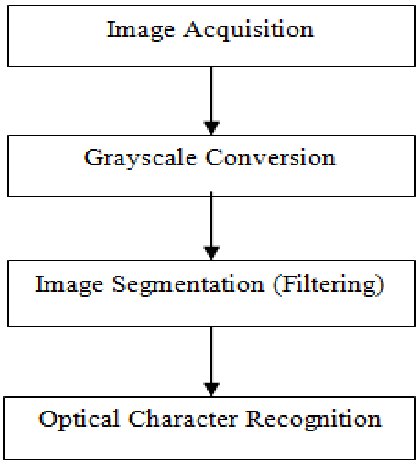


Fig. 4: Dataflow of proposed work

A. Image Acquisition

This is main component of NPR system. In this component following steps are performed:

1) Image Capturing
2) Image Color Conversion

B. Gray Scale Conversion

A gray scale or gray scale digital image is an image in which the value of every pixel is a single sample. These Images also known as black-and-white are made of totally of shades of gray, changeable from black at the weakest strength to white at the strongest intensity. Black-and-white images or gray scale are different from black and white metaphors, which in the perspective of computer imaging are images with no more than the two colors. Gray scale images have large number of shades of gray in between. Gray scale images are frequently the result of measuring the intensity of light at every pixel in a single band of the EM spectrum (e.g. infrared, detectable light, ultraviolet,) and in these cases they are monochromatic proper when only a specified frequency is captured or taken[3]. But also they can be synthesized from a full color image. This is most critical process in Number Plate Recognition System.

C. Character Segmentation

In this part further image processing is done on extracted number plate to remove unnecessary data. After character segmentation, the extracted number plate has only those characters that belong to license number[5]. This also achieved by means of the width height ratios matching with the contours detected on extracted number plate. After character segmentation, the extracted number plate has only those characters that belong number[2].

D. Optical Character Recognition

OCR is utilized to extract characters from an image or a scanned document. These characters are used for additional processing such as it can be edited, formatted, searched, indexed and automatically translated or converted to speech[4]. In this part, character segmented number plate is passed to optical character recognition engine.OCR engine returns number plate in text format.

Tesseract-ocr is an open source engine. It is written using C/C++ and developed at Hewlett-Packard (HP) lab from 1985 and 1996. They didn't utilized it in their products. It was one of the best engines in the 1995 UNLV Accuracy test. In 2005, it was released as open source engine. From 2007, Google has started the supervising tesseract-ocr for further development and maintenance. It takes gray or color image as input and gives output in text format. IN early stages, only .tiff type of image was used to support but now it also supports additional types of images like .png .jpg, etc. It is capable of reading data in any language from image like English, Swedish, Danish, Chinese etc. and developers or users can train their own language if the support for a particular language is not offered[5]. Google Developer experienced it on Windows and Ubuntu operating system but it can also installed on other Mac and Linux platforms, etc. We can also utilized this library on mobile platforms like Android and iPhone etc.

III. Flowchart

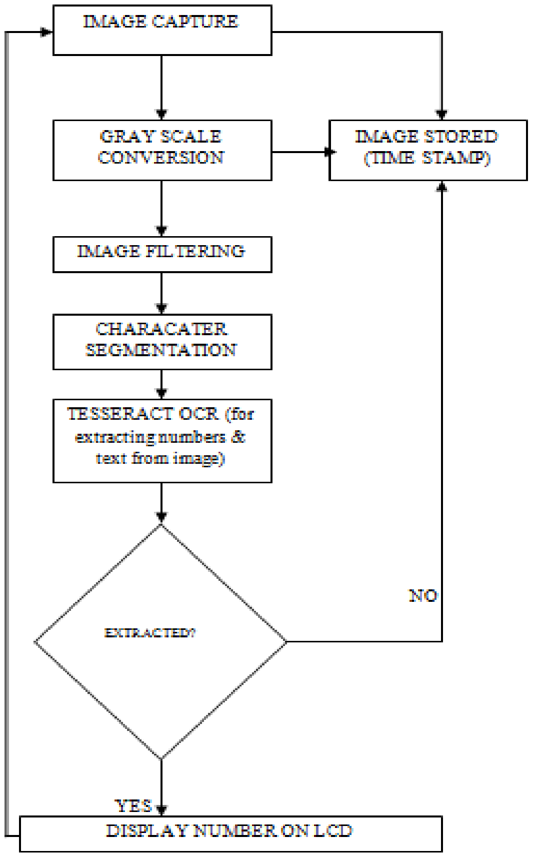


Fig. 5: Flowchart

IV. Algorithm

1) Capture image with camera using python script.
2) Converting image to grayscale and saving the captured Image as a time/date stamp.
3) Filtering and denoising of image.
4) Applying Tesseract OCR on filtered image.
5) Displaying extracted numbers and text on 16x2LCD display.

V. Methodology

A. Capture The Frame

First we import cv2 module then initialized a camera. Now we initialize the camera capture object with the cv2.VideoCapture class. All it requires is the index to camera port. After entering capture.py command image is captured.Fig.6 shows the captured image.

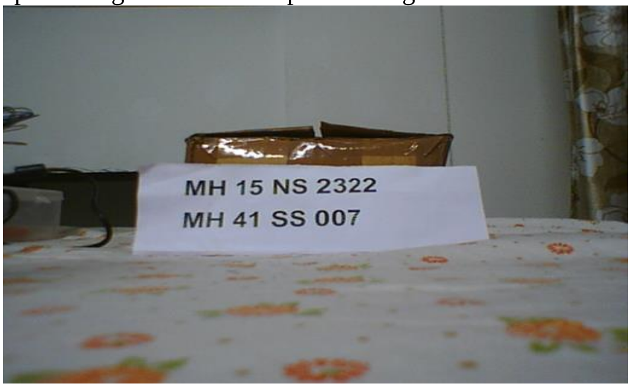


Fig. 6: Captured image

B. Grayscale Conversion of Color Image

First we import the cv2 module by using following function: import cv2.After that,read the image to a variable named “image”: image = cv2.imread ('frame.png')[7].Images are converted into Grayscale. Now, to store it to another variable named “gray_image” use the function cv2.cvtColor () alongside with parameters arranged as the “image” variableand“cv2.COLOR_BGR2GRAY”:gray_image=cv2.cvtColor(image, cv2.COLOR_BGR2GRAY).So,the variable “gray_image” will hold the gray transformation of the input image. Now, for writing/saving the converted gray-scale image to the hard disk, “cv2.imwrite ()” and the variable “gray_image” to which the transformed image is stored:cv2.imwrite ('gray_image.png', gray_image).This image is shown in Fig.7. Now, to display the original and the gray-scale, we utilize function “cv2.imshow ()” with parameters similar to the “windowtitle”.The“imagevariable”:cv2.imshow('color_image',image) cv2.imshow ('gray_image', gray_image).

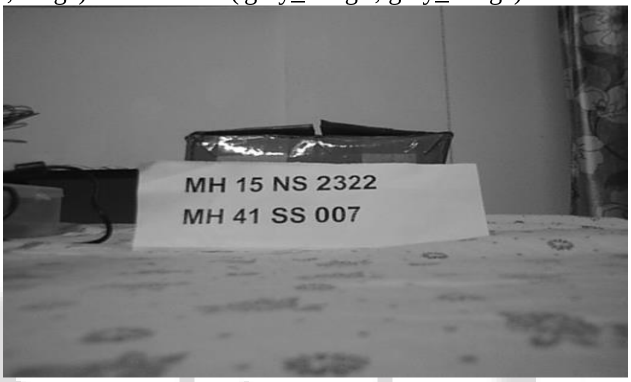


Fig. 7: Captured image converted into grayscale

C. Tessaract OCR

It consists of three parts: OCR software often "pre-processes" images to recover the odds of successful recognition. These are:

1) Binarization

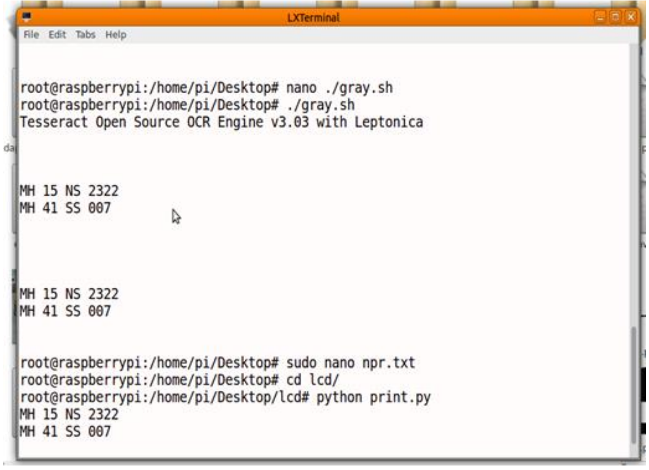
Image is converted from color to black and white . The task of binarization is performed as a easy way of unscrambling the text (or any other preferred image part) from the background[6]. The task of binarization itself is essential as most commercial recognition algorithms work only on binary images as it proves to be simpler.The convenience of the binarisation step decreases to a significant amount the quality of the character acknowledgment stage and the cautious decisions are made in the option of the binarization employed for a given input picture type; since the class of the binarization method employed to achieve the binary result depends on the type of the input picture (scanned paper, scene text image, historical degraded document etc.).

2) Character Recognition

Matrix matching consists of comparing an image to a stored set of symbols on a pixel-by-pixel base; it is also called as "prototype matching", "prototype recognition", or "image correlation". This depends on the input set of symbols being properly separated from picture, and on the stored set of symbols being in a similar font and at the similar scale. This technique works best with text and does not work fine when new fonts are occurred.

3) Post Processing

Tesseract utilizes its dictionary to control the character segmentation step, for enhanced accuracy. The yielded stream may be a plain text or file of typescript, but more complicated OCR systems can protect the original layout of the page and created.Fig.8 shows extracted numbers.



```
root@raspberrypi:/home/pi/Desktop# nano ./gray.sh
root@raspberrypi:/home/pi/Desktop# ./gray.sh
Tesseract Open Source OCR Engine v3.03 with Leptonica


MH 15 NS 2322
MH 41 SS 007



MH 15 NS 2322
MH 41 SS 007


root@raspberrypi:/home/pi/Desktop# sudo nano npr.txt
root@raspberrypi:/home/pi/Desktop# cd lcd/
root@raspberrypi:/home/pi/Desktop/lcd# python print.py
MH 15 NS 2322
MH 41 SS 007
```

Fig. 8: Extracted Numbers using Tesseract

VI. RESULT AND DISCUSSION

Distance from camera (ft.)	Accuracy (%)
1	100
2	100
3	98
4	90
5	70
6	35
7	20
8	15

Table 1. System accuracy readings with respect to distance

Eight images were captured using PS3 camera. These images were captured at different distances (feet's). Readings are shown in table 1.PS3 camera comes with resolution of 640*480 pixel having frame rate of 120 frames per seconds. This camera can capture images at wide angle (150 degree).Captured image is first converted into grayscale. Further filters like Guassianblur,median and bilateral filter, image segmenting filters like thresholding ,adaptive mean thresholding and binary mean thresholding are applied on image to segment captured image.After segmentation Tesseract OCR is applied for character recognition and numbers and texts are extracted from captured image.From graph we can conclude thatupto 3 feet accuracy is 100%. As distance increases accuracy goes on decreasing. Also camera pixel count play important role in this system i.e. more the pixel better accuracy will be obtained.

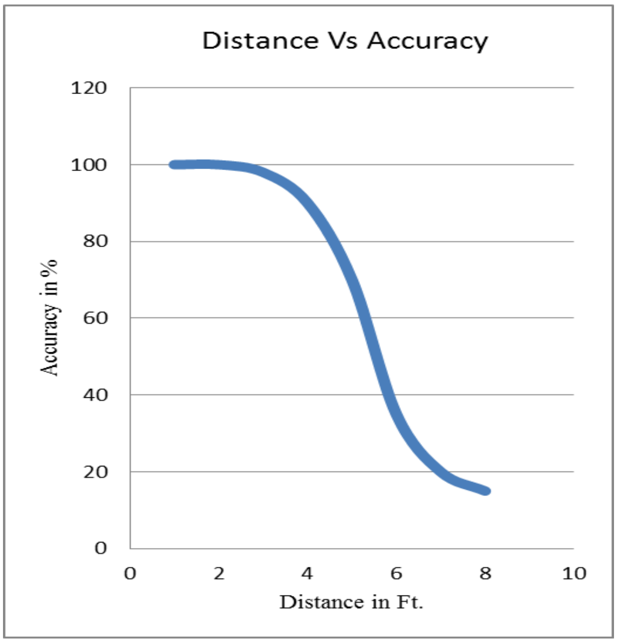


Fig. 9: System Accuracy with respect to Distance

VII. CONCLUSION

A system is designed to manage traffic on road, parking system, it is often time consuming typing in even so short alphanumeric combinations as number plate registration numbers and time is a factor, we wantto improve this limitation by utilizing existing detection and recognition technique by adapting them to new technique. Implemented real time detection and recognition system which grants the fast and correct processing of number plate information without the need of given input and which is running under the limited constraints (processing power and space) of Raspberry pi.With the help of Raspberry pi we can make this system small and portable.Also we are using open source tools like Python and Opencv having advantages over other proprietary paid tools like Matlab.This system does not give 100% accuracy as camera connected to Raspberry Pi gives accuracy about 70%.Thus overall accuracy of this system is about 60-70%.Due to open source nature of our system it will be easy for future development in this area by other researchers and developers.

REFERENCES

[1] Pratiksha Jain, Neha Chopra, Vaishali Gupta "Automatic License Plate Recognition using OpenCV" Department of CSE IGIT, GGSIPU New Delhi, India International Journal of Computer Applications Technology and Research Volume 3– Issue 12, 756 - 761, 2014.

[2] Siqi Chen, Daniel Cremers, and Richard J. Radke, "Image segmentation with one shape prior - A template-based formulation" Volume 30, December 2012.

[3] B.SC. Thesis Ondrej Martinsky "Algorithmic And Mathematical Principles Of Automatic Number Plate Recognition Systems" BRNO 2007.

[4] [Smith R, "An Overview of the Tesseract OCR Engine," in IEEE Ninth Intenational Conference Proceeding of Document anay and Recognition, 2007.

[5] Chirag Patel, Dipti Shah, Atul Patel "Automatic Number Plate Recognition System (ANPR".

International Journal of Computer Applications (0975 – 8887) Volume 69– No.9, May 2013.

[6] Kumar Parasuraman and P.Vasantha Kumar, "An Efficient Method for Indian Vehicle License Plate Extraction and Character Segmentation", IEEE International Conference on Computational Intelligence and Computing Research 2010.

[7] Dr. Khalid Nazim S. A., Mr. Adarsh N.:-"A Robust Automated Process for Vehicle Number Plate Recognition" ,Volume. 5 (3) , 2014.

[8] Ahmad Gull Liaqat."Mobile Real-Time License Plate Recognition" Linnaeus University 2011-12-05.

[9] [K.M Sajjad, "ALPR Using Python and Open CV" Dept. Of CSE, M.E.S College of Engineering Kuttipuram, Kerala.2008-06-21.

[10] Nicole Ketelaars "Final Project : ALPR", 2007- 12-11

[11] Roy and D.P Ghoshal, "Number Plate Recognition for use in different countries using an improved segmenation," in 2nd National Conference on Emerging Trends and Applications in Computer Science(NCETACS), 2011.

[12] Anju K Sadasivana, T.Senthilkumarb "Automatic Character Recognition in Complex Images" International Conference on Communication Technology and System Design 2011.

[13] A Conci, J. E. R. de Carvalho, T. W. Rauber, " A Complete System for Vehicle Plate Localization, Segmentation and Recognition in Real Life Scene" , IEEE Latin America Transaction, Volume. 7, No. 5, September 2009.