International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-05, Issue-03, June 2019

Raspberry PI Based OCR System For Vehicle

[*]Piyush D. Umbare, [#]Santosh G. Bari, [$]Priti J. Rajput

[*]Student, [#,$]Professor, E&TC (Signal Processing), D. Y. Patil school of engineering academy, Ambi, Talegaon, Pune, Maharashtra, India.

[*]piyushumbare225@gmail.com, [#]santosh.bari@dyptc.edu.in, [$]priti.rajput@dyptc.edu.in

Abstract -**As the evolution of technology increases with upcoming new technology, the Digital image processing, i.e. the use of computer systems to process pictures, has applications in many fields, including of medicine, space exploration, geology and oceanography and continues to increase in its applicability. The main objective of this paper is to demonstrate the ability of image processing algorithms on a small computing platform. Specifically, we created a road sign recognition system based on an embedded system that reads and recognizes speed signs. The paper describes the characteristics of speed signs, requirements and difficulties behind implementing a real-time base system with embedded system, and how to deal with numbers using image processing techniques based on shape and dimension analysis. The paper also shows the techniques used for classification and recognition. Color analysis also plays a specifically important role in many other different applications for road sign detection, this paper points to many problems regarding stability of color detection due to daylight conditions, so absence of color model can led a better solution. In this project lightweight techniques were mainly used due to limitation of real- time based application and Raspberry Pi capabilities. Raspberry Pi is the main target for the implementation, as it provides an interface between sensors, data base, and image processing results.**

Keywords — Digital Image Processing, Raspberry Pi, Embedded System, OCR, and Vehicles Speed

I. INTRODUCTION

The field of traffic sign recognition is not very old with the first paper on the topic published in Japan in 1984 when the aim was to try computer vision methods for the detection of objects. Since then however, the field has continued to expand at an increasing rate. Traffic sign recognition is used to maintain traffic signs, warns the distracted driver, and prevent his/her actions that can lead an accident.

A real-time automatic speed sign detection and recognition can help the driver, significantly increasing his/her safety. Traffic sign recognition also gets an immense interest lately by large scale companies such as Google, Apple and Volkswagen etc. driven by the market needs for intelligent applications such as autonomous driving, driver assistance systems (ADAS), mobile mapping, Mobile-eye, Apple, etc. and datasets such as Belgian, German mobile mapping. Methods of recognizing and detecting traffic signs continue to be published as the number of systems and tools to interpret images increases across multiple platforms.

Our worked focused on a low cost, off the shelf solution, specifically, a mini embedded computer Raspberry Pi that is capable of doing everything you would expect a desktop computer to do, from word processing, image processing to playing games. The system has originally been developed by Raspberry Pi Foundation in an effort to give young people an easy solution to learn coding and computer programming. Raspberry Pi Foundation was founded in 2009, by a game developer David Braben, and supported by a tech firm, Broadcom, and the University of Cambridge Computer Labs. In order to provide fast processed results, this project aimed to demonstrate use of the simple shape recognition algorithms and open source optical character recognition (Tesseract OCR) on Raspberry Pi. Tesseract OCR is an open source optical character recognition module for various operating systems. And its development supported by Google since 2006. Tesseract OCR is one of the top character recognition engines in terms of accuracy. Tesseract can detect letters in various forms of images, and it uses the open source C library Leptonica library. In this project we will be able to pass images to Tesseract OCR and read them. To improve accuracy we had to do pre-processing on images before pass them Tesseract OCR engine. The system can be scaled down to improve the conditions of highly automated driving systems.

 DOI : 10.35291/2454-9150.2019.0194

II. PROPOSED METHOD

2.1 System Requirements:

To design a good recognition system, the system needs to

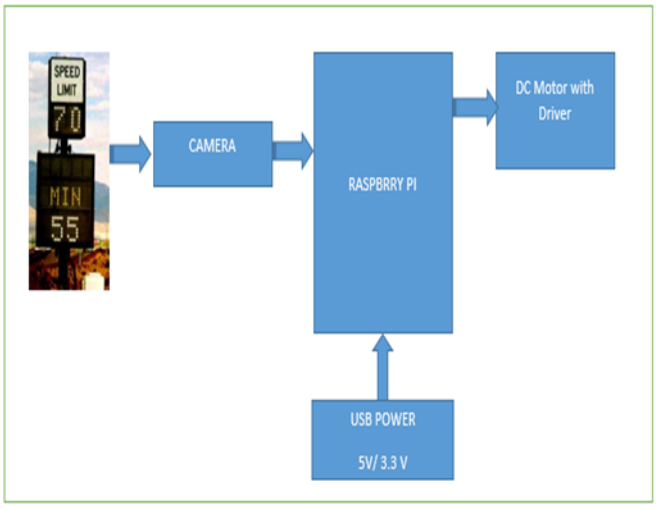

Figure 1: System design-Block diagram

have a good discriminative power and a low computational cost The system should be robust to the changes in the geometry of sign (such as vertical or horizontal orientation) and to image noise in general. Next the recognition should be started quickly in order to keep the balanced flow in the pipeline of Raspberry Pi allowing for processing of data in real time. Finally, the optical character recognition engine must be able to interpret a pre-processed image into a text file. The general diagram of the proposed method is shown in Figure 1.

The identification of the speed signs is achieved by two main stages: detection and recognition. In the detection phase the image is pre-processed, enhanced, and segmented according to sign properties such as colour, shape, and dimension. The output of segmented image contains potential regions, which can be recognized as possible speed signs. The effectiveness and speed are the important factors throughout the whole process, because capturing images from the video port of Raspberry Pi and processing images as they come into to the pipe should be synchronized.

In the recognition stage, each of the images is tested with K-Nearest algorithm according to their dimensions, which is an important factor to detect the speed signs, since we want to process images only once as they come; it also emphasizes the differences among the other rectangle shapes. The shape (rectangle) of the signs plays a central role in this stage. When a speed sign detected, it is passed into the optical character recognition engine to be converted and written out as a text file. The GPS module is used to add to the results with time and location. To retrieve GPS data and write those into text file another open source platform Arduino-Uno used. This platform consists of both a physical programmable circuit board (micro-controller) and its integrated development environment (IDE). Arduino was used for this project because, unlike most complex programmable circuit boards, the Arduino does not need a separate piece of hardware to program instead, the only thing it needs a USB cable to load new code onto the board.

2.2 Capturing Images:

A raspberry Pi is capable of capturing a sequence of images rapidly by utilizing its video- capture port with JPEG encoder.

However several issues need to be considered:

- The video-port capture is only capturing when the video is recording, meaning that images may not be in a desired resolution/size all the time (distorted, rotated, blurred etc.).
- The JPEG encoded captured images do not have exit information (no coordinates, time, not exchangeable).
- The video-port captured images are usually “more fragmented” than the still port capture images, so before we go through pre-processed images we may need to apply more de-noising algorithms.
- All capture methods found in OpenCV (capture, capture continuous, capture sequence) has to be considered according their use and abilities. In this project, the capture sequence method was chosen, as it is the fastest method by far. Using the capture sequence method our Raspberry Pi camera is able to capture images in rate of 20fps at a 640×480 resolution. One of the major issues with the Raspberry Pi when capturing images rapidly is bandwidth.

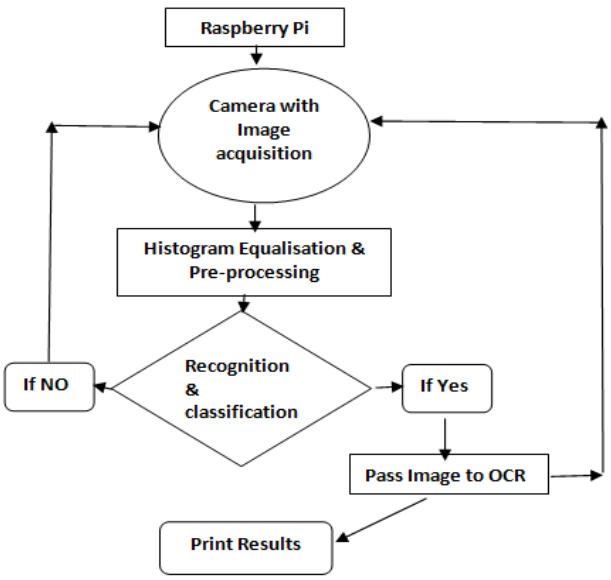

Figure 2: The Speed Sign Detection & Recognition Flow Chart

 DOI : 10.35291/2454-9150.2019.0194

The I/O bandwidth of Raspberry Pi is very limited, and the format we are pulling pictures makes the process even less efficient. In addition, if the SD card size is not large enough, the card will not be able to hold all pictures that are being captured by camera port, leading to cache exhaustion.

2.3 Detecting Speed Signs:

When we look at the pictures of speed signs, the most defining feature of a speed sign is rectangular shape with mostly round edges. Before finding the rectangles in a captured image, we retrieve the contours, thus the shape detection algorithm employed loops through a subset of shape detection is based on the OpenCV's Python implementation preceded by filtering and edge detection. To prevent the noise from being mistaken as edges, and produce wrong results, the noise must be reduced to certain level. Thus, the images are smoothed by applying Gaussian Filter. In a two dimensional space an isotropic Gaussian form F(x, y) equals to: Using the capture sequence method.

$$F(x,y)=\frac{1}{(2\Pi\sigma^2)}e^{\left(\frac{-x^2+y^2}{2\sigma^2}\right)}$$

2.4 Open CV Contour Features and Edge Detection

The purpose of the edge detection is to significantly reduce the amount of data in the image by converting the mage into binary format. Even though canny edge detection is quite an old method, it's become one of the standard edge detection algorithms. In doing image processing and especially shape analysis it is often required to check the objects shape, and depending on it perform further processing of a particular object. In our application it is important to find the rectangles in each of frames as these may potentially correspond to road speed signs. This shape detection must be done at least once in every 40 frames to ensure close to real processing. Once we check all the contours retrieved, we should look for closed loops then that closed loop should meet the following conditions to become a rectangle.

Contour approximation is an important step for finding desired rectangles, since because of distortions other issues in the image perfect rectangle may not be visible. The contour approximation might be in this case better choice for finding convex hulls. After this step we can approximate the rectangular shape in the captured image as shown in Figure 3. Once the individual regions for the signs are identified, they are rotated to fit a common alignment and then OCR is performed. To align the signs that first we find 4 max and min points in a rectangular shape.

III. RECOGNISING SPEED SIGNS

3.1 k-Nearest Neighbour Algorithm (kNN):

The kNN algorithm is a non-parametric, basic algorithm that does not make assumptions on the underlying data Distribution such as Gaussian mixtures etc.

Figure 3: Bounding rectangles for a set of speed signs

For kNN it is assumed that the given data is in a feature space (geometrical metric space) and the points are in a 2D geometrical space, where they have a notion of distance. Each of the training data consists of a set of vectors and classes assigned to each vector. k stands for number of neighbours that influences the classification and is usually an odd number.

3.2 Open Source Optical Character Recognition Engine:

In order to read the speed signs accurately, Tesseract Optical Character Recognition (OCR) is chosen for the project. Tesseract OCR is and open source engine started as PhD research project at the HP Labs, Bristol. After a joint project with HP Bristol Labs and HP scanner division, Tesseract OCR was shown to outperform most commercial OCR engines. The processing within Tesseract OCR needs a traditional step-by-step pipeline. In the first step the Connected Component Analysis is applied, and the outlines of the components stored. This step is particularly computational intensive, however it brings number of advantages such as being able to read reversed text, recognizing easily on black text on white background. After this stage the outlines and the regions analyzed as blobs.

The text lines are broken into characters cells with the algorithm nested in the OCR for spacing. Next, the recognition phase is set two parts. Each word is passed to an adaptive classifier, and the adaptive classifier recognizes the text more accurately. Adaptive classifier has been suggested for use of OCR engines in deciding whether the font is character or non-character. Like most of the OCR engines Tesseract does not employ a template (static) classifier. The biggest difference between a template classifier and an adaptive classifier is, that adaptive classifiers use baseline x-height normalization whereas static classifiers find positions of characters based on size normalization. The

 DOI : 10.35291/2454-9150.2019.0194

baseline x-height recognition allows for more precise detection and recognition of upper case, lower case characters and digits, but it requires more computational power in return.

To improve quality of results, the images ideally should be given to the OCR module in form of clear black text and white background. By default, Tesseract OCR applies Otsu's thresholding method to every image; however since we have our custom pre-processing algorithm, this step was bypassed in order to improve speed. To disable internal thresholding of Tesseract OCR; the tesseract delegate option should be set as "self" (tesseract. Delegate = self**).**

IV. CONCLUSION

The work described in this paper is splits into two parts, similar to other applications in the field, as "detection" and recognition". For the detection part, shape-based algorithms were used because colour-based segmentation is much less reliable than shape-based segmentation. After the detection of speed sign Motor Will be controlled.

Due to limited computation power of Raspberry Pi, complex techniques were not chosen despite their availability within the OpenCV libraries (such as Eigen faces, SURF-SIFT template matching, and Fuzzy matching). In order to keeps Raspberry Pi running smoothly, other classifiers like k nearest neighbour, and Euclidian distance were chosen for this project. Speed sign detection in different conditions like lighting, disorientation was not tested well.

By using a pre-built OCR engine, a detailed study of recognition techniques is outside the scope of this paper. Detection, tracking and recognition are interwoven, while recognition reduces false positives and detection narrows the choices and increases accuracy of a system. Keeping the results is another important part; as a result, it needs to be improved as more databases are being developed by the time.

This project's implementation focused on real-time video processing, however, for future work, the use of car's dynamics (direction, trajectory, speed changes etc.) should be considered to improve the system's robustness of the speed sign reading process.

A comparison of the performance within an embedded system of this project will provide the baseline of the improvements. However, the lack of an open source evaluation framework for similar systems (datasets for speed signs, labelled data etc.) makes it hard to perform that comparison. Our work supports claims that the complexity of traffic sign recognition systems will continue to be reduced in the near future as embedded technology advances.

V. ACKNOWLEDGEMENT

We (authors) wish to thank the management of the institute and head of the department for providing the facilities to carry out the work.

REFERENCES

[1] Vengadesh, K. Sekar, "Automatic Speed Control Of Vehicle In Restricted Areas Using RF And GSM", International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056,Volume: 02 Issue: 09,Dec-2015

[2] Vishal Pande, Malhar Mohite, Supriya Mhatre, Siddhesh Desai, Anjali Kumari, "Autonomous Speed Control of Over Speeding Vehicles Using Radio Frequency", IJAREEIE, Vol. 4, Issue 4, April 2015.

[3] K. Govindaraju, S. Boopathi, F. Parvez Ahmed, S. Thulasi Ram, M. Jagadeeshraja, "Embedded Based Vehicle Speed Control System Using Wireless Technology", IJIREEICE, Vol. 2, Issue 8, August 2014.

[4] Gurjashan Singh Pannu, Mohammad Dawud Ansari, Pritha Gupta, "Design and Implementation of Autonomous Car using Raspberry Pi", International Journal of Computer Applications (0975–8887), Volume 113 – No. 9, March 2015.

[5] R. Mithe, S. Indalkar and N. Divekar. "Optical Character Recognition" International Journal of Recent Technology and Engineering (IJRTE)", ISSN: 2277-3878, Volume-2, Issue-1, March 2013.

[6] P. Sai Chaitanya, V. Vikram, B. Kalesh, "Automatic Vehicle Speed Control System Using Wireless fidelity", International Journal of Advance Electrical and Electronics Engineering (IJAEEE), Volume-3 Issue-4, 2014.

[7] K Nirmala Kumari, Meghana Reddy, "Image Text to Speech Conversion Using OCR Technique in Raspberry Pi", IJAREEIE, Vol. 5, Issue 5, May 2016.

[8] Rohit Tiwari, Dushyant Kumar Singh, "Vehicle Control Using Raspberry Pi and Image Processing", Innovative Systems Design and Engineering, [ISSN 2222-1727 (Paper), ISSN 2222-2871 (Online)], Vol.8, No.2,2017.

 DOI : 10.35291/2454-9150.2019.0194