# *Detection and Recognition of Traffic Signs based on RGB to RED Conversion*

*Mr.Mohit Bhairav Mahatme*
Electronics and Telecommunication Department
Goa College of Engineering
Farmagudi, Goa, India- 403401
mohitm2602@gmail.com

*Mrs.Sonia Kuwelkar*
Assistant Professor
Electronics and Telecommunication Department
Goa College of Engineering
Farmagudi, Goa, India- 403401
sonia@gec.ac.in

*Abstract*— **This paper presents a different approach for detecting and recognizing traffic signs. Traffic signs are designed such that they can be easily observed by humans because of their shape and shading which are not precisely the same as surrounding environment. Traffic sign recognition is a fundamental component in autonomous cars and it can similarly be utilized as an integral part of driver assistance frameworks. The system is designed to perform two tasks, traffic sign detection and recognition. The algorithms demonstrated include RGB to Red conversion for getting the red component in the image, filters which perform the task of noise reduction as well as edge detection, thresholding and segmentation to get the desired traffic sign from image and then an artificial neural network for recognition. The neural network used here is a Single Layer Perceptron neural network. Implementation of the system has been successfully done in Matlab and it works quite productively and efficiently with the database of Indian traffic signs created as a part of the project.**

***Keywords—Traffic sign recognition; RGB to Red conversion; Single Layer Perceptron; image processing;***

## I. Introduction

Traffic sign detection and recognition progressively has been a vital issue for self-governing vehicles and driver help frameworks. Traffic signs have two roles to accomplish; the first is to manage the traffic and, second is to show the condition of the road, controlling and cautioning drivers and pedestrians. Because of the presence of complex environment, such as climatic conditions, illumination and geometric distortions, the Traffic Sign Recognition system has dependably been a challenging task.

The task of Traffic Sign Recognition involves two main phases, traffic sign detection and classification. In the detection phase, the main challenge is finding a way to take out all the non-sign objects in the picture while holding the signs in the picture such that their position and size does not get corrupted. The task of recognition phase is to take these detected signs and recognize them keeping in mind the end goal to give relevant information to the user[1]. A number of algorithms have been proposed for detecting and identifying traffic signs to assist the driver and furthermore avert accidents. However, an algorithm that accomplishes high precision as well as quick execution speed is required.

The detection stage comprises of different procedures which wipe out the non-sign objects while retaining the road signs in the picture. Detection algorithms usually utilize shape or color as a strategy for deciding the regions of interest in the picture. Color based strategy is a clear comparison by searching particular hues in a picture, yet challenged by illumination and sign condition. The shape based technique are invariant to illumination and sign condition, additionally challenged by signs blocked in part or signs situated in background with comparative shading which influences the color based strategy too[2]. Once the traffic signs are detected, recognition is achieved by employing Support Vector Machines or neural network[3],[4] or template matching[5]. Other than sign identification performance, real-time operation is another challenge for the traffic sign recognition system. Only a framework which can recognize signs sufficiently quick to inform the driver can be viewed as valuable for driver assistance[2].

This paper aims to demonstrate a system which employs algorithms that are very much different from the standard color segmentation or shape detection algorithms. The algorithms used achieve quicker execution speed and higher accuracy. In this system, the picture is acquired initially and then it is pre-processed to convert it from RGB to a Red segmented image. Filtering is then carried out inorder to remove the noise and for edge detection by utilizing non-linear filters. Then it undergoes thresholding and segmentation. To recognize the traffic sign in the image, the Single layer perceptron neural network is employed.

## II. Related work

In [6] an algorithm was developed for detecting and identifying the road signs in Vietnam with real-time processing capability and high accuracy. The color model that has been used for color segmentation is HSV (Hue-Saturation-Value)[7]. Incase of outdoor images, the color is sensitive to variation in lightning[6]. Hue has been employed to segment the color regions in conventional methods. But, sometimes when Value is low or high, Hue becomes meaningless. So, most researchers use Saturation and Value to find out the chromatic zone for red.

In [6] recognition is done using an SVM in OpenCV[8] software. This system has been designed to run on a FriendlyARM Tiny4412 board. The system employed Multi-threading method for quad-core ARM Cortex-A9 processor on FriendlyARM board inorder to enhance the real-time capability of the system[6].

[9] employs Color-based Method with CIELab + hue as better results on localizing traffic signs are obtained with it. It first preprocesses the picture to give a binary picture. Inorder to get more accurate result on detecting traffic signs, the binary picture is processed with canny. The preprocessed picture is then checked by using Ramer-Douglas-Peucker algorithm to get the road sign shape. To detect occluded and attached signs, the method proposed in [9] uses the shape-arc algorithm. Approximated circle, square, or triangle images are detected as traffic signs and processed with linear c-SVM[9].

The techniques utilized in [2] include hue detection for road sign detection, morphological filters for noise reduction, and Hausdorff distance calculation for template recognition. Zynq-7000 FPGA processing system and custom IP cores are employed together in the hardware platform presented[2].

The system presented in [10] consists of three steps: 1) segmentation in accordance with the color of the pixel; 2) road sign detection by classifying shapes and 3) recognizing the content. The color segmentation scheme used here is Hue Saturation Intensity (HSI). Shape classification is performed by using linear SVMs and recognition phase uses SVMs with Gaussian kernels. Each color and shape classification uses a different SVM[10].

## III. Proposed Algorithm

The task of Traffic Sign Recognition consists of two fundamental phases, detection and recognition. The processes performed in the detection phase include acquiring the image, pre-processing the image which is followed by thresholding and segmentation. The overview of Traffic Sign Recognition system is shown in Fig. 1.

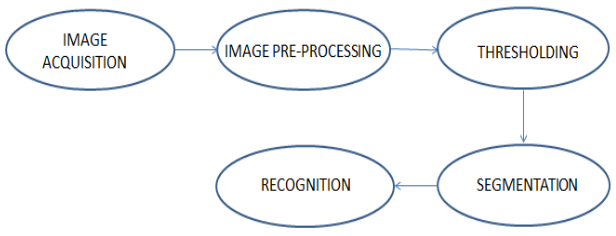


Fig. 1 Overview of Traffic Sign Recognition System

The picture is acquired initially and then it is pre-processed. Then it undergoes thresholding and segmentation. Then to recognize the traffic sign in the image, the Single layer perceptron neural network is employed.

### A. *Image Pre-processing*

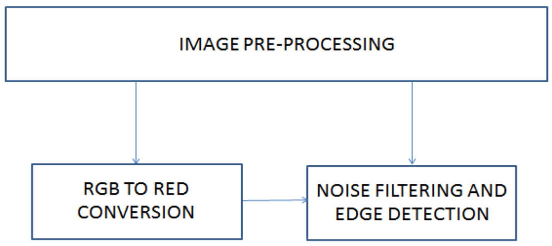


Fig. 2 Image pre-processing

The acquired picture is first converted from RGB to Red. Then filtering is carried out inorder to remove the noise and for edge detection by utilizing non-linear filters and a blurring mask.

.

#### 1) *RGB to Red Conversion*

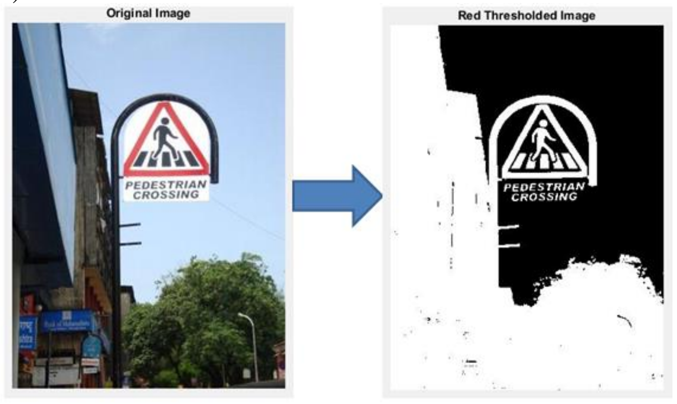


Fig. 3 RGB to Red Component converted image

Every picture consists of three components, Red, Green and Blue which are often abbreviated as RGB. For the purpose of our project, we show more interest in the Red color present in the image because traffic signs have a dominant Red component. So we convert the image from RGB to Red by setting a threshold value in every pixel of the image for Red color present in it and then selecting only those pixels where red component is dominant and exceeds the threshold value.

#### 2) Noise Filtering and Edge Detection

Filtering is a technique used to adjust or improve a picture. You can filter a picture to underscore certain elements or remove certain components. Image processing operations executed with filtering incorporate smoothing, sharpening, and edge enhancement[11].

The median filter is a nonlinear digital filtering technique which is used to remove noise. This type of noise reduction is a step of the pre-processing phase which improves the results of later processing[12].

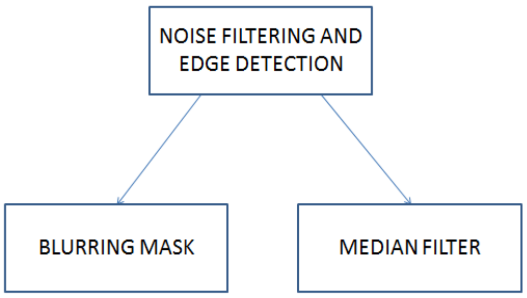


Fig. 4 Noise filtering and edge detection

It removes the noise in the image but preserves the edges[12]. Therefore it plays a crucial role in detecting the edges. The median is computed by first arranging all the pixel values in ascending order, and then replacing the pixel value being considered with the middle or intermediate pixel value[13].

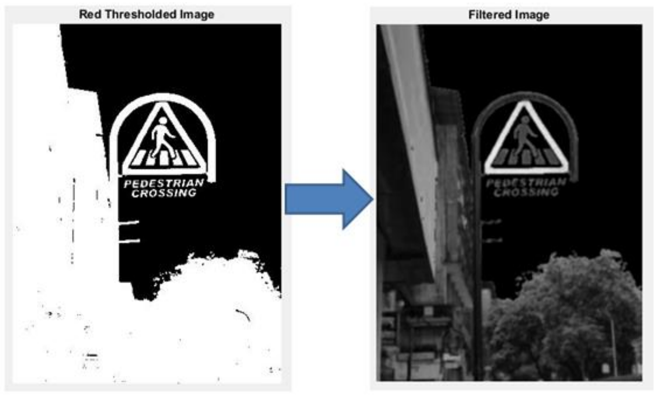


Fig. 5 Noise filtered and Edge detected image

The blurring mask or the blurring filter blurs the background objects in the picture and also normalizes the pixel values. As a result, the edges become sharper.

This mask is employed for detecting the edges along with the median filter which does the task of edge preserving as well as reduces the noise.

## B. Thresholding

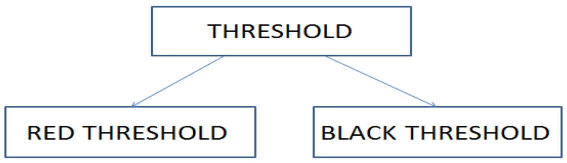


Fig. 6 Thresholding

Thresholding is used to convert a grayscale image to a binary image[14]. When we first convert the image from RGB to Red, we set a threshold value for Red component present in every pixel, and choose only those pixels where red component is dominant and exceeds the threshold value. Red threshold is used in this case.

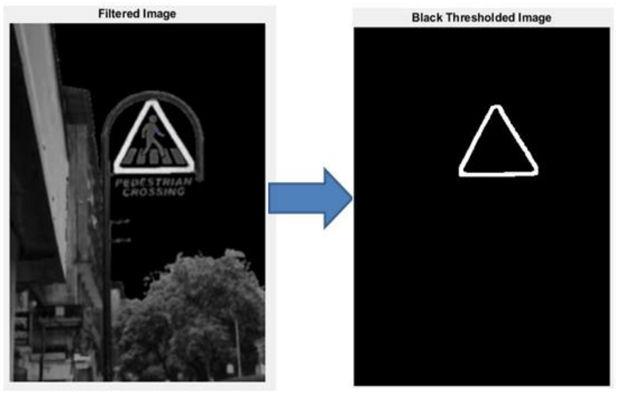


Fig. 7 Thresholded Image Output

Once filtering and edge detecting is performed, thresholding is done to convert the image to a binary image where every pixel having intensity above the threshold value will be replaced by a white pixel and a pixel having intensity below the threshold value will be replaced by a black pixel. Black threshold is used in this case. Hence we get a thresholded binary image at the end of thresholding phase, with our region of interest depicted by white pixels, which is then extracted out by performing segmentation.

## C. Segmentation

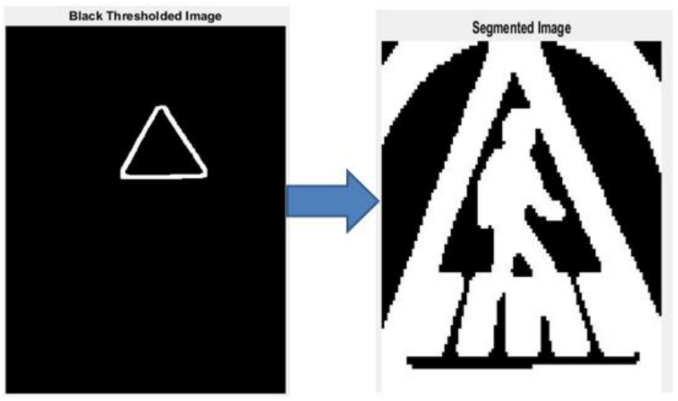


Fig. 8 Segmented image

To get the desired part of the picture containing the traffic sign, segmentation is performed. The technique of segmentation that we have utilized is somewhat different from the usual segmentation technique.

Usually for segmentation, scanning is done only in one direction and once a white pixel is encountered, certain pixels around that white pixel are extracted to yield the segmentation output. But in this case, the thresholded image is scanned in all the four directions to find the first white pixel in each direction.

Then we draw a box taking the four points discovered from the scans performed and dot multiply the contents within the box with the red thresholded image. This results in reconstruction of the pixels within the box. This is done to get the exact road sign from the image preventing chances of other objects being included or incomplete traffic sign being included at the output of the segmentation phase. This is the method of segmentation that is followed.

### *D. Recognition*

Recognition is carried out, once the road signs are detected. This is usually performed by using Neural Networks or Support Vector Machines or template matching. The technique used for recognition in this system is a Single layer perceptron network.

The Perceptron is an algorithm for learning a binary classifier. It is a function which maps its input x which is a real-valued vector to an output f(x) which is a single binary value[15].

$$f(x) = \begin{cases} 1 & \text{if } w \cdot x + b > 0 \\ 0 & \text{otherwise} \end{cases}$$

where w is a vector of real-valued weights, w.x is the dot product where

$$\sum_{i=1}^{m} w_i x_i,$$

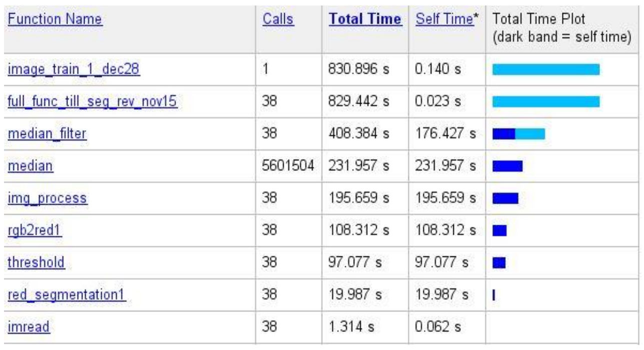
m is the number of inputs to the perceptron and b is the bias[15].

To differentiate it from a multilayer perceptron, which is a more complicated neural network, the perceptron algorithm is also called as the single-layer perceptron. Single-layer perceptron is the simplest feedforward neural network as a linear classifier[15].

## IV. Results

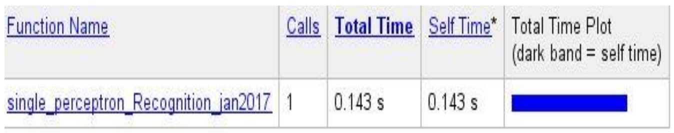
The proposed system for detecting and recognizing Indian Traffic signs based on RGB to Red conversion and Single Layer Perceptron neural network was designed and implemented successfully. Training of the network is required to be done only once when the system is designed and then it recognizes any number of test images fed to the system. Detection accuracy of 93.25% was observed. The network requires a time of 829.644secs to be trained over the database which is shown in Fig. 10.

A test image fed to the system is recognized in 0.143secs as shown in Fig. 11.



| Function Name | Calls | Total Time | Self Time* | Total Time Plot (dark band = self time) |
|---|---|---|---|---|
| image_train_1_dec28 | 1 | 830.896 s | 0.140 s | ▬ |
| full_func_till_seg_rev_nov15 | 38 | 829.442 s | 0.023 s | ▬ |
| median_filter | 38 | 408.384 s | 176.427 s | ▬ |
| median | 5601504 | 231.957 s | 231.957 s | ▬ |
| img_process | 38 | 195.659 s | 195.659 s | ▬ |
| rgb2red1 | 38 | 108.312 s | 108.312 s | ▪ |
| threshold | 38 | 97.077 s | 97.077 s | ▪ |
| red_segmentation1 | 38 | 19.987 s | 19.987 s | ▏ |
| imread | 38 | 1.314 s | 0.062 s | |

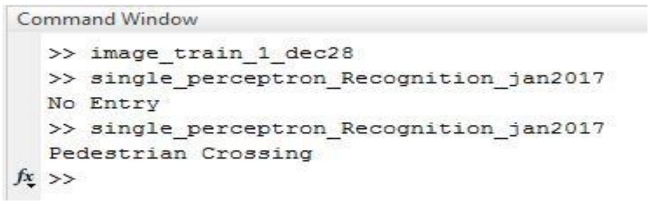
Fig. 10 Time required for training the network



| Function Name | Calls | Total Time | Self Time* | Total Time Plot (dark band = self time) |
|---|---|---|---|---|
| single_perceptron_Recognition_jan2017 | 1 | 0.143 s | 0.143 s | ▬ |

Fig. 11 Time for Recognizing test image

Fig. 12 shows the output of the recognition phase which identifies the traffic sign.



```
Command Window
  >> image_train_1_dec28
  >> single_perceptron_Recognition_jan2017
  No Entry
  >> single_perceptron_Recognition_jan2017
  Pedestrian Crossing
fx >>
```

Fig. 12 Output of the Recognition phase

## V. Conclusion

The proposed system for detecting and recognizing Indian traffic signs was implemented successfully with high accuracy over a database of Indian traffic signs created as a part of the project. Due to the use of RGB to RED color segmentation, fewer amounts of data were required to be handled and processed which resulted in a much higher computation speed and also a high detection accuracy of 93.25% was obtained. This system could be later implemented on a parallel processing hardware such as a GPU or an FPGA for high speed real-time implementation.

## References

[1] M. Russell and S. Fischaber, "OpenCV Based Road Sign Recognition on Zynq," in 11th IEEE Int. Conf. on Ind. Inform., Bochum, July 2013.

[2] Y. Han and E. Oruklu, "Real-Time Traffic Sign Recognition Based on Zynq FPGA and ARM SoCs," IEEE Int. Conf. on Electro/Informative Technology, Milwaukee, June 2014.

[3] A. d. l. Escalera, L. E. Moreno, M. A. Salichs and J. M. Armingol, "Road Traffic Sign Detection and Classification," IEEE Trans. Ind. Electron., vol. 44, no. 6, pp. 848–859, 1997.

[4] A. Broggi, P. Cerri, P. Medici, P. P. Porta and G. Ghisio, "Real Time Road Signs Recognition," in Proc. IEEE Intell. Veh. Symp., Istanbul, June 2007, pp. 981–986.

[5] S. Waite and E. Oruklu, "FPGA-Based Traffic Sign Recognition for Advanced Driver Assistance Systems," J. Transp. Technologies, vol. 3, no. 1, pp. 1–16, 2013.

[6] T. Q. Vinh, "Real-Time Traffic Sign Detection and Recognition System Based on FriendlyARM Tiny4412 Board," in Int. Conf. on Commun., Manage. and Telecomun., DaNang, Dec. 2015.

[7] E. Oruklu, D. Pesty, J. Neveux, and J.E. Guebey, "Real-Time Traffic Sign Detection and Recognition for In-Car Driver Assistance Systems," IEEE 55th Int. Midwest Symp. on Circuits and Syst., Boise, Aug. 2012.

[8] P. Shopa, N. Sumitha, Dr. P.S.K Patra, "Traffic Sign Detection and Recognition Using OpenCV", Int. Conf. on Inform. Commun. and Embedded Syst., Chennai, Feb. 2014.

[9] D. Soendoro and I. Supriana, “Traffic sign recognition with Color-based Method, shape-arc estimation and SVM,” in Int. Conf. Elect. Eng. and Informatics, Bandung, July 2010, pp. 1–6.

[10] S. Maldonado-Bascón, S. Lafuente-Arroyo, P. Gil-Jiménez, H. Gómez-Moreno and F. López-Ferreras, “Road-Sign Detection and Recognition Based on Support Vector Machines,” IEEE Trans. on Intell. Transp. Syst., vol. 8, no. 2, pp. 264–278, June 2007.

[11] Mathworks, “What Is Image Filtering in the Spatial Domain?,” [Online]. Available:https://in.mathworks.com/help/images/what-is-image-filtering-in-the-spatial-domain.html?requestedDomain=www.mathworks.com [Accessed: May. 27, 2017].

[12] Wikipedia, “Median filter,” [Online]. Available: https://en.wikipedia.org/wiki/Median_filter [Accessed: May. 27, 2017].

[13] The University of Auckland, “Image Filtering Median Filtering,” [Online]. Available: https://www.cs.auckland.ac.nz/courses/compsci373s1c/PatricesLectures/Image%20Filtering_2up.pdf [Accessed: May. 27, 2017].

[14] Wikiwand, “Thresholding (image processing),” [Online]. Available: http://www.wikiwand.com/en/Thresholding_(image_processing) [Accessed: May. 27, 2017].

[15] Wikipedia, “Perceptron” [Online]. Available: https://en.wikipedia.org/wiki/Perceptron [Accessed: May. 27, 2017].