ECE 513 Project: Traffic Sign Classification

Ram Rohit Gannavarapu

14th May 2020

1 Introduction

Automation is playing a very keen role in human life especially in the 21st century. One such field which is quickly evolving is Autonomous Cars, as of today a lot of work and research is being done on various pursuits to make automobiles more intelligent, safer and more convenient [1]. Advanced driver assistance systems or ADAS plays a big role in implementing safety and better driving. Safety is given the topmost priority and the systems are designed to avoid collisions and accidents by alerting the drivers or taking over control of the vehicle. Road sign recognition is an important subpart of ADAS.

2 Theory

There has been a lot of research on road sign recognition, different researchers have approached to the problem with different methods and constraints. Road sign detection methods mostly depend on the nature of data available. Vision Systems have been largely incorporated using visual sensing devices in detection, classification, tracking and recognition. But according to my research I found that most of the work published in the area of road and traffic sign recognition has been separated in two different phases, detection and classification.

The detection of road signs are mainly done in two ways such that the characteristics of road signs (RS) e.g. shapes and colours of the sign could be exploited. During this stage the region of interest is highlighted. As we have studied, some papers have only used colour segmentation, while some have used shape as the main detection criterion while some papers have used both of these techniques in parallel. The next step after detecting a road sign is to classify it i.e. to identify what that sign board actually signifies. To do that we have used a Machine learning model to classify various sign boards.

Algorithm 1 Traffic Sign Recognition

```
Load the Image
Convert Image from RGB to HSV
Create Mask based on the Threshold
Remove areas smaller than 100pixels - using bwareaopen(MATLAB)
Smooth the borders of the mask - using imclose (MATLAB)
Fill in holes as they also have same color -using imfill (MATLAB)
Extract the image based on the mask
Find Various shapes in the masked image
Displaying Region of interest based on the detected shape.
```

2.1 Road Sign Detection

Detecting the roads signs is the first step in any traffic sign recognition system. To be able to get the exact location or the Region of interest in a image where the traffic sign is present plays a crucial role as this region of interest is fed to the machine learning model to predict the traffic sign. The detection is divided into two parts Color based and Shape based detection which are further discussed in the next sections.

2.1.1 Colour based Detection

Image segmentation is a process which assigns a label to each pixel of an image so that the pixels with the same labels share similar visual characteristics. The simplest method of image segmentation is thresholding: every pixel with a value above a certain threshold is marked with the appropriate label. Also in the mid-1970s researchers in computer graphics developed the HSL and HSV colour models, which rearrange the RGB colour space in cylindrical coordinates so that the resulting representation is closer to human visual perception. A very similar model is HSI, commonly used in computer vision. HSL, HSV and HSI differ only in the definition of the third component – L stands for lightness, V for value and I for intensity. The first two components are hue and saturation. Color threshold is one of the fastest methods to extract specific color pixels from an image.

Figure 1: Original Image

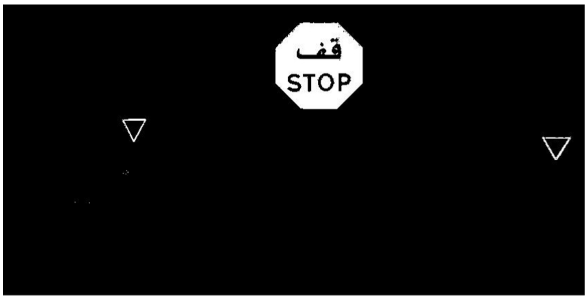

Figure 2: Mask Image after Thresholding

2.1.2 Shape based Detection

Many researchers have used shape identification methods to categorize the road signs as colour based detection has its limitation, the main being that colour vary with daylight and their reflection property can change making hard for system to interpret. More over proper care of road signs is not taken due to which the colour erodes away and thus they become unrecognisable. Hence shape detection comes to the rescue, below are some of the techniques which we studied from different papers.

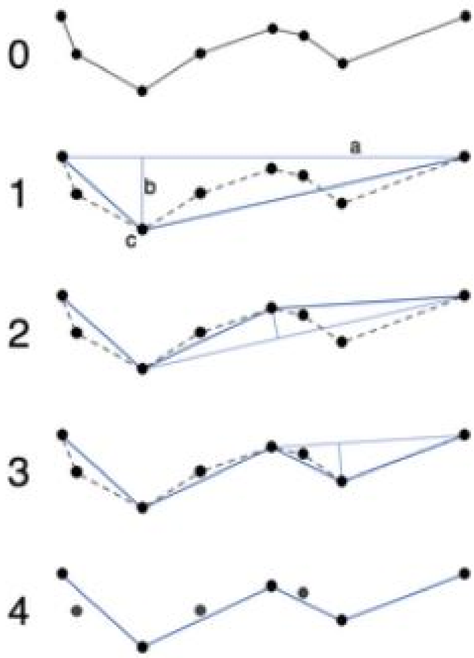

Figure 3: Ramer-Douglas-Pecker shape Approximation

To detect the shape of the object, contour approximation technique is used. Contour is a curve joining all the continuous points (along the boundary), having same colour or intensity. Contour approximation is performed for reducing the number of points along a curve into reduced set of points. This algorithm is commonly known as Ramer-Douglas-Pecker algorithm or split and merge algorithm. The algorithm uses recursion methods to approximate a curve by making a series of small line segments. As a result approximated curve having a subset of the original curve is made. Figure 3-0 shows the input curve, in the first step it marks the first and the last point by measuring maximum distance of contour, it then finds the point which is farthest from the line segment connecting first and last point. (Distance b in figure 8-1) This approach is followed recursively until the whole curve is linearly approximated and any point less than (epsilon distance) is discarded. Figure 3-4 shows the contour approximated curve of Figure 3-0.

Algorithm 2 Ramer-Douglas-Peucker algorithm

1: Take one base edge. (The one with the most extreme X or Y value - A).
2: Take the farthest edge from A(N)
3: Make line from A to N.
4: Take farthest orthogonal distance edge from line AN
5: Is M to AN ¿ epsilon?
6: If true, make to lines AM and MN, repeat to 1.
7: Else, all edges can be simplified to line AN

MATLAB's reducepoly function was used for reducing the number points to represent a polygon.[5]

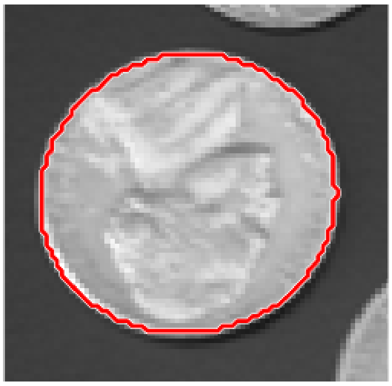

Figure 4: Mask Outline in Red[5]

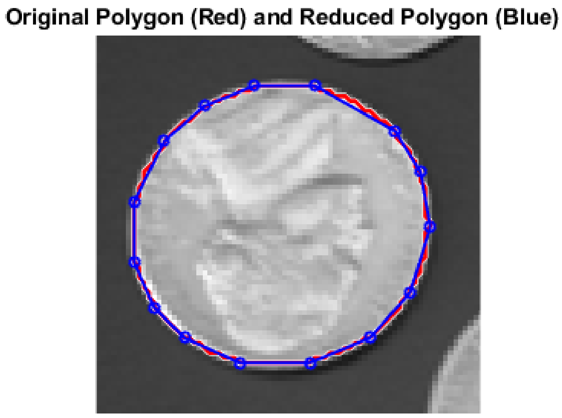

Figure 5: Reduced polygon using Ramer–Douglas–Peucker algorithm[5]

The shape is then estimated based on the geometric properties of the mask. As most traffic signs have a circular or octagonal shape and there are exception as we also have sign boards in square and triangular shapes. the regionprops function in MATLAB gives us details about various geometric properties of the mask. We can make predictions based on multiple factors. One such method was to find the compactness or also called as circularity of the mask i.e how close is it to the shape of a circle.

$$Compactness = \frac{4 * \pi * Area}{Perimeter^2} \tag{1}$$

elongation is another factor by which we can estimate the shape of an arbitrary object.it is the ratio between the length and width of the object bounding box[9]. If the ratio is equal to 1, the object is mostly square or circularly shaped. As the ratio decreases from 1, the object becomes more elongated[9]:

$$Elongation = \frac{Width_{bounding}}{Length_{bounding}} \tag{2}$$

2.2 Road Sign Classification

To Classify the traffic signs detected from the above steps I have used a Machine Learning algorithm, which have high accuracy in classifying. The German Traffic Sign Dataset[2] was used in this project to train the machine learning model.The data from the German Traffic Sign Dataset has real life images of 50,000 trafffic signs with 43 different classes. The Image sizes vary between 15x15 to 250x250 pixels and are not always in a square format. The actual traffic sign is also not necessarily centered within the image. Annotation is provided for each image in which a bounding box arround the traffic sign are provided.

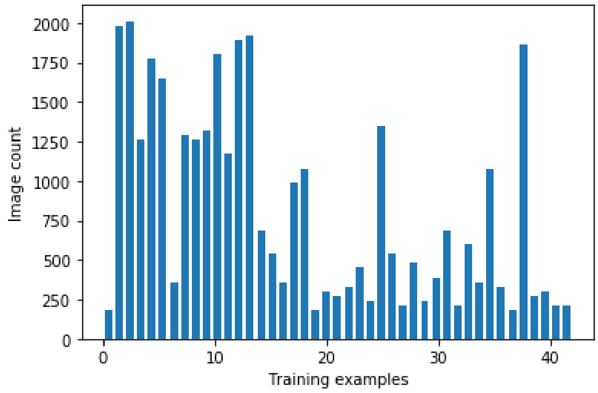

Figure 6: Training Images Distribution

Figure 7: Sample Images from Training set

As you can observe from the sample images that there are a lot of imperfections in those images. for example the keep right symbol in Figure 7 is not well illuminated. Data pre-processing plays a vital role in Machine learning as the quality of the data directly affects the ability of a model to learn. The following image processing steps are applied to the input images so that the machine learning model achieves better accuracy.

1. **Gray-scaling**: P. Sermanet and Y. LeCun in their paper [3] stated that Convolutional Neural Networks accuracy increases by using grayscale images instead of color images.

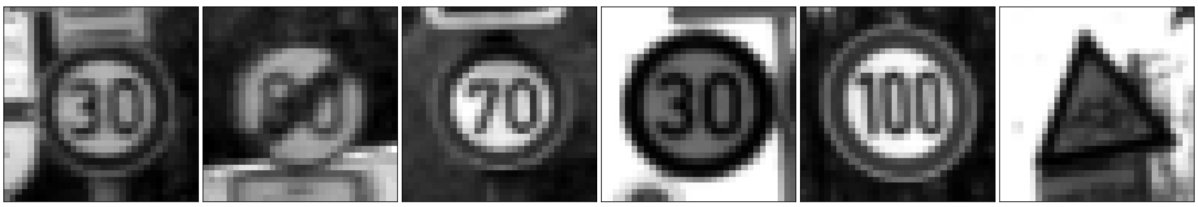

Figure 8: Images after Gray Scaling

2. **Histogram Equalization**: This technique is a widely used image contrast enhancement method. It spreads out the most frequent intensity values in an image, which enhances the images with low contrast. The current data set has a lot of real world images and have many images with low contrast and by applying local histogram equalization we will be able to improve the quality of the images.

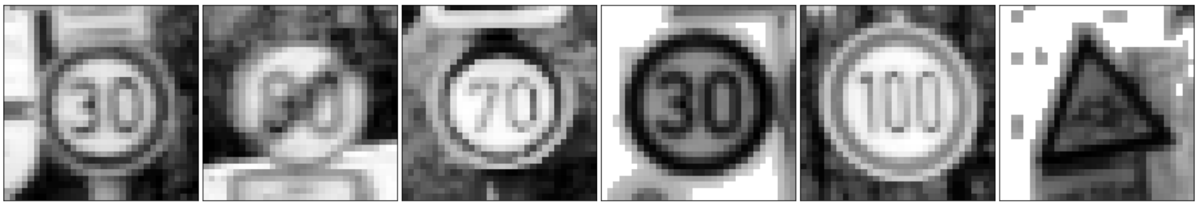

Figure 9: Images after histogram equalization

3. **Normalization**: Normalization is a process that changes the range of pixel intensity values. Usually the image data should be normalized so that the data has mean zero and equal variance.

VGGNet was proposed by K. Simonyan and A. Zisserman from the University of Oxford in 2014 [4]. VGGNet has a better accuracy and it has many successful real world implementations. Their main contribution from this paper was that, Evaluation of networks of increasing depth using an architecture with very small (3x3) convolution filters, which shows that a significant improvement on the previously designed configurations can be achieved by pushing the depth to 16-19 weight layers.[4]

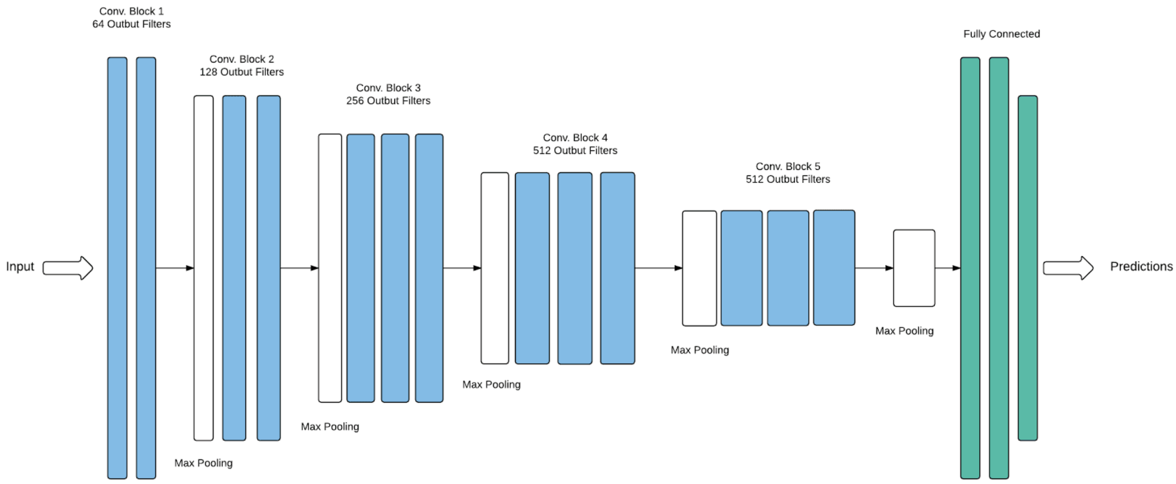

Figure 10: VGG Net Architecture

The original VGGNet architecture has 16-19 layers, In the current implementation we have excluded few layers and implemented a modified version with only 12 layers so that it can smoothly run on my system.

This Convolutional Network has the following layers:

Input$\Rightarrow$ $Convolution \Rightarrow ReLU \Rightarrow Convolution \Rightarrow ReLU \Rightarrow Pooling \Rightarrow Convolution \Rightarrow ReLU \Rightarrow Convolution \Rightarrow ReLU \Rightarrow Pooling \Rightarrow Convolution \Rightarrow ReLU \Rightarrow Convolution \Rightarrow ReLU \Rightarrow Pooling \Rightarrow FullyConnected \Rightarrow ReLU \Rightarrow FullyConnected \Rightarrow ReLU \Rightarrow FullyConnected$

3 Results

3.1 Road Sign Detection

3.1.1 Test Image 1

Figure 11: Original Image

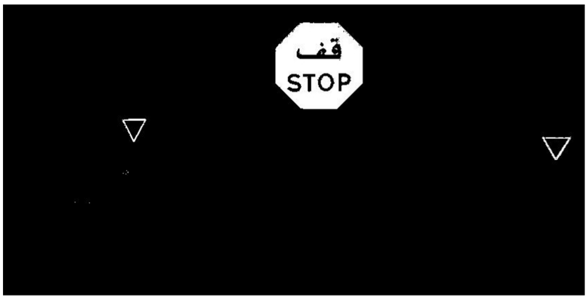

Figure 12: Color Threshold Output(db)

Figure 13: Removed objects smaller than 100 px

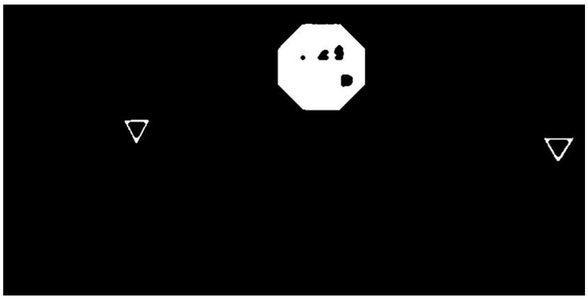

Figure 14: smoothing the borders

.

Figure 15: Fill in holes

Figure 16: smoothing the borders

Figure 17: Fill in holes

3.1.2 Test Image 2

Figure 18: Original Image

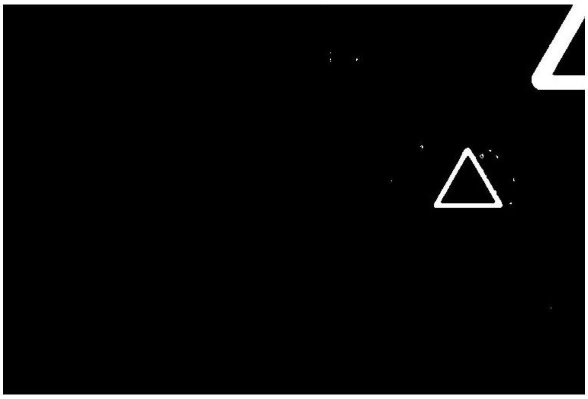

Figure 19: Color Threshold Output(db)

Figure 20: Removed objects smaller than 100 px

Figure 21: smoothing the borders

.

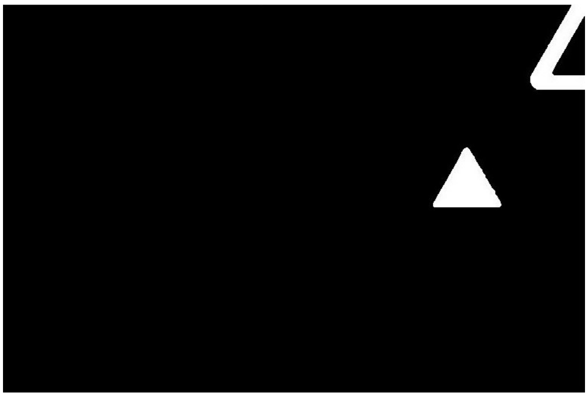

Figure 22: Fill in holes

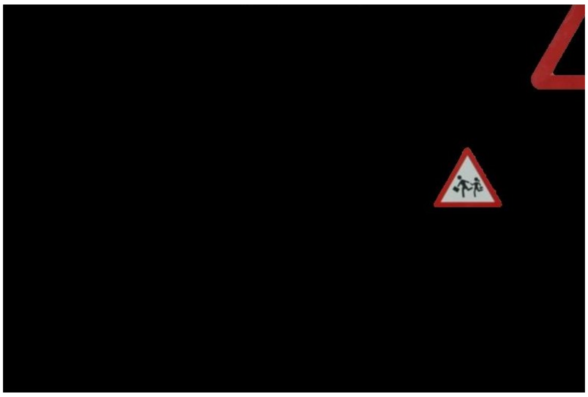

Figure 23: smoothing the borders

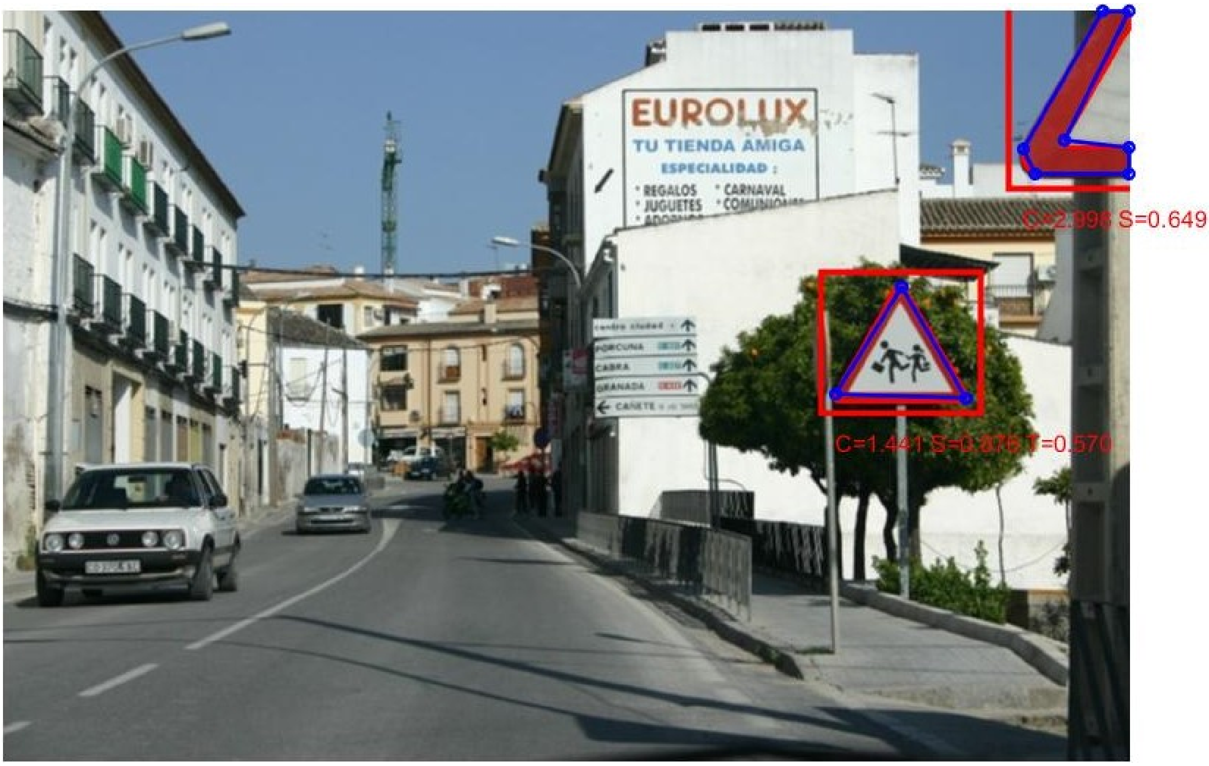

Figure 24: Processed Image with Region of Interest

3.2 Road Sign Classification

By usign VGGnet, we've been able to reach a validation accuracy of 99.3% and a testing accuracy of 97.6 %. The model has a good accuracy of 100 % when predicting Yield, Stop and No entry signs. These signs are nearly distinct and the model has learned them quite well. On the other hand The model has some poor accuracy on the speed limit signs because various speed limits are sometimes mispredicted with other speed limits this is due to the similarities they cosesly have. There are also few cases of speed signs with triangular shape that are incorrectly classified among themselves.

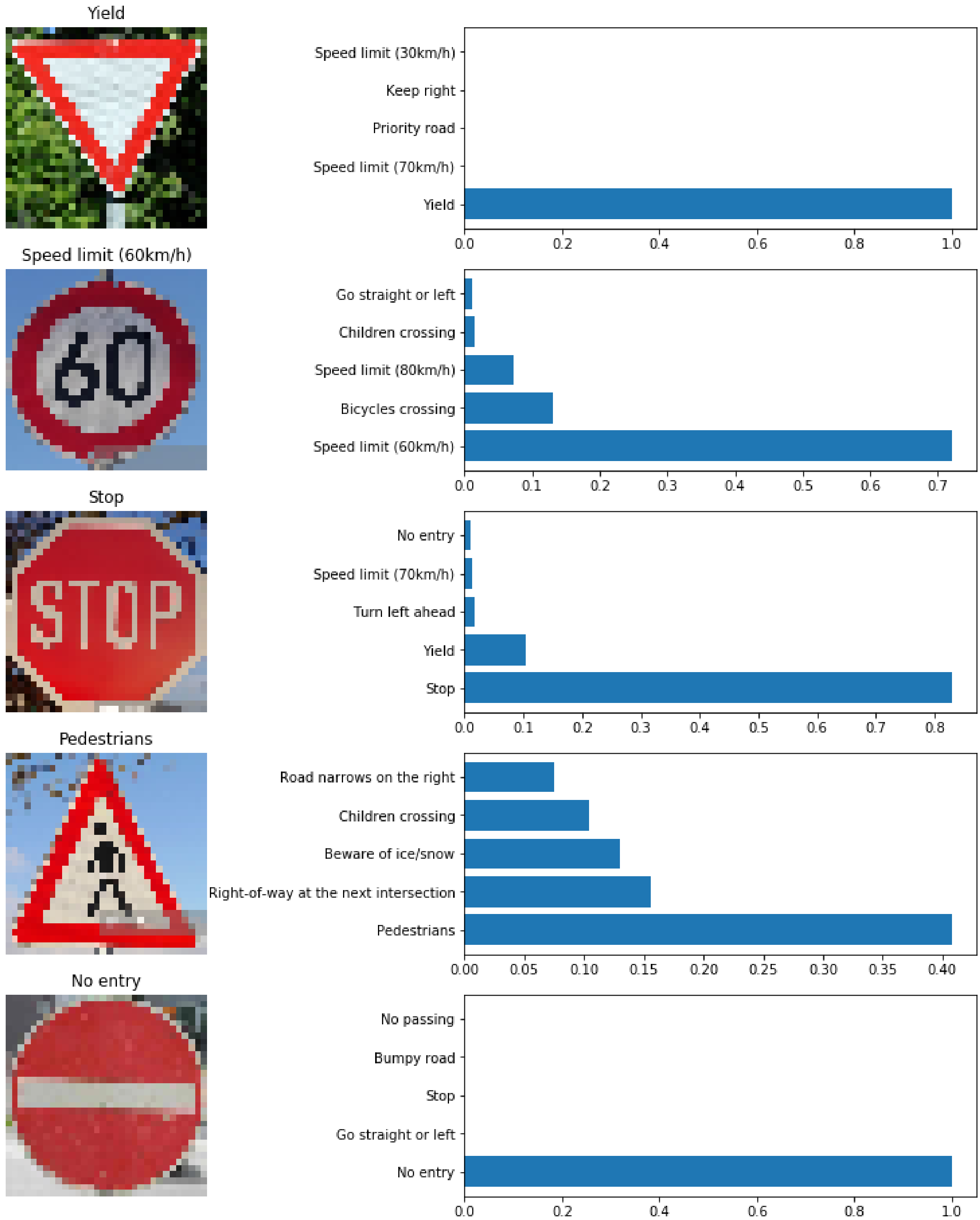

Figure 25: Machine Learning Model Performance

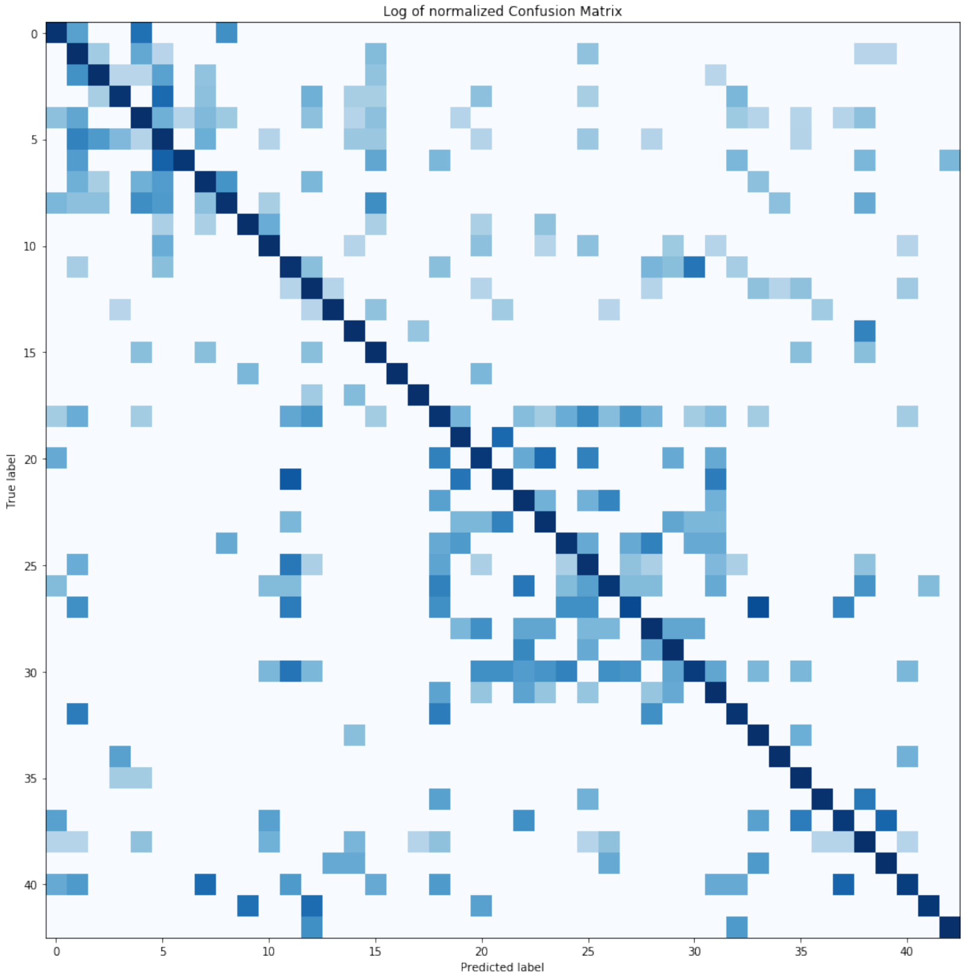

Figure 26: Confusion Matrix

Class Number and their Respective Names

0	'Speed limit (20km/h)'	22	'Bumpy road'
1	'Speed limit (30km/h)'	23	'Slippery road'
2	'Speed limit (50km/h)'	24	'Road narrows on the right'
3	'Speed limit (60km/h)'	25	'Road work'
4	'Speed limit (70km/h)'	26	'Traffic signals'
5	'Speed limit (80km/h)'	27	'Pedestrians'
6	'End of speed limit (80km/h)'	28	'Children crossing'
7	'Speed limit (100km/h)	29	'Bicycles crossing'
8	'Speed limit (120km/h)'	30	'Beware of ice/snow'
9	'No passing'	31	'Wild animals crossing'
10	No passing for vehicles over 3.5 metric tons'	32	'End of all speed and passing limits'
11	'Right-of-way at the next intersection'	33	'Turn right ahead'
12	'Priority road'	34	'Turn left ahead'
13	'Yield'	35	'Ahead only'
14	'Stop'	36	'Go straight or right'
15	'No vehicles'	37	'Go straight or left'
16	'Vehicles over 3.5 metric tons prohibited'	38	'Keep right'
17	'No entry'	39	'Roundabout mandatory'
18	'General caution	40	'End of no passing'
19	'Dangerous curve to the left'	41	cell6
20	'Dangerous curve to the right'	42	'End of no passing by vehicles over 3.5 metric tons'
21	'Double curve'		

4 Conclusion

At this point, after detailed study and implementation of real time road sign recognition it can be concluded that the techniques used for colour and shape detection have relatively less computation time then studied from different papers in literature review but the technique is limited to constant thresholding of HSV values and due to difference in lighting condition throughout the day constant threshold values would not always provide desired output.

References

[1] D. Soendoro and I. Supriana, "Traffic sign recognition with Color-based Method, shape-arc estimation and SVM," Proceedings of the 2011 International Conference on Electrical Engineering and Informatics, Bandung, 2011, pp. 1-6, doi: 10.1109/ICEEI.2011.6021584.

[2] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In Proceedings of the IEEE International Joint Conference on Neural Networks, pages 1453–1460. 2011.

[3] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale Convolutional Networks," The 2011 International Joint Conference on Neural Networks, San Jose, CA, 2011, pp. 2809-2813, doi: 10.1109/IJCNN.2011.6033589.

[4] Simonyan, Karen Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.

[5] https://www.mathworks.com/help/images/ref/reducepoly.html

[6] G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system," 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), Sendai, 2004, pp. 70-75 vol.1, doi: 10.1109/IROS.2004.1389331.

[7] https://www.mathworks.com/help/images/ref/imclose.html

[8] https://www.mathworks.com/help/images/ref/bwareaopen.html

[9] http://www.cyto.purdue.edu/cdroms/micro2/content/education/wirth10.pdf