Traffic Sign Recognition System

BSc in the Internet of Things
Waterford Institute of Technology
Waterford, Ireland
danieljcollins17@gmail.com

***Abstract* – This document outlines the design and implementation of a Traffic Sign Recognition System (TSR). This is an important Advanced Driver Assistance System (ADAS) for an automated vehicle. Vison based sign recognition are used extensively throughout the automotive industry to provide this ADAS System**

I. INTRODUCTION

Traffic Sign Recognition Systems are a huge part of the automotive industry's endeavour to produce fully automated vehicles. Current TSR systems use image processing to parse incoming video feed from a forward facing camera to determine what signs are presented during the drive. The detection methods can generally be divided into colour based, shape based and learning based methods. This project investigates using image processing to accurately detect traffic signs present in a video input. The main goal is to correctly identify the traffic signs and present these to the user in a clear and informative approach.

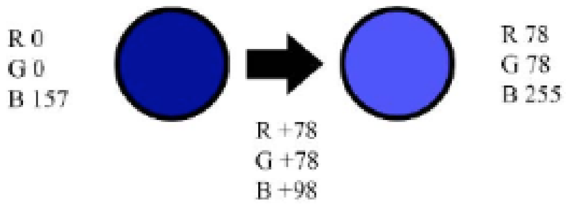
II. PROPOSED METHOD

TSR Systems are commonly developed in three main phases;

- *Image Pre-Processing Phase*
- *Detection Phase*
- *Recognition Phase*

A. Image Pre-Processing Phase

The main objective of the pre-processing phase is to enhance the input using several processing techniques, such as RGB to Greyscale conversion for dimensionality reduction (Liyanage & Udawatta) and image cropping. A colour in the RGB space is comprised of a mixture of Red, Green and Blue colours.



The figure above shows how two shades of blue can be achieved by changing the RGB colours. Difference between two colours can be hard for a computer to learn because a slight change in the illumination of a colour causes all three RGB values to change. In the Greyscale colour space, changes of illumination can be characterized by intensity, which in turn reduces the computational cost of handling different lighting conditions. Another essential method to be implemented in the Pre-Processing phase is to enhance the contrast of the image. Images can contain dark areas which could potentially obscure essential information. When an image is in the Greyscale space, histogram equalization can be used to change the contrast of an image.

This can be carried out efficiently by changing only the intensity value (Histogram Equalization)

$$J = histeq(I, n) \dots \dots \dots (1)$$

In the equation above, the “histeq” function transforms the intensity of Image I, returning an intensity image with n discretion levels to the variable J.

Image before histeq transformation

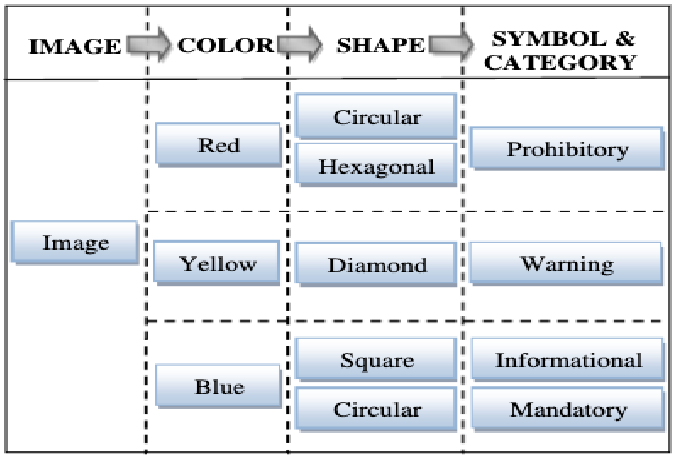
Image after histeq transformation

Many input images contain large portions that are not required for the sign recognition system to work. Cropping out the areas that do not contain traffic signs reduce the false detection rate.

Images before and after cropping

B. Detection Phase

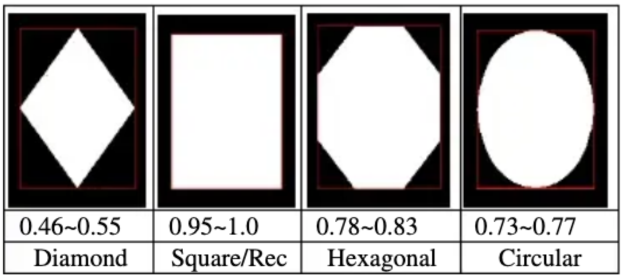
Road signs can be classified using the following model;



Since we are altering the images to be in the greyscale space, the colour column in redundant for us. The system uses bounding box’s to calculate the extent of the object of interest. A bounding box is the smallest box that the object of interest can fit in.

The extent of the object is then calculated. Extent is a specific ratio of pixels in the object of interest to the pixels in the bounding box. The total pixels in the region of the object of interest refers to the area of the object of interest (Sarah, Hussin, & Yusoff)

$$Extent = \frac{Total\ Pixels\ of\ OOI}{Total\ Pixels\ of\ Bounding\ Box}$$



The above shows four shapes and their respective ranges of extent obtained from standard images of the shapes. The systems identifies the objects of interest, and then calculates the Extent of the object using the methods described above. Once the extent is calculated, the object can be defined as one of 4 categories, Diamond, Square/Rectangle, Hexagonal or Circular.

C. Recognition Phase

Once the objects have been classified, their area's will be compared with standardised images that have been calculated as part of the training section. The training phase takes in standard images of road signs and calculates the area ration of the signs. The area ratio is calculated using the following (Sarah, Hussin, & Yusoff);

$$AreaRatio = \frac{Area\ of\ Symbol\ Region}{Area\ of\ Shape\ Region}$$

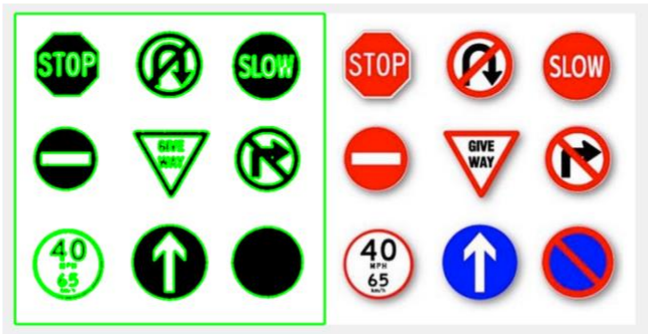
The ratios for each shape are stored as variable in MATLAB, and are then compared with the ratio of the detected object. The closest match is then determined to be the correct sign. Once the sign has been correctly determined, this can be outputted to the Driver's Heads-Up-Display (HUD) to indicate the sign and it's corresponding message via the CANbus network.

III. BENCHMARK TESTING

The following benchmarks were set out before implementing the system

A. Shape Recognition

This benchmark relates to the system being able to detect objects in an arbitrary static image. Being able to identify the correct key objects (such as squares, circles and hexagons) is imperative for the system to work.



As you can see in the image above, the code detects the objects correctly, however, it also detects the content of the sign, which is not expected. The system is expected to detect just the outline of the sign and then uncover the signs category based on this. However, a more advanced version of this TSR system could read the sign and its content and for faster classification and understanding of the purpose of the signs.

B. Recognition in Real-Life Signs

This benchmark relates to the system being able to identify and detect sign objects from a still image from a vehicles perspective. The image has been cropped, as described in the Image Pre-Processing phase above, and the sign has been detected. In spite of this, the system also detects irrelevant noise as the bottom of the image, which is presumably due to the threshold set in the MATLab code. I attempted to alter the threshold but the noise was still picked up.

IV. CONCLUSION

Due to the Simulink model construction failing, progress was halted quite early on in this project. However, the code for the system at the minute passed the first benchmark set out and partially passes the second. More work needs to be done to implement a Simulink model, and with that in place and some more refinements on the system's code, the system would work

I. Bibliography

Histogram Equalization. (n.d.). Retrieved from Mathworks: https://uk.mathworks.com/help/images/histogram-equalization.html

Liyanage, S., & Udawatta, N. (n.d.). *Online Traffic Sign Recognition Systems for Intelligent Driver Assistance.*

Sarah, S., Hussin, F., & Yusoff, M. (n.d.). *Road Sign Detection and Recognition Systemfor Real-Time Embedded Applications.*