International Journal of Engineering Trends and Technology Volume 69 Issue 12, 47-55, December, 2021
ISSN: 2231 – 5381 /doi:10.14445/22315381/IJETT-V69I12P207

Low-cost Driver Assistance System for drivers suffering from Dyslexia or Color-blindness using Machine Learning

Shikher Srivastava[1], Ayush Nigam[2], Dr. Madiajagan Muthaiyan[3]

[1,3]*School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India*

[1]shikher707@gmail.com, [2]ayushnigam97@gmail.com

Abstract — *This paper proposes a Driver Assistance System with the capability of Traffic Sign and Traffic Light Detection from complex background images in a real-time environment with high efficiency. The proposed methodology aims to design an intelligent and self-sufficient model to extract the traffic sign and traffic light from a complex natural image using a camera with the help of Image Processing and Machine Learning. To make the traffic sign recognition perform efficiently in a real-time environment, a Machine Learning Model is trained with actual traffic sign data, i.e., images of real-life traffic signs. Traffic Lights are detected using color and shape differentiation techniques to extract a traffic post from the background and then extract the light from the post. This model can easily be installed in any vehicle with a mounted camera on the front or be a part of a more sophisticated vehicle controlling system. It will recognize traffic lights and road signs and then accordingly instruct the required actions to the driving agent, especially helpful for drivers suffering from color-blindness or dyslexia.*

Keywords — *Driver Assistance System, Iterative End-Point Fit Algorithm, Traffic Sign Recognition, Traffic Light Recognition, Support Vector Machine (SVM)*

I. INTRODUCTION

Over the last decades, technology has come a long way, with automation playing a crucial role in making many of these advances feasible. Transportation is one such field that has been highly influenced by automation. Across all countries of the world, critical information on the road's condition and limitation is provided to drivers as visual signals, such as traffic lights and traffic signs. They play a vital part in road infrastructure by providing navigational information about the current state of the route, limitations, warnings, notices, and other valuable information for navigation purposes. This information is encoded in the visual features of traffic signs and lights like symbols, colors, and shapes. Neglecting or failing to spot these traffic lights or signs can lead directly or indirectly to an accident[1][2]. In these scenarios, if there is a driver assistance system installed in the vehicle, it will compensate for the potential inattention or negligence of the driver by helping him obey the traffic rules and thereby making driving safer and easier. This paper proposes a driver assistance system which is an intelligent system that is installed inside the vehicle to assist the driver while driving in multiple ways. It can be used to provide vital information to the driver about the closure and blockage of roads ahead, speed limits, diversions, and traffic lights, etc. This system is capable of detecting traffic signs and traffic lights from complex background images in a high-performance real-time environment. The proposed model is designed using concepts and techniques to maintain high efficiency on high-end as well as low-end hardware, i.e., Physical Camera. Traffic Road Sign and Traffic Light detection is a fairly complex procedure and requires extensive training and testing due to the high risk of failure or misclassifications. The process of detection works in three phases: 1) Pre-processing of an image using Image Processing. 2) Localization of the image of the Traffic Sign/Light in the Detection phase. 3) Predicting the Label of the extracted image of the Traffic Sign or Light using Machine Learning in the Recognition phase. An overview of the Traffic Light and Traffic Road Sign detection is shown in Figure 1, which explains the Process flow starting from the image input from cameras and output to the environment through a speaker. The architecture proposed in Figure 1 showcases the concepts and methods used in our approach in Traffic Sign and Traffic Light detection.

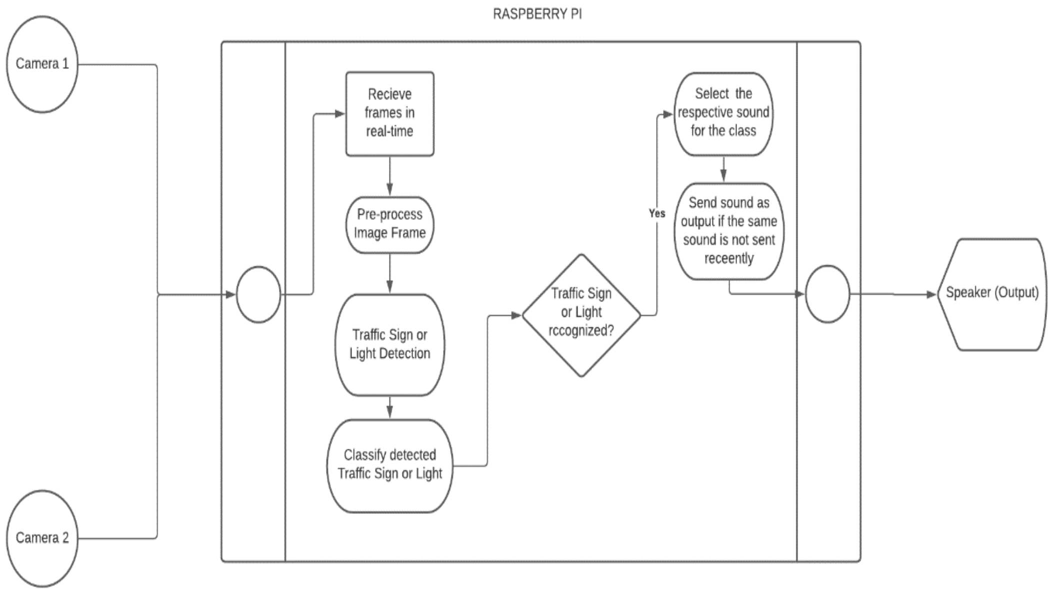


Fig. 1: Architecture of Driver's Assistance System with Traffic Sign and Traffic Light Recognition

Two separate Cameras are used for both Traffic Sign and Traffic Light detection to avoid any detection errors. Pre-processing phase deals with preparing the frames received from the Camera for the detection phase. This includes concepts of Color Thresholding, Edge Detection (LOG method), and Binarization applied to an HSV Color Space[3][4][5]. Localization is performed on the processed frame to extract the region of interest, i.e., Traffic Sign or Light. Localization works by finding large-sized connected contours and using these contours to find the shape of the Sign or to detect the Traffic Light post. After the Image is localized and the Traffic Sign is detected, the Image is converted to HOG (Histogram of Oriented Gradients) feature description, which is then passed for prediction of Traffic Sign to an SVM model.

II. MOTIVATIONS

Driving is a pretty arduous task and requires the driver to maintain constant undivided attention to the driving environment for a long time. This may include roads, traffic conditions, road signs, other vehicles, weather conditions, and a bunch of other factors. With so much to process, it is very easy to get distracted due to some external factors. The National Highway Traffic Safety Administration (NHTSA) findings show that 94% of major accidents are caused due to human error, such as drunk or distracted driving [6] [7] [8] [9].

It is very common for the driver to miss reading traffic Signs or Traffic Lights as both of this information are generally at the sides of the road, and the driver is typically focused on the road and other vehicles on the road.

The findings of a study by Pilar Tejero et al., 2018 [10] show that people with dyslexia may have trouble reading traffic signs when driving. When attempting to read traffic signals, people with dyslexia read them less accurately and had a harder time maintaining a consistent speed. Additionally, these issues are exacerbated when the time available to read the traffic sign is insufficient due to driving task-related variables. Moreover, in the results of research by Izautino P. Oliveira et al., 2015 [11], it is revealed that color-blind people do not feel safe driving alone in critical situations; in these cases, they always travel with friends or family who can aid them in traffic.

Through our research, we aim to develop a Driving Assistance System to detect traffic signs and traffic lights from the environment and inform the driver of the same to relieve the driver of fatigue of searching Traffic Signs and Traffic Lights.

III. IMPLEMENTATION

A. Pre-Processing Phase

Traffic road sign and Traffic light detection is a pretty complex task and thus has a very small margin of error. Designing a model that performs Traffic sign and light detection that works in real-time environment with high accuracy requires having a way to focus on the Region where the Traffic Sign/Light is present and avoid any noise in the frames. Pre-processing phase deals with exactly that and prepares the image for the detection phase. The Pre-processing phase is used to filter the image so that

Background of the Image can be minimized and the Traffic Sign or Traffic Light, if present, can be highlighted fromthe complex Image. Frames are extracted from the input video and used as input for the model. Since the detection process is a heavy process, frame resolution is decreased to 640 X 480 to make the process faster.

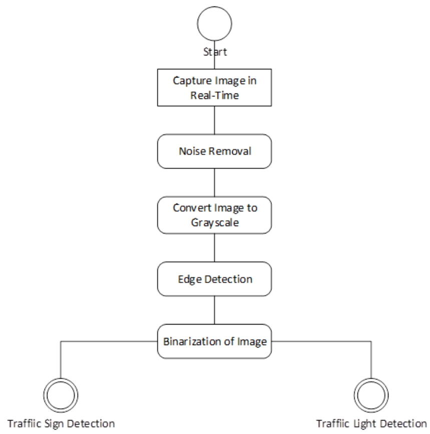


Fig. 2: Pre-processing phase flow diagram

a) Noise Removal

A traffic sign or light detection model operating in real life has to deal with various disturbances in the image frames. These noises need to be smoothed out so that the next steps in the process, like edge detection, do not form unwanted boundaries of the image. Noise has a significantly negative impact on edge detection as it can make a pixel of the image look very different from its surrounding pixels and thus form an inaccurately high amount of unwanted edges.

Gaussian Filter is used to removing these unwanted details and noise from the image. Gaussian filter is a low-pass filter that convolves the image with a user-defined or predefined Gaussian kernel[12]. It is preferred over other low-pass filters as, in contrast to uniform filters like Mean filter, which computes weighted average uniformly over the whole image, Gaussian filter computes the average of pixel values weighted more toward the center in the neighborhood of current pixel, which leads to more preserved edges and better noise reduction. This property of Gaussian Filtering makes it a very useful method in the Object detection domain.

The proposed approach uses a 2-D isotropic convolution gaussian filter to blur or smooth the image for noise removal of the given form with a standard deviation (σ) of 1.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

A Gaussian kernel of size 3 X 3 is used, which is a square array of pixel values corresponding to the values of the 2-D isotropic gaussian curve. The idea of performing a Gaussian filter is to compute a weighted average of the pixel in the neighboring area. Our current pixel is carried out by a sliding window approach in which the non-uniform kernel is slid over parts of the image and smoothing the image by computing the average based on the isotropic curve.

b) Edge Detection

A traffic sign or light has to be separated from the boundary in order to be detected. For the sign or light to be detected, its boundaries need to be found. Edge detection is performed to find these boundaries by highlighting the edges between areas with comparatively distinctive properties.

The Laplacian of Gaussian (LOG) technique is used for edge detection. The LOG approach is a 2-step process involving 1) a Laplacian operator being applied over 2) a Gaussian Filter on the image in a single pass. The importance of the Gaussian Filter has been explained in section 3.1. The Edge Detection process works most efficiently when the image is in grayscale color space. The Laplacian operator, when applied to an image, highlights the pixel regions with sharp changes in intensity between a pixel and its neighbors, which are used to form a distinctive region separated by boundaries.

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

The Laplacian is an isotropic operator which calculates a 2nd order spatial derivative of an image [13]. The function shown above gives an output of zero (represents color black) for the pixel region with constant intensity or no change. However, for the regions with rapid change in intensity, it results in a positive value when moving from a lighter pixel to a darker pixel color and vice-versa [14][15][16].

Other Edge Detection operators like Sobel or Gradient operators are slower in comparison to Laplacian operators; they use multiple kernels and require many iterations to detect edges, whereas Laplacian uses only one kernel and a single pass hence providing better performance speed.

Laplacian Operator works along with a Gaussian filter as a sliding window technique in a single pass where Gaussian filter removes unnecessary noises from the window and Laplacian operator then highlights the edges in the same window. The combined formula for the Laplacian of Gaussian Operation is shown below.

$$LoG(x, y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2 + y^2}{2\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

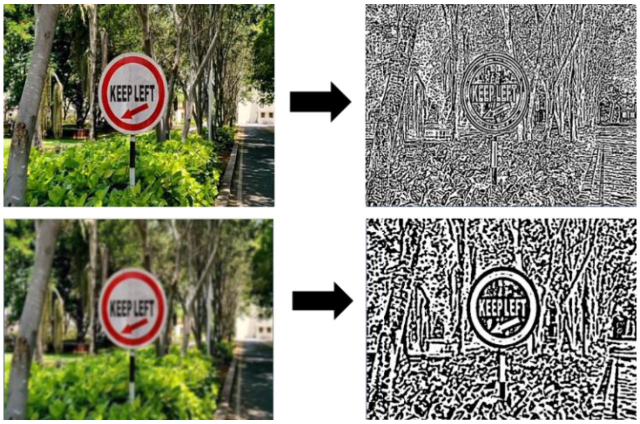


Fig. 3: Edge Detection with and without Gaussian Blur

Once the edges are highlighted in the grayscale color space, to further highlight the more significant edges from the faint once, the image is binarized, i.e., all pixel intensities below a certain threshold are converted to 0 (numerical representation of black), and the rest is converted to 255 (numerical representation of white).

$$\mathrm{Bin}(x, y) = \begin{cases} 1 & if\ gray(x, y) \geq Threshold \\ 0 & otherwise \end{cases}$$

B. Detection Phase

Once the pre-processing phase has prepared the phase for the detection phase, the image has already been binarized, and the edges have been highlighted. However, it is possible that many edges that are highlighted in the pre-processing phase are formed by undesired colors that are not present in traffic signs and traffic lights. In the Detection Phase, Various techniques like color masking, contour detection, and shape analysis have been explained. Working on them for Traffic signs and Traffic Light detection will be discussed in the later sections.

a) Color Masking

As explained earlier, undesired colors need to be removed from the background in order to make the detection process faster and more accurate. Color masking is one of the most efficient methods for extracting the region of our interests based on color differentiation. The regions with the required color(s) can be separated from the rest of the image by applying a color mask corresponding to the desired color.

Usually, an image is represented in the RGB (Red, Green, and Blue) color space. However, RGB is highly sensitive to lighting conditions, hence is inefficient and error-prone for the task of object detection. Hence, it is preferred to convert the image from default RGB to HSV (Hue, saturation, Value) color space. HSV allows ranges of different illuminations of a particular color, which decreases the effect of varying lighting conditions expected in the real environment and thus, provides better object detection [17].

Fig. 4: Color Masking applied on a bottle cap

The process of Color Masking or Color Filtering consists of 2 steps: 1) Creating a color mask and 2) Applying the mask to the image [18]. At first, a mask is created, which consists of a single color or a combination of multiple colors. The mask is created using a range of Hue, Saturation, and Value. The mask is finally created when the range is applied to the original colored image that converts the regions in the image which fall in the range of the mask as white, denoting true areas in the mask, whereas other areas are converted to black denoting false areas. Thus, the mask contains white pixels where desired color(s) is present, and the rest of the image is black. In the second step, the mask is applied to the pre-processed image using a bitwise AND operation, which compares the mask image and the original image and extracts the pixels from the original image, which are of white color value in the mask image. Figure 4 explains the process of Color Masking on a Traffic Sign.

b) Object Identification and Extraction using Contour Features

Color Masking makes the image free of the unwanted background, but still, we need to find the Traffic Sign/Light in the image. Contour detection is the first step towards locating the region of interest from the background, be it a Traffic Sign or a Traffic Light.

$$M_{ij} = \sum_x \sum_y x^i y^j$$

$$Area\ of\ Contour = M_{00}$$

$$Centroid\ \{\bar{x}, \bar{y}\} = \left\{\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}\right\}$$

The iterative End-Point Fit Algorithm is used for the process of detecting shapes of the contours. [19]. The idea behind this algorithm is to find a similarly connected curve with a smaller number of points than the original curve, which is achieved by recursively dividing the curve into smaller parts which are then rejoined as a straight line to reform a single connected component in each iteration [20]. The extent to which a curve will be straightened depends upon the tolerance factor (ε). The shape is then computed by counting the number of sides in the resulting connected component. For example, a triangular shape has three sides, as shown in Figure 5.

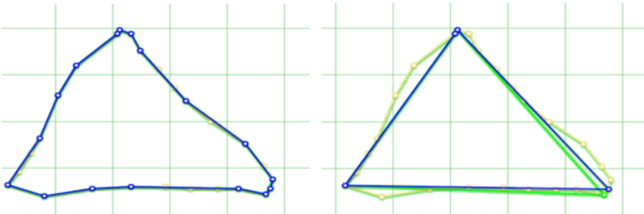


Fig. 5: Iterative End-Point Fit Algorithm applied to the triangle-like object

This algorithm is quite efficient while finding shapes with finite sides but has low accuracy in finding non-edged shapes like a circle. Traffic Lights and some Traffic Signs are circular in shape; therefore, a different method has been used for detecting circular shapes [21].

To detect circles, the distance of each point on the contour is calculated from its centroid (calculated using Image Moments), and the maximum distance is computed. This maximum distance is used to find the mean variation of the distances of all other points with respect to the maximum distance. The object with a mean value above the given threshold is identified as a circle-shaped object.

As a real-life application, if more than a Traffic sign or Traffic Light is present in the image at different distances, the one nearer to the vehicle should be given preference over the farther one. As explained earlier in Image Moments, the 0th Moment of an object gives the area of the object; this value can be used to estimate the distance of the object and also to compare which Traffic Sign or Light is closer to the vehicle.

- ***Traffic Sign Detection***

As per NHTSA guidelines, Traffic Signs are always bounded with Red and Blue colored boundaries. Hence, red and blue color masks are applied to the image so as to extract the traffic signs from the pre-processed image. The range of Hue, Saturation, and value for both red and blue are set so as to achieve similar detection efficiency in varying lighting conditions.

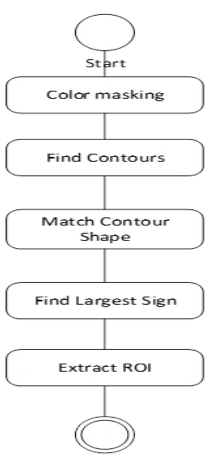


Fig. 6: Traffic Sign Detection flow diagram

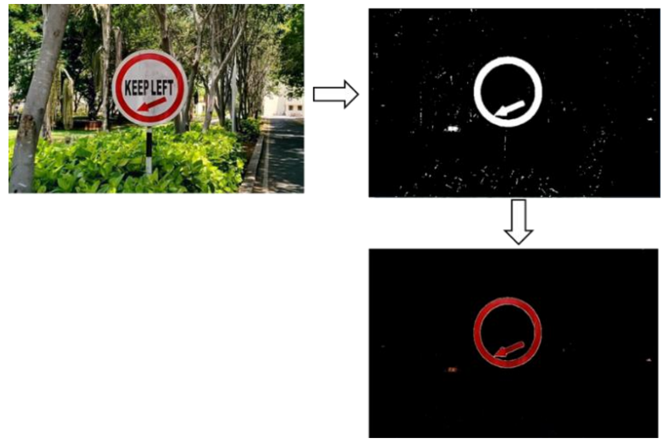


Fig. 7: Color mask applied on a Traffic Sign

Figure 7 illustrates the use of a Color Mask to Filter out colors from Traffic Sign Images. Even after color masking, some unwanted boundaries still remain that are not small enough to be considered as noise but not big enough to be considered for detection. These components are considered small components and are needed to be removed from the background. Hence, a component size threshold is set, and any component below the size of the given threshold is removed from the image.

Traffic signs are available in certain shapes, each denoting different actions. For example, Triangle-shaped traffic Signs are used to denote Cautionary Actions, and Round shaped signs are used to denote Regulatory actions. Hence, we have used contours features for finding contours and identifying shapes for traffic sign detection.

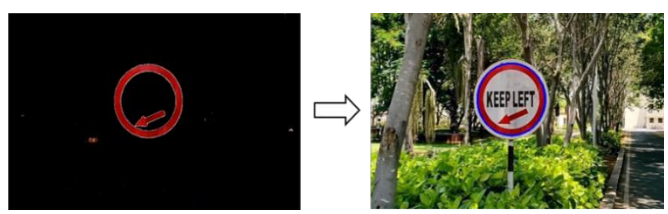


Fig. 8: Contour (blue color) is shown around a Traffic Sign

Figure 8 shows the main contour drawn around the Traffic Sign Image.

After finding all contours, Iterative End-Point Fit Algorithm is used to detect if the found contours resemble the shape of a Traffic sign, i.e., Circle, Triangle, Square, or Hexagon. The Contour is considered a traffic sign only if its shape is detected to be matching the mentioned criteria.

As previously explained, if there is more than one Traffic Sign in an image, the area of the identified signs is compared so that the Traffic Sign, which is greater in size and also greater than a certain threshold, is extracted while others are ignored. Once the Traffic sign is detected, a cropped image of the region, including the Traffic Sign, is sent to the recognition phase.

- ***Traffic Light Detection***

Similar to Traffic Sign Detection, Color Masking is performed in the first step of Traffic Light detection to extract the regions with yellow or black color as the traffic signposts containing the traffic lights are generally of either yellow or black color.

Traffic Posts are rectangular in shape with a positive height to width ratio. A range of ratio between height and width is used to detect the rectangular shape of Traffic Light, which provides the most accurate results with the least false negatives and false positives. The contours found meeting these shape requirements are considered as Traffic Posts. Among these identified traffic posts, the largest sized post is cropped and used for further traffic light detection process.

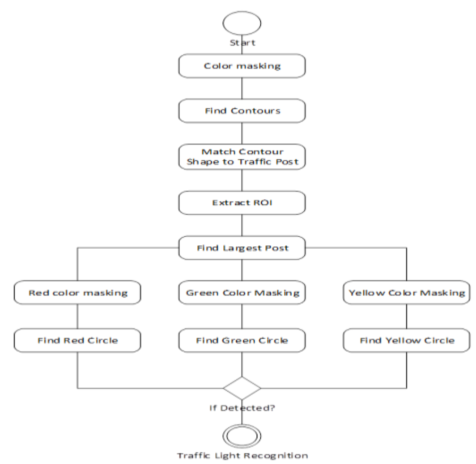


Fig. 9: Traffic Light detection flow diagram

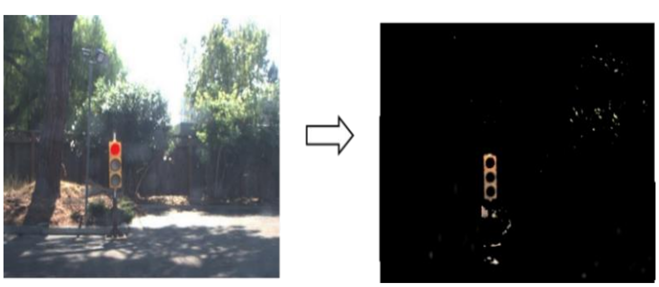


Fig. 10: Color mask applied on a Traffic Light

Figure 10 illustrates the use of a Color Mask to filter out colors from Traffic Light images.

There are three types of traffic lights; Red for `Stop', Green for `GO', and Yellow for `Wait'. From the cropped traffic post image, the active color of the traffic light is detected. Another Color Masking is performed in all three cases: 1) Red mask is used for detection of Red Light, 2) a Green mask is used for detection of Green Light, and 3) a Yellow mask is used for detection of Yellow Light. After extracting the respective colored region for each case, contours are found, and if there exists a circular shape within the traffic post greater than a certain threshold, the traffic light is detected.[22]

To improve the accuracy and reduce errors in the detection of traffic lights, the cropped image is sent to the recognition phase.

C. Recognition Phase

The Recognition receives the cropped image from the Detection phase. The Recognition Phase is a Machine Learning Model trained using Support Vector Machine (SVM), which classifies the Traffic Signs and Traffic Lights and labels them.

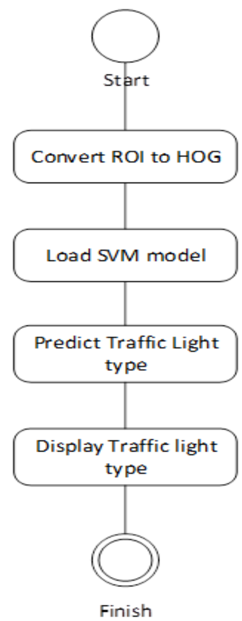


Fig. 11: Traffic Sign Recognition flow diagram

- ***Traffic Sign Recognition***

The detection phase for traffic signs dealt with localizing and detecting the region was and whether the traffic sign existed in the image. The main role of the recognition phase is to classify the detected traffic sign into different Traffic Sign Classes like Stop sign, No Parking sign, etc., using a machine learning model. Before the image can be sent to the Machine Learning SVM model for classification, the cropped image needs to be converted into its HOG Description [23]. Before HOG conversion, the image is fit into the desired aspect ratio of 32 X 32 Dimensions [24]. The various parameters required by the HOG descriptor for conversion are defined like window size as 20 X 20 and cell size as 10 X 10. HOG representation of Traffic signs can be seen in Figure 12.

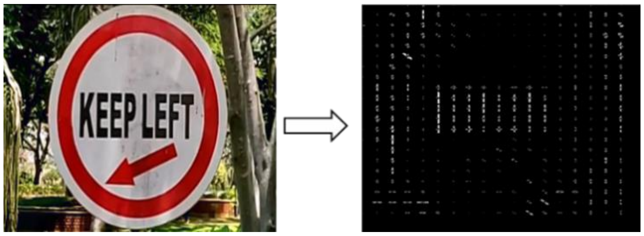


Fig. 12: HOG representation of a cropped Traffic Sign

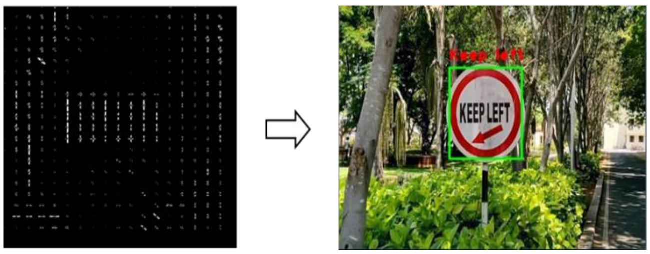


Fig. 13: Classification of Traffic Sign using SVM

After converting the image to its HOG representation, the image is provided as input to the SVM model for classification [25]. And then, after the successful recognition of class for the traffic sign, a green bounding box is drawn around the traffic sign with its class label on the original image. Figure 13 shows the recognized Traffic Sign using SVM and HOG.

- ***Traffic Light Recognition***

In contrast to the traffic sign recognition where the classification was done to find the Traffic Sign label, the recognition phase in the traffic light is mainly concerned with confirming whether the detection phase result and SVM model output are the same or not [26]. This will help in improving the accuracy of detection and reduce the number of false positives of the model. Similar to traffic sign recognition, the cropped image from the Detection Phase is converted to its HOG representation before passing it to the SVM for classification. Since the Aspect ratio of a rectangular Traffic Post is not symmetrical, the HOG parameters and the aspect ratio is defined accordingly.


Fig. 14: HOG representation of a cropped Traffic Light


Fig. 15: Classification of Traffic Light using SVM

The HOG descriptor of the image preserves the shape of the image, which is used to confirm the detected location of the colored light in traffic post as the HOG descriptor of the image works on gradients of color intensity and thus preserves the shape and structure of the traffic signpost and colored circle of traffic light inside the post. HOG representation of traffic light can be seen in Figure 14. Since the detection has already determined the Traffic Light, which is active, the recognition phase makes sure that the result of the detection phase and the output of the SVM model is in accordance.

Since the HOG representation of the image is based on the change in intensity and not on the color, the SVM model predicts the traffic light using the relative positioning of the Traffic Light circle and the post. Figure 15 shows the recognized Traffic Light using SVM and HOG.

IV. DATASET DESCRIPTION

The dataset is composed of images from the Udacity Self-Driving Car Simulator and the LISA Dataset. For traffic signs, we used nine classes, with eight of them being traffic signs and the final one being erroneous visuals that may be classified as traffic signs. Similarly, for traffic lights, we used four classes, with three of them being traffic lights and the final one being erroneous visuals that may be classified as traffic lights. A total of 1680 images were used with a split ratio of 70% train, 15% validation, 15% test images.

V. RESULTS AND DISCUSSION

A Traffic Sign and Traffic Light detection system has been successfully trained and tested. The SVM Machine Learning Model has been trained with images of different Traffic Signs and Traffic Lights, and the Generated Model is used to Detect Traffic Signs or Light from the Complex Natural Environment. The system has been tested against a good number of Globally Standard Traffic Signals and Traffic Light Designs. Tests are made at various angles and lighting conditions for a reliable accuracy score. Tables representing detection metrics of Traffic Signs and Traffic Lights in various environmental conditions is shown in Figure 16 and Figure 17 below:

CLASS	IMAGE	ACCURACY	F1	PRECISION	RECALL	SENSITIVITY	SPECIFICITY
STOP SIGN		0.967937	0.881587	0.848449	0.917419	0.917419	0.975492
SPEED LIMIT		0.95904	0.85122	0.804147	0.904145	0.904145	0.967213
NO PARKING		0.965419	0.873309	0.854392	0.893082	0.893082	0.976559
HORN PROHIBITED		0.960551	0.819923	0.838558	0.802099	0.802099	0.980529
PEDESTRIAN CROSSING		0.937385	0.766144	0.760897	0.771465	0.771465	0.962827
KEEP LEFT		0.982709	0.907623	0.949343	0.869416	0.869416	0.994977
ROUND ABOUT		0.924794	0.704485	0.724559	0.685494	0.685494	0.960796
SLIPPERY ROAD		0.934531	0.740691	0.785614	0.700629	0.700629	0.970554

Fig. 16: Evaluation metrics for Traffic Sign Model

The proposed system depicts a slightly better detection accuracy for circular-shaped traffic signs (≈96% to ≈98%) than triangular-shaped signs (≈92% to ≈94%) due to a smaller Region of Interest in the triangular signs. Also, the system had a very high accuracy in detecting traffic signs with solid fill inside the shape (≈98%). The detection accuracy of the model does not reduce even in the case of similar-looking traffic signs like No Parking (96.54%) and Horn Prohibited (96.05%).

For Traffic Light, the detection accuracy for green light was 95.19% as compared to slightly less for red light (94.04%) and lesser for yellow light (90.96%). The reason for the relatively low accuracy for yellow light was the slight gradient difference between the Traffic Light and the Traffic Light post. The similarity between red and yellow colors at different lighting conditions also led to some false positives.

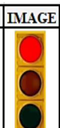
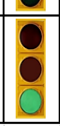
CLASS	IMAGE	ACCURACY	F1	PRECISION	RECALL	SENSITIVITY	SPECIFICITY
RED LIGHT		0.940462	0.927619	0.887079	0.97204	0.97204	0.920061
YELLOW LIGHT		0.909643	0.839486	0.921676	0.770754	0.770754	0.971044
GREEN LIGHT		0.951903	0.921733	0.903202	0.94104	0.94104	0.95658

Fig. 17: Evaluation metrics for Traffic Light Model

VI. CONCLUSIONS

A cost-efficient Traffic Road Sign and Traffic Light Detection System have been proposed in this paper capable of accurate detection using Real-Time Image Processing and Recognition. The aim of the research was to design a Detection and Recognition approach that can be implemented on low-end devices with processing power that cannot support the high-power requirements of Neural Networks. With this being the goal, every concept of image processing and object recognition applied in the proposed method has been optimally designed to ensure a minimal delay in the detection and recognition times without trading off on the high accuracy of detection required in an application of such nature. A high degree of pre-processing is performed on the image before the detection phase to minimize the search area and determine the Region of Interest (ROI) in the Image. Processes like Noise Removal and Small Components removal make sure that other heavier processes don't have to unnecessarily search for the region of interest in unimportant areas of an image and cost more time and processing. There is still some scope for future improvements:

- The accuracy of detection and recognition can be increased in a moving vehicle with the help of object tracking. This will prevent the same sign or light from being detected again and its respective sound being played by the speaker.
- Detection in cases of occlusions, shape deformation, and shadows can be improved.

REFERENCES

[1] Regan, Michael A., Charlene Hallett, and Craig P. Gordon., Driver distraction and driver inattention: Definition, relationship, and taxonomy. Accident Analysis & Prevention, 43(5) (2011) 1771-1781.

[2] R.R. Thoker, R. Gupta, E.N. Kumar, Study on congestion of national highways due to traffic volume, International Journal of Engineering Trends and Technology, 68 (2) (2020) 36–41.

[3] Y. Jie, A new traffic light detection and recognition algorithm for electronic travel aid., (2013) 644-648.

[4] G. Mu, Traffic light detection and recognition for autonomous vehicles., The Journal of China Universities of Posts and Telecommunications, 22 (1) (2015) 50-56.

[5] Y. K. Kim, K. W. Kim, X. Yang, Real-time traffic light recognition system for color vision deficiencies, (2007) 76-81.

[6] Ranney, Thomas A., W. Riley Garrott, and Michael J. Goodman. NHTSA driver distraction research: Past, present, and future. No. 2001-06-0177. SAE Technical Paper, (2001).

[7] A. Thierer, R. Hagemann, Removing roadblocks to intelligent vehicles and driverless cars., Wake Forest JL &Pol'y, 5 (2015) 339.

[8] Ascone, Debra, Tonja Tonja Lindsey, and Cherian Varghese. An examination of driver distraction as recorded in NHTSA databases. No. DOT HS 811 216. The United States. National Highway Traffic Safety Administration, (2009).

[9] Angell, Linda, Miguel Perez, and W. Riley Garrott. Explanatory Material About the Definition of a Task Used in NHTSA's Driver Distraction Guidelines and Task Examples. No. DOT HS 811 858. (2013).

[10] Tejero, Pilar, Beatriz Insa, and Javier Roca, Difficulties of drivers with dyslexia when reading traffic signs: analysis of reading, eye gazes, and driving performance, Journal of learning disabilities, 52(1) (2019) 84-95.

[11] Olivera, I. P., et al. A Vision of Traffic Lights for Color-Blind People, The Fourth International Conference on Smart Systems, Devices and Technologies (SMART), Brazil., (2015) 34-36.

[12] M. Sharifi, M. Fathy, M. T. Mahmoudi, A classified and comparative study of edge detection algorithms, Proceedings. International conference on information technology: coding and computing, IEEE, (2002) 117-120.

[13] O. R. Vincent, O. Folorunso, A descriptive algorithm for sobel image edge detection., Vol. 40, Informing Science Institute, California,

(2009) 97-107.
[14] Bhardwaj, Saket, and Ajay Mittal., A survey on various edge detector techniques., Procedia Technology, 4 (2012) 220-226.
[15] Liqin, Shen, Shen Dinggang, and Qi Feihu., Edge detection on real-time using LOG filter, Proceedings of ICSIPNN'94. International Conference on Speech, Image Processing, and Neural Networks. IEEE, (1994) 37-40.
[16] Kumar, Sunil, Amit Kumar Upadhyay, Preeti Dubey, and Sudeep Varshney, Comparative analysis for Edge Detection Techniques, In 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), IEEE, (2021) 675-681.
[17] Mohd Ali, Nursabillilah, NahrulKhairAlang Md Rashid, and Yasir MohdMustafah, Performance comparison between RGB and HSV color segmentations for road signs detection, Applied Mechanics and Materials., Trans Tech Publications Ltd, 393 (2013) 550-555.
[18] S. Vitabile, Road signs recognition using a dynamic pixel aggregation technique in the HSV color space., IEEE, (2001) 572-577.
[19] Nevatia, Ramakant, and K. Ramesh Babu., Linear feature extraction and description, Computer Graphics and Image Processing, 13(3) (1980) 257-269.
[20] Saalfeld, Alan., Topologically consistent line simplification with the Douglas-Peucker algorithm., Cartography and Geographic Information Science, 26(1) (1999) 7-18.
[21] Salhi, Abderrahim, BrahimMinaoui, and Mohamed Fakir. "Robust Automatic Traffic Signs Detection using a fast polygonal approximation of digital curves, 2014 International Conference on Multimedia Computing and Systems (ICMCS). IEEE, (2014) 433-437
[22] Omachi, Masako, and Shinichiro Omachi., Traffic light detection with color and edge information. 2009 2nd IEEE International Conference on Computer Science and Information Technology. IEEE, (2009).
[23] Chapelle, Olivier, Patrick Haffner, and Vladimir N. Vapnik., Support vector machines for histogram-based image classification, IEEE transactions on Neural Networks, 10(5) (1999) 1055-1064.
[24] Suleiman, Amr, and Vivienne Sze., Energy-efficient HOG-based object detection at 1080HD 60 fps with multi-scale support. 2014 IEEE Workshop on Signal Processing Systems (SiPS). IEEE, (2014).
[25] Tzotsos, Angelos, and DemetreArgialas., Support vector machine classification for object-based image analysis, Object-Based Image Analysis. Springer, Berlin, Heidelberg, (2008) 663-677.
[26] Chapelle, Olivier, Patrick Haffner, and Vladimir N. Vapnik., Support vector machines for histogram-based image classification, IEEE transactions on Neural Networks, 10(5) (1999) 1055-1064.