Warning Traffic Sign Detection Using Learning Vector Quantization & Hough Transform and Recognition Based On HOG

Moumita Roy Tora

ID: 11301025

Supervisor

Rubel Biswas

Department of Computer Science & Engineering

January 2014

BRAC University, Dhaka, Bangladesh

Abstract

Traffic Sign Recognition (TSR) is used to regulate traffic signs, warn a driver, and command or prohibit certain actions. Fast real-time and robust automatic traffic sign detection and recognition can support and disburden the driver and significantly increase driving safety and comfort. Automatic recognition of traffic signs is also important for an automated intelligent driving vehicle or for driver assistance systems. This paper aims to present a color segmentation approach for traffic sign recognition based on LVQ neural network and also focuses on triangular edge detection and feature extraction based on Hough transformation and HOG respectively. At first samples of images in different weather conditions are collected and then RGB images are converted into HSV color space. The samples are then trained using LVQ depending on the hue and saturation values of each pixel and then tested for color segmentation. The edges of the triangular segmented images are then detected using Hough Transformation. Then samples are taken to extract features using HOG. Finally they are trained and tested using SVM to get the output image. The algorithms were applied to around 100 sampled images taken in different countries and varied weather conditions. Despite the varying conditions, the algorithms worked almost accurately in all situations and the success rate was quite satisfactory with a very good response time of a few milliseconds.

ACKNOWLEDGEMENT

I wish to acknowledge the efforts of my supervisor Rubel Biswas for his contribution, guidance and support in conducting the research and preparation of the report. I also wish to acknowledge my chairperson, Professor Mohammad Zahidur Rahman for his direct or indirect support and inspiration. I am very thankful to my parents, family members and friends for their support in completing the work.

Finally, I am very thankful to BRAC University, Bangladesh for giving me a chance to complete my B.Sc degree in Computer Science and Engineering.

Supervised By

……………………

Table of Contents

List of Figures

List of Tables

Abbreviation

RGB- Red Green Blue

HSV- Hue Saturation Value

LVQ- Learning Vector Quantization

HOG- Histogram Oriented Gradient

SVM- Support Vector Machine

NN- Neural Network

1. Introduction

Traffic signs or road signs are signs erected at the side of or above roads to give instructions or provide information to road users. With traffic volumes increasing since the 1930s, many countries have adopted pictorial signs or otherwise simplified and standardized their signs to overcome language barriers, and enhance traffic safety. Such pictorial signs use symbols in place of words and are usually based on international protocols. Such signs were first developed in Europe, and have been adopted by most countries to varying degrees [1]. Traffic signs play very important role in reducing accidents but unfortunately most of the people hardly follow the signs, resulting in accidents. To prevent such mishap, my paper aims to find a solution by building software based on images that automatically will warn drivers about the type of sign ahead. There are many types of traffic sign as shown below-

FIG: 1.1 Danger Warning Sign

FIG: 1.2 Prohibitory Sign

FIG: 1.3 Mandatory Sign

There are many more other types of traffic signs such as special regulation signs, signs for direction and position, welcome sign etc. but my concern is only triangular warning signs as these are the most important ones in order to avoid accidents. The main techniques used in my paper are color segmentation and information extraction. Color segmentation is a process of separating one or more connected regions from the image domain. This is a very critical issue specially when there are continuous changes in the natural scenes such as illumination changes, moving objects, point of view etc. This is very crucial to properly segment the images because the accuracy of the output depends on how perfectly the color segmentation is done. There are various methods to apply but the performance and accuracy differs. So the method must be selected very carefully in order to satisfy the objective accurately. For my work, before LVQ was applied, I had to analyze different methods such as SVM, Probabilistic Methods, Soft Computing Techniques etc. in order to compare the performance of each. Since my work was not limited to segmentation, I had to carefully observe and select the algorithm that would maximize my performance since the rest of the parts i.e. information extraction depended on how precisely the segmentation was done. LVQ is a supervised learning approach that learns to recognize groups of similar input vectors in such a way that neurons physically near to each other in the neuron layer respond to similar input vectors. It involves neural networks that operate directly on image pixels and the target class is chosen by the user. The applications of the algorithm include human face detection, traffic sign detection etc. The biggest advantage of LVQ is that it works efficiently for images in a variety of conditions such as good/ bad light, foggy/ rainy/ sunny weather, for all the colors (red/blue/yellow) and images with different illumination across many countries. Moreover, the execution time of LVQ is very small because it directly works on image pixels. Once the segmentation was done successfully, I moved on to information extraction. At first, Canny Edge Detection and Hough Transformation were applied to find the triangular edges of warning- traffic signs. My focus was only triangular sign detection, so I used basic geometric principles such as the size of 2 sides of a triangle is greater than the 3rd side etc. The last step was to detect the inner object i.e. what type of sign it is. For this part, the concept of Dalal and Triggs was used. The concept is based on HOG and SVM which they used for pedestrian detection. HOG is an image descriptor based on the image’s gradient orientation. The 2 phases of the algorithm are- descriptor formation, and training and classification. The training requires the use of SVM which is used extensively in data mining, machine learning etc.

1.1 Proposed System

The aim of my thesis is to get better results in color detection and recognition using LVQ and also to find a solution in extracting information using several algorithms such as Canny Edge Detection, Hough Transformation, Bounding Box, HOG etc. Therefore there are mainly 2 parts in my paper- traffic sign detection and information extraction.

For the color segmentation, at first the images are taken by digital camera and the RGB images are then converted into HSV. The main reasons why we need this conversion are-

1. In RGB, it is difficult to separate the color information from the brightness one
2. HSV uses color descriptions instead of color primaries which have more intuitive appeal to the user.

Then the input vector is created based on the hue and saturation of sample images and these input vectors are used with their target classes in training the LVQ network. Next, the network is tested by taking input images in different environmental conditions for testing the accuracy.

Once the color segmentation is done successfully, the edges of all the objects in the image are found by Canny Edge Detection algorithm, objects are detected by bounding box, triangular lines are detected by Hough Transformation and finally the information extraction is done using HOG (Histogram- Oriented Gradient). The 2 phases of HOG are described below-

1. Descriptor Formation- Normalize gamma and color, 1D gradient filters compute orientation, Form histograms of gradient orientations over spatial cells, Group cells into overlapping blocks (RHOG or C-HOG) and normalize, Descriptor - HOGs of all blocks within the window.
2. Training and classification- Dataset is split intro training and test sets, HOG descriptors formed, Support Vector Machine (SVM) training, Classification using the trained SVM

1.2 Thesis Outline

The structure of the paper is as follows. Section 2 presents the related work as my system and the drawbacks. In section 3, the technical overview of my system i.e. the details of the algorithms and methods used are explained. The detailed procedures of how I applied the algorithms and designed my system are mentioned in section 4. In section 5, the result and analysis is given. Finally the conclusion and future work is mentioned in section 6.

2. Background of Research

2.1 Previous Work

In recent studies, the detection and recognition of traffic signs have been developed in many research centers. A vision system for the traffic sign recognition and integrated autonomous vehicle was developed as part of the European research project PROMETHEUS at DAIMLER-BENZ Research Center [2]. Moreover, many techniques have been developed for road sign recognition such as fuzzy approach, spatial K-Means clustering algorithm, SOM etc. A genetic algorithm was also proposed by Aoyagi and Askura to identify road sign from gray-level images. But in all cases the accuracy and performance were not of the highest standard and hence it is still a major concern and important topic of discussion in emerging science. Although there are many approaches in order to segment color such as SOM, probabilistic method, soft computing techniques etc, all have their disadvantages as well specially the accuracy is very poor. Therefore the main objective of my paper is to reduce the search space and indicate only potential regions for increasing the efficiency and speed of the system. Apart from color segmentation, my paper also extracts features from warning traffic signs based on HOG. HOG was mainly used by Dalal and Triggs for pedestrian detection but my paper applies the principle of HOG to extract the features which is a completely new and different approach, and there is no such related work done in the past.

2.2 Review of Color Segmentation

This section describes some of the approaches of color segmentation.

2.2.1 Histogram Thresholding

Histogram Thresholing is mainly used for monochrome image segmentation. It works on different gray level of images. This method assumes that a histogram is divided into two main classes- the background and the foreground. It then tries to find the optimum threshold level that divides the histogram in two classes. This method weighs the histogram, checks which of the two sides is heavier and removes weight from the heavier side until it becomes the lighter. The operation is repeated until the edges of the weighing scale meet. The major drawback of this method is that it cannot deal with very noisy images [3].

2.2.2 Clustering Methods

The k- means algorithm is an iterative technique that partitions an image into k- clusters [4]. The algorithm is as follows-

1. Pick *K* cluster centers, either randomly or based on some heuristic
2. Assign each pixel in the image to the cluster that minimizes the distance between the pixel and the cluster center
3. Re-compute the cluster centers by averaging all of the pixels in the cluster
4. Repeat steps 2 and 3 until convergence is attained (i.e. no pixels change clusters)

Although the algorithm guarantees to converge, but it may not return the optimal solution. The quality of the solution depends on the initial set of clusters and the value of K.

2.2.3 Edge Detection

Edge detection is a well-developed field on its own within image processing. Region boundaries and edges are closely related, since there is often a sharp adjustment in intensity at the region boundaries. Edge detection techniques have therefore been used as the base of another segmentation technique. The edges identified by edge detection are often disconnected. To segment an object from an image however, one needs closed region boundaries. The desired edges are the boundaries between such objects. Segmentation methods can also be applied to edges obtained from edge detectors. Lindeberg and Li developed an integrated method that segments edges into straight and curved edge segments for parts-based object recognition, based on a minimum description length (MDL) criterion that was optimized by a split-and-merge-like method with candidate breakpoints obtained from complementary junction cues to obtain more likely points at which to consider partitions into different segments.

2.2.4 Proposed System LVQ

Learning Vector Quantization (LVQ) network is a supervised learning approach that learns to recognize similar input vectors in such a way that neurons having place nearby to others in the neuron layer respond to similar input vectors. In LVQ the transformation of input vectors to target classes are chosen by the user. This algorithm has been applied to spotting and tracking human faces, and shows more robustness than other algorithms for the same task. Moreover the information extraction is an unresolved issue till date. So my paper shows a methodology that can be used to identify the signs completely in a fast and efficient way.

2.3 Review of HOG

HOG is a feature descriptor used for object detection. It uses the concept of gradient orientation in order to find the position of an object. Navneet Dalal and Bill Triggs, researchers for the French National Institute for Research in Computer Science and Control (INRIA), first described Histogram of Oriented Gradient descriptors in their June 2005 CVPR paper [5]. In this work they focused their algorithm on the problem of pedestrian detection in static images, although since then they expanded their tests to include human detection in film and video, as well as to a variety of common animals and vehicles in static imagery.

3. Technical Overview

3.1 Segmentation

Image segmentation is the process of partitioning a digital image into multiple homogeneous regions based on some properties in order to analyze it where each region shares some common characteristics such as intensity, color, texture etc. In my paper segmentation is done to separate red/ yellow/ blue color from the rest as these colors are used to represent a traffic sign. Either of the colors is used based on the sign's global position or type, so segmentation is done based on in which country or region the algorithm is run. For instance, I have segmented red portion as in Bangladesh, this color is used for triangular signs.

3.2 RGB vs HSV

RGB- Cones of retina perceive color of 3 visual pigments- red, blue and green, and compare their intensities. The colors are represented as a cube where the vertices on the axes represent the primary colors and remaining vertices represent the complimentary for each of the primary colors.

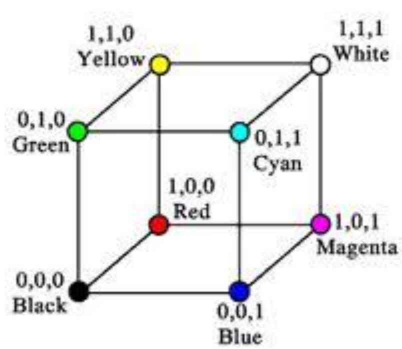


FIG: 2.1 RGB Color Model

HSV- The boundary of the hexagon represents various hues and it is used as the top of the HSV hex-cone. Saturation is measured along a horizontal axis and value along a vertical axis through the center of the hex-cone. Hue is represented as an angle about the vertical axis from 0-360 degrees. Vertices of the hex-cone are separated by 60^0 intervals.

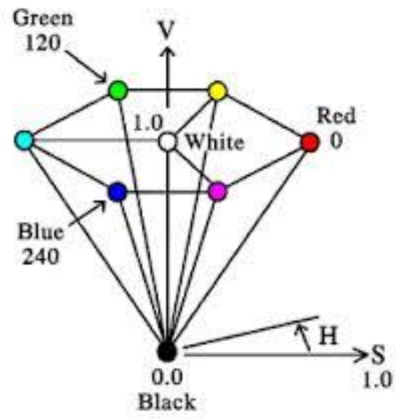


FIG: 2.2 HSV Color Model

3.3 RGB to HSV Conversion

The RGB values are divided by 255 to change the range from 0..255 to 0..1:

R' = R/255

G' = G/255

B' = B/255

Cmax = max(R', G', B')

Cmin = min(R', G', B')

Δ = Cmax - Cmin

Hue calculation:

$$H = \begin{cases} 60^\circ \times \left(\frac{G'-B'}{\Delta} mod 6\right) & , Cmax = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & , Cmax = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & , Cmax = B' \end{cases}$$

Saturation calculation:

$$S = \begin{cases} 0 & , \Delta = 0 \\ \frac{\Delta}{Cmax} & , \Delta <> 0 \end{cases}$$

Value calculation:

V = Cmax

3.4 Learning Vector Quantization

LVQ is an algorithm based on feed forward neural network and potential applications include pattern recognition, speech recognition, image processing etc. A neural network is a network of simulated neurons that can be used to recognize instances of patterns. NNs learn by searching through a space of network weights. LVQ has 2 layers- competitive and linear layer. Competitive layer's classes are called subclasses and linear layer's classes are called target classes. Competitive layer's neurons are equal to the number of subclasses and linear layer's neurons are equal to target classes. Besides LVQ has one hidden layer of neurons connected with input layer and an output layer of neurons. First-layer weights are initialized with midpoint. The second-layer weights are set so that each output neuron i has unit weights coming to it from some percent of the hidden neurons. It uses target output classification for each input data set and that is why it is called supervised method. For my paper, inputs of LVQ are connected directly with the pixel-vector of the image I (in RGB format) and its outputs are connected directly to the decision function fd, which produces an output of 1 or 0 depending of if the color corresponds to the object to be segmented forming in this way a new vector image I' [6].

The LVQ network architecture is shown below.

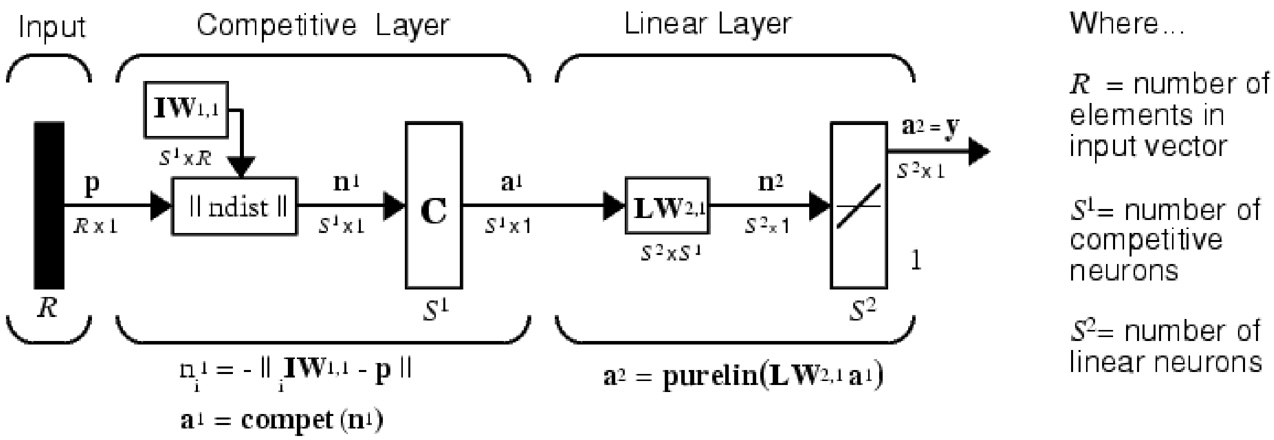


FIG: 3.1 LVQ Network

An LVQ network has a first competitive layer and a second linear layer. The linear layer transforms the competitive layer's classes into target classifications defined by the user. We refer to the classes learned by the competitive layer as *subclasses* and the classes of the linear layer as *target classes*.

Both the competitive and linear layers have one neuron per (sub or target) class. Thus, the competitive layer can learn up to S^1 subclasses. These, in turn, are combined by the linear layer to form S^2 target classes. (S^1 is always larger than S^2.)

For example, suppose neurons 1, 2, and 3 in the competitive layer all learn subclasses of the input space that belongs to the linear layer target class No. 2. Then competitive neurons 1, 2, and 3, will have $LW^{2,1}$ weights of 1.0 to neuron n^2 in the linear layer, and weights of 0 to all other linear neurons. Thus, the linear neuron produces a 1 if any of the three competitive neurons (1,2, and 3) win the competition and output a 1. This is how the subclasses of the competitive layer are combined into target classes in the linear layer.

In short, a 1 in the i^{th} row of a^1 (the rest to the elements of a^1 will be zero) effectively picks the i^{th} column of $LW^{2,1}$ as the network output. Each such column contains a single 1, corresponding to a specific class. Thus, subclass 1s from layer 1 get put into various classes, by the $LW^{2,1}a^1$ multiplication in layer 2.

We know ahead of time what fraction of the layer 1 neurons should be classified into the various class outputs of layer 2, so we can specify the elements of $LW^{2,1}$ at the start. However, we have to go through a training procedure to get the first layer to produce the correct subclass output for each vector of the training set.

3.5 Hough Transform

The Hough Transform is a feature extraction technique which transforms an image from its Cartesian coordinates to its Polar form. Consider a line segment y=mx+b in the image space. The main idea of Hough Transform is to consider this line segment by its parameters *m* and *b*. For a vertical line *m* becomes infinity. To avoid this problem, the line segment can be considered by $r=x \cos \vartheta + y \sin \vartheta$ where $r \in R$ is a vector representing the shortest distance of the line from the origin and $\vartheta \in [0, \pi]$ is the angle of this vector from the *x*-axis. Any point in the image space is represented by a sinusoidal curve in the Hough space. Moreover, two points on a line segment generate two curves which are superimposed at a location which corresponds to a line in the image space.

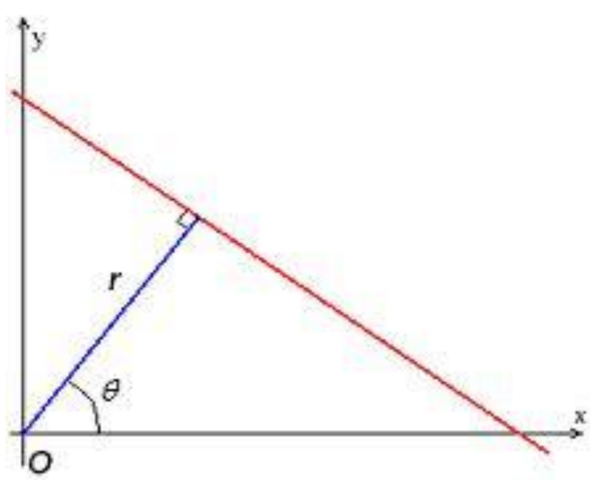


FIG: 3.2 Line segment representation

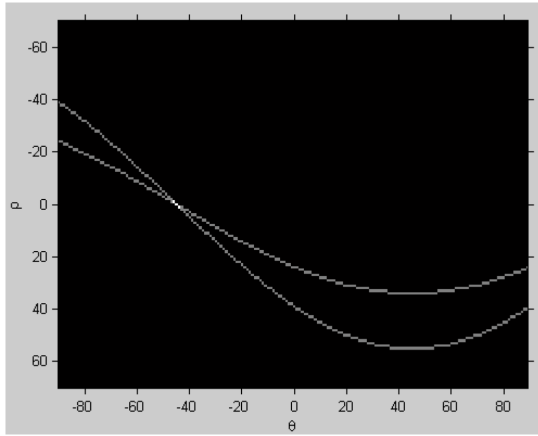


FIG: 3.3 Hough Transform of Two Points on the Line

3.6 Histogram Oriented Gradient

It is a feature descriptor used for object detection. It uses gradient orientation in localized portions of an image. The steps of this method are discussed below and Figure 3.4 shows the steps in detail.

1. Compute Gradient Values- Apply 1D centered, point discrete derivative mask in one or both sides of horizontal and vertical directions.
2. Orientation Binding- It involves creating the cell histograms. Each pixel within a cell casts a weighted vote for an oriented- based histogram channel based on the values found in the gradient computation. For vote weight, pixel contribution is gradient magnitude itself.
3. Descriptor Blocks- To adapt with the changes in illumination and contrast, the gradient strengths must be locally normalized, which requires grouping the cells together into larger spatially connected blocks. The HOG descriptor is then the vector of the components normalized cell histograms from all of the block regions. These blocks typically overlap meaning that each cell contributes more than once to the final descriptor. There are 2 block geometries-

- Rectangular R-HOG blocks- these are square grids with 3 parameters: number of cells per block, number of pixels per cell and number of channels per cell histogram.
- Circular C- HOG blocks- There are 4 parameters: number of angular and radial bins, radius of center bin, and expansion factor for radius of additional radial bins.

4. Block Normalization- Dalal and Triggs explore four different methods for block normalization. Let v be the non-normalized vector containing all histograms in a given block, $\|v\|_k$ be its *k*-norm for $k = 1, 2$ and e be some small constant (the exact value, hopefully, is unimportant). Then the normalization factor can be one of the following:

L2-norm: $$f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}}$$

L2-hys: L2-norm followed by clipping (limiting the maximum values of v to 0.2) and renormalizing, as in

L1-norm: $$f = \frac{v}{(\|v\|_1 + e)}$$

L1-sqrt: $$f = \sqrt{\frac{v}{(\|v\|_1 + e)}}$$

In addition, the scheme L2-Hys can be computed by first taking the L2-norm, clipping the result, and then renormalizing. In their experiments, Dalal and Triggs found the L2-Hys, L2-norm, and L1-sqrt schemes provide similar performance, while the L1-norm provides slightly less reliable performance; however, all four methods showed very significant improvement over the non-normalized data.

5. SVM Classifier- The descriptors are fed into some recognition system based on supervised learning. The SVM classifier is a binary classifier which looks for an optimal hyper plane as a decision function. Once trained on images containing some particular object, SVM can make decisions regarding the presence of an object.

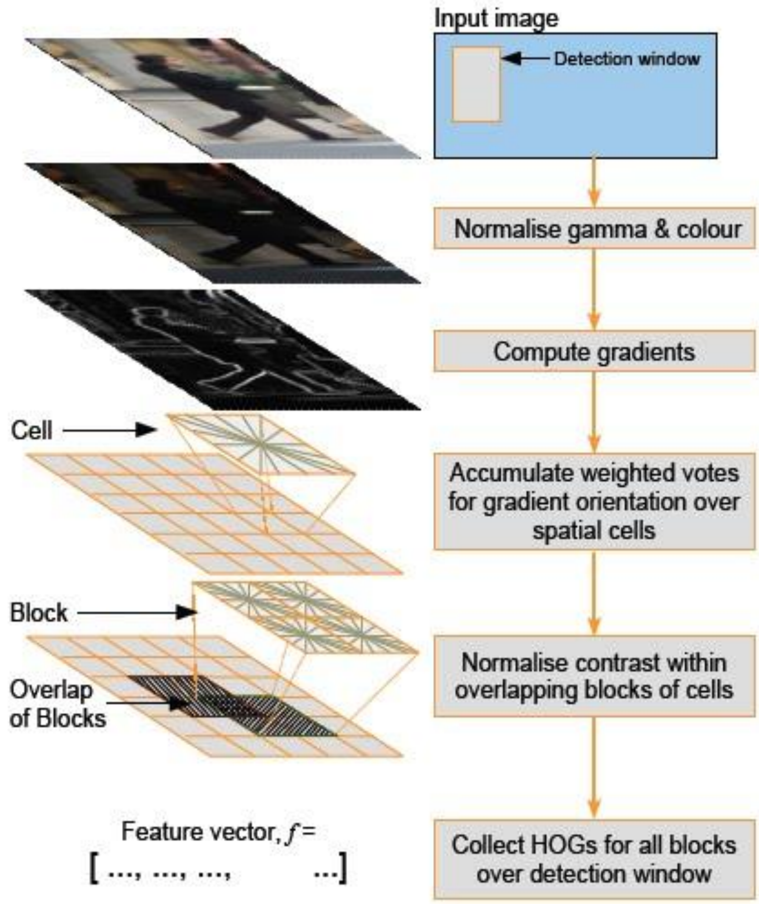


FIG: 3.4 An overview of static HOG Feature Extraction. The Detector Window is tiled with a Grid of Overlapping Blocks. Each Block Contains a Grid of Spatial Cells. For Each Cell the Weighted Vote of Image Gradients in Orientation Histograms is performed. These are Locally Normalized and Collected in One Big Feature Vector

3.7 Support Vector Machine

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces [9].

4. System Design

4.1 Tools and Methodology

Tools used are MATLABR2009a and Image Processing Toolbox 7

Algorithms used are LVQ, Canny Edge Detection, Hough transformation, Bounding Box, HOG etc.

Methodology- Data collection, input vector and target class based on hue and saturation, network formation and training, red/blue/yellow/green color segmentation, edge detection, object detection, triangle detection, information extraction (whether a turn/ speed-breaker/ no- parking sign).

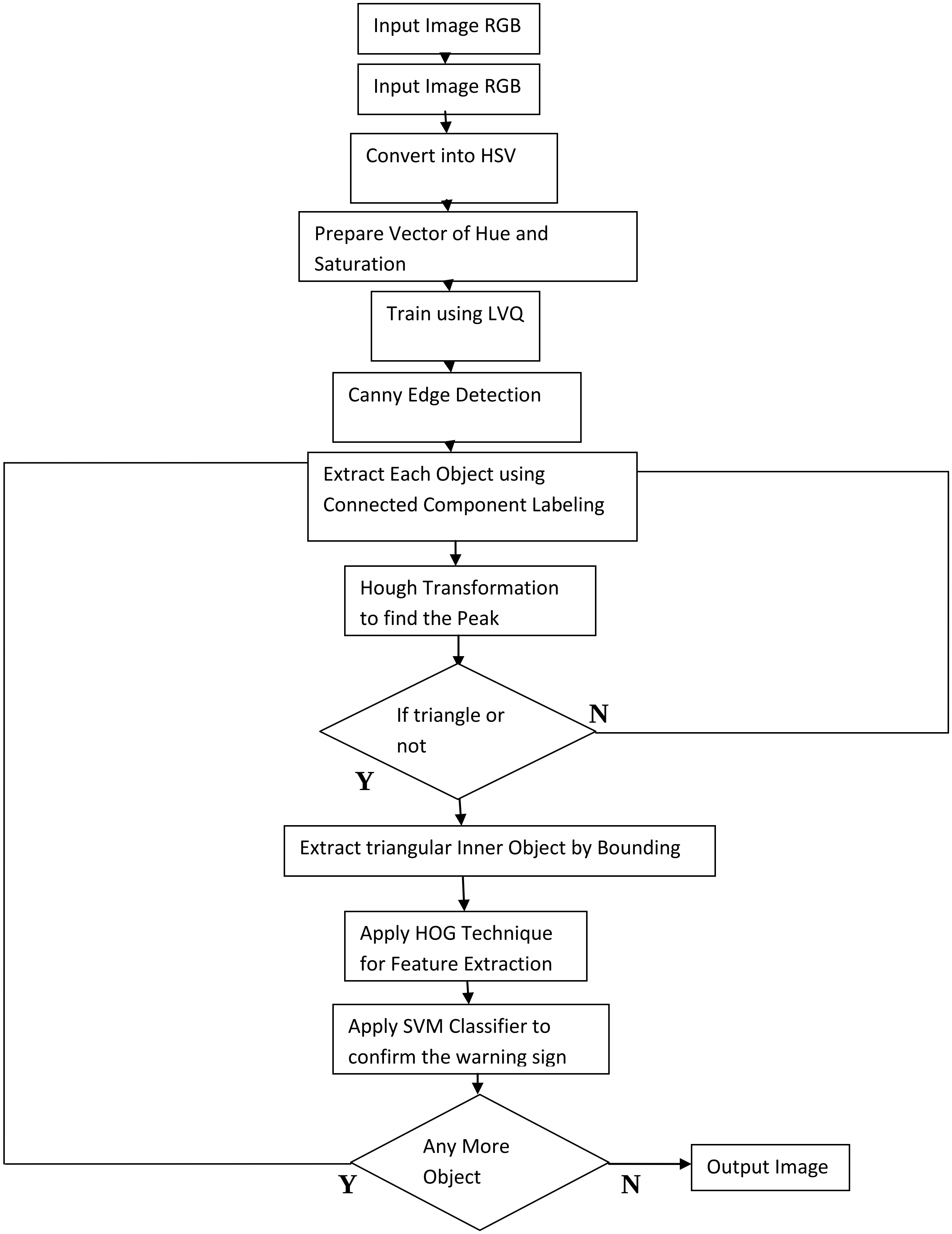


FIG: 4.1 Flow-chart of Steps

4.2 Data Collection

In the field of image processing, data collection is the most crucial part because if the images are not accurate, we will not get proper output. In my paper, the sample images were collected across different nations in a large variety of weather conditions. Some images were in poor weather conditions such as snowy or rainy, some were in bad light geometry and so on. The number of samples could not be more than 200 as the amount of triangular signs is limited. After the images were collected, blocks of pixels with different sizes were extracted from different digital images as shown below-

FIG: 4.2 Block of Pixel Extracted From JPG Files

4.3 Creating the input vector

Following steps were executed-

1. All images were taken one by one.
2. Their size was calculated in the form of rows and columns

3. The RGB images were converted into HSV
4. The input vectors H and S for each pixel of each image were prepared by defining arrays and putting one after the other
5. An input vector P was created having 2 vectors, H and S
6. A target vector was created for each input vector's indices having true class among 1, 2 and 3 for red, blue and yellow color. Its length is equal to the length of H or S

4.4 Formation and Training of LVQ neural network

The LVQ net created has two inputs with six neurons in the competitive layer as there are two input vectors and number of classes for each is three, three neurons in the learning layer with default learning rate and default learning function.

The function for creating the net is:

net = newlvq(PR,S1,PC,LR,LF)

where

PR= an R x 2 matrix of min and max values for R input elements

S1= Number of hidden neurons

PC= S2 element vector of typical class percentages

LR= Learning rate (default = 0.01)

LF= Learning function (default = 'learnlv2')

After creating the neural network, it needs to be trained with respect to the target value. The training function is given below:

Train (net,P,T,Pi,Ai) takes

Net= Network

P= Network inputs

T= Network targets (default = zeros)

Pi= Initial input delay conditions (default = zeros)

Ai= Initial layer delay conditions (default = zeros)

and returns new network.

4.5 Triangle Detection using Hough Transformation

The Hough Transform in its original form is not suitable for complex object detection such as triangles, circles, and rectangles due to noise and shape imperfection. Consider a perfect equilateral triangle shown in Figure 4.3. Since the triangle has three sides, there are three points of intersection in the Hough space which are represented by θ_a, θ_b and θ_c where $\theta_a > \theta_b > \theta_c$. The corresponding normal distances are ρ_a , ρ_b and ρ_c.

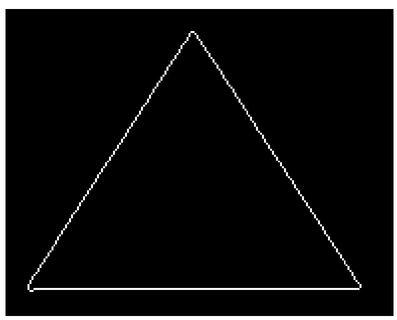


FIG: 4.3 A Perfect Equilateral Triangle

$|\vartheta_a - \vartheta_b| >= T_1$, $|\vartheta_b - \vartheta_c| >= T_2$,

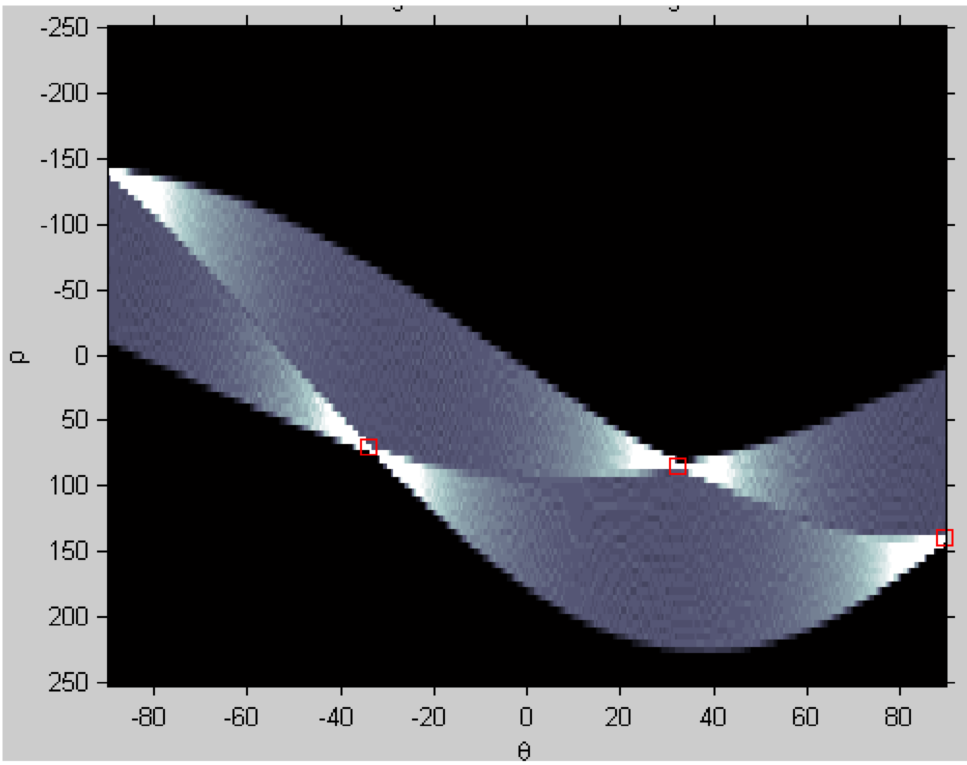


FIG: 4.4 Hough Transform of the Triangle

Consider the triangle depicted in Figure 4.5 which is located in the standard position. It contains one circle whose center is located in the centroid of this triangle. Let *a*, *b*, and *c* be the lengths of the sides *BC*, *AC*, and *AB*, respectively.

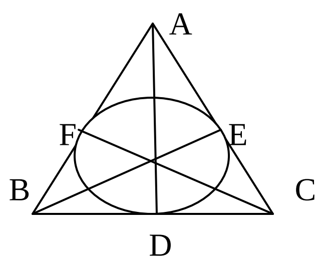


FIG: 4.5 Equilateral Triangle with a Circle

The set of rules to detect any triangle in the image are as follows:

1. Rule 1: The difference between two adjacent ϑs is more than a certain threshold. For a perfect equilateral triangle $T1=T2=60^{o}$.

$|\vartheta_a - \vartheta_b| >= T_1$, $|\vartheta_b - \vartheta_c| >= T_2$,

$|\vartheta_a - \vartheta_c| >= T_1 + T_2$

2. Rule 2: The summation of differences between any two adjacent angles equals the difference between ϑ_a and ϑ_c. This rule minimizes the number of false positive objects created in the image.

$| \vartheta_a - \vartheta_c | = | \vartheta_a - \vartheta_b | + | \vartheta_b - \vartheta_c |$

3. Rule 3: The summation of any two sides of a triangle is always greater than the third side

$a+b>c, a+c>b , b+c>a$

4. Rule 4: The normal distances ρ_a , ρ_b and ρ_c are those indicated by OD, OE, and OF in Figure 4.3. The algebraic difference between any two normal distances is less than a threshold.

$| \rho_a - \rho_b | <T_3, | \rho_b - \rho_c | <T_3, | \rho_a - \rho_c | <T_3$

5. Rule 5: The summation of three angles of the triangle is π. Therefore, the difference between any two angles is less than π.

$\vartheta_a - \vartheta_b <= \pi, \vartheta_b - \vartheta_c <= \pi, \vartheta_a - \vartheta_c <= \pi$

6. Rule 6:

$|a^2-(b^2+ c^2 - 2bc \cos (\pi- \vartheta_b + \vartheta_c))| < T_4$

$|b^2-(a^2+ c^2 - 2ac \cos (\pi- \vartheta_a + \vartheta_b))| < T_4$

$|c^2-(b^2+ a^2 - 2ab \cos (\vartheta_a - \vartheta_c -\pi))| < T_4$

7. Rule 7:

- $|a-(\rho_a/\tan(0.5(\pi- \vartheta_a + \vartheta_b))) - (\rho_a/\tan(0.5(\vartheta_a + \vartheta_c- \pi)))| < T_5$
- $|b-(\rho_b/\tan(0.5(\pi- \vartheta_b+ \vartheta_c))) - (\rho_b/\tan(0.5(\vartheta_a + \vartheta_c- \pi)))| < T_5$
- $|c-(\rho_c/\tan(0.5(\pi- \vartheta_b + \vartheta_c))) - (\rho_c/\tan(0.5(\pi- \vartheta_a + \vartheta_b)))| < T_5$

4.5.1 Detection of Warning Signs

The detection of a warning traffic sign is not exactly the detection of a triangle. In the simple case, Canny edge detection generates two triangles inside each other as depicted in Figure 4.5. Moreover, the inner triangle shown in Figure 4.5 consists of a number of broken lines which makes the detection task more difficult.

FIG: 4.6 Detection of a Warning Traffic Sign

Based on the number of Hough peaks and their distribution, one of the following three scenarios will be employed to detect Warning traffic signs:

4.5.1.1 Scenario 1:

If the base of the triangle is located horizontally then the Hough Transform shows three Hough peaks which mean three line segments. If these three line segments satisfy the rules mentioned in 6.2, then they form a triangle.

4.5.1.2 Scenario 2:

If the base of the triangle is not horizontal then it is shown as a set of broken lines. The Hough Transform in this case shows more than three Hough peaks which mean more than three line segments, Figure 4.7. The selection of proper Hough peaks is achieved as follows:

- Search through the ϑs and select the maximum ϑ as ϑ_a and the minimum as ϑ_c.
- Search in the list among the remaining ϑs and select ϑ_b - which fulfill rule 2.
- Check whether line segments fulfill the other rules.

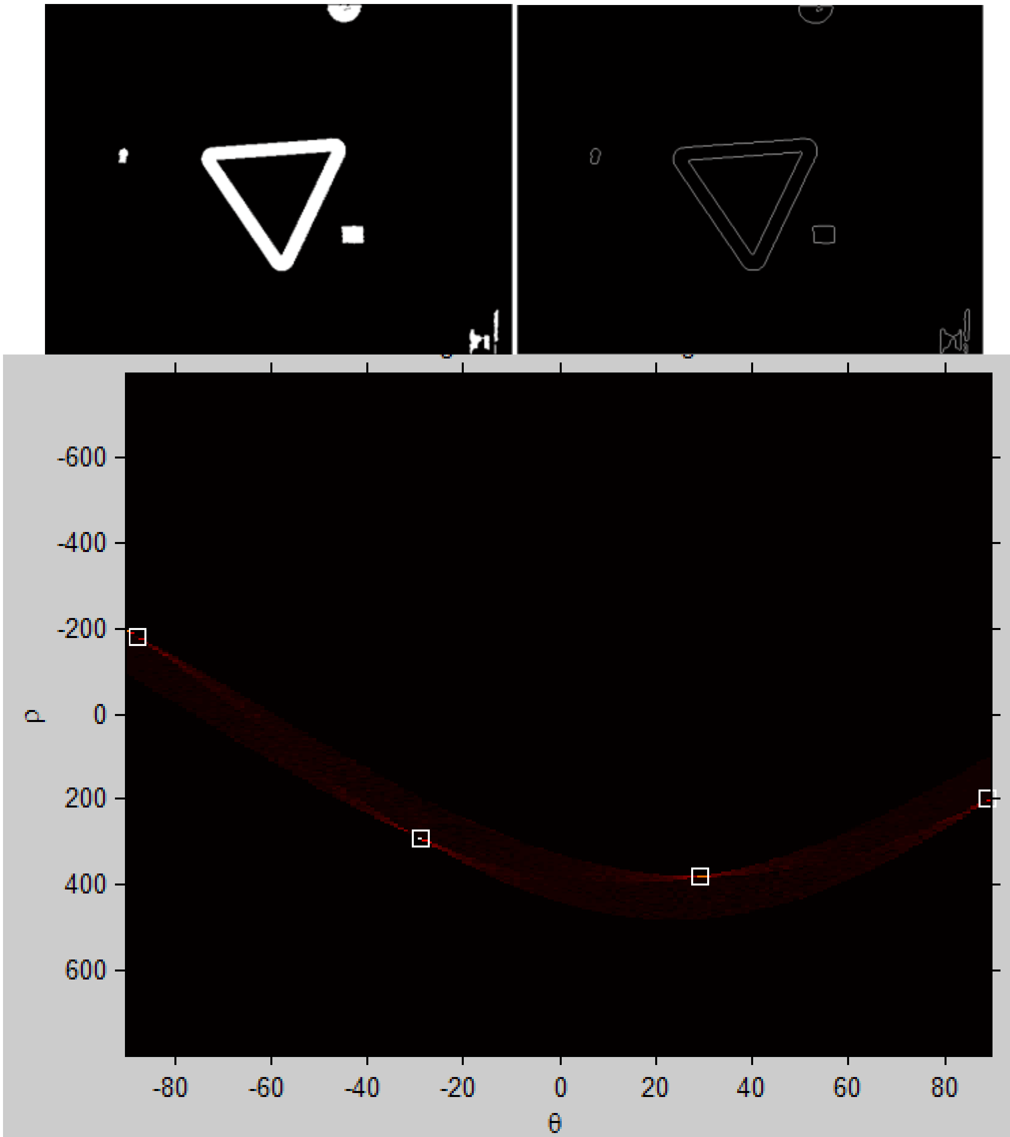


FIG: 4.7 Detection of Rotated Warning Signs

4.5.1.3 Scenario 3:

When the image is noisy, color segmentation will produce objects which are highly imperfect. The Hough Transform in this case shows many Hough peaks which in most cases are clustered in several places on the Hough space, as shown in Figure 4.8. To deal with this scenario, all Hough peaks within a certain threshold distance are averaged and replaced by the resultant one. Then follow scenario 2 to select the suitable Hough peaks among the list of generated ones.

As mentioned in the beginning of this section Canny edge detection generates, in the best case, two triangles inside each other. However, in many cases, one of the two triangles is missing because of imperfection, noise, and the presence of other objects with the same color as the traffic sign. Figure 4.8 depicts such cases.

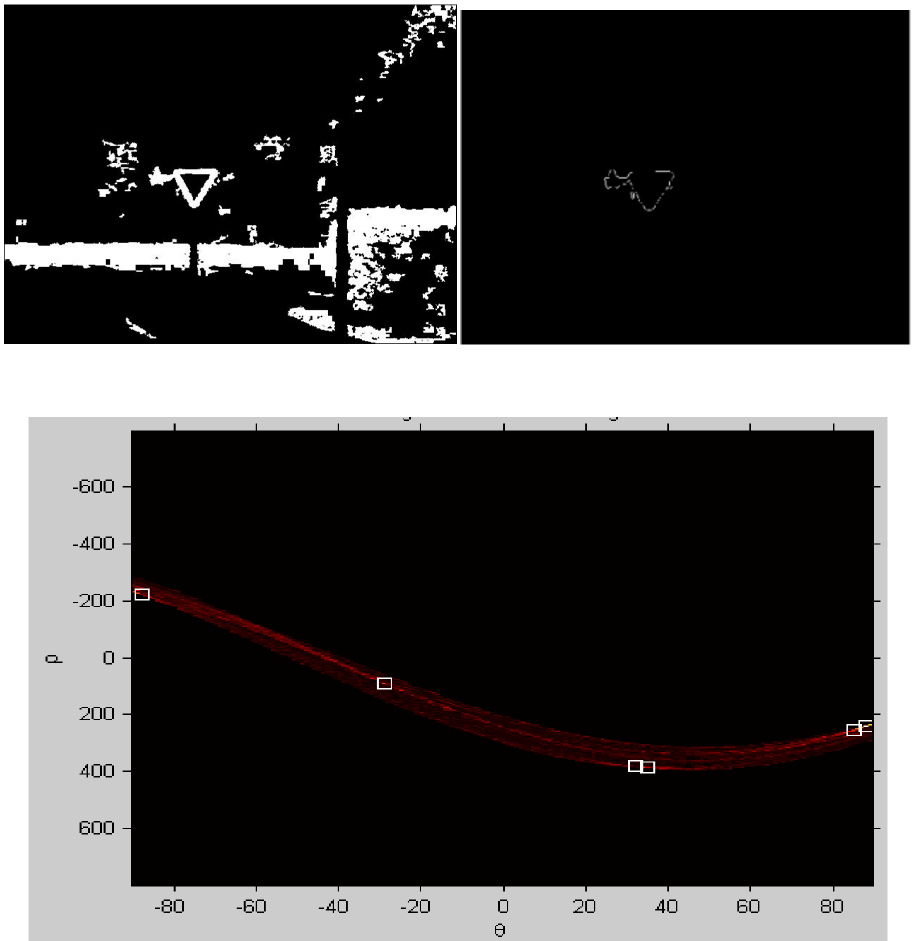


FIG: 4.8 Detection of Imperfect Warning Signs.

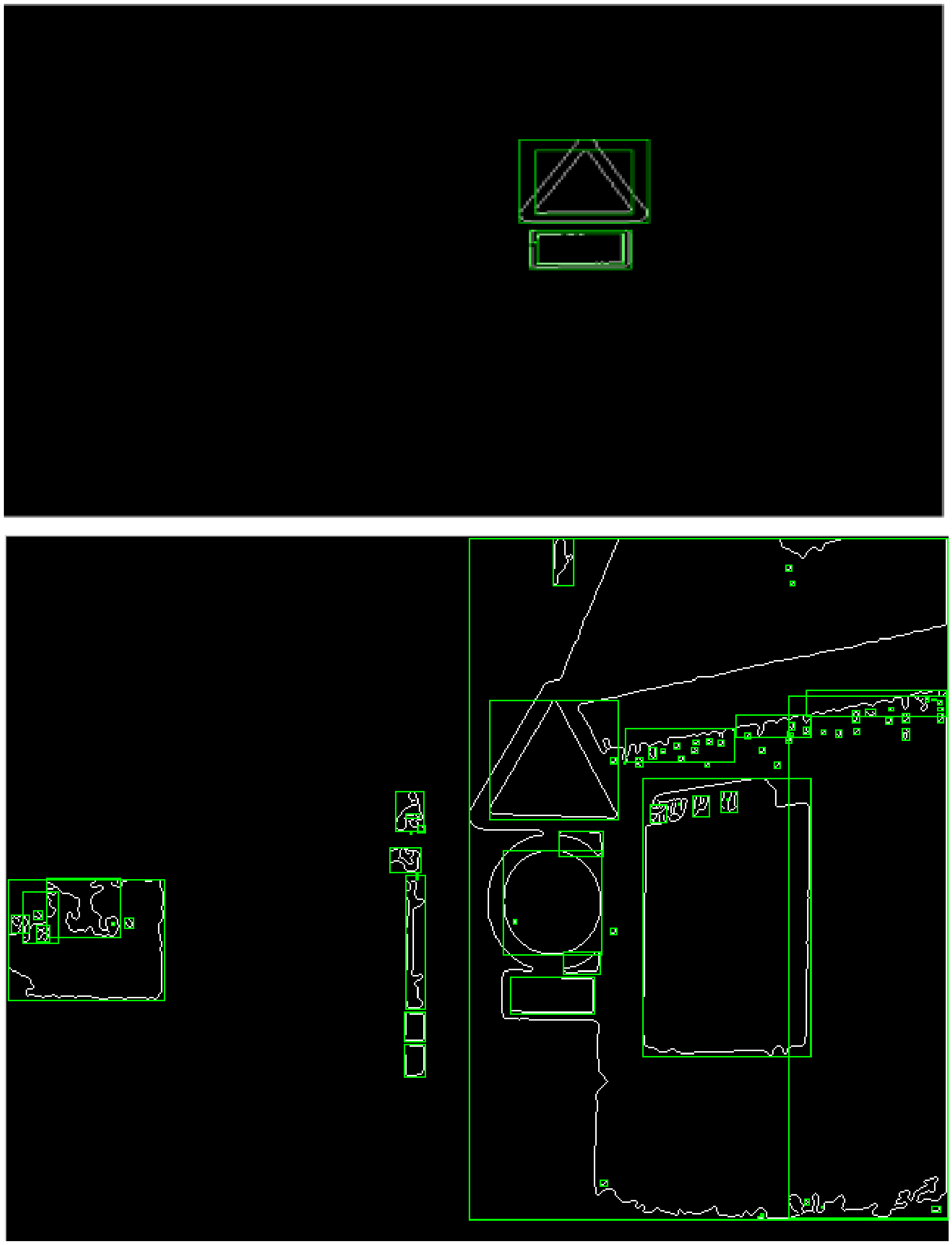


FIG: 4.9 Imperfection of Triangle Edges.

To solve this problem, the following procedure is applied:

- If two triangles, which are concentrated in each other, are detected, then this mean that one warning traffic sign is detected in this location and a bounding box is drawn.
- If one triangle which is located inside another object is detected, then this means that the outer edge of the triangle is destroyed for some reason and a bounding box based on the detection of the inner edge of the warning sign is drawn.

- If one triangle which is not located inside another object is detected, then this means that the inner edge of the triangle is destroyed and a bounding box based on the detection of the outer edge of the warning sign is drawn.

4.6 Feature extraction using HOG

The steps are as follows [7].

1. Compute horizontal and vertical gradients
2. Compute gradient orientation and magnitude. For color image, pick highest gradient magnitude color channel for each pixel.
3. The images are resized into 48*48
4. Divide into 6*6 blocks with 50% overlapping, thus 5*5=25 blocks in total
5. Each block contains 2*2 cells with size 8*8
6. Quantize gradient orientation into 9 bins-
    - The vote is the gradient magnitude
    - Interpolate votes bi-linearly between neighboring bin center. For example, if $Ø=85^0$ and bins are 70 and 90, then distance to bin centers are 15 and 5 respectively. Hence ratios are 5/20 and 15/20
    - The vote can also be weighted with Gaussian to down-weight the pixels near the edges of the block
7. Concatenate histograms in 1D matrix of length 14400 (4*9*5*5*16) i.e. the feature dimension.

5. Experiment and Result Analysis

5.1 LVQ Color Segmentation

Images are taken in different weather conditions. The table below shows the number of images along with the success rate of segmentation and Figure 5.4 shows the segmentation of the images that were obtained through the experiment.

Table 1

Results of LVQ based Color Segmentation

Weather Type	**Number of Images**	**Success %**
Sunny	28	98
Cloudy	20	95
Snowy	15	91
Foggy	20	94
Vague	15	93

5.2 LVQ Training Parameters

Input vectors= 2

Number of subclasses= 3

Number of hidden neurons= 6

Number of target classes=3

Default learning rate=0.01

Default learning function=learnlv1

Training epochs= 6

From the above parameters, the LVQ network has 2 input vectors, H(Hue) and S(Saturation). The network has three subclasses and two input vectors. So total number of neurons in the competitive layer is six and in learning layer is three as it extracts red, blue and yellow colors. The network was tested with different number of epochs. The performance of training depends on number of training sets. So preparing data for training should be done carefully.

5.3 Training

Training was done with six epochs. In training, number of epochs, number of training sets and quality of the data taken to prepare training sets are very important. If the number of epochs is increased, the network will return better results. In the same way if the number of training sets is increased, the network will learn more and will give better results.

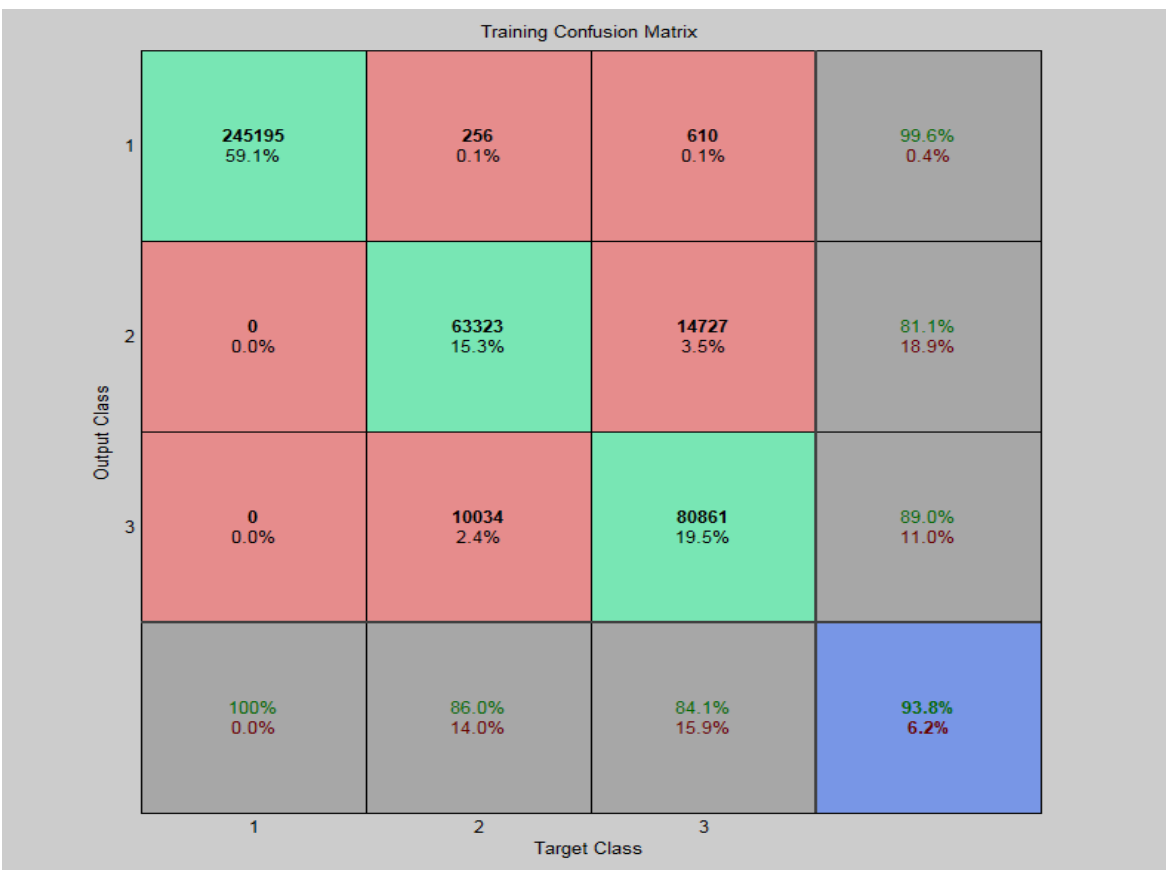


FIG: 5.1 Training Confusion Matrix

Here the coordinates are target class and output class. The network was trained on the base of target classes and the output classes are the outputs of the network. Class 1 belongs to blue, class 2 to yellow and class 3 to red. The results of the colors are 99.6%, 81.1% and 89% which is pretty good.

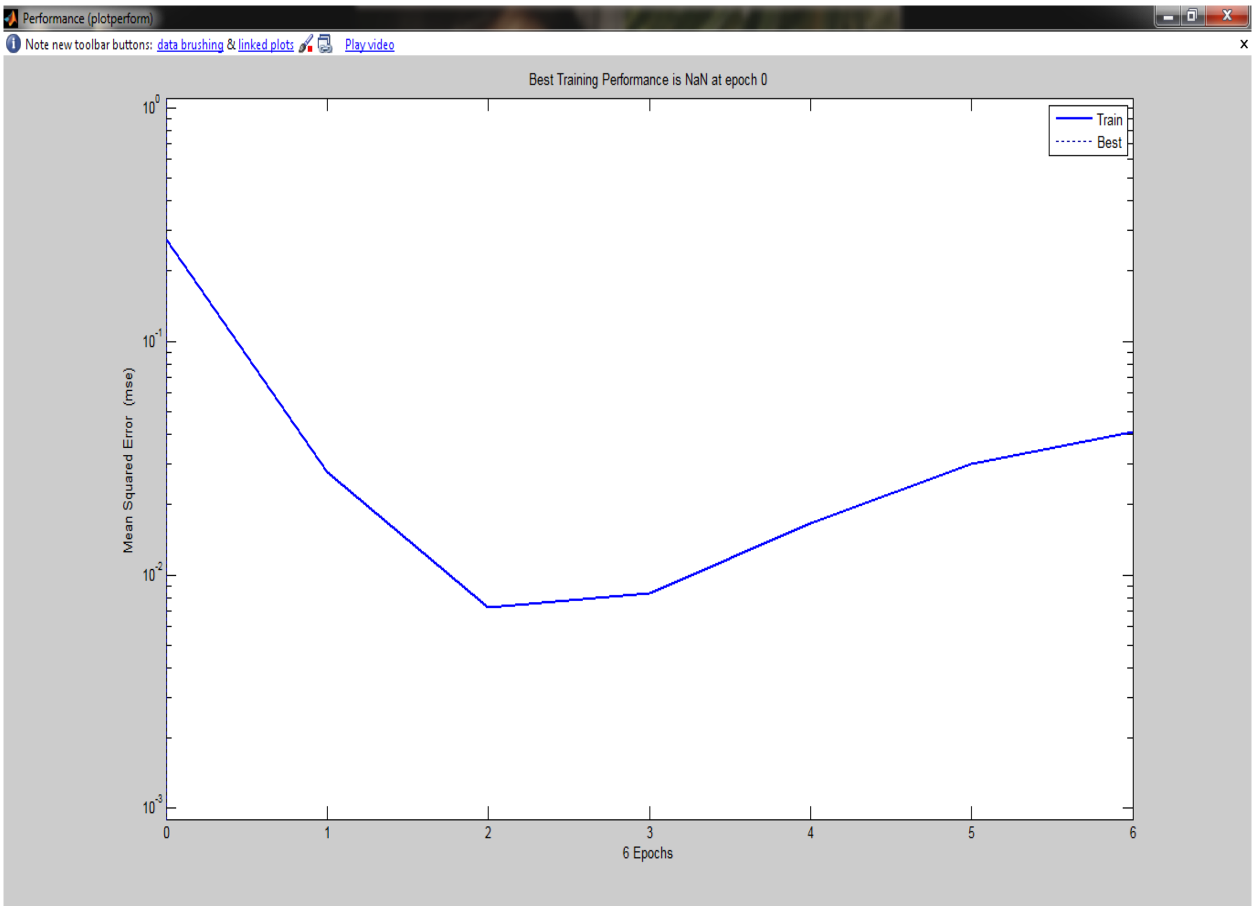


FIG: 5.2 The Training Performance

The figure shows the performance of the network in terms of Mean Square Error (MSE). During training it gained minimum MSE at epoch number two and then again increased slightly.

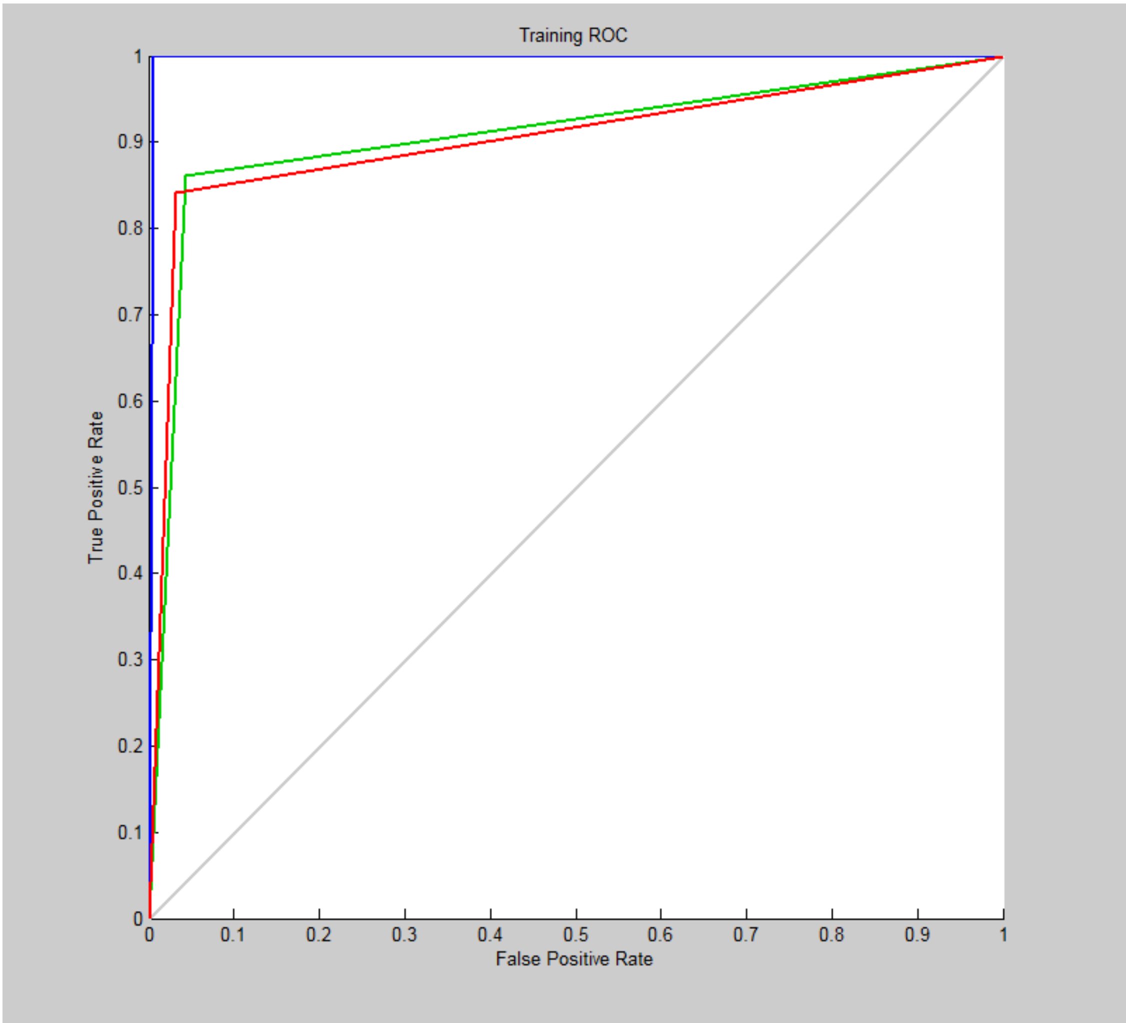


FIG: 5.3 Receiver Operating Characteristics

The figure shows True Positive Rate training against False Positive Rate and the graph shows quite satisfactory result.

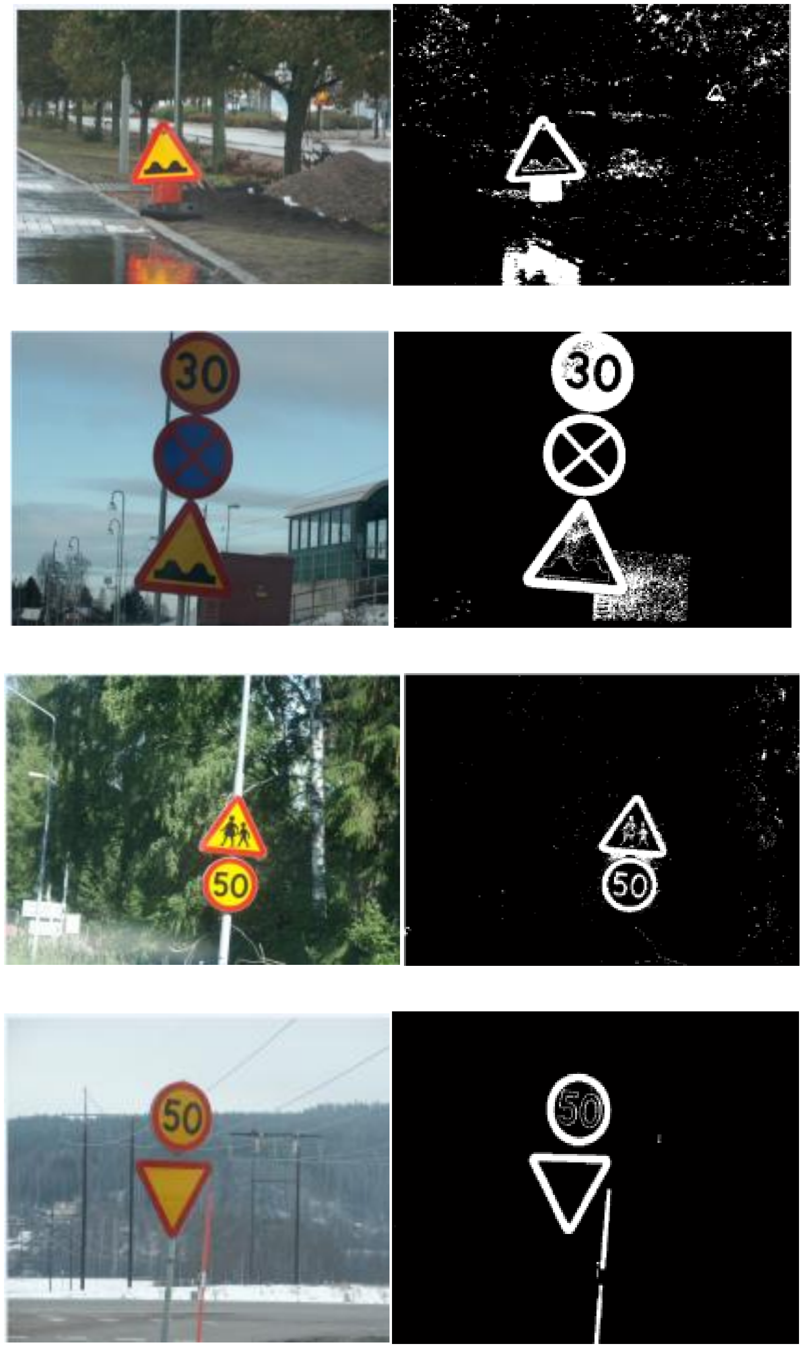


FIG: 5.4 Results of Color Segmentation Based on LVQ

5.4 Detection of Warning Signs

The proposed approach was tested on two different sets of images each of which comprises 57 images. The first set is the Yield sign and the second one consists of a combination of other warning signs. Table 2 depicts the detection results achieved under different light conditions. The average detection rate was 96.5%. The same test was employed for the other warning signs. The detection rate achieved for this set of signs was 94.7%. Table 3 presents the detection results of this set. Figure 5.5 shows a number of warning sign detected by this approach. Table 2 shows that the system produces one false positive in the case of Yield signs while Table 3 shows 2 false positives in the case of other warning signs.

Table 2

Detection of Yield Signs

Condition	Number of Images	Correct Detection	False+
Sunny	15	15	0
Highlight	5	5	0
Blurred	4	4	0
Bad Lighting	13	12	0
Noisy	6	6	1
Snow Fall	2	2	0
Fog	3	3	0
Dusk/Dawn	9	9	0
Total	57	56	1

Table 3

Detection of Other Warning Signs

Condition	Number of Images	Correct Detection	False+
Sunny	9	9	0
Highlight	4	4	0
Blurred	11	11	0
Bad Lighting	10	10	0
Noisy	12	11	1
Snow Fall	4	4	0
Fog	4	4	0
Dusk/Dawn	3	3	1
Total	57	56	2

FIG: 5.5 Results of Warning Sign Detection

5.5 Feature Extraction of Warning Signs

The HOG features were extracted based on five classes- class 1 for straight line, class 2 for speed breaker, class 3 for warn for school and class 4 for road under construction. If the input image does not match any of the above classes, then it falls under the 5th class which is 'unknown class'. Table 4 shows the number of images tested for each class and the success rate, and Figure 5.6 shows the extracted images.

Table 4

Results of HOG based Feature Extraction

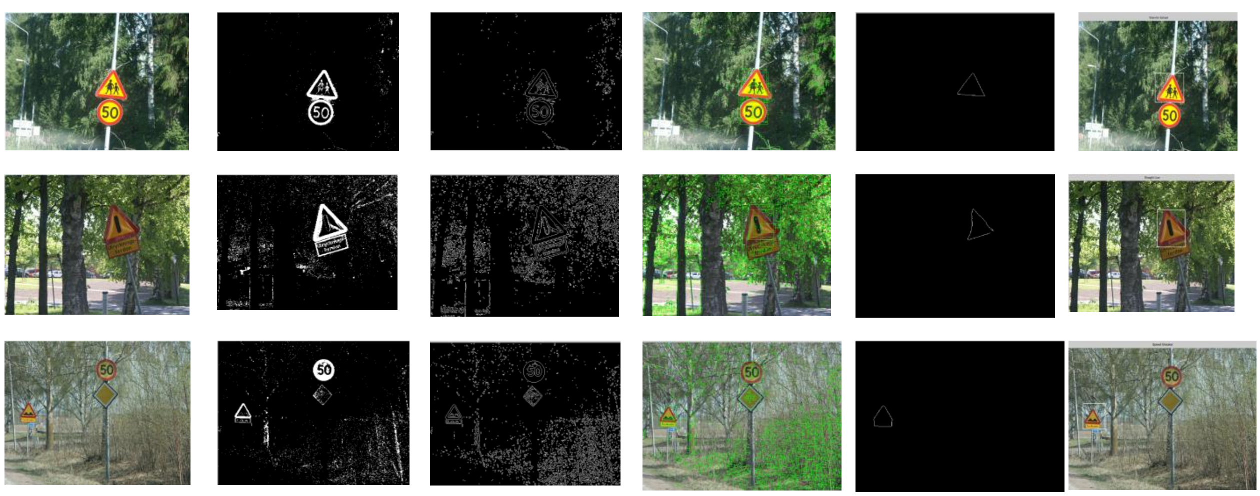
Class Type	**Number of Images**	**Success %**
1	15	95
2	15	93
3	10	93
4	10	91
5	10	90

FIG: 5.6 Feature Extraction of Warning Sign

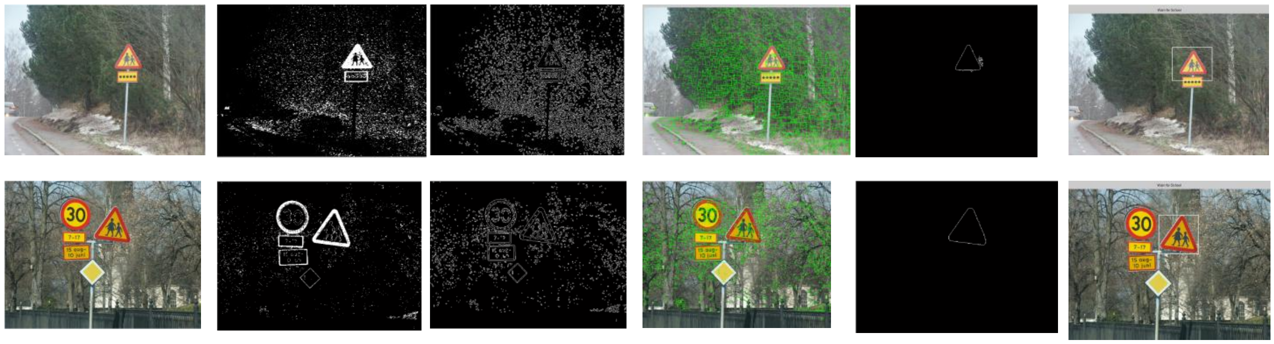
5.6 Final Output

Cloudy

Sunny

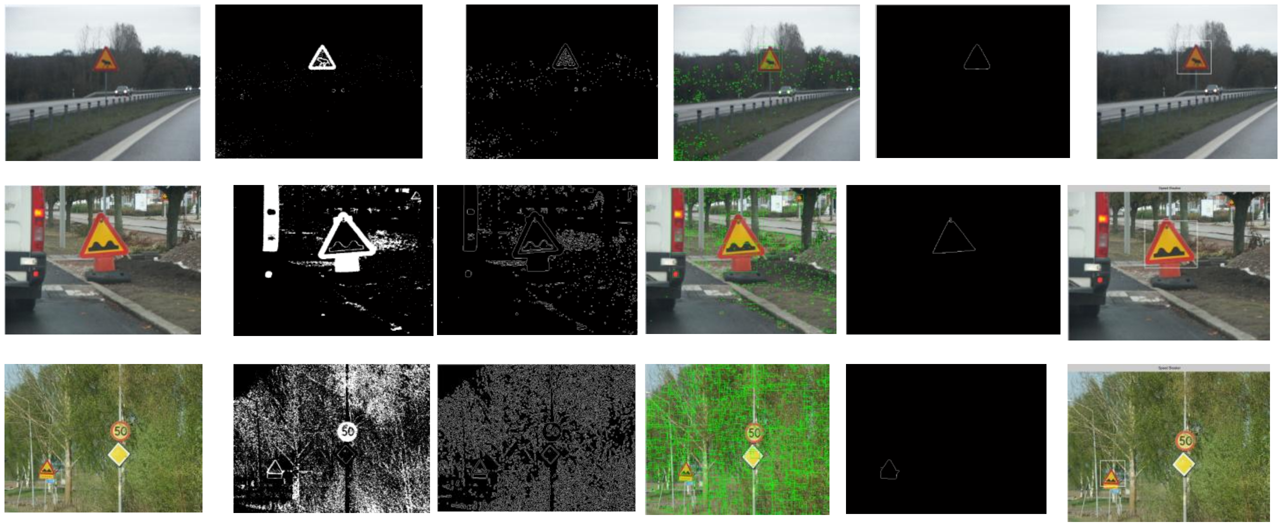


Foggy



Snowy

Vague



6. Conclusion and future work

This paper presents a new color segmentation approach for traffic sign recognition based on LVQ neural network. Moreover it tries to find a solution to the edge detection and information extraction based on Canny Edge, Hough Transformation and HOG. The biggest advantage of LVQ is that it works efficiently for images in a variety of conditions such as good/ bad light, foggy/ rainy/ sunny weather, for all the colors (red/blue/yellow) and images with different illumination across many countries. Moreover, the execution time of LVQ is very small because it directly works on image pixels. Not only that, the paper aims to detect the warning sign edges by using simple mathematical rules and Hough Transformation. The paper also found a satisfactory solution in determining the type of traffic sign using the concept of Dalal and Triggs. Future work will be to improve the performance by using the later versions of LVQ such as LVQ3 and extract more precise information from a traffic sign. Also, a better result for information extraction will be tried to achieve by increasing the number of cells and blocks for HOG. Moreover, signs will be detected based on videos rather than still images that will make the process of detecting traffic signs more efficient.

References

[1]. http://en.wikipedia.org/wiki/Traffic_sign

[2]. Traffic Sign Recognition Using Neural Network on OpenCV: Toward Intelligent Vehicle/Driver Assistance System Auranuch Lorsakul1 and Jackrit Suthakorn*, 2 Center for Biomedical and Robotics Technology (BART LAB), Department of Biomedical Engineering, Faculty of Engineering, Mahidol University 25/25 Putthamonthon 4 Road, Salaya, Putthamonthon, Nakornpathom, 73170 THAILAND

[3]. http://en.wikipedia.org/wiki/Balanced_histogram_thresholding

[4]. http://en.wikipedia.org/wiki/Image_segmentation

[5]. http://en.wikipedia.org/wiki/Histogram_of_oriented_gradients

[6]. Color Segmentation using LVQ, Learning Vector Quantization, Jabbar Hussain. 2011-01-10

[7]. Review and MATLAB implementation of 'Histogram og Oriented Gradients for Human Detection' by Dalal and Triggs

[8].http://users.du.se/~hfl/traffic_signs/Color%20segmentation/LVQ%20Color%20Segmentation

[9] http://en.wikipedia.org/wiki/Support_vector_machine