Real-Time Recognition System For Traffic Signs

M. Taha Khan

2008

Master Thesis
Computer Engineering
Nr: E3610D

HÖGSKOLAN Dalarna

DEGREE PROJECT
Computer Engineering

Program **Master of Science in Computer Engineering**	Reg Number **E3610D**	Extent **30 ECTS**
Name of Student **M. Taha Khan**	Year-Month-Day **2008-10-22**	
Supervisor **Hasan Fleyeh**	Examiner **Mark Dougherty**	
Company/Department **Department of Computer Engineering, Dalarna University**	Supervisor at Company/Department **Hasan Fleyeh**	
Title **Real-Time Recognition System for Traffic Signs**		

ABSTRACT

The aim of this thesis project is to develop the Traffic Sign Recognition algorithm for real time. In real time environment, vehicles move at high speed on roads. For the vehicle intelligent system it becomes essential to detect, process and recognize the traffic sign which is coming in front of vehicle with high relative velocity, at the right time, so that the driver would be able to pro-act simultaneously on instructions given in the Traffic Sign. The system assists drivers about traffic signs they did not recognize before passing them. With the Traffic Sign Recognition system, the vehicle becomes aware of the traffic environment and reacts according to the situation.

The objective of the project is to develop a system which can recognize the traffic signs in real time. The three target parameters are the system's response time in real-time video streaming, the traffic sign recognition speed in still images and the recognition accuracy. The system consists of three processes; the traffic sign detection, the traffic sign recognition and the traffic sign tracking. The detection process uses physical properties of traffic signs based on a priori knowledge to detect road signs. It generates the road sign image as the input to the recognition process. The recognition process is implemented using the Pattern Matching algorithm. The system was first tested on stationary images where it showed on average 97% accuracy with the average processing time of 0.15 seconds for traffic sign recognition. This procedure was then applied to the real time video streaming. Finally the tracking of traffic signs was developed using Blob tracking which showed the average recognition accuracy to 95% in real time and improved the system's average response time to 0.04 seconds. This project has been implemented in C-language using the Open Computer Vision Library.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENT

First and foremost, I would like to thanks my supervisor Dr. Hasan Fleyeh for his kind support and guidance. Because of him, the span of time I have spent in Hogskolan Dalarna became very smooth. He creates a wonderful and dynamic environment to learn and gives the freedom and flexibility to explore insights into the field of Computer Vision. His unique way of explaining makes the complex things simple to understand. He is one of the few persons I will always remember. For me he is a source of motivation.

I would also like to thanks Professor Mark Dougherty and Dr. Pascal Rebreyend and Dr. Siril Yella for reviewing the report and kindly serving on the committee.

I would like to express my gratitude to my beloved, parents and my brothers for their love, support and encouragement in the pursuit of my career.

Finally I would like to thank all of my friends with whom I have spent good times during my stay in Borlange, specially Mevludin Memedi and Rahul.

CHAPTER 1

INTRODUCTION

1. INTRODUCTION

1.1 BACKGROUND

The design and functioning of the road transport system is intended to ensure socio-economically efficient and sustainable transportation. Individuals and the business communities depend upon the safe, efficient, and environmentally sound transports.

Intelligent Transport System (ITS) is the integration of Information Technology and automation into the traditional transportation infrastructure and vehicles. ITS cover the aspects of car navigation system, traffic management and monitoring.

There are two main branches of ITS

1. Information communication technologies.
2. Intelligent detection and recognition.

Information communication technologies, which are based on the distributed applications, are responsible to circulate data in the network of intelligent transport system, while the intelligent detection and recognition system is based on the artificial intelligence and image analysis to navigate and monitor the traffic.

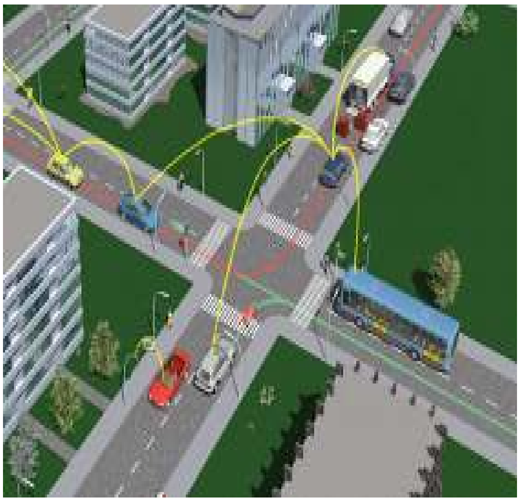


Figure 1.1: Intelligent Transportation [1].

In an Intelligent Transport System, to drive an automated vehicle, the system extracts the data from the traffic signs. Normally the drivers need to be very vigilant and cautious to identify road signs at the right time. But to identify road signs especially in poor weather conditions depends very much on the physical and mental health of the drivers. The visual perception abilities can be affected by stress, tension, physical illness or toxicities. There fore predictive techniques are being developed to maximize transportation safety and efficiency by Intelligent Transport Systems in which the traffic signs detection and recognition holds the most important position.

A digital infrastructure is vital for ITS to be able to function at all. Traffic data must be collected and there must be well-functioning traffic databases, such as the national road database (NVDB). Once the digital infrastructure is working, ITS can be used for road traffic management and other services.

REPORT STRUCTURE

This thesis report is divided in 5 chapters. Chapter 1 is the introduction and the background of the thesis work. Chapter 2 is the theory of the methodology that has been applied to implement the system. Chapter 3 covers the implementation of the methodology. Chapter 4 is about the evaluation and results that have been achieved by the system. Detailed evaluation has been mentioned with the results, and comparisons are given with the previous research conducted in Dalarna University on real time traffic sign recognition systems. Chapter 5 consists of the conclusion and the future works respectively.

1.2 PREVIOUS RESEARCH AT DALARNA UNIVERSITY

Since this thesis work had been conducted in Hogskolan Dalarna and this thesis project being the extension of the research work going on here, the research and the past achievements of the professors and the students in the area of Intelligent Transport Systems is worth to be mentioned. The details are given in the table 1.1.

Table 1.1: Previous Research in Dalarna University.

Year	Authors	Research Conducted
2004	Fleyeh & Dougherty	Three new methods for *color detection and segmentation of road signs* were developed [2]. The images were taken by a digital camera mounted in a car. The RGB images were converted into IHLS color space. New methods were applied to extract the colors of the road signs for detection. The methods showed high robustness when tested on hundreds of outdoor images in different light conditions

2005	Fleyeh	A novel *fuzzy approach to determine the shape of the traffic signs* was presented [3]. Every RGB image was converted into HSV color space and segmented by using a set of fuzzy rules depending on the hue and saturation values of each pixel. Four shape measures were used to decide the sign shape i.e. rectangular, triangular, elliptical or octagonal.
2005	Fleyeh	An algorithm for color detection and segmentation of road signs using *Color Constancy* [4] in poor light conditions was presented. The RGB channels of the digital images were enhanced separately by histogram equalization. Then a color constancy algorithm was applied to extract the true colors of the sign. The resultant image was converted into HSV color space, and segmented to extract the colors of the road signs. The method showed success when tested in different light conditions.
2006	Fleyeh, Dougherty & Gilani	*An approach using Fuzzy ARTMAP Neural Networks to recognize and classify Swedish road and traffic sign was developed* [5]. A new color detection and segmentation algorithm was presented in which the effects of shadows and highlights were eliminated.
2007	Fleyeh & Dougherty	*Support Vector Machines (SVMs) and Legendre Moments for traffic sign recognition was used for traffic sign classification* [6]**.** Legendre moments of sign borders and speed-limit sign images were computed and the SVM classifier was trained with these features. An accuracy of about 99% is achieved in sign border recognition.
2007	Fleyeh, Dougherty Dinesh & Sruthi	This research was enhanced by *training a Fuzzy ARTMAP neural network to recognize the traffic signs using Zernike moments for the sign borders as features* [7] . A fuzzy ARTMAP is trained directly with features, or by using PCA for dimension reduction, or by using LDA algorithm as dimension reduction and data separation algorithm. The method showed high robustness when tested in different conditions.

1.3 SWEDISH TRAFFIC SIGNS

SRA (Swedish Road Administration) maintains a standard for the traffic signs easily recognizable by drivers because of their physical properties that vary them from the environment. They are the control devices used to regulate traffic and provide useful information to make the driving safe and convenient. Road signs define a visual language used to convey messages to drivers and to create user friendly road system which can be understood by everyone. The information on traffic signs may be in the form of a pictogram, characters or both. Road Signs are designed with standard shapes and colors that can be easily differentiated from each other which becomes an essence in detection and recognition process. The wavelength of the colors of road signs is pre determined and has the standard set by SRA.

Although the traffic signs can enhance the driver's capabilities and comfort, they can also create potential distractions if they do not conform to standards or if placed at wrong places. Signs that are unwarranted or ineffective may distract road users from more important traffic control devices. It is better that sign and the background should be in contrast colors so as to recognize the signs easily.

1.3.1 TYPES OF ROAD SIGNS

Swedish Road Administration (Vägverket VV) is in charge of defining the appearance of all signs and road markings in Sweden. VV defines four different types of Signs described below

1. Warning Sign:

A Warning Sign alerts the drivers with a hazard ahead or the road lay out on the roadway as shown in the figure 1.2. VV designed the Warning Sign with usually Red/Yellow combination whereas the borders of the Signs are colored Red with the background color yellow which becomes easy to visualize and interpret specially in poor weather conditions like snowing.

Warning Signs are in the shape of equilateral triangle. They may in the form of Yield Sign in which the triangle has the tip downwards. While in the rest of the warning signs the tip is upwards. Some other signs like the distance to level crossing signs and track level crossing also belong to this category. Figure 1.2 displays Warning Signs:

Figure 1.2:Warning Signs [8].

2. Prohibitory Signs:

Prohibitory sign prohibits certain types of maneuvers, restricts the actions of the drivers and road users or unwanted traffic depending on the symbols on the sign. The category includes No Entry, No Parking, and Speed Limit Signs etc. They are designed in a shape of a circle with yellow background and red borders.

There are few exceptions, that is, the international standard STOP sign is an octagon with red background and white borders, and ‘No Parking’ and ‘No Standing’ which have blue background color instead of yellow falls in this category. Figure 1.3 displays the Prohibitory Signs:

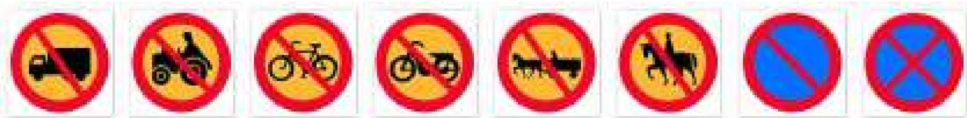


Figure 1.3: Prohibitory Signs [8].

3. Mandatory Signs:

They are round blue signs with thin white borders. They are termed as ‘Regulatory Signs’. They control the actions of the drivers and the road users. Figure 1.4 below displays the Regulatory Signs:

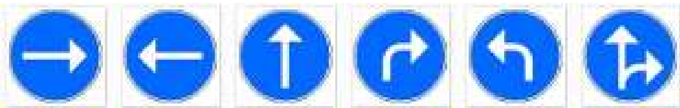


Figure 1.4: Regulatory Signs [8].

4. Information Signs:

These Traffic Signs displays the services along the roads or the important information that that could be in driver’s priorities. These signs are blue, white, black, yellow or green. Figure 1.5 below displays Information Signs.

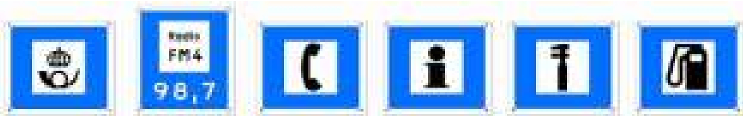


Figure 1.5: Information Signs [8].

1.3.2 PHYSICAL PROPERTIES OF THE TRAFFIC SIGNS

In all the traffic signs the two utmost properties are their 'Shape' and 'Color' using which the real-time system becomes capable to detect and recognize the Traffic Signs. The table 1.2 depicts the shape description of the traffic sign and the table 1.3 explains the colors of traffic signs. Combining the shape and the colors, the main physical characteristics of the traffic signs are achieved based on which they are recognized.

Table 1.2: Shape description of traffic signs.

Shape	Traffic Sign Description
Circle	Prohibitory or Regulatory
Octagon	Stop
Diamond	Construction and Maintenance
Rectangle	Information
Equilateral Triangle, Pointing up	Warning
Equilateral Triangle, Pointing down	Yield

Table 1.3: Color description of traffic signs

Color	Traffic Sign Description
Red	Prohibitory and Warning
Blue	Directive/Information
Green	Guidance and mileage
Orange	Construction and maintenance
Yellow	Warning
Black	Information
White	Auxiliary

1.4 THE APPROACH

A Real-Time Traffic Sign Recognition system has been proposed that is based upon the four main steps given below:

1. *Image Acquisition stage;* the camera mounted in front of the car has been used to collect the data which is a live video stream consisting of frames.
2. *Image Pre-processing Stage/Traffic Sign Detection Stage;* which consists of following steps:
    a. The frames are grasped one by one and converted into the Hue, Saturation and Value (HSV) model from the Red, Green, and Blue (RGB) model for segmentation.
    b. Noise filter is applied to the frames to remove the noise from the scene.
    c. Connected Components Labeling is applied to the objects left in the frame.
    d. Size Filter is applied to remove the inappropriate sized objects.
3. *Traffic Sign Recognition Stage* (feature extraction); a new Traffic Sign recognition technique is applied named as “Pattern Matching”, which found to be much faster, and efficient according to the real time requirements.
4. *Traffic Sign tracking;* Blob tracking have been used to predict the search region for the sign in the next frame to improve the traffic sign detection time.

The flow chart of the proposed approach is given below in figure 1.6:

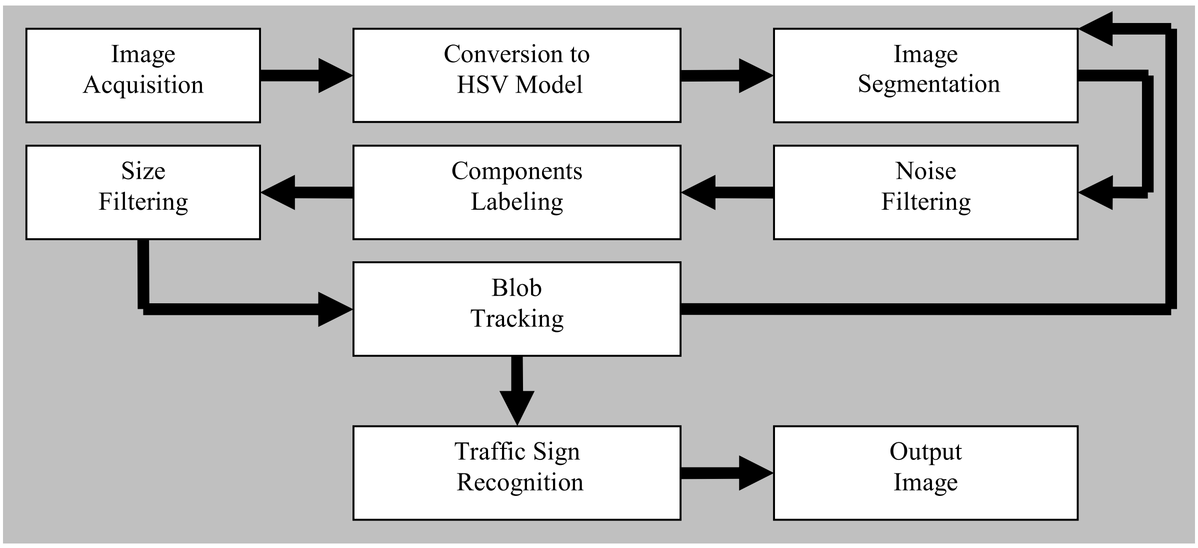


Figure 1.6: Flowchart of the approach

The resulted procedure led to the creation of the subsystem of Real-Time Traffic Sign recognition for Driver Support System (DSS). The aim is to provide the DSS with the ability to understand its environment and to assist him by giving predictions regarding collision, speed and directions.

There can't be assured absolute system's reliability and a system which is 100% "safe-failure". The aim is to provide a level of safety similar to that of human drivers. It could assist drivers about the signs they did not recognize before passing them. For example speed limit sign recognition could provide the present speed limit to the driver and alarm if the car is going faster than the speed limit.

1.5 POTENTIAL CHALLENGES

Smart vehicles will operate in real traffic conditions. Therefore the algorithm should be robust enough to get good results under adverse illumination and weather conditions which are a great challenge for the developers. Identification of traffic signs at correct time and place is important. Due to the change of weather conditions or viewing angles, traffic signs are difficult to be identified.

The following are the potential challenges in real-time traffic sign recognition system shown in the figure 1.7:

1. Color of the sign fades with time as a result of long exposure to the sunlight and reaction of the paint with air.
2. Signs may be damaged, disoriented or occulted.
3. There are variations in the lighting conditions according to the time of the day, the season, cloudiness, fog, rain and snow which causes shadowing or highlighting that effects the sensitive color information.
4. Obstacles like buildings, poles, trees, vehicles and pedestrians occlude road signs partially.
5. Sign boards often reflect the light from an approaching car during the weak day light hours.
6. Video images of road signs that are acquired in a moving car often suffer from motion blurring due to car vibration since the camera is mounted on a moving vehicle.

Many solutions have been developed to solve these problems.

- Escalera et al. dealt with the road sign recognition [9] in the environments where lighting conditions couldn't be controlled or objects could be partially occluded and their position and orientation is not known a priori. He used genetic algorithm for detection, allowing invariance localization to changes in position, scale, rotation, weather conditions, partial occlusion, and the presence of other objects of same color. He used Neural Networks for classification.

- Fleyeh [2] proposed an algorithm for traffic sign detection and segmentation in poor light conditions. The RGB channels of road images were enhanced separately by histogram equalization and the true colors of the signs were extracted by a color constancy method. Resultant image was converted into HSV and segmented to extract the road sign's colors.

- Fleyeh [2] developed a color detection and segmentation algorithm for road signs in which the effect of shadows and highlights were neglected to get better color segmentation results. The RGB images of road signs were converted into HSV color space and the shadow-highlight invariant method was applied to extract the colors of road signs.

a. Faded Sign **b. Snow** **c. Reflection from Car**

d. Occluded Sign **e. Noisy Sign** **f. Highlight**

g. Fog **h. Damaged Sign** **i. Bad-Light**

Figure 1.7: Potential challenges [8]

Dalarna University
Röda vägen 3S-781 88
Borlange Sweden
Tel: +46(0)23 7780000
Fax: +46(0)23 778080
http://www.du.se

1.6 AIM OF THE RESEARCH PROJECT

Since the research has been done in the real time environment for traffic sign recognition, there are three utmost aspects considered, to be achieved by the real-time traffic sign recognition system:

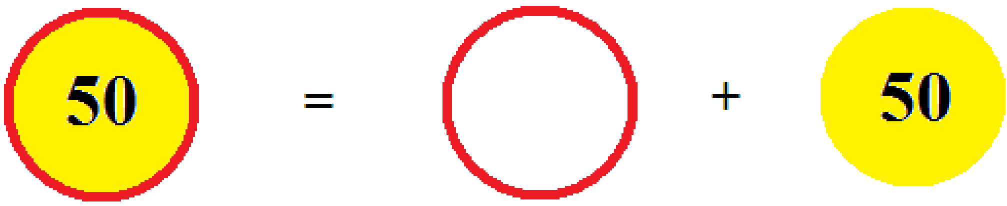
1. **Speed**: *Processing Time in Traffic Sign recognition.*
2. **Efficiency**: *Overall System's Response Time.*
3. **Reliability**: *Accuracy in Traffic Sign Recognition.*

The basic area of this thesis project is the shape based detection and recognition of the Traffic Signs in Real-Time, while the pictogram information has been used to aid in the correct shape detection.

1.6.1 SHAPE-BASED RECOGNITION:

The research is based upon Real-Time Traffic Sign Recognition System that accounts for speed and accuracy of recognition process. The main area of concern had been Shape Recognition of Traffic Signs with a predictive algorithm for traffic sign tracking called "Blob Tracking". Traffic Sign Classification has been left as the future work.

The classification of image of a traffic sign requires two steps:



Traffic Sign Classification = Shape Recognition + Pictogram Recognition

Each sign differs with the other in terms of shapes (see table 2.1). Therefore to recognize a sign, the shape of the sign must be recognized first, disregarding the internal information in the pictogram.

500 images were captured, in which total traffic signs found were 1028. The research is based on those signs which are commonly found on the Swedish Streets. They are shown in the table 1.4.

Table 1.4: Survey on traffic signs

Traffic Signs	Shape	Border Colors	Number of signs found
Information	Rectangle	Blue and Red	324
Prohibitory	Circle	Red	196
Regulatory	Circle	Blue	114
Yield	Triangle, pointing upwards	Red	107
Stop	Octagon	Red	104
Warning	Triangle, pointing downward	Red	96
Other Signs	-	-	87

THE SCENARIO

Now let suppose a car is moving on a highway with 110 km/hr which equals to 30.5 m/sec which is the maximum speed limit allowed in Sweden. The scenario is shown in the figure 1.8.

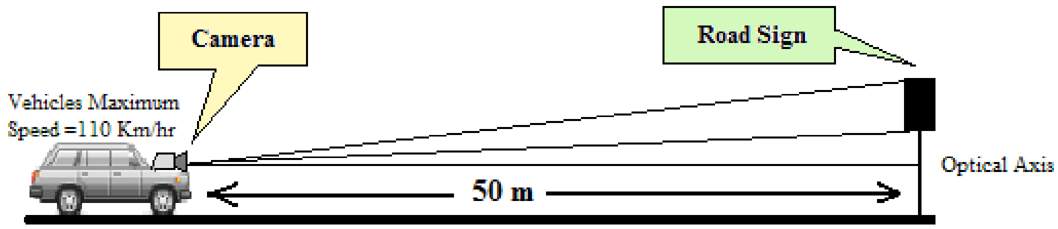


Figure 1.8: A Scenario for Traffic Sign Recognition

Now there appears a Traffic Sign at about 50m which is the normal distance for a camera to snap the clear image of a traffic sign. Time is calculated using the average velocity formula which is given as

$$time = \frac{distance}{velocity} = \frac{50}{30.5} = 1.61 \sec$$

The car is supposed to pass that traffic sign in nearly 1.6 seconds. The target is now to produce an efficient algorithm that can recognize the Traffic Sign accurately in less than 1.6 seconds in a real time situation and alert the driver based on the information displayed inside the traffic sign.

Dalarna University
Röda vägen 3S-781 88
Borlange Sweden
Tel: +46(0)23 7780000
Fax: +46(0)23 778080
http://www.du.se

CHAPTER 2

THE THEORY

2. THE THEORY

2.1 IMAGE ACQUISITION

Image acquisition is the first step of the Traffic Sign Recognition. An input image can be either taken by the live stream from the camcorder mounted on the vehicle's deck or taken from the video for an experimental purpose. The video format, acceptable by the OpenCV platform, should be in the AVI format. Each frame of the video is in a RGB Image format. The dimension of captured image is set to be 320 x 240 pixels. The pictures taken in outdoor environment are usually much cluttered. Figure 2.1 shows such an example. Image Pre-processing is necessary to simplify the image before further processing.

Figure 2.1: A Frame of the Live Stream Video [8]

The first problem in image acquisition is the interlacing problem. It arises due to camera quality of the digital camcorder. Because of this problem, the traffic sign in the image becomes unrecognizable. Bob De-Interlacing method has been used to solve the interlacing problem.

2.2 BOB DE-INTERLACING / PROGRESSIVE SCAN

- Bob-de interlacing is much faster, easier to program and is compatible to the Real-Time requirements as compared to the other de-interlacing techniques and assures 100% interlacing effect removal in real-time image acquisition from a video.

- It is used to resolve interlacing effected images, because correct traffic sign recognition with an interlacing effected camera cannot be guaranteed as it may produce blurred and interlaced objects for the detection of traffic signs.

2.2.1 THEORY DETAILS

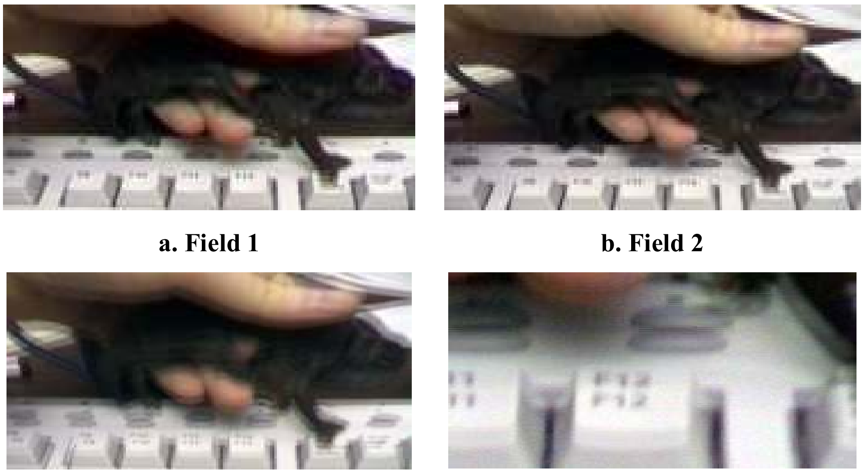
A digital camcorder records 50 pictures per second, intermixing every 2 consecutive pictures with half the height into 1 frame. In fact, it is not called a frame, but field, that is two fields are mixed into 1 frame. This mixing is called interlacing [10].

The time-line of a digital camcorder in which it produces the stream of fields is as follows:

1. Record field 1.
2. Record field 2.
3. Interlace (or mix) the field 1 and field 2 into one frame and save the frame as frame 1.
4. Record field 3.
5. Record field 4.
6. Interlace (or mix) the field 3 and field 4 into one frame and save the frame as frame 2.

The digital camera captures the field 1 and then field 2 at half the height as shown in the figure 2.2.a and 2.2.b and then stretches down to be the full height image.

The two figures look alike but if the positions of the thumb and keyboard keys are compared they would be different. Now these two fields are mixed (or interlaced) into Frame1 which is a complete full height image by joining the two fields without stretching them, as shown in the figure 2.2.c.



a. Field 1 **b. Field 2**

c. Frame 1 **d. Zoomed View of Frame 1**

Figure 2.2: Interlacing problem [10]

This frame 1 is an exact frame as on tape of a digital camcorder. A zoomed view of the above Frame 1 is shown in the figure 2.2.d. It could be clearly seen that the Frame1 consists of Field1 and Field2.

The way it looks is called saw tooth type edge distortion or interlaced lines. In other words a single frame consists of two different moments in time, that is Field1 = Time1 and Field2 = Time2. Because of this time intermix (that is 1 frame = time1 + time2), it is possible to

- De interlace a frame
- Keep 25 frames per second
- Keep the full quality; that is all the information of a picture will be saved.

The technique to achieve a high quality image by solving the interlacing problem in the image is called Bob De Interlacing or Progressive Scan.

Once the Interlacing problem is solved in the image, the image is converted into HSV model for image segmentation.

2.3 THE HSV COLOR MODEL

- The image acquired by the camera is in RGB format is greatly sensitive to chromatic variation of the daylight. The coordinates of three colors are highly correlated. As a result of this any variation in the ambient light intensity affect the RGB system by shifting the cluster of colors towards the white or the black corners. As a result, it will be hard to recognize the object.
- HSV was the ideal color model for the recognition problem since it decouples the chromatic and achromatic notion of light. This method is also preferable because Hue feature is invariant to shadows and highlights.
- HSV represents the colors in a similar way by which human eye senses the color.

2.3.1 THEORY DETAILS

Every Color in this space is represented by three components:

1. Hue (H): the apparent light color (determined by dominant wavelength).
2. Saturation (S): the purity of light.
3. Value (V): the total light across all frequencies.

The HSV model is illustrated as a conical object. The cone is usually represented in the three-dimensional form. The hue is represented by the circular part of the cone. The saturation is calculated using the radius of the cone and value is the height of the cone. Advantage of the conical model is that it is able to represent the HSV color space in a single object.

THE HUE

Red, Green and Blue (RGB) are the three primary colors used by computer monitors. 180° away from a primary, none of it is mixed in. These colors are the complement hues i.e. Cyan, Magenta and Yellow. The next level colors are between the secondary and primary colors, are called the tertiary hue colors. This process continues, creating a solid ring of colors around the primaries (figure 2.3). This definition of color describes just one dimension of color that is hue.

Hue is more specifically described by the dominant wavelength. Hue describes a dimension of color readily experienced by the eye. Hence it is the dimension of color interpreted by the human brain.

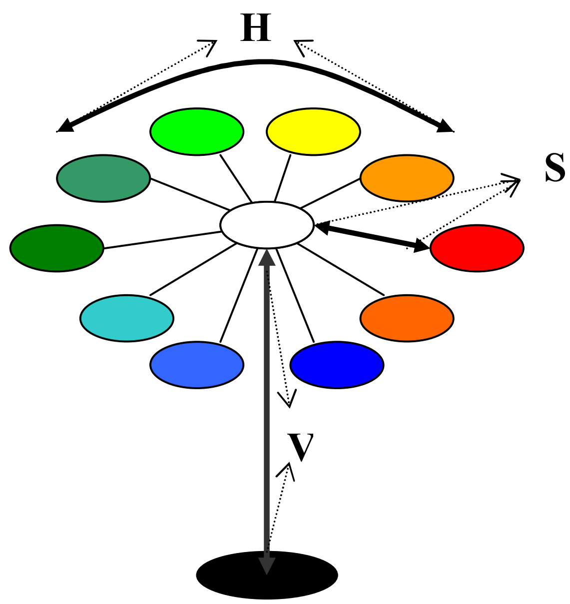


Figure 2.3: The Hue, Saturation, Value (HSV) color model

THE VALUE

Value is the brightness of the color, ranges from 0 to 100% and varies with color saturation. When the value is 0, the color space will be completely black. In terms of a spectral definition of color, value describes the overall intensity or strength of the light. If hue can be thought of as a dimension going around a wheel, then value is a linear axis like an axis running through the middle of the wheel as shown in figure 2.3.

THE SATURATION

Saturation refers to the dominance of hue in the color. On the outer edge of the hue wheel, are the 'pure' hues. Near the centre of the wheel, the hue to describe the color dominates less and less. Exactly in the centre of the wheel, no hue dominates. These colors directly on the central axis are considered **de-saturated**. These de-saturated colors constitute the gray scale ranges from 0 to 100%, running from white to black with all of the intermediate grays in between, perpendicular to the Value axis (see figure 2.3).

In terms of a spectral definition of color, saturation is the ratio of the dominant wavelength to other wavelengths of color. White light is white because it contains an even balance of all wavelengths.

2.4 IMAGE SEGMENTATION

- Image Segmentation is a process by which the specific objects in the image are distinguished from the background. Based on the color information (see table 1.3) candidate traffic signs needs to be separated from the rest of the image.
- By segmenting the image in the binary image, only two types of pixels are left to be processed, those are “white and black”. In this way the complexity of the image processing will be reduced for Traffic Sign Recognition.
- The processing time will be improved too, because only two intensity levels will be used for processing the image.

2.4.1 THEORY DETAILS

The basic factor on which the object is distinguished in the Traffic Sign detection system is color.

- Colors represent an important part of the information provided to the driver to ensure the objectives of the road signs. Therefore road signs and their colors are different from the nature or from the surrounding and man-made backgrounds in order to be distinguishable.

- Colors are regulated to the sign category, “Red for Stop Signs” or “Yellow for danger”. The information in the sign has one color while the rest of the signs have different colors. The tint of the paint which covers the sign borders corresponds to a specific wavelength.

Using the values of the colors spectrum and also that the different parts of the objects are exposed to different illumination levels, the preferable method for segmentation adopted is a modified version as described by de la Escalera [9] i.e. the image is segmented by taking the AND of the values of Hue, Value and Saturation. The HSV threshold regions are depicted in figure 2.4.

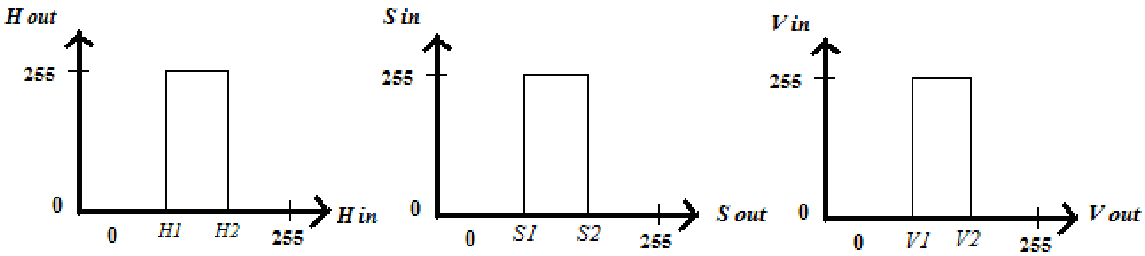


Figure 2.4: Image segmentation in HSV model.

The pixels with values above or below the threshold ranges are eliminated by assigning the value 0, while the pixels with values inside the threshold range are assigned value 1. The resulted image is binary. The segmented image based on Yellow color Hue threshold is shown in the figure 2.5 below.

a. Sample Image [8] **b. Yellow Color Segmentation**

Figure 2.5: Image Segmentation.

Figure 2.5.b shows the output image with lots of noise. To clean the image, an efficient noise filter is needed which can keep the traffic sign’s pixels safe while remove the maximum background noise.

2.5 NOISE FILTERING

When measurements are corrupted by random variations in the camera, they are said to be affected by noise. These random variations can be characterized by the standard deviation of the pixel values in the image. That is, the larger the standard deviation, the noisier is the measurement. The procedure of reducing or attenuating the noise components is commonly known as **filtering [11]**.

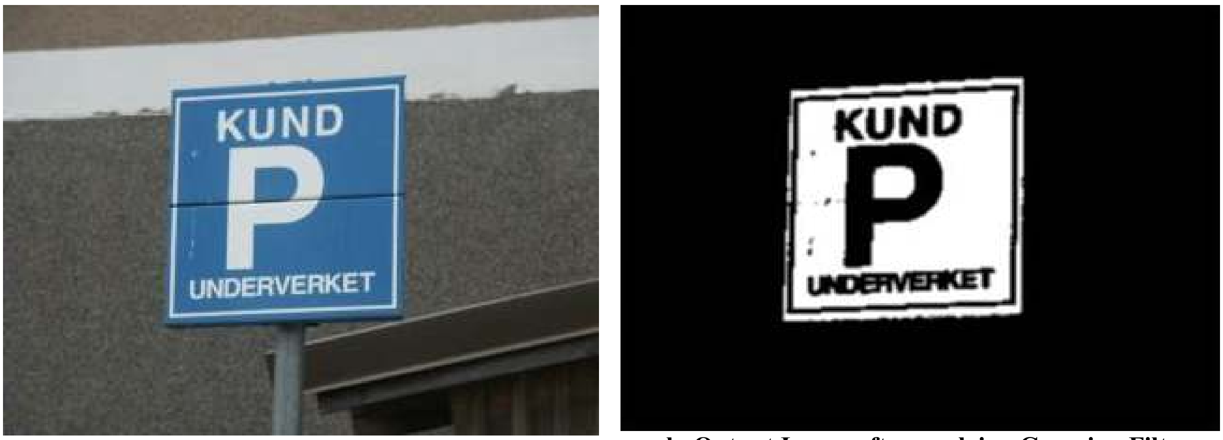
There are three types of noise filters which were developed to test in the system:

1. Linear filters for noise removal.
2. Non-Linear filters for noise removal.
3. Split and merge for noise removal.

Amongst them the modified split and merge was found to be the most efficient in noise removing.

➢ PROBLEMS WITH NOISE FILTERS

❖ Linear Smoothing filters tend to blur an image, as it can be seen in the figure 2.6.b. This is because pixel intensity values that are significantly higher or lower than the surrounding neighborhood would smear across the area. Because of this blurring, linear filters are not useful for the recognition system, as edge information is strictly not to be lost.



a. Sample Image [8] **b. Output Image after applying Gaussian Filter**

Figure 2.6: Linear noise filter.

❖ The output image after applying the median filter that is a Non-Linear Filter is shown in the figure 2.7. The median filter though removed the noise but it has also removed the internal information of the Traffic Sign which create problem in the detection of candidate traffic sign.

Figure 2.7: Output image on applying median filter.

SPLIT AND MERGE NOISE FILTER

SPLIT AND MERGE solved the problem of noise removal without blurring the image. Also it does not remove the internal information in the Traffic Signs as it was with linear and non linear filters. The original Split and Merge algorithm may eat up small blocks of information due to its nature, therefore the algorithm is modified in a way that restores the information related to the traffic sign.

Splitting and merging attempts to divide an image into uniform regions [12] as shown in the figure 2.8. The basic representational structure is a pyramid (figure 2.9) i.e. a rectangular region of size m by n. At one level of a pyramid have 4 sub-regions of the parent of size m/2 by n/2.

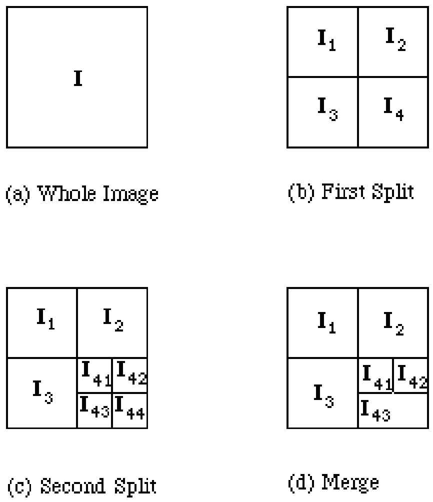


Figure 2.8: Quad splitting of an image [12]

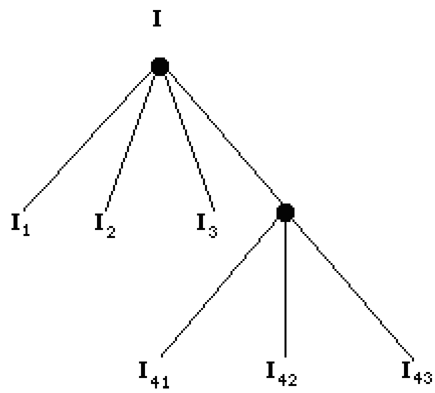


Figure 2.9: Pyramid form of Split & Merge [12]

The basic idea of region splitting is to break the image into a set of disjoint regions which are coherent within themselves. Following are the steps taken for splitting and merging.

- Initially take the image as a whole to be the area of interest.
- Decide if all pixels contained in the area of interest satisfy some similarity constraint.
- If TRUE then the area of interest corresponds to a region in the image.
- If FALSE split the area of interest (usually into four equal sub-areas) and consider each of the sub-areas as the area of interest in turn.
- This process continues until no further splitting occurs. In the worst case, this happens when the areas are just one pixel in size.
- This is a 'divide and conquer' or top down method. If only a splitting schedule is used, then the final segmentation would contain many neighboring regions having identical properties.
- Thus, a merging process is used after each split which compares adjacent regions and merges them if necessary.

The process is started with the whole image. The process of splitting is simple. A list of current regions to be processed, i.e. regions defined as not homogeneous, is maintained. When a region is found to be homogeneous it is removed from the Process List and placed on a Region List.

A region is said to be homogeneous if the standard deviation of the pixels in the region is less than threshold value where standard deviation is given by equation 2.1 as

$$\sigma = \left[\frac{1}{N-1} \sum_{j=1}^{N} \left(x_j - \bar{x} \right)^2 \right] \tag{2.1}$$

And $\bar{x}$ is the mean intensity of the N pixels in the region which could be used instead of the standard deviation. It is given as equation 2.2

$$\bar{x} = \frac{\sum x}{n} \tag{2.2}$$

Whereas splitting is quite simple, merging is more complex because due to the nature of the split and merge, the algorithm may eat up some blocks of pixels by considering them non-homogenous, therefore the algorithm is modified to resolve this problem. The modifications are discussed in the implementation (section 3.5) of split and merge.

2.6 CONNECTED COMPONENTS LABELING

- Connected components labeling scans an image and groups its pixels into components based on pixel connectivity that is all pixels in a connected component share similar pixel labeling.
- For the candidate traffic Sign detection, it is important to merge the connected pixels together and call it one object. These objects will become a candidate for the traffic Sign Recognition.

2.6.1 The Algorithm

1. For each pixel on the line, first check to see if the pixel to the left has the same pixel value. If so, you're in the same blob, and the current pixel is so labeled. If the pixel at the top has the same value as the pixel to the left but not the same blob ID, it is known at once that the pixels to the left and to the top are in the same region and that these regions should be merged.
2. If a pixel is found with left and top neighbor having different pixel values, a new blob id is created and assigned to this pixel. The algorithm continues this way, creating new blobs and merging them whenever they are discovered to be the same.

2.7 TRAFFIC SIGN RECOGNITION

In the recognition process, features for the region of interest are extracted and tested against the sets of patterns to determine which group the feature vectors belong to. There are various algorithms for the recognition of the Traffic Signs. Two algorithms were developed and tested for the system:

1. Image recognition using the Distance Function with Cross Correlation
2. Image recognition using Pattern Matching

In both the methods there is a need to normalize the object so that the object which is a possible candidate to be the traffic sign is surrounded by the blob as shown in Figure 2.10 below.

Figure 2.10: Normalized object.

2.7.1 THE DISTANCE FUNCTION WITH THE CROSS CORRELATION

a. The Distance Function

Based on the fact that all traffic signs are homogenous about their vertical axis, the algorithm intends to recognize the traffic sign from any other non homogenous object by using the similarity between the right side and the left side of the object under consideration [13].

To extract the contour of the object, the binary image is scanned horizontally line by line from the top to the bottom and the Y coordinate of the leftmost and the rightmost white pixels are computed to create the left and right distance vectors as shown in figure 2.11.

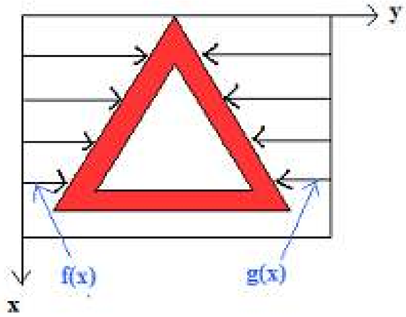


Figure 2.11: Distance Function

In this way, the left boundary of the object can be described by a distance function f(x), while the right side of the object by a distance function g(x). The shape of the distance function is related to the shape of the object under consideration.

b. Similarity Measurement using Cross Correlation

The second step of algorithm is the Similarity Function. By folding the left side of the object onto its right side, it will be easy to recognize the presence of a traffic sign because if the object is a traffic sign (or their combinations) then the two parts must match each other [13]. The improved formula of cross correlation is based on the measured distance function *f(x)* and *g(x)* given in equation 2.3

$$corr(f,g) = \frac{\sum_{i=1}^{H}\left(\left(f_i - \overline{f}\right)\times\left(g_i - \overline{g}\right)\right)}{\sqrt{\sum_{i=1}^{H}\left(f_i - \overline{f}\right)^2}\times\sqrt{\sum_{i=1}^{H}\left(g_i - \overline{g}\right)^2}} \qquad \textbf{(2.3)}$$

Where $\overline{f} = \frac{\sum_{i=1}^{H} f_i}{H}$, $\overline{g} = \frac{\sum_{i=1}^{H} g_i}{H}$ and H is the height of the image.

The Cross Correlation is a good way to find the similarity between the two sides of the object, but it requires scanning the entire object's surface pixel by pixel. This means it consumes high computational time. This solution has the computational time complexity O (WH), where W and H are the image's width and height respectively in the worst case, which is huge. The reason why this algorithm is discarded is because there is no need to process the whole image. Rather the search should be concentrated on the region of interest where the candidate traffic sign lays. The method discussed next has acquired this property and therefore it is much more efficient and accurate.

2.7.2 PATTERN MATCHING RECOGNITION FOR TRAFFIC SIGN

During the development of the Cross-Correlation and the Hausdorff Distance algorithm, a new method (named Pattern Matching by the author), was found which proved to be more robust, accurate and fast than the other image recognition methods.

The principle applied was to find the Euclidean distance between the patterns of the two images to find the matching difference between them, as described by Hausdorff as *the Hausdorff Distance*. Mathematically the Hausdorff distance between the two patterns A and B can be written as

$$H(A,B) = \max\{\min\{d(a,b)\}\} \quad \textbf{(2.4)}$$

Where 'a' and 'b' are the sets of points which belongs to the pattern A and B respectively.

Here pattern A could be a Yield Sign, a Warning Sign or an information sign etc and pattern B is a labeled object. Euclidean distance will be calculated between the vertices of these patterns. For example a yield sign has 3 vertices as it is a triangle; then Pattern A and pattern B can be mapped as shown in figure 2.12; here pattern A is the original yield sign while the pattern B is a labeled object.

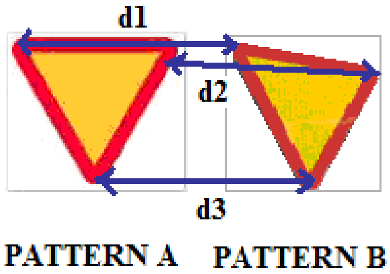


Figure 2.12: Pattern Matching

Since only the vertices are calculated rather than all the pixels of the whole image, the process time will reduce drastically for traffic sign recognition and therefore the speed and efficiency of the system will increase.

ADVANTAGES OF USING PATTERN MATCHING RECOGNITION

The problem with the previous Traffic Sign Recognition algorithms had been their computational time. They used to process each and every pixel in the image for recognition. Therefore, there is a need for a better algorithm that could process the recognition very fast in the real time.

1. The Pattern matching technique has proved to be very efficient, accurate and fast because the processing of image is concentrated around the area where the candidate traffic sign lays.
2. The image recognition process that used to take 1.5 seconds or more are now taking less than 0.15 seconds using pattern matching.
3. Since only the vertices of Traffic Signs will be processed, if the traffic signs are damaged from the middle or from the side ways, but not damaged near the vertices, they will be recognized correctly.
4. The Pattern matching technique eliminated the need of three algorithms namely, cross correlation, normalization and Hausdorff distance. Therefore it saves a great amount of time.
5. This method involves very general rules of trigonometry.
6. The problem due to the aspect ratio in image recognition is solved using Image mapping.
7. The results are very accurate and 97% of the images on average were recognized correctly.

Further details of the algorithm are given in the implementation section 3.8 of Pattern Matching algorithm.

2.8 TRACKING OF THE TRAFFIC SIGNS

To improve the efficiency of detection process while maintaining a low false hit rate, the technique of tracking is used to predict road sign by camera geometry information. By applying this technique, the detection process can significantly narrow down the search space and therefore reduce a large number of false hits as it occurred in the case of using traditional image processing [14].

There are two types of Tracking Algorithms developed and evaluated for the system which were

1. Kalman Filters
2. Blob Tracking

2.8.1 DRAWBACKS WITH THE KALMAN FILTER

Unfortunately Kalman Filter failed at occasion due to the mathematical constraints it possesses. The reasons are

- Kalman Filters work well only when an accurate model of the problem is available [15] i.e.
    - The vehicle moves straight with a constant speed.
    - The physical size of road sign is known.
- In Kalman filters, 2D trajectories, as observed in arbitrary video sequences, are projections of the 3D trajectories of possibly autonomous and independent objects [15]. When used for tracking, the trajectory changes are modeled as noise by the Kalman filters. Therefore it doesn't allow tracking small fast objects that are more likely to be lost in noise.
- Kalman filters are unable to represent multiple simultaneous hypotheses for multiple objects.
- Kalman filters have the higher cost of computational load.

To get rid of these problems, the focus was on another algorithm for traffic sign tracking which is called Blob tracking.

2.8.2 BLOB TRACKING ALGORITHM

Blob Tracking is used to assign blobs in frame (t+1) to the ones in the previous frame (t) which have already been recognized as traffic sign candidates. Using this method, traffic signs could be tracked without developing an accurate model for prediction as it was required in Kalman filters.

To overcome the holes in the Kalman filters as well as for the record keeping of the blobs identification, two blob tracking methods were merged, which were

1. Multi-Resolution Blob Tracking [14]
2. Two-Layered Blob Tracking (used in Ring Partition Method [16])

In multi-resolution Blob Tracking, tracking is performed by matching blobs between the frames. The sizes of blobs are expected to increase from frame to frame and the positions might change according to the movement of the vehicle. Therefore Blob Matching is based on the overlapped area for slow blobs using the confidence measure that the same blob will appear in the next frame (t+1).

Due to the temporal occlusion or missed sign detection, blobs in frame (t+1) may not have a corresponding blob in frame (t). To solve this problem, the blob identification needs to be maintained to be able to track the exact blob which may occur in frame (t+1). Therefore the Distance Formula used in the Ring Partition Method [16] was added to match the blob appearing in the frame(t+1) to the one which appeared in frame (t).

Using these two methods, temporal relationship between blobs in successive frames are established. The difficulty here is to generate all relevant hypotheses without generating all possible hypotheses especially due to occlusion and miss detection. The tracking of the blob is shown in the figure 2.13 for 5 overlapping frames. It can be seen that the size of the blob is increasing with time t.

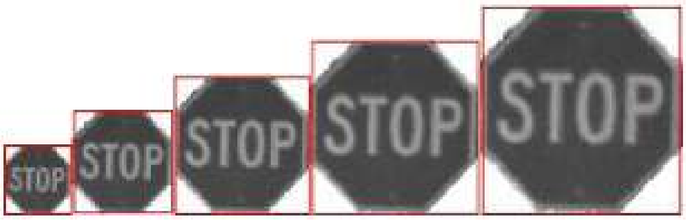


Figure 2.13: Blob Tracking

Using these characteristics, the system's response time is enhanced because the exact location of traffic sign which may occur in the next frame (t+1) is predicted. The time complexity of the real-time system with tracking is **O (nwh)** where n is the number of candidate traffic signs in the image, w and h are the widths and heights of the blobs of each candidate traffic sign respectively. This is a great improvement in terms of time reduction in the traffic sign recognition in real time. Blob tracking is discussed in detail in the implementation (section 3.9).

CHAPTER 3

THE IMPLEMENTATION

3. THE IMPLEMENTATION

3.1 REAL-TIME TRAFFIC SIGN RECOGNITION FLOWCHART

As discussed in section 1.4, the system is based upon the four main steps (including sub-steps) which include one more step of 'Tracking' for faster search by the prediction of next search region. The flow chart in figure 3.1 depicts the final design of the *real-time traffic sign recognition system*:

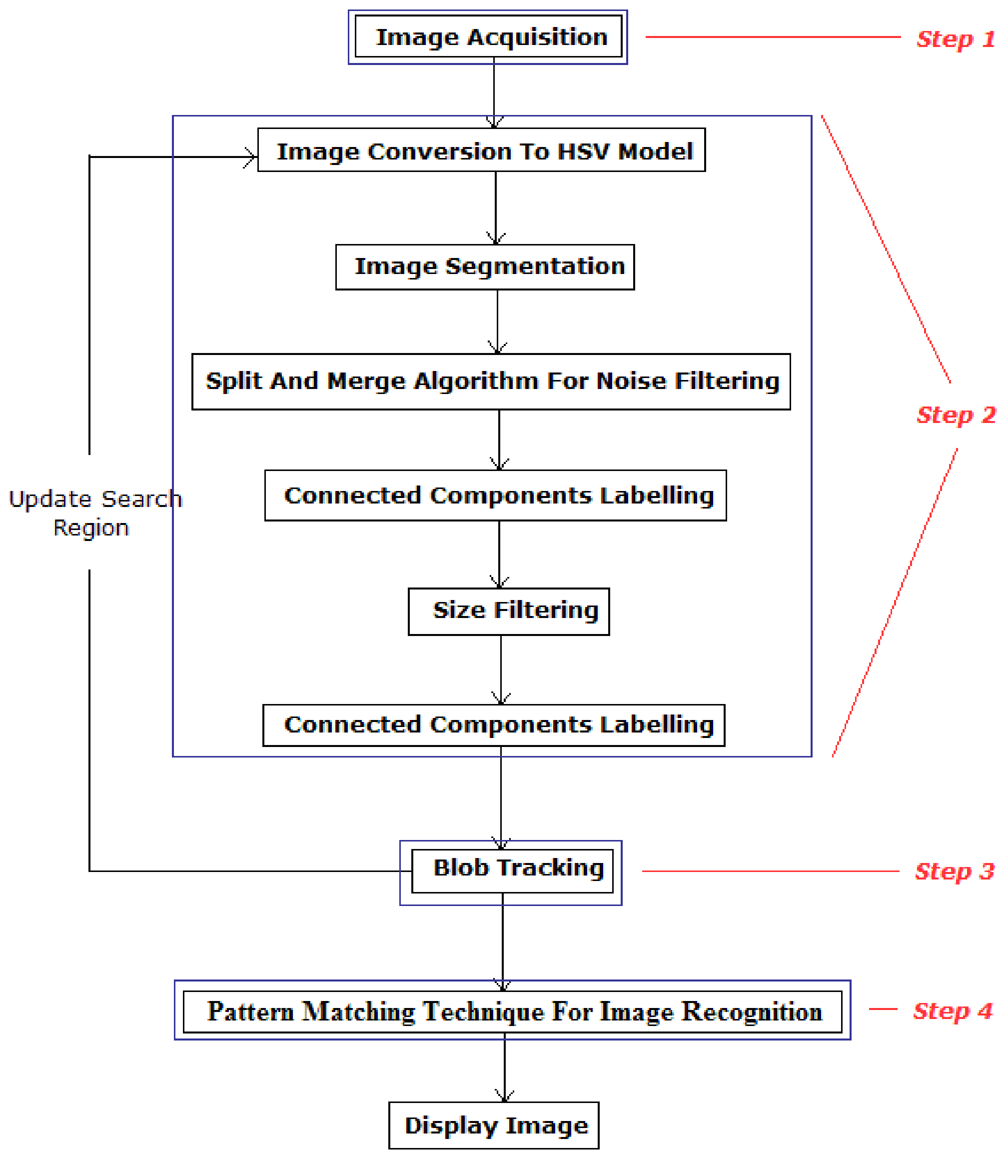


Figure 3.1: Flowchart of Real-Time Traffic Sign Recognition

3.2 IMPLEMENTATION OF BOB DE-INTERLACING

According to this technique:

1. Split each interlaced image into two frames (one of each field) of half the height.
2. Stretch each frame vertically to make a full frame by interpolation. Use simple line doubling for interpolation.
3. Display every field with 50 fps one after the other so that no information is lost without interlacing.

a. Interlaced Image [10] **b. De Interlaced Image**

Figure 3.2: Bob-De interlacing

Figure 3.2.a displays an Interlaced Image. After applying Bob-De Interlacing method, a very smooth picture for the Traffic Sign Recognition is achieved as shown in figure 3.2.b. No information is lost because both the fields are displayed one after the other.

3.3 IMPLEMENTATION OF CONVERSION (RGB to HSV)

HSV is defined mathematically by transformations between the r, g, and b coordinates [22]. Let r, g, $b \in [0, 1]$ be the red, green, and blue coordinates in RGB color space. Let ***max*** be the greatest of r, g, and b, and ***min*** the least of r, g, and b. To find the hue angle $h \in [0, 360]$ for HSV, compute the following equation (3.1):

$$h = \begin{cases} 0 & \text{if } \max = \min \\ \left(60 \times \dfrac{g - b}{\max - \min}\right) \bmod 360 & \text{if } \max = r \\ 60 \times \dfrac{b - r}{\max - \min} + 120, & \text{if } \max = g \\ 60 \times \dfrac{r - g}{\max - \min} + 240, & \text{if } \max = b \end{cases} \tag{3.1}$$

The values for *s* and *v* of an HSV color are defined in equations 3.2 and 3.3 respectively:

$$s = \begin{cases} 0 & \text{if } \max = 0 \\ \dfrac{\max - \min}{\max} & \text{otherwise} \end{cases} \tag{3.2}$$

$$v = \max \tag{3.3}$$

These functions are readily available in the OpenCV Library and is given as

```
cvCvtColor (const CvArr* src, CvArr* dst,int code);
```

Where, code is CV_RGB2HSV, is an OpenCV code to convert the source image 'src' from RGB to HSV which is displayed in the destination image 'dst'.

Figure 3.3.a shows a sample image displaying a yield sign and a regulatory sign in RGB color format while the figure 3.3.b shows the color after the conversion into HSV color space.

a. Sample Image in RGB [8] **b. Image in HSV model**

Figure 3.3: HSV color space

3.4 IMPLEMENTATION OF IMAGE SEGMENTATION

A coding scheme for image segmentation has been designed according to which the selected color is assigned a different code depending on which the color represents the white pixels in the segmented output image. This color coding scheme is shown as in the following table 3.1:

Table 3.1: Segmentation Color Codes

Code	Color
0	Black
1	White
2	Red
3	Yellow
4	Blue

The values of Hue, Saturation and Value are normalized in the range [0, 255] based on the codes for each color. Hue can be calculated as given in equation 3.1.1:

$$H_{out} = \begin{cases} 255 & H_{min} < H_{in} < H_{max} \\ 0 & \text{otherwise} \end{cases} \quad \textbf{(3.1.1)}$$

Saturation can be calculated as given in equation 3.2.1

$$S_{out} = \begin{cases} 0 & 0 \leq S_{in} \leq S_{min} \\ S_{in} & S_{min} \leq S_{in} \leq S_{max} \\ 255 & S_{max} \leq S_{in} \leq 255 \end{cases} \quad \textbf{(3.2.1)}$$

Whereas the value can be calculated as given in equation 3.3.1

$$V_{out} = \begin{cases} 0 & 0 \leq V_{in} \leq V_{min} \\ V_{in} & V_{min} \leq V_{in} \leq V_{max} \\ 255 & V_{max} \leq V_{in} \leq 255 \end{cases} \quad \textbf{(3.3.1)}$$

A logical AND between S_{out} and H_{out} and V_{out} will generate a binary image containing the road sign with the desired color. Since there are three main colors that are red, blue and yellow, the hue, value and the saturation values are tuned as shown in the table 3.2.

Table 3.2: Normalized Hue, Saturation and Value ranges after tuning.

Color	Hue Range [0 ,255]	Saturation Range [0,255]	Value Range [0,255]
Red	210-255,0-10	127-255	127-255
Yellow	20-45	110-255	90-255
Blue	120-180	127-255	127-255
White	-	166-255	90-255
Black	-	0-230	0-63

Threshold values are achieved for each color by using this table. The pixel color values under this threshold range will be assigned 1 (which is a white pixel) and the others will be assigned 0.

- There are two ranges of the red color, because as shown in the figure 2.3, it has the highest proportion in the hue spectrum wheel. Usually after the tuning, the Hue ranges of Red are taken as [0, 10] and [210, 255] as shown in the figure 3.4.
- The blue color has one range around 170. The light blue starts from 120 and the dark blue starts at 160 and both end up till 180 after tuning. It is shown in the figure 3.5.
- The yellow color has the smallest range among the three colors which is around 37. Usually it is tuned as [20, 50] for segmentation as shown in the figure 3.6.
- The white and black colors are only used for pictogram recognition. The white has the values of Saturation and the Value in the ranges as [166-255] and [90-255] respectively. While the black has the ranges of saturation and value as [0-230] and [0-63].

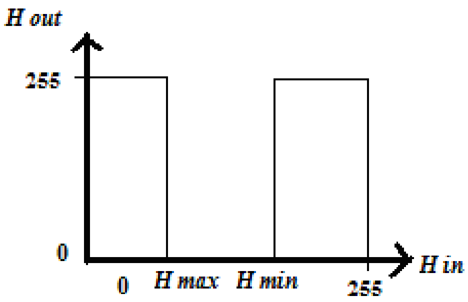


Figure 3.4: The Hue Transfer Function For Red [2].

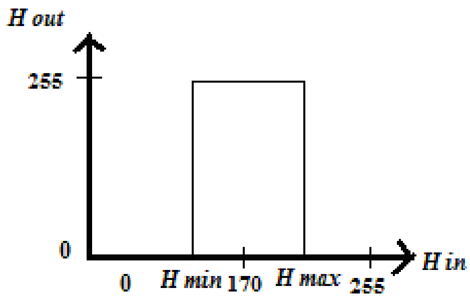


Figure 3.5: The Hue Transfer Function For Blue [2].

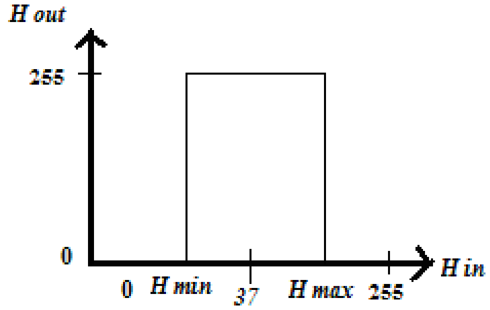


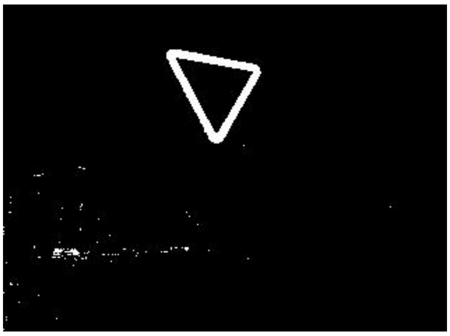
Figure 3.6: the Hue Transfer Function for Yellow[2].

The segmentation results for the image are shown below in figure 3.7.a, for the colors black, white, red, blue and yellow respectively shown in the figures 3.7.b, 3.7.c, 3.7.d, 3.7.e and 3.7.f.

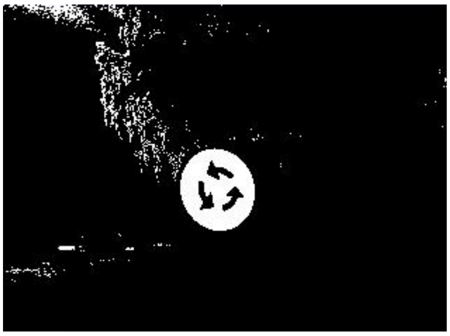
a. Sample Image for Segmentation [8]

b. Black Color Segmentation

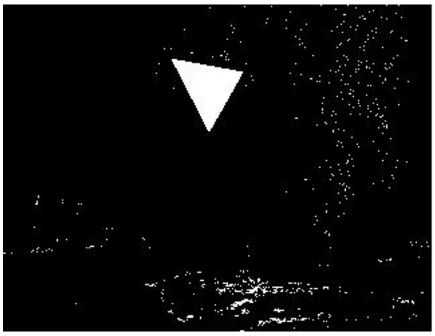
c. White Color Segmentation



d. Red Color Segmentation



e. Blue Color Segmentation



f. Yellow Color Segmentation

Figure 3.7: Image Segmentation Output.

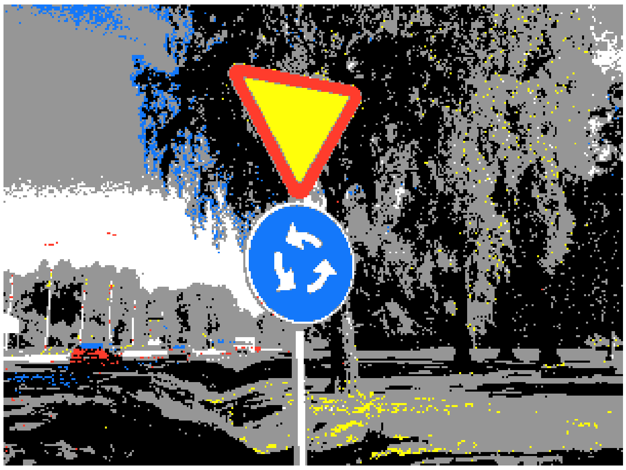
After applying the segmentation, five different images are achieved, each of the respective colors, for the traffic signs. White and black are used for pictogram recognition. While red, blue and yellow are kept for the shape recognition.

- Image Segmentation improves the accuracy of sign detection as the segmented image has the specific colors assigned to the Traffic Signs by the VV.

- After the segmentation, in the figures 3.7 d, 3.7 e and 3.7 f, the borders of the traffic signs are very evident for red and blue traffic sign. These borders across the traffic Signs with typical red, blue and yellow colors are very useful for feature extraction and Traffic Sign Recognition.

Since the colors of the pixels are coded with unique number, the grey-level images can be achieved by multiplying the codes with 50. The figure 3.8.a below shows the images in grey-level values. Using grey-level values, the colors can be separated as shown in figure 3.8.b, where the segmented colors are represented with their pseudo colors; black, white, yellow, red and blue are shown in their respective colors while the rest of the hue in the image is shown as grey.

a. Grey-level image



b. Pseudo Colored image

Figure 3.8: Segmented Image.

3.5 IMPLEMENTATION OF SPLIT & MERGE NOISE FILTER

The images in the figures 3.7 have noise in the surrounding coming from the objects whose hue lie in the same ranges as the traffic sign's hue. This is a big problem as the noise has to be removed taking care of the information inside the traffic signs. Otherwise this noise will generate false alarms. Also the formal Split and Merge Algorithm may eat up the regions during the merging process which should be preserved (see figure 3.9). This created a need to modify the algorithm.

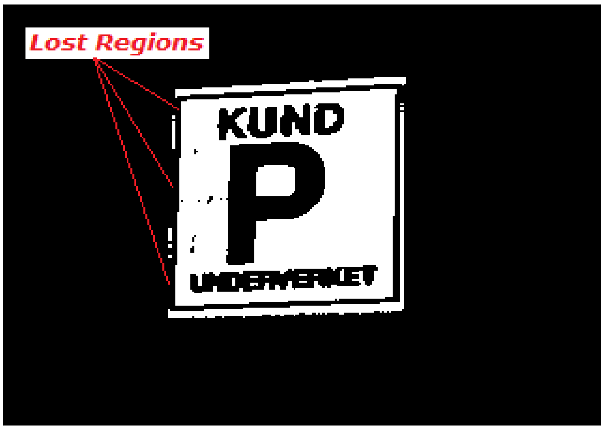


Figure 3.9: Lost Regions Due to Formal Split and Merge Algorithm.

3.5.1 HOW THE MODIFIED SPLIT AND MERGE WORKS

In the modified form of split and merge, the homogeneity of the 4-diagonal neighbors of the 8-neighbours of the non-homogenous regions, is checked before deleting the region under process. If any region in the neighborhood of the neighboring region is homogenous then the non-homogenous region under consideration will be merged to that neighboring region as shown in the figure 3.10.

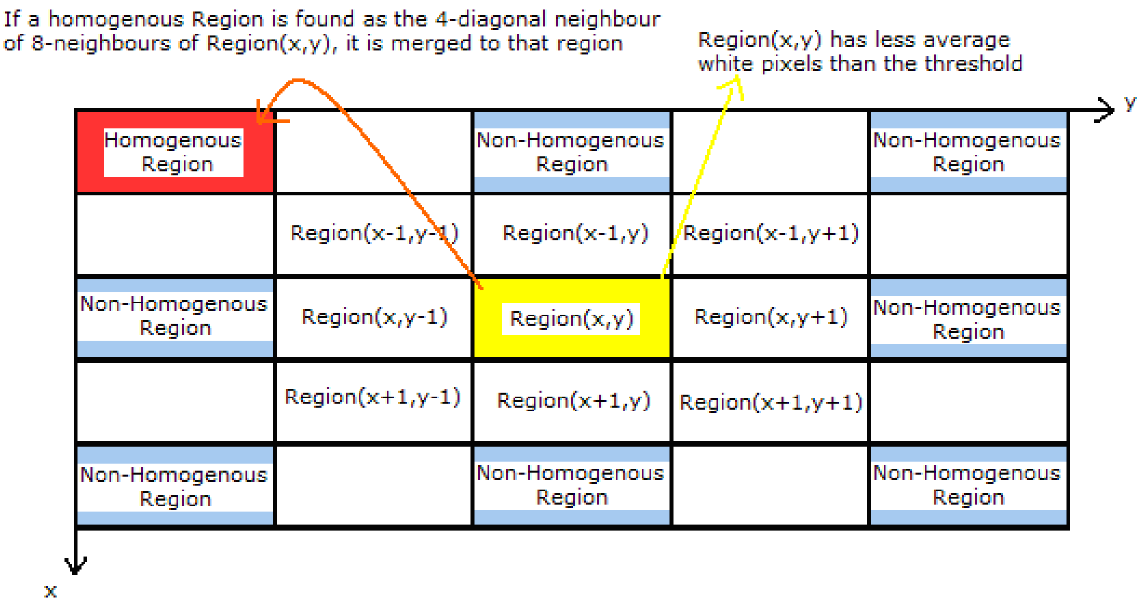


Figure 3.10: Modified Split and Merge.

To illustrate the working steps of this method, consider an image I as shown in the figure 3.11 (a).

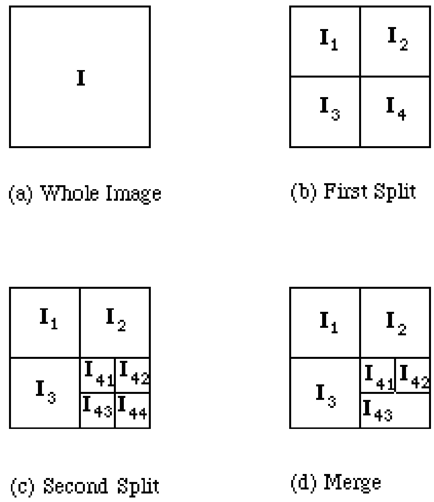


Figure 3.11: Split and Merge.

1. Let I denote the whole image which is considered as one region (see figure 3.11a).
2. If the average white pixels of the region are less than the mean threshold value so the region is split into 4 regions as I_1, I_2, I_3 and I_4 (see figure 3.11b).
3. Now regions I_1, I_2 and I_3 respectively have average white pixels greater than the mean threshold value but region I_4 has less average white pixels than the threshold mean value, therefore I_4 is split next. The algorithm will mark I_1, I_2 and I_3 as homogenous regions and I_4 as non-homogenous region which is then split further into I_{41}, I_{42}, I_{43} and I_{44} (see figure 3.11c).
4. Now assume that I_{41}, I_{42} and I_{43} are homogenous, so they are kept in the list of homogenous regions, while I_{44} is non-homogenous and that the region I_{44} cannot be further split into sub-regions, therefore it is kept in a separate list of non-homogenous regions.
5. The list of non-homogenous regions is now checked one by one. If the 4-diagonal neighbors of the 8-neighbouring regions of any non-homogenous region are homogenous, then the non-homogenous region under consideration will be labeled as homogenous and it will be included into the list of homogenous regions.
    - Otherwise if there is no homogenous region found in the 4-diagonal neighbors of the 8-neighbours of the region, then the non-homogenous region will be considered as noise and will be deleted from the image.

6. After deleting all the non-homogenous regions, all the homogenous regions will be merged together to form the complete image I.

The mean intensity value of the white pixels in the region is the basis of homogeneity of that region. For n number of total pixels in the region where x be the white pixels, the mean is calculated as equation 2.2.

$$\bar{x} = \frac{\sum x}{n} \tag{2.2}$$

Note that the minimum size of the sub-region is kept as 2x2 while the mean threshold value is kept as 0.6 which is set after tuning the image in order to remove the maximum noise. Figure 3.12.b. illustrates the output image of modified split-and-merge algorithm applied to the Information Sign.

a. Information Sign [8]

b. Output Image

Figure 3.12: Modified Split and Merge.

The figure 3.12.b can be compared with figures 2.6.b and 2.7 which were filtered using linear and non linear filters. The difference is evident that the image is not blurred as in 2.6.b and there is no internal information loss as in 2.7.b. Also note that the regions that were lost in the figure 3.9 have been preserved in figure 3.12.b.

As soon as the image is cleaned from the noise by split and merge algorithm the objects are labeled using the connected components labeling.

3.6 IMPLEMENTATION OF COMPONENTS LABELING

Components labeling method scans an image pixel-by-pixel, from top to bottom and left to right to identify connected pixel regions. Since it will work with the binary image the intensity value V is taken as 1 and the 8-connectivity neighborhood of the pixels for labeling have been used.

The connected components labeling operator scans the image by moving along a row until it comes to a point *p* (where *p* denotes the pixel to be labeled at any stage in the scanning process) for which intensity value $V=1$. When this is true, it examines the four neighbors of *p* which have already been encountered in the scan [17]. Based on this information, the labeling of *p* occurs as follows:

- If all four neighbors are 0, assign a new label to *p*, else
- if only one neighbor has $V=1$, assign its label to *p*, else
- If more than one of the neighbors has $V= 1$, assign one of the labels to *p* and make a note of the equivalences.

After completing the scan, the equivalent label pairs are sorted out into equivalence classes and a unique label is assigned to each class. In the figure 3.13 the *labels 1, 2 and 3* are assigned to the single class after the first pass. Later, a second scan is made through the image, during which each label is replaced by the label assigned to its equivalence classes in the integration table and the *labels 1, 2 and 3* are assigned to one class which is 5.

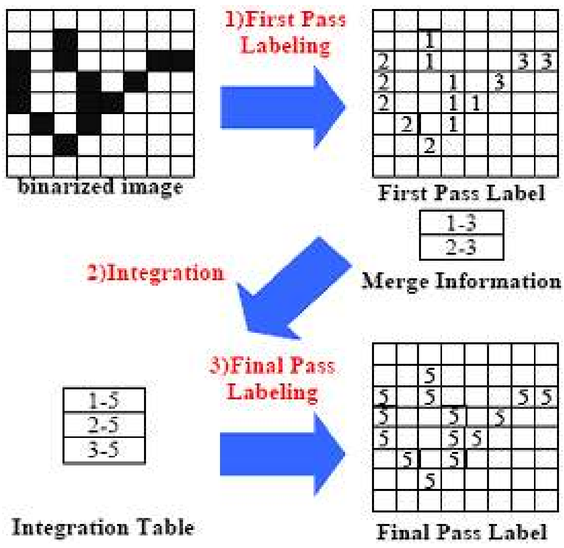


Figure 3.13: Labeling Flow [18].

Now since class 5 is the one and only class in the connected classes it is assigned the class label of 1 as shown in the figure 3.14 below

		1					
1		1				1	1
1			1		1		
1			1	1			
	1		1				
		1					

Figure 3.14: Final Label.

When the above designed algorithm is applied to the image for blue colored traffic signs, the following output image is achieved as shown in the figure 3.15.

a. Sample Image [8] **b. CCL Output Image**

Figure 3.15: Components Labeling.

The figure 3.15.b shows seven connected components which could be the possible candidate traffic signs. As it can be seen in figure 3.15.a, the sample image shows three information signs with the blue objects in the background which are blue colored vehicle and the sign boards on the buildings.

The connected components are labeled from left-to-right and top-to-bottom, therefore the top-left object is labeled as 1 and the bottom-right object will be labeled as 7.

Later on Size Filter is applied to remove very big as well as very small sized objects based on the distance of the camera with the object.

3.7 IMPLEMENTATION OF SIZE FILTER

Size Filtering is the process of the selection of labeled objects based on the size of the object. The size of the object is calculated by the number of white pixels in the binary image.

3.7.1 WHY AND HOW TO APPLY SIZE FILTER

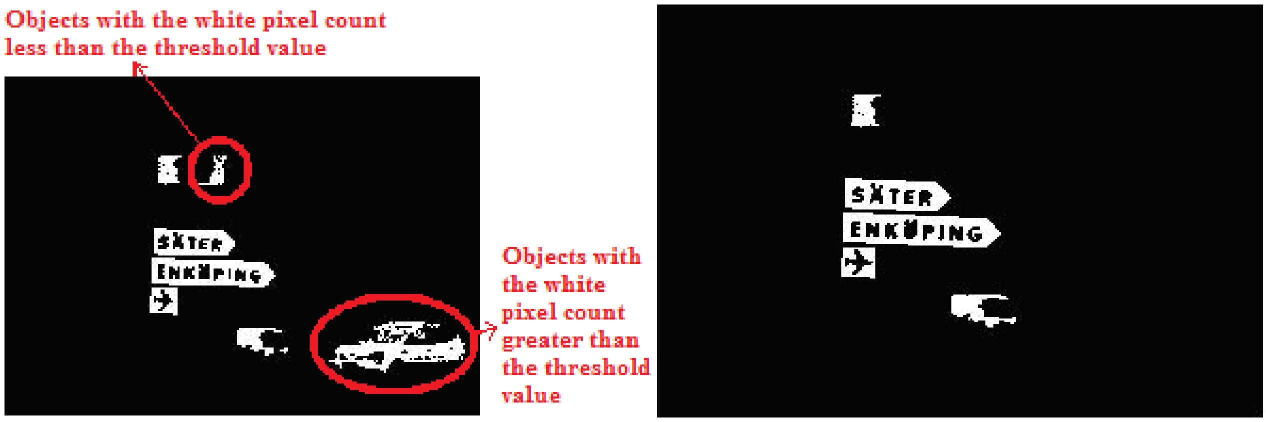
The concept behind Size Filtering is that, those traffic signs which are far away from the camera mounted on the vehicle’s deck are ignored because the system can’t read the information from that distance. Such objects have very small white pixels count, which is less than the minimum threshold value (taken as 32 x 32 pixels) and will be deleted from the image. Also there is logic that any small object is not possibly the traffic sign, and it is useless to process such an object.

As soon as the traffic sign comes nearer it becomes worthy to be recognized because the next decision of the driver will depend on the directions given by that traffic Sign. The size of these objects will be under the threshold ranges. Therefore these objects will be preserved.

Similarly, when the traffic signs come very near to the vehicle, there is no need to process them because the driver does not have enough time to make the decision. The size of these images will be very large, that is they have larger number of white pixels count which is greater than the maximum threshold value of the size which is given as equation 3.4

$$Max.Threshold\ .Value = \frac{\text{image height}}{4} \times \frac{\text{image width}}{4} \qquad \textbf{(3.4)}$$

The objects with the size greater than the Maximum Threshold value will be deleted from the image.



a. Labeled Objects **b. Deletion of objects by the Size Filter**

Figure 3.16: Applying size filter.

Figure 3.16.a shows the labeled objects that are above and below the threshold values of the white pixel's count. Figure 3.16.b displays the deletion of those two objects.

The Size Filter improves a lot of time in real time processing as un-important deleted objects are not processed in the image for traffic sign recognition. Objects like the sky in the background with blue hue will be deleted from the image using the size filters. There is a need to apply connected components labeling again after applying Size Filters because the Size Filter reduces the number of un-important objects drastically. Thus it increases the efficiency and the speed of the system.

3.8 IMPLEMENTATION OF PATTERN MATCHING RECOGNITION ALGORITHM

According to the equation 2.4 in section 2.7.2, the matching difference between the two patterns can be calculated by using the Hausdorff distance formula. Here Pattern A is one of the traffic sign from the list of six traffic signs under investigation as shown in table 1.4. Pattern B is the unknown object extracted from the image after Components Labeling which is a candidate Traffic Sign.

Pattern Matching Algorithm, using this logic, consists of 7 steps:

1. Isolate the pattern B by a minimum rectangle containing this pattern.
2. Select the traffic sign shape for pattern A.
3. Find the outer edge of pattern B and check the geometrical relationship with pattern A.
4. Calculate the distance between the corresponding vertices of pattern A and pattern B. This step is called Matching.
5. Calculate the threshold distance for checking the level of matching error.
6. Calculate the Percentage of Resemblance between pattern A and pattern B.
7. If pattern B is matched with pattern A, display the name of the recognized Traffic Sign as Pattern B, else
   - Select the next traffic sign in the list as Pattern A and go to step 3.

Algorithm is repeated until pattern B is recognized as pattern A. If all the traffic signs in the list have been compared with pattern B and still it is un-recognized, then it will be deleted from the image.

3.8.1 ALGORITHM DETAILS

An example of Warning Sign Recognition is given to explain the working of the algorithm. Later on this algorithm is generalized for the rest of the traffic signs in section 3.8.2. A warning sign is selected for matching if the candidate traffic sign is generated from the red color hue segmentation. A sample warning sign used for the matching algorithm as pattern B is shown in figure 3.17.

Figure 3.17: Warning Sign for matching [8]

Step-1: Isolation of Pattern B

In the first step, the candidate labeled object is extracted from the image and isolated by a minimum rectangle as shown in the figure 3.18.

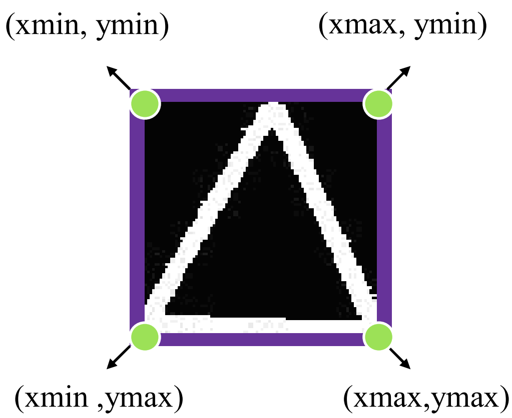


Figure 3.18: Isolation of Pattern B.

Note that the coordinates of the rectangle vertices are (xmin, ymin), (xmin, ymax), (xmax, ymin) and (xmax, ymax).

Where

xmin is the minimum x-value of the blob,

ymin is the minimum y-value of the blob,

xmax is the maximum x-value of the blob,

ymax is the maximum y-value of the blob.

Step-2: Choose the shape of Pattern A

Now, the shape of pattern A is selected as one of the traffic signs and virtually it is considered it to be drawn inside the isolated region**.** Figure 3.19 shows the red colored borders of the warning sign acting as pattern A**.**

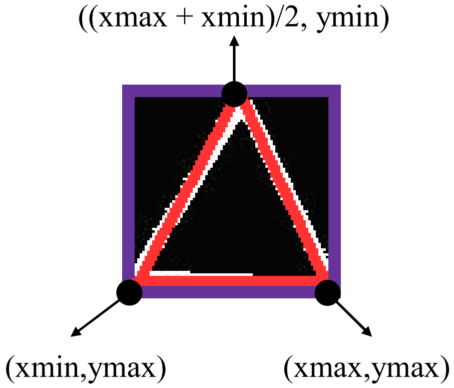


Figure 3.19: Choosing pattern A

The three vertices of the Pattern A are ((xmax + xmin)/2, ymin), (xmin, ymax) and (xmax, ymax).

Step-3: Match the features of pattern B with the features of pattern A

The shape of pattern B is checked next by drawing its outer edges. Then the shape and the color of pattern B are compared with the shape and color of pattern A.

- Color of pattern B is saved during the segmentation process (see section 3.4).

- Shape resemblance can be calculated by matching the number of vertices of pattern B with the number of vertices of pattern A. For example, a triangle has 3 vertices and a warning sign is a triangle; a rectangle has 4 vertices and an information sign is a rectangle and so on. The vertices' of Pattern B are highlighted with dots in the figure 3.20 which are located very near to the coordinates of the warning sign.

Figure 3.20: The coordinates of Pattern B.

Note that the coordinates of Pattern B are (x, ymin), (xmin, y1) and (xmax, y2).

- The upper coordinate (x, ymin) has x which is very near to (xmax + xmin)/2.
- The left bottom coordinate (xmin, y1) has y1 could be either ymax or any value near to ymax.
- The right bottom coordinate (xmin, y2) has y2 could be either ymax or any value near to ymax.

Step-4: Calculate the matching distance between Pattern A and Pattern B using Hausdorff distance formula

The distance between the vertices of pattern A and the vertices of pattern B is calculated to find the matching difference between them**.**

The generalized formulae to calculate the overall distance is given by equation 3.6

$$d_n = \sqrt{(a_n.x - b_n.x)^2 + (a_n.y - b_n.y)^2} \qquad \textbf{(3.6)}$$

Where, $1 \ldots n$ are the number of vertices of pattern A and pattern B.

If a warning sign is considered as the pattern A, then n is 3 and the coordinates can be given as

Point a_1 = (xmin, ymax) and point b_1 = (xmin, y1) or b_1 = (xmin, ymax)

Point a_2 = (xmax, ymax) and point b_2 = (xmax, y2) or b_2= (xmax, ymax)

Point a_3 = (xmid, ymin) and point b_3 = (x, ymin)

Where, xmid is the mid-point of the width of the blob given as equation 3.6.1.

$$xmid = \frac{x\min + x\max}{2} \qquad \textbf{(3.6.1)}$$

Step-5: Calculate Threshold Value to check the difference between patterns

This step proved to be very useful to solve the ***aspect ratio problem*** for the traffic signs. The technique is that, a virtual grid map with 1 unit pixel is drawn inside the blob as in the figure 3.21.

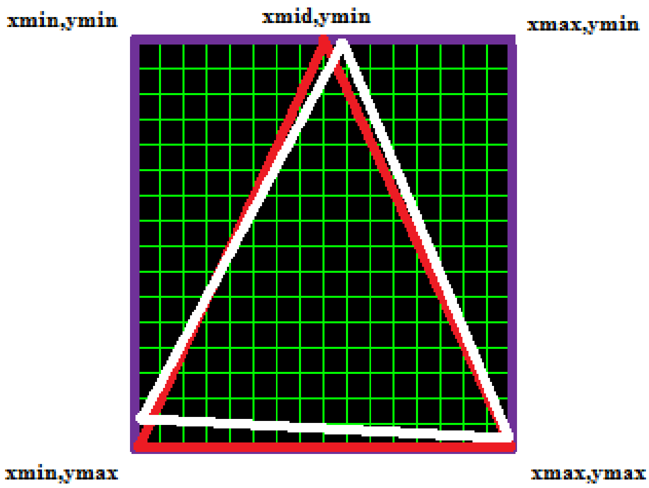


Figure 3.21: Grid Map inside the Blob

Since d_1, d_2......d_n are measured in unit pixels for n vertices, therefore to calculate the error percentage in the recognition of the image, a distance threshold value T_D is needed. The Euclidean distance between pattern A and pattern B should be less than the threshold value T_D, that is $d_1 < T_D$, $d_2 < T_D \ldots d_n < T_D$, is necessary for correct shape recognition of a traffic sign. Also the greater the distances between the vertices' coordinates means the greater is the matching difference between the pattern A and pattern B. Mathematically the formula that justified scientifically the threshold value on the grid space can be written as equation (3.6).

$$T_D = \frac{Blob\_Area}{Mask} \quad \textbf{(3.6)}$$

Where

- Blob area = (xmax – xmin) x (ymax – ymin)
- The Mask is the grid square area around the pattern A's vertices. This is an experimental value for each of the traffic sign shape which is tuned to solve the aspect ratio problem. The greater the aspect ratio the higher the mask value should be kept. This value is tuned to prevent false alarms. Here the mask for warning sign is kept as 16x16.

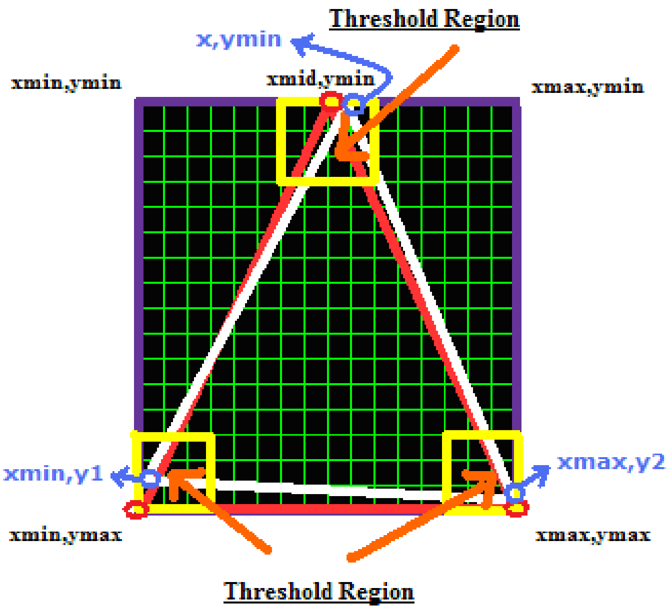


Figure 3.22: Threshold regions marked with yellow.

The figure 3.22 above shows the threshold area around the vertices of pattern A in yellow color. If all the vertices of pattern B lie under their respective threshold areas than there is a higher percentage of resemblance of pattern B with pattern A, which will result in correct Traffic Sign Recognition. The formula for the T_D shows that the threshold value is adaptive to the size of the object; that is, the greater the blob area the greater will be the threshold value.

Step-6: Calculate Percentage of Resemblance between Pattern A and B

Now the difference between pattern A and pattern B is calculated by calculating first the average Euclidean distance achieved from the step 4.

Average distance can be calculated as given by the equation (3.7)

$$d_{avg} = \frac{\sum d_n}{n} \tag{3.7}$$

Where n is the total number of vertices of the sign shape selected. For instance n = 3 are the vertices of the Warning Sign.

In order to find the error percentage, take the percentage of the average distance divided by the threshold distance. Logic is simple; the coordinates of pattern B that are inside the threshold region of pattern A have no error. The formula for error% is given below in equation (3.8)

$$Error\,\% = \frac{d_{avg}}{T_A} \times 100 \tag{3.8}$$

Where

$\boldsymbol{T_A}$ is the maximum threshold area around the vertices.

Step-7: Recognize the Candidate Traffic Sign

The system recognizes the Pattern B as the Traffic Sign based on its resemblance with the Pattern A. Otherwise if the percentage of resemblance is less, then it rejects the object. This is done by setting a limit for the Error percentage. The author has kept 30% error for rejection of the object. The candidates having percentage greater than the 30% are termed as recognized traffic sign as their respective pattern A.

The algorithm is repeated if a candidate traffic sign is not recognized until all the traffic signs are being compared with the candidate traffic sign. If there are more than one traffic signs matching with the object, then the traffic sign with more percentage of resemblance will be selected as the matched traffic sign with the object. The object under consideration is matched with the warning sign with the resemblance percentage of 91.66% in 0.13 seconds.

Note that the pattern A is an imaginary traffic sign. In reality only the vertices of Pattern A in the blob are processed without concerning the other pixels of the image. This is one of the elements that make the recognition process very fast.

3.8.2 GENERALIZATION OF ALGORITHM FOR OTHER SHAPES

Pattern matching algorithm had been applied to the Warning Sign as an example. If the candidate traffic sign is not recognized, as the warning sign, then algorithm will check for the rest of the traffic signs in the list, until it finds it match to the closest shape.

In this aspect, steps 1, 4, 5, 6 and 7 will remain the same in the recognition process. In steps 2 and 3 the geometrical characteristics of the candidate traffic sign were used which are different for the every other traffic sign pattern to be matched with pattern B. In the table 3.3, it is shown that the numbers of vertices of the traffic signs inside the blob are varying which is one property used for recognition of the candidate traffic sign.

Table 3.3: Geometrical Characteristic of Traffic Signs.

Traffic Signs	Border Colors	Number of Vertices	Colors inside the borders	Threshold Mask n x n
Warning	Red	3	Black and yellow	16 x 16
Yield	Red	3	Yellow	16 x 16
Information	Red Or Blue	2-to-4	White, Black and Yellow	8 x 8
Prohibition	Red	4	Blue, yellow and black	8 x 8
Stop	Red	8	White	8 x 8
Regulatory	Blue	4	White	8 x 8

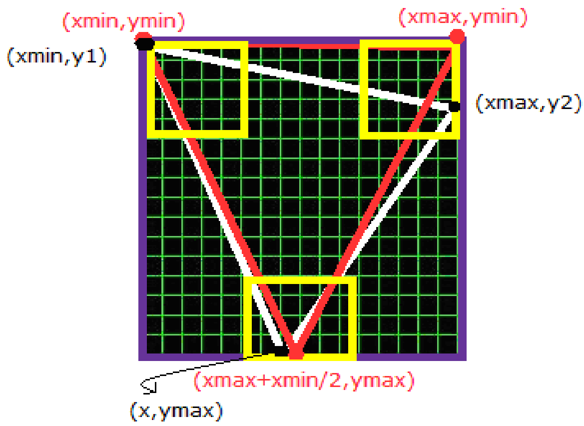
Based on these characteristics, the actual Traffic Sign is selected as pattern A in step 2 of the algorithm for its matching with pattern B. Note that the threshold mask is different for each of the traffic signs. Mask values are tuned according to the aspect ratio of the sign.

3.8.2.1 ALLOCATION OF VERTICES

Selection of vertices' coordinates is an important issue because the pattern matching is based upon it. It is easier to grasp the vertices for Yield signs, Warning Signs and Stop Sign but this step becomes complex in Prohibitory/Regulatory Signs and the Information Signs.

<u>Yield Sign</u>

A yield sign is selected for matching if the candidate traffic sign is segmented for the red color hue. In the yield Sign the coordinates of the pattern B are (xmin, y1), (xmax, y2) and (x, ymax), are matched with the coordinates of the pattern A in the isolating rectangle which are (xmin, ymin), (xmax, ymin) and ((xmax + xmin)/2, ymax). It is shown in the figure 3.23.b.



a. Sample Image [8] **b. Yield Sign Image Mapping**

Figure 3.23: Yield Sign Recognition.

Yield Sign in figure 3.23 above has 89% Resemblance Percentage and is recognized in 0.16sec.

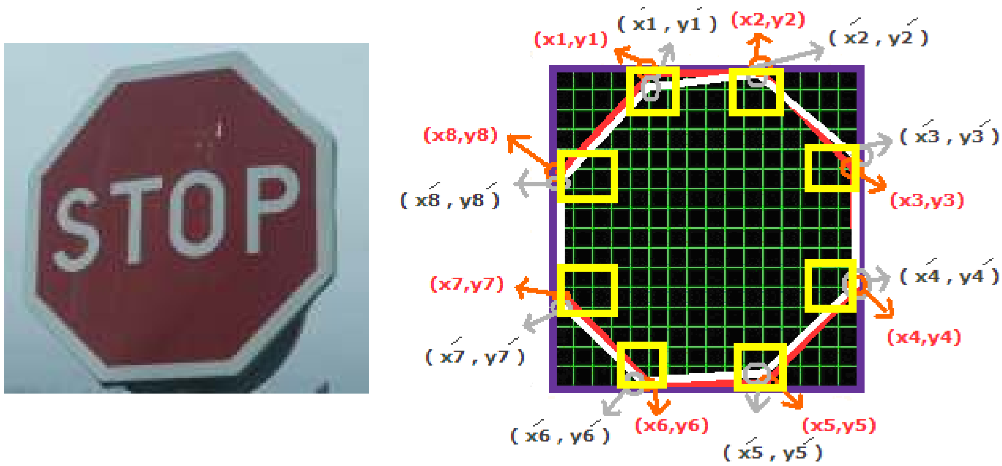
<u>Stop Sign</u>

Stop signs, being pattern A, have eight coordinates which are intersecting with the borders of the isolating rectangle. The table 3.4 below depicts the 8-vertices of pattern A and B which will be taken as the tuples in the distance formula to calculate the matching between them, based on which the system decides whether the object under consideration is a traffic sign. The coordinates are searched in 8 threshold regions.

Table 3.4: Coordinates of Stop Sign.

Region	Coordinates of Pattern A	Coordinates of Pattern B
1	$(x_1, y_1) = ((x\min + \frac{x\max - x\min}{3}), y\min)$	(x_1', y_1')
2	$(x_2, y_2) = ((x\min + 2\times\frac{x\max - x\min}{3}), y\min)$	(x_2', y_2')
3	$(x_3, y_3) = (x\max, (y\min + \frac{y\max - y\min}{3}))$	(x_3', y_3')
4	$(x_4, y_4) = (x\max, (y\min + 2\times\frac{y\max - y\min}{3}))$	(x_4', y_4')
5	$(x_5, y_5) = ((x\min + 2\times\frac{x\max - x\min}{3}), y\max)$	(x_5', y_5')
6	$(x_6, y_6) = ((x\min + \frac{x\max - x\min}{3}), y\max)$	(x_6', y_6')
7	$(x_7, y_7) = (x\max, (y\min + 2\times\frac{y\max - y\min}{3}))$	(x_7', y_7')
8	$(x_8, y_8) = (x\max, (y\min + \frac{y\max - y\min}{3}))$	(x_8', y_8')

Figure 3.24.b shows the coordinates of Pattern A in red and Pattern B in grey with the threshold regions highlighted by the yellow lines.



a. Sample Stop Sign [8] **b. Coordinates of Pattern A and Pattern B**

Figure 3.24: Stop sign recognition

Prohibitory or Regulatory Signs

The points taken for circular shapes in eight regions for matching in the grid space are the 8-polar coordinates at 45, 90, 135, 180, 225, 270, 315 and 360 degrees of the circle respectively. The coordinates of the circular Prohibitory/Regulatory Signs are given in the table 3.5, where

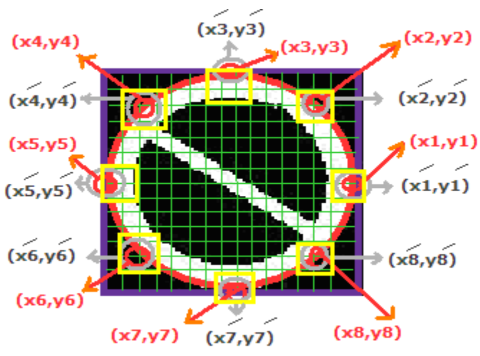
$$Radius \Rightarrow r = x\max - x\min \quad \textbf{(3.9)}$$

Table 3.5: Coordinates of Circular Signs.

Region	Coordinates of Pattern A	Coordinates of Pattern B
1	$(x_1, y_1) = (x\max,\ y\min + \frac{r}{2})$	(x_1', y_1')
2	$(x_2, y_2) = ((x\min + r\cos 45^o, y\min + r\sin 45^o)$	(x_2', y_2')
3	$(x_3, y_3) = (x\min + \frac{r}{2}, y\min)$	(x_3', y_3')
4	$(x_4, y_4) = ((x\min + r\cos 135^o, y\min + r\sin 135^o)$	(x_4', y_4')
5	$(x_5, y_5) = (x\min,\ y\min + \frac{r}{2})$	(x_5', y_5')
6	$(x_6, y_6) = ((x\min + r\cos 225^o, y\min + r\sin 225^o)$	(x_6', y_6')
7	$(x_7, y_7) = (x\min + \frac{r}{2}, y\max)$	(x_7', y_7')
8	$(x_8, y_8) = (x\min + r\cos 315^o, y\min + r\sin 315^o)$	(x_8', y_8')

Figure 3.25.b below shows the coordinates of Pattern A in red and Pattern B in grey with the threshold regions as yellow squares.

a. Sample Circular Traffic Sign [8]



b. Coordinates of Pattern A and Pattern B

Figure 3.25: Prohibitory Sign Recognition.

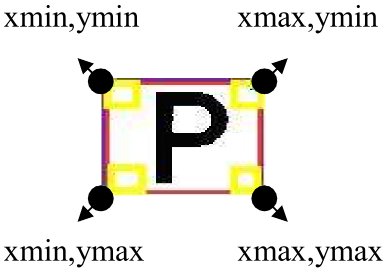
Information Signs

Rectangular Signs seems easiest to be recognized in the normal conditions (without higher aspect ratios) yet they are most difficult in the cases of higher aspect ratios.

a. In the normal case

In the normal case all 4 vertices of the pattern B will intersect the rectangular blob border; in fact the borders of the Pattern A are the lines of the isolating rectangular covering it as shown in the figure 3.26 b

a. Sample Information Sign [8]



b. Rectangular Blob

Figure 3.26: Information Sign Recognition.

Figure 3.26.b shows the coordinates of the blob vertices, and the pattern A in red. The coordinates of the candidate traffic Sign and the coordinates of actual traffic sign are the same. The four threshold regions are marked with yellow. The coordinates in such situation are depicted in the table 3.6 below. The above pattern matching will give 90% or above matching resemblance.

Table 3.6: Coordinates of pattern A and pattern B for Information Sign Recognition

Region Number	Coordinates of Pattern A	Coordinates of Pattern B
1	$(x_1, y_1) = (x\min, y\min)$	$(x_1', y_1') = (x\min, y\min)$
2	$(x_2, y_2) = (x\max, y\min)$	$(x_2', y_2') = (x\max, y\min)$
3	$(x_3, y_3) = (x\max, y\max)$	$(x_3', y_3') = (x\max, y\max)$
4	$(x_4, y_4) = (x\min, y\max)$	$(x_4', y_4') = (x\min, y\max)$

b. With higher Aspect Ratios (worst case)

There would be the cases where 2 or 3 vertices may intersect the isolating rectangle as shown in the figure 3.27. Therefore it is very difficult to grasp which vertices to match with the vertices of Pattern A. If by mistake wrong vertices are grasped for matching, it will result in the wrong calculations of the distances and therefore very high percentage of error will be generated resulting in the rejection of the pattern B.

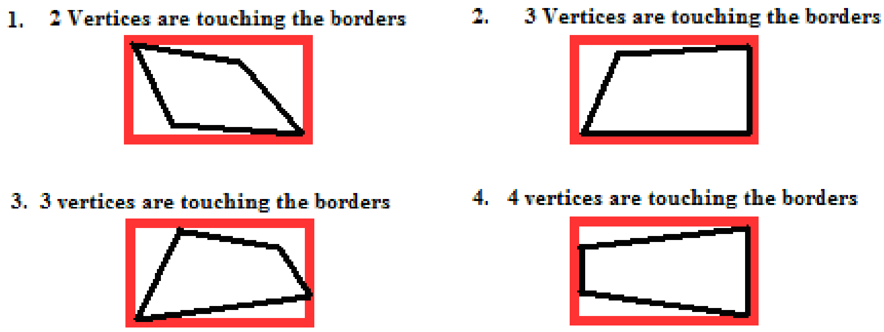


Figure 3.27: Shapes of Information Sign in different aspect ratios.

Solution to the problem:

A method to grasp the points in such situation is developed. According to this technique the isolated rectangle of blob is divided in four regions as shown in the figure 3.28. The coordinates are shown for each region, where

$$xmid = \frac{x\min + x\max}{2}, \; ymid = \frac{y\min + y\max}{2}$$

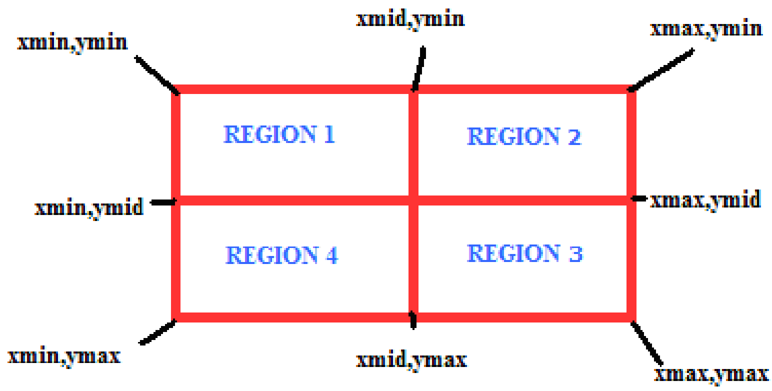


Figure 3.28: region splitting for Information Sign Recognition.

Following additional conditions were applied

i. (x_1', y_1') will be searched and captured always in Region-1 if it is not (xmin, ymin). Otherwise the candidate object is not the traffic sign.

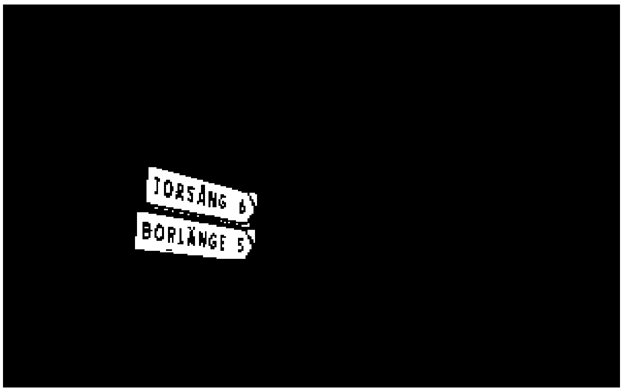
ii. (x_2', y_2') will be searched and captured always in Region-2 if it is not (xmax, ymin). Otherwise the candidate object is not the traffic sign.

iii. (x_3', y_3') will be searched and captured always in Region-3 if it is not (xmax, ymax). Otherwise the candidate object is not the traffic sign.

iv. (x_4', y_4') will be searched and captured always in Region-4 if it is not (xmin, ymax). Otherwise the candidate object is not the traffic sign.

An image with a very high aspect ratio is shown in figure 3.29.b in which two objects that were successfully recognized are shown as the output as the two Information Signs.

a. Sample Image [8]



b. Output Information Signs

Figure 3.29: Successful recognition of Information Signs under high aspect ratio.

Candidate Information Signs in figure 3.29.a never seems to be a rectangle nor a triangle or any other shape. By applying the additional checks, the candidate traffic signs will be recognized correctly under higher aspect ratios. The percentage of resemblance of the Information Signs in the above figure 3.29.b are 67.25% and 72.75% and are calculated in 0.15 seconds, which is a very efficient response time for the recognition of two information signs in the image.

3.9 IMPLEMENTATION OF BLOB TRACKING

To overcome the holes in the Kalman Filters and in traditional Blob Tracking, a new method of Blob Tracking is developed with the merger of two algorithms.

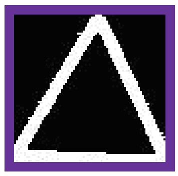
1. Two-Layered Blob Tracking[16].
2. Multi-Resolution Blob Tracking [14]

Following were the steps taken to get the predictive search region

Step 1: Isolate the candidate traffic sign

The candidate traffic sign is isolated by a minimum rectangle as shown in the figure 3.30.b; it is shown that the object is surrounded by the purple colored rectangle. Using these blobs, the spatial information (bounding boxes) of signs is determined.

a. Sample Warning Sign Image [8]



b. Isolated Labeled Object

Figure 3.30: Isolation of the candidate traffic sign.

Step 2: Blob Matching

Blob Tracking is performed by matching blobs between the frames. The sizes of blobs are expected to increase from frame to frame and the positions might change according to the vehicle movement.

In addition, due to the temporal occlusion or missed sign detection, blobs in *frame (t+1)* may not have a corresponding blob in *frame (t)*. In order to solve this problem, the algorithm was designed by merging two tracking methods.

1. Matching of blob using the **confidence measure** [14].
2. Blob labeling using the **distance function** [16].

Dalarna University
Röda vägen 3S-781 88
Borlange Sweden
Tel: +46(0)23 7780000
Fax: +46(0)23 778080
http://www.du.se

Blob Matching Using Confidence Measure

A tracking algorithm is developed that maintains the temporal relationships of the blobs detected in successive frames. Through this, multiple small blobs are preserved with a measure of confidence derived from a blob dissimilarity measure.

The confidence measure used here is an approximation, based on normalized area difference and centroid distance. The description for a target instance (blob) a_t at time t includes its area $A(a_t)$ (pixel count) and a centroid $c(a_t)$ are computed as equation (3.10):

$$c(a_t) = \frac{1}{A(a_t)} \sum_{i=0}^{A(a_t)-1} P_i \tag{3.10}$$

Where, *Pi* is the blob white pixels.

The size-based dissimilarity $D(a_t, b_{t-\Delta t})$ between two target instances (blobs) a_t and $b_{t-\Delta t}$ in two consecutive time slices t and $t-\Delta t$ is defined as equation 3.11

$$S(a_t, b_{t-\Delta t}) = \left| \frac{A(a_t) - A(b_{t-\Delta t})}{A(a_t) + A(b_{t-\Delta t})} \right| \tag{3.11}$$

Introducing the centroid distance in equation (3.12)

$$D(a_t, b_{t-\Delta t}) = \sqrt{\left(c(a_t) - c(b_{t-\Delta t})\right)^2} \tag{3.12}$$

And a maximum distance threshold Dm Proportional to the image dimensions given as in equation (3.13)

$$D_m = \frac{blob\_width + blob\_height}{2} \tag{3.13}$$

Equation 3.10, 3.11, 3.12 and 3.13 yields the following ***confidence measure*** equation (3.14) for the blob matching

$$K\ (a_t, b_{t-\Delta t}) = 1 - \frac{D(a_t, b_{t-\Delta t})}{D_m} S(a_t, b_{t-\Delta t}) \tag{3.14}$$

Where $D(a_t, b_{t-\Delta t}) \le D_m$ and $K\ (a_t, b_{t-\Delta t})$ =0 otherwise.

Intuitively, the measure K depicts the matching of blobs, in consecutive frames, that are of similar size and spatially close.

The fundamental principle behind the algorithm is to bring those blobs that must be matched, closer together in image space, to the point where the corresponding match confidence measure K will be in the threshold range [0.1 – 3.0]. Once the hypothesis is generated at a given level, they are propagated to the refinement of tracking based on the distance formula.

Keeping the Blob labeling record using the Distance Function

The difference between the blobs in frame (t) and frame (t+1) are found using the distance function. If the distance is under given threshold value 0.1, it will be assured that the blob in frame (t+1) is same labeled component which was present in frame (t). The matching distance between the two corresponding blobs in the current frame and the corresponding one is given by equation 3.15

$$d_{NO} = \frac{P_N}{P_N + Q_N} - \frac{P_o}{P_o + Q_o} \tag{3.15}$$

Where

P_N is the number of white pixels of the current blob.

Q_N is the number of black pixels of the current blob.

P_O is the number of white pixels of the corresponding blob

Q_O is the number of black pixels of the corresponding blob

Step 3: Update the Search Region

As soon as the blobs in *frame (t+1) and frame (t)* are matched the search region will be updated.

Let $(x\min, y\min)$, $(x\min, y\max)$, $(x\max, y\min)$ and $(x\max, y\max)$ are blob coordinates in frame (t).

Let $(x\min', y\min')$, $(x\min', y\max')$, $(x\max', y\min')$ and $(x\max', y\max')$ be the blob coordinates in *frame (t+1).*

The search region for the sign detection, which is a blob in the next *frame (t+2)* will be given as $(x\min'', y\min'')$, $(x\min'', y\max'')$, $(x\max'', y\min'')$ and $(x\max'', y\max'')$.

This search region updating process is shown (figure 3.31). The coordinates will be updated as

$$x\min'' = x\min' + x\min$$

$$x\max'' = x\max' + x\max$$

$$y\min'' = y\min' + y\min$$

$$y\max'' = y\max' + y\max$$

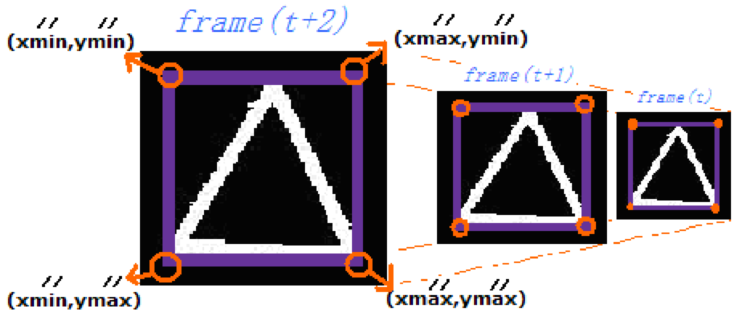


Figure 3.31: Blob Tracking.

The figure 3.32 below depicts the functioning of Blob Tracking with respect to the system.

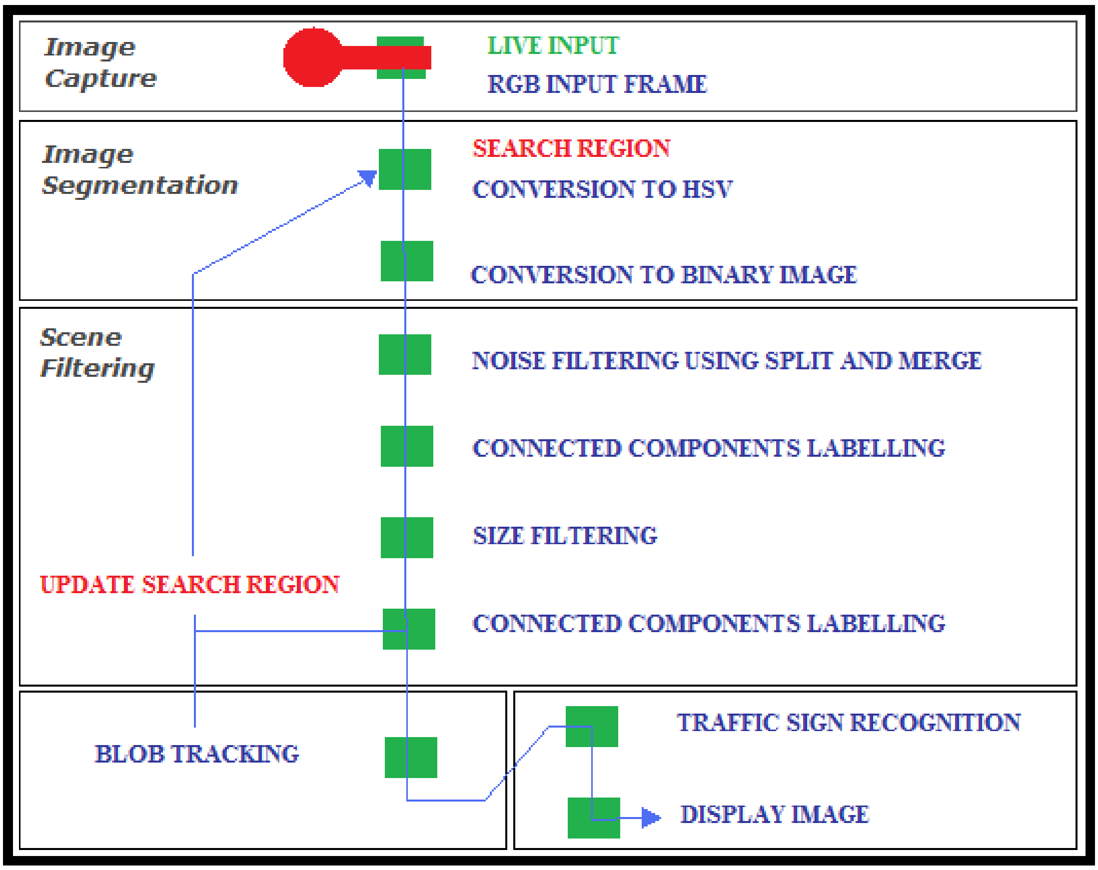


Figure 3.32: Functioning Of Blob Tracking In Real-Time Traffic Sign Recognition.

It can be seen that, as the frame (t) passes through the detection phase, blob coordinates are secured from labeled components. The system waits for frame (t+1) to appear. As soon as it appears the blob coordinates of the candidate traffic sign are captured. Based on the information from previous frames, Blob Matching is done and the blob coordinates in the frame (t+2) are updated which now acts as the next search region for the Detection and Recognition Process of the traffic sign. This process is kept on until the last frame of that traffic Sign remains in the scene. As soon as the traffic sign vanishes from the scene, the information is recollected in the same way for next traffic sign.

The figures 3.33.a and 3.33.b display the blob tracking for two successive frames (frame (t) and frame (t+1)) in time *t* respectively*,* when blob tracking was applied to the video. The red color rectangular blobs show the predicted search region in the image. While the yellow colored rectangular blobs shows the actual presence of the traffic sign that has been recognized.

a. Traffic Sign recognized in *frame (t)* is shown in the yellow rectangle, while the predictive region is shown in red for the traffic sign detection in the *frame (t+1).*

b. Traffic Sign recognized in *frame (t+1)* is shown in the yellow rectangle, while the predictive region is exhibited as red rectangle for traffic sign detection in *frame (t+2).*

Figure 3.33: Blob tracking for two successive frames; *frame (t) and frame (t+1).*

In this way enormous time can be saved by reducing the search time in detection and recognition of traffic signs from *frame (t+2)* onwards till the traffic signs vanishes from the scene. After the traffic sign vanishes from the screen, the search region will be extended to the whole image until the next candidate traffic sign is detected in the succeeding frames.

CHAPTER 4

RESULTS & ANALYSIS

4. RESULTS AND ANALYSIS

4.1 INTRODUCTION

As stated earlier in section 1.6, the objectives of Real-Time traffic Sign Recognition were

1. Systems' Response Time
2. System's Reliability
3. System's Efficiency

These factors are the performance criterion of our system. In this project 500 images of size 320 x 240 were used for testing. The traffic signs were assigned the recognition codes (table 4.1).

Table 4.1: Traffic Sign Recognition Codes.

Traffic Sign	Recognition Code	Border Color	Shape
Yield	1	Red	Triangle pointed downwards
Warning	2	Red	Triangle pointed upwards
Prohibitory	3	Red	Circle with X inside
Regulatory	4	Blue	Circle
Information	5	Blue	Rectangle
Stop	6	Red	Octagon

4.2 ANALYSIS OF BOB DE-INTERLACING ALGORITHM

➢ **Performance:**

After applying Bob De-interlacing, the acquisition process is ended up with an image which had

- Smooth picture movement is there, scrolling messages inside the traffic signs are cleared and they are easy to recognize with no jagged edges or combing.
- Super fluid movie achieved for recognition.
- Sharp picture produced with sharp edges of the traffic signs.
- 100% de interlacing achieved that is there won't be any interlaced lines left.

➢ Side-Effects of Bob De Interlacing Method

Following were the side effects in the image after applying Bob De Interlacing method

- Still horizontal lines appeared to be up or down in the image.
- Needed to resize the movie during play so higher computational time is there.
- Needed to play 50fps, so a faster processor or a faster codec is needed.

4.3 ANALYSIS OF HSV COLOR MODEL IMAGE

➢ Advantages of converting to HSV

- The HSV color space is quite similar to the way in which humans perceive color. The other models define color in relation to the primary colors. The colors used in HSV are clearly defined by human perception which is not always the case with RGB or CMYK.
- Hue played the central role in the color detection because it is invariant to the variations in light conditions as its scale invariant, shift invariant and invariant under saturation changes.
- HSV model has been very helpful to resolve the problems of Shadows and Highlights[2] or the chromatic variation of the day light. For example a faded image is considered as one with the low saturation; the value of saturation can be tuned of that color as per the weather conditions. Therefore it is able to preserve the maximum image information.

➢ Problems with hue in HSV Color Space

The hue coordinate is unstable and small changes in the RGB caused strong variation in hue. It suffered from three problems as stated by Fleyeh [2]:

- When the intensity is very low or very high the hue is meaningless.
- When the saturation is very low, the hue is meaningless.
- When the saturation is less than the threshold value, the hue becomes unstable.

4.4 ANALYSIS OF IMAGE SEGMENTATION

➢ Performance

Segmentation of the image is the most critical part of the Real-Time traffic Sign recognition. The whole process of Traffic Sign Recognition is like a funnel as shown in the figure 4.1 in which the Image segmentation holds the most significant position because if important data related to the traffic sign is lost in segmentation then the traffic sign will be lost.

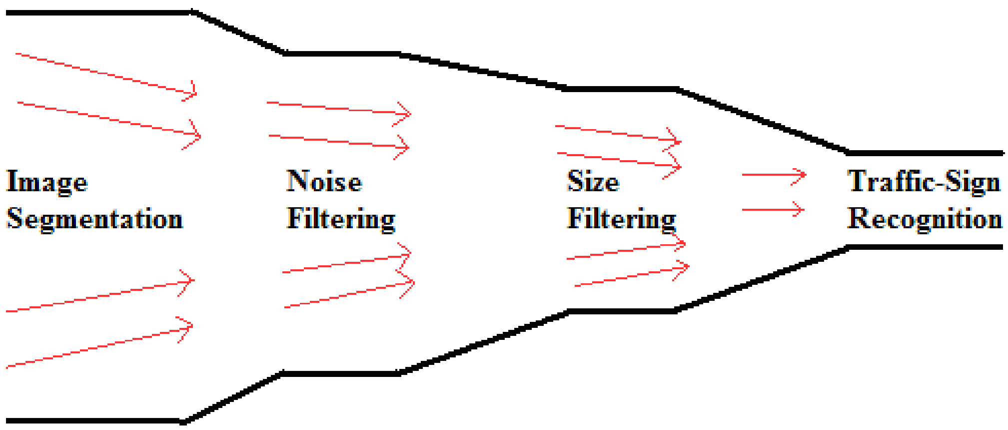


Figure 4.1: the recognition funnel.

- In order to cope up with shadows and highlights in the image, values of Hue, saturation and Value had to be tuned to get the segmentation output for the traffic sign detection.

- Red color segmentation is most problematic because of its huge spectrum in the hue's 360 degrees circle. Whereas the image segmentation for black, blue, white and yellow was very successful in the normal conditions as well as in the shadows and highlights.

- 500 images were tested for image segmentation, 100 images for each respective color. The graph in figure 4.2 below shows the success rate of image segmentation for each sign color.

- The reason why the red color segmentation is not successful (as depicted in the graph 4.2) under abnormal conditions is that, the red has the huge hue spectrum. Also the red borders of the warning sign, yield sign or prohibitory signs become very thin due to motion blurring or under adverse illumination, which could be considered as noise by the filters applied later on.

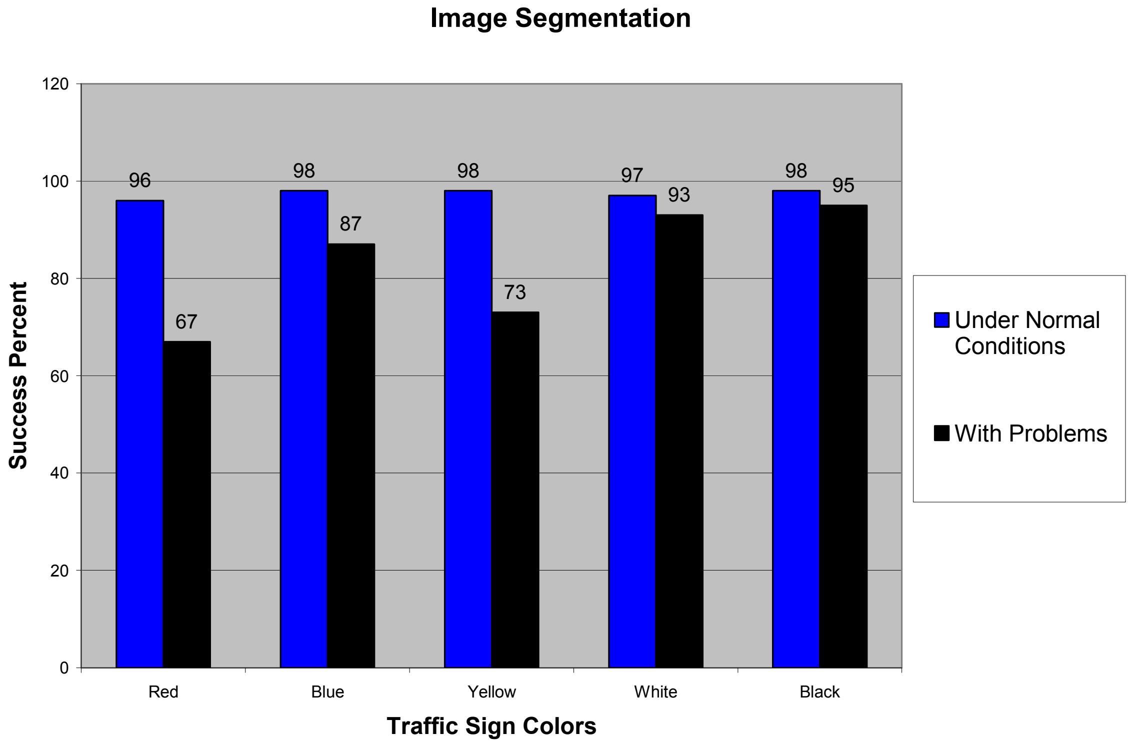


Figure 4.2: Segmentation Success Percentage.

- The blue signs, however, have good success percentage under abnormal conditions because they have broad blue color borders which exist even under poor conditions.

- The success percentage of segmentation of white and black colors is not varying because there is no hue present in both colors. The only difference is because of the noisy picture captured by the camera.

4.5 ANALYSIS OF MODIFIED SPLIT AND MERGE FILTER

➢ Performance:

Filtering with modified split and merge reduced the number of objects in the image. Since there would be less number of objects in the image, there would be less time to be taken by recognition procedure. Therefore it will increase the efficiency and speed of the traffic sign recognition in the system.

500 images were tested using the modified split and merge algorithm. The graph in figure 4.3 depicts the average percentage of object reduction with the noise filters for red, blue and yellow colors in each of the images.

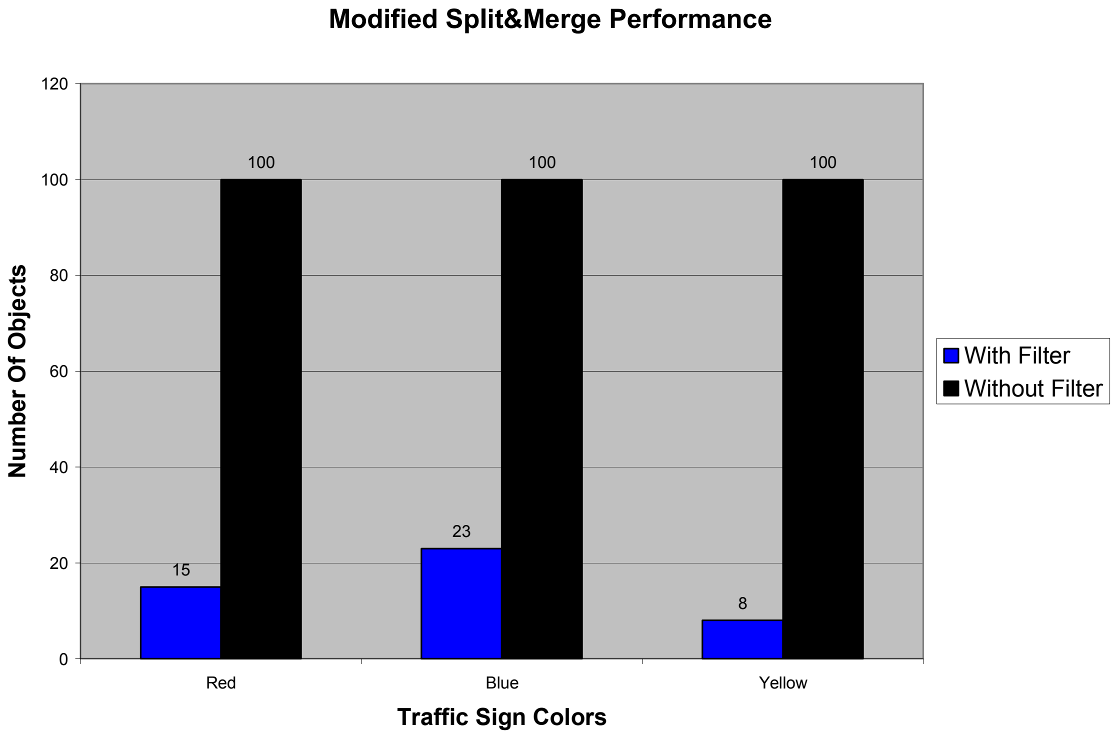


Figure 4.3: Split and Merge Efficiency.

- ❖ The graph 4.3 shows the percentage reduction of the noise. The figures in the graph show very efficient results for the modified split and merge noise filter for red, blue and yellow objects which are 15%, 23% and 8% noise reductions for each color respectively.

- ❖ The reason why the blue color has still more noise after applying the filter, is because of the color of the sky which covers a very large part of the image.

- ❖ The yellow color has the highest noise reduction percentage because it has the maximum small particles of noise coming from the sunlight in the background as shown in figure 3.8.b.

- ❖ When modified split and merge algorithm is applied, noise is removed without blurring the image and without losing the pictogram information. It will also prevent the system from making False alarms that could be possible by the incorrect Traffic Signs detection.

➢ Problem with the Split and Merge:

The problem with split and merge is that any two regions may be merged if adjacent and if the larger region satisfies the homogeneity criteria, but regions which are adjacent can create problem because they may have different parents or may have different size.

4.6 ANALYSIS OF SIZE FILTERS

➢ Performance:

Size Filters proved to be very efficient to reduce false alarms. The blue colored traffic sign detection has the problem with the background because of the sky color which is blue. Since the sky covers a big region which is usually 1/4th of the overall image therefore it is cut off using the size filter. Similarly with the red color signs, the big problem was the occlusion created by the red colored building in the backgrounds. These are removed because of their large size. Following graph in figure 4.4 depicts the averages percentage reduction of un-important objects using the Size Filters.

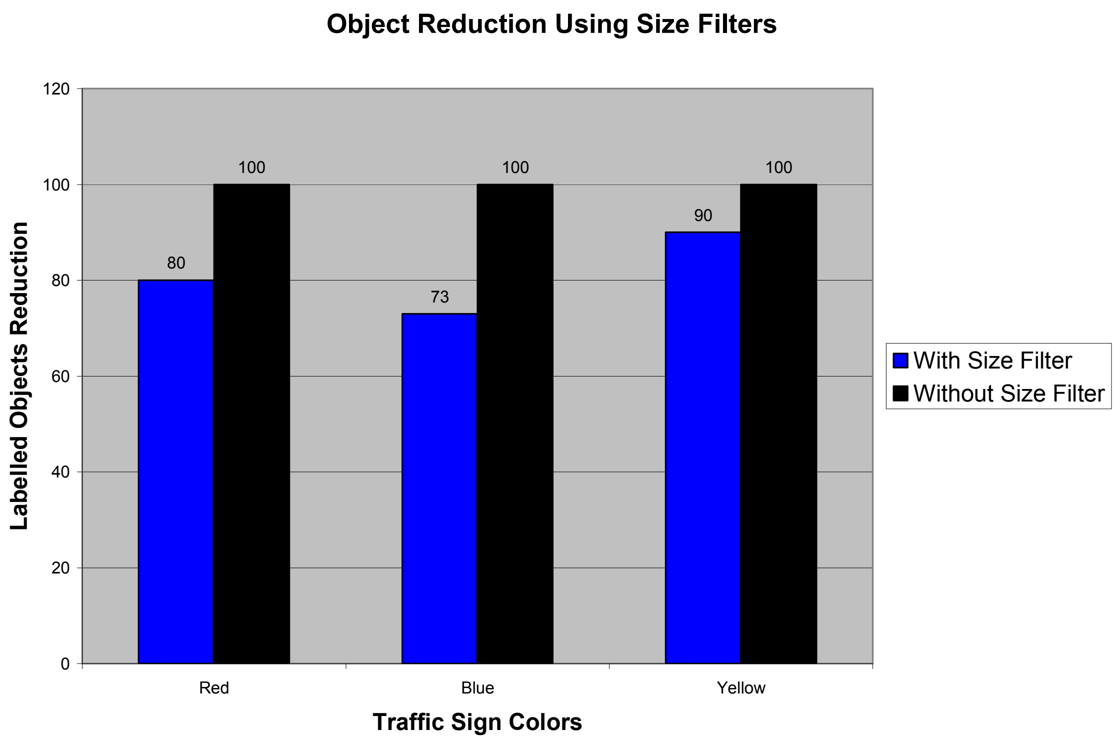


Figure 4.4: size filter efficiency.

The graph shows the average percentage reduction of the number of objects after applying size filters when it is tested for 500 images. Note that the ratio is taken between the already reduced objects by split and merge filter before labeling, with the number of labeled objects further reduced by the Size Filters.

4.7 ANALYSIS OF PATTERN MATCHING ALGORITHM

➢ Performance

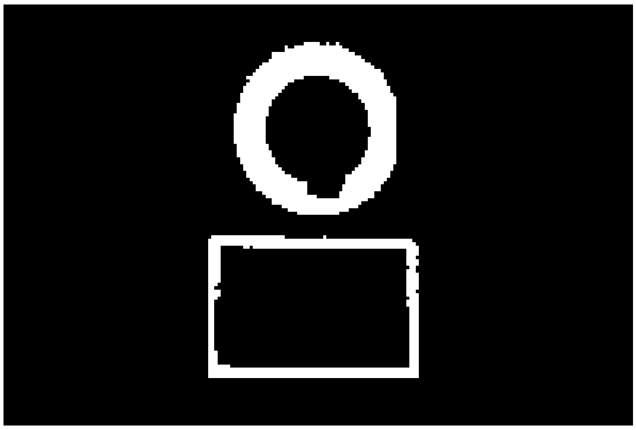
Pattern matching proved to be a great success in this project for traffic sign recognition. In this method only the vertices of the objects were processed instead of comparing each and every pixel for recognition which has increased the systems efficiency by reducing the recognition time. Imagine that there is an object of 64x64 = 4096 pixels which was processed in the previous methods, while here an object with 3x3=9 vertices for patterns A and pattern B (for triangles) to 8x8=64 vertices (for an octagon) is processed. Therefore the system's response time is increased. The average time calculated using pattern matching for the traffic signs recognition are given in the table 4.2.

Table 4.2: Efficiency of Pattern Matching Recognition.

Traffic Sign	Overall average Time for recognition without tracking (seconds)	Percent Accuracy In Normal Conditions	Recognition with higher aspect ratios	Percent Accuracy in real time
Yield	0.137	98.5%	88%	93%
Warning	0.133	98%	87%	93%
Prohibitory	0.144	98%	90.5%	91%
Regulatory	0.111	97%	90%	98%
Information	0.131	96%	92%	99%
Stop	0.158	95%	83%	95%

Pattern matching algorithm has solved the problem of damaged signs at a great extent. Example of a distorted candidate traffic sign is given in figure 4.5.b below in which the candidate traffic sign has 71% resemblance with the prohibitory sign and it has been recognized correctly.

a. Distorted Prohibitory Sign [8]



b. Output With Successful Recognition

Figure 4.5: Damaged sign successful recognition.

Speed Comparisons

The time taken by the pattern matching algorithm can be compared with the method developed by Gilani [19] who worked with the real time speed-limit sign detection using fuzzy art map. The recognition timings are given in the table 4.3 below which was for speed-limit traffic sign recognition:

Table 4.3: Fuzzy ARTMAP for Outer Edge of Sign Recognition [16].

	Minimum Time (seconds)	**Maximum Time (seconds)**	**Average Time (seconds)**
Training	0.17	0.42	0.295
Testing	0.08	0.231	0.155
Recognition	0.01	0.03	0.02
Overall	0.26	0.681	**0.475**

The comparison between pattern matching recognition with the fuzzy art map is given in the graph 4.6 below where pattern matching algorithm speed beats the speed of ART MAP with 0.325 seconds. Also note that the fuzzy ARTMAP was applied only to recognize the speed-limit traffic signs, which is included in the category of prohibitory signs in this thesis work.

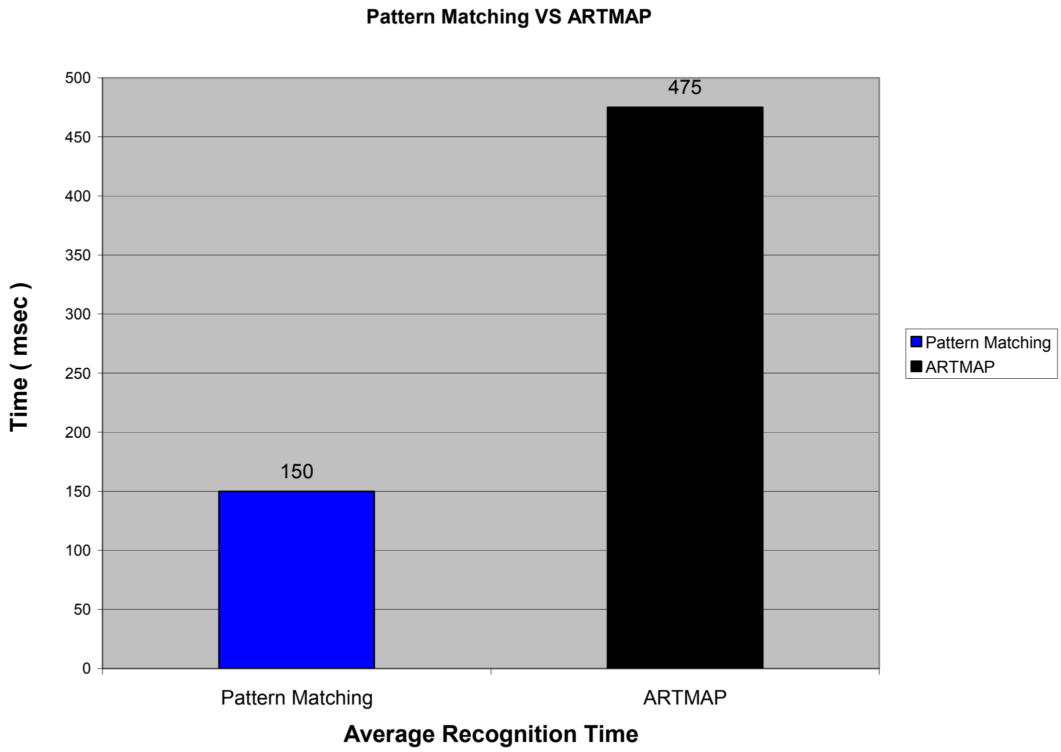


Figure 4.6: Efficiency Comparison of Pattern Matching VS Fuzzy Artmap

Accuracy Comparisons

The recognition accuracy of this thesis work can be compared with the accuracy of the research work of Hassan Gilani [20] who worked on road sign classification based on SVM. Accuracy of the shape recognition part of the traffic signs has been compared. He implemented four methods

1. Haar Features
2. Geometric Moments
3. Effective FT Coefficient.
4. Orthogonal Fourier Mellin.

It is worth mentioning that these methods were applied on the stationary images in the offline mode as they required training and the testing of the data for classification.

The table 4.4 shows the results achieved by the system of Hasan Gilani [20]

Table 4.4: Research of Hasan Gilani [20]

Feature Extraction Methods	Shape Recognition Accuracy	Speed-Limit Recognition Accuracy
Haar Features	97.77%	96.00%
Effective FT Coefficients	99.04%	90.67%
Orthogonal Fourier Mellin	92.22%	50.67%
Geometric Moments	92.22%	62.67%

Following figure 4.7 shows the graph with the difference between the accuracy of the thesis comparing to the four methods developed by Hasan Gilani[20] for speed limit sign shape recognition which is a prohibitory sign in the thesis work.

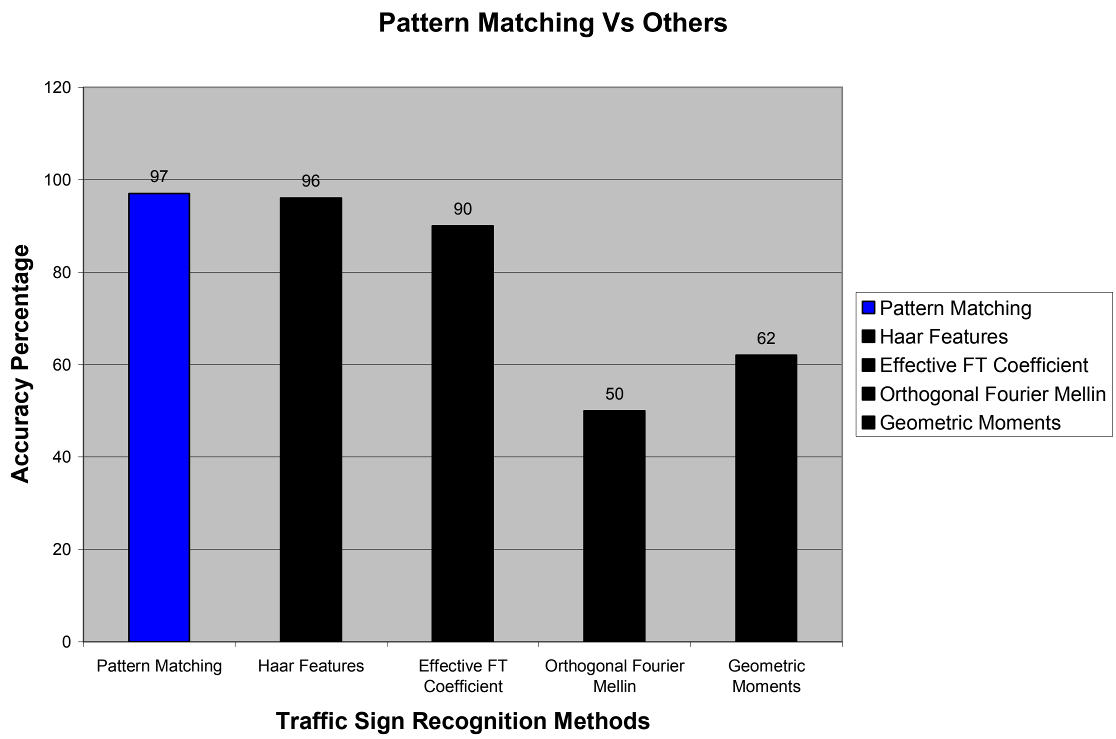


Figure 4.7: Pattern Matching Vs Other methods

The figure 4.7 shows the accuracies of shape recognition of various methods developed before. 97% accuracy achieved with the pattern matching method on stationary images for the traffic signs recognition is a very good achievement.

4.8 ANALYSIS OF BLOB TRACKING ALGORITHM

Blob Tracking enhanced the system's response time by producing 80% reduction of the traffic sign recognition time. The system takes average 0.15 seconds to scan the whole image until it receives a blob. At this time the search area is updated and in the next frame only this area is processed for the traffic sign detection. *The average response time after applying Blob Tracking is 0.04 seconds.* Comparing to the Fuzzy ARTMAP, the system's overall response time is shown in the figure 4.8.

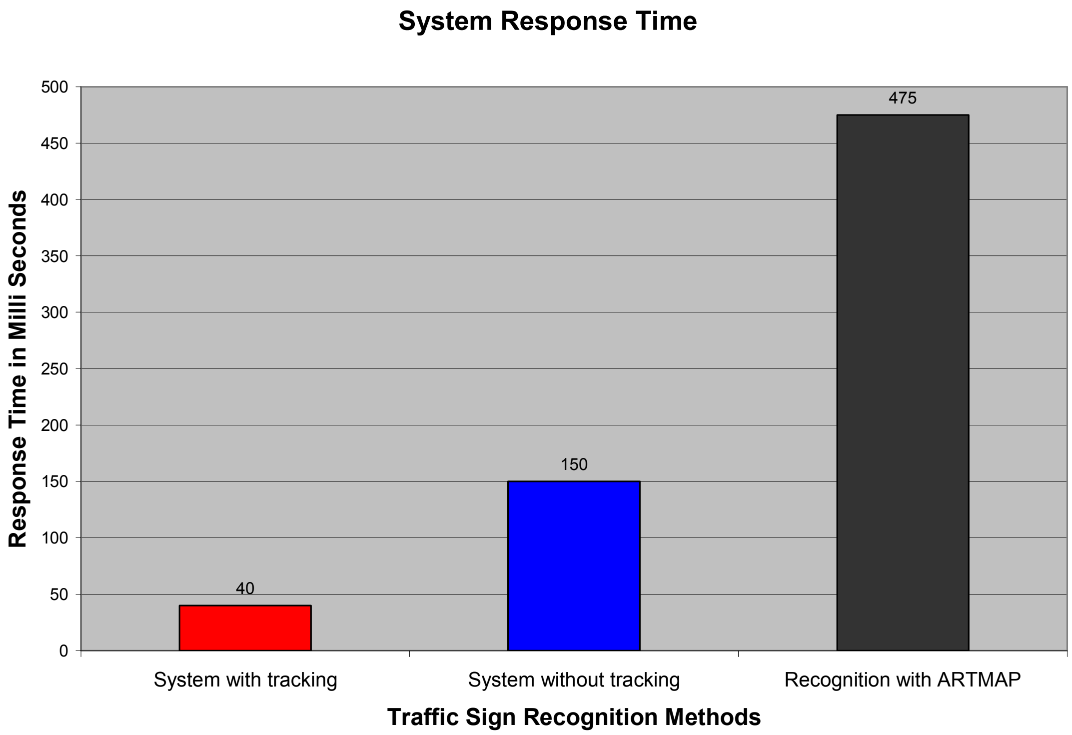


Figure 4.8: Comparison of Systems Response Time.

It can be seen in the graph that there is a big improvement in the overall system's response time which is now just 0.04 seconds, that is, it is the 10% of the time ARTMAP had taken for recognition.

The only problem in the blob tracking algorithm is miss detection of blobs which is kept in the future works. Due to the miss detection of the blob, the system has to search the whole image once again until it finds the object. If blob is correctly detected, the search space could be concentrated around the same trajectory which could maintain the search space for tracking.

CHAPTER 5

CONCLUSION
&
FUTURE WORKS

5. CONCLUSION AND FUTURE WORKS

5.1 CONCLUSION

To conclude the thesis report, the Real-Time Traffic Sign Recognition System presented here has been a success. The three targets based on Speed, Reliability and Efficiency which was the necessary parameters for the real-time environment has been accomplished. The average target values achieved by our system are illustrated in the table 5-1 below

Table 5.1: Achieved Target Values.

Parameters	Average Values
Recognition Average Speed	0.15 seconds
Average Percent Reliability in real time	95% correct recognition
System's Response Time	0.04 seconds

- Image Segmentation is found to be the most critical task during the whole project. This is because of the illumination conditions especially due to highlights and shadows. There is always a need to tune the parameters in segmentation during the process. However the Image Segmentation that has been done in the project was satisfactory.

- Noise filtering with modified Split and Merge algorithm has been very efficient. The success of traffic sign recognition accuracy and time mostly depends upon the image filtering by modified split and merge, as it reduces the number of objects to be appointed as candidate traffic signs for recognition. Modified Split and Merge is harmless to the internal information carried by the traffic sign which could be used for the classification. Therefore it is very compatible for the traffic sign recognition system in real time.

- Size Filtering of the objects also reduced the number of objects present in the image which could be un-necessary information which can waste the recognition time. Eliminating them thus improved the response time of the system.

- The Pattern Matching and Hausdorff Distance concepts are not new but the way they are applied to the project for recognition is a new technique and it proved to be very efficient and fast as well as accurate for Traffic Sign Recognition. Secondly it has solved the problem of recognition of traffic signs with higher aspect ratios at a great extent. It is also observed that the damaged signs were recognized correctly most of the time using this method. Thus the two previous problems with the traffic sign recognition have been solved by this method.

- Blob tracking has been very useful to enhance the system's response time by narrowing down the search space for traffic sign detection. The only problem was the misdetection of the blobs in the successive frames which could refine the response time. This has been kept as the future work.

In this project a complete package for Traffic Sign Recognition has been developed. Since the Traffic Sign Recognition is in the real-time the most important thing considered here was time and speed with a bit of compromise with the accuracy of recognition especially under adverse conditions. Thus a very good platform has been achieved to extend this work for traffic sign classification in real time which used to take a huge computational time before.

5.2 FUTURE WORKS

- Image Segmentation is the most critical phase of the recognition process. There is a need for a more robust segmentation. This could be accomplished by applying fuzzification techniques to the hue, saturation and value parameters, which could be adaptable to the environmental conditions and could be tuned according to the conditions.

- *Occlusion* is found to be a big problem. Occlusion with the traffic signs occur if a car, buildings or sign boards with the same hue, saturation and value occurs in the background of a traffic sign. It is possible some how to remove such objects using size filters. But if the traffic sign is present in the middle of that object then it is a big challenge to separate them. In this case the only choice left was to reject such traffic signs. Therefore author has recommended this issue to be solved as the future work.

- A problem is faced with the recognition of No-Entry signs. The problem is, when the image is segmented for red color hue, the No-Entry signs are divided into two different labeled objects in the form of unrecognizable semi-circles. The broken parts of the No-Entry Sign are shown in the figure 5.1.b. This problem is kept as the future work.

a. No-Entry Sign [8]

b. Segmentation Of No-Entry Signs

Figure 5.1: No-Entry Signs.

- Blob Tracking has the problem with the miss detection of the blobs in the successive frames. If this problem could be solved for real-time, then it can save a lot of time by refining the search space for the traffic sign recognition. To maintain the labeling of the blobs in the successive frames is difficult because at any instant of time there is a possibility that the labeled object will be lost or there could be many more labeled objects appearing in the next frame as the blobs of the candidate traffic sign. To keep the identification for every blob is therefore a difficult task and it is kept as the future work.

- This thesis work has been applied on the six different types of the traffic signs. The scope of the project for the other traffic signs can be enhanced in the future. The other signs may include the cross-bars or signs with the green color inside which is not included in the scope of our thesis work are shown in figure 5.2.

Figure 5.2: Traffic Signs out of our project's scope.

- This project can be extended by applying classification methods for character recognition in the pictograms. The classification methods like Support Vector Machines or ART maps may be used to classify the traffic signs.

REFERENCES

[1] "Intelligent Transportation", http://rtm.science.unitn.it/intelligent-optimization/research.html.

[2] Fleyeh.H, "Color detection and segmentation for road and traffic signs," IEEE Conference on Cybernetics and Intelligent Systems, vol. 2, pp. 809-814, 2004.

[3] Fleyeh.H, "A novel fuzzy approach for shape determination of traffic signs," Second Indian International Conference on Artificial Intelligence, Pune, India, 2007.

[4] Fleyeh.H., "Shadow and Highlight Invariant Color Segmentation Algorithm for Traffic Signs," 2006 IEEE Conference on Cybernetics and Intelligent Systems, pp. 1-7, 2006.

[5] H. Fleyeh, S. Gilani, and M. Dougherty, "Road sign detection and recognition using fuzzy ARTMAP : a case study Swedish speed-limit signs," The 10th IASTED International Conference on Artificial Intelligence and Soft Computing at Palma de Mallorca, Spain, 2006.

[6] Fleyeh.H and M. Dougherty, "SVM Based Traffic Sign Classification Using Legendre Moment," Third Indian International Conference on Artificial Intelligence, 2007.

[7] Fleyeh.H, M. Dougherty, D. Aenugula, and S. Baddam, "Invariant Road Sign Recognition with Fuzzy ARTMAP and Zernike Moments," 2007 IEEE Intelligent Vehicles Symposium, pp. 31-36, 2007.

[8] "Hasan Fleyeh's Traffic Signs Record Database", http://users.du.se/~hfl/traffic_signs_rec/.

[9] D. L. Escalera., J. Armingol, and M. Mata, "Traffic Sign Recognition and Analysis for Intelligent Vehicles," Image and Vision Computing, vol. 21, pp. 247-258, 2003.

[10] "Bob De Interlacing",www.100fps.com.

[11] "Noise Filters", http://lorien.ncl.ac.uk/ming/filter/filter.htm.

[12] "Split and Merge algorithm", http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/ MARBLE/medium/segment/split.htm.

[13] P. Z. Hasan Fleyeh, "A Contour Based Separation of Vertically Attached Traffic Signs," 2007.

[14] A. R. J. Francois, "Real-Time Multi Resolution Blob Tracking," 2004.

[15] K. C., P. Wang, Zhiqiong, W. G., and R. McCann., "Kalman Filter based Tracking System for Automated Inventory of Roadway Signs," 2006.

[16] S. Aryuanto and Y. Koichi, "Improving the performance of traffic sign detection using blob tracking," IEICE Electronics Express, vol. 4, pp. 684-689, 2007.

[17] "Components labeling", http://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm.

[18] N. M. and M. Y., "High Speed Connected Component Labeling as a Killer Application for Image Recognition Systems by Dynamically Reconfigurable Processor," IPFlex Inc. Tokyo, Japan, 2003.

[19] S. O. Gilani, "Swedish Speed-limit Sign Detection and Recognition using Fuzzy ARTMAP," 2005.

[20] H. Gilani, "Road Sign Recognition based on Invariant Features using SVM," 2007.