6-10-2019 10:30 AM

Assessment of Driver's Attention to Traffic Signs through Analysis of Gaze and Driving Sequences

Shabham Shabani, *The University of Western Ontario*

Supervisor: Bauer, Michael A., *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Science degree in Computer Science

Abstract

A driver's behavior is one of the most significant factors in Advance Driver Assistance Systems. One area that has received little study is just how observant drivers are in seeing and recognizing traffic signs.

In this contribution, we present a system considering the location where a driver is looking (points of gaze) as a factor to determine that whether the driver has seen a sign. Our system detects and classifies traffic signs inside the driver's attentional visual field to identify whether the driver has seen the traffic signs or not. Based on the results obtained from this stage which provides quantitative information, our system is able to determine how observant of traffic signs that drivers are. We take advantage of the combination of Maximally Stable Extremal Regions algorithm and Color information in addition to a binary linear Support Vector Machine classifier and Histogram of Oriented Gradients as features detector for detection. In classification stage, we use a multi class Support Vector Machine for classifier also Histogram of Oriented Gradients for features. In addition to the detection and recognition of traffic signs, our system is capable of determining if the sign is inside the attentional visual field of the drivers. It means the driver has kept his gaze on traffic signs and sees the sign, while if the sign is not inside this area, the driver did not look at the sign and sign has been missed.

Keywords: traffic sign detection, traffic sign recognition, Advance Driver Assistance System, drivers' gaze, drivers' behavior, traffic sign data set.

Acknowledgements

First and foremost, I would like to express my very profound gratitude to my supervisor, Dr. Michael A. Bauer for providing me continuous support and teaching countless lessons throughout my research study. His patience, encouragement and valuable advice made me motivated to have the best performance. He is not only an outstanding mentor, but a compassionate friend. I highly appreciate his time, consideration and ideas to make me have a pleasant research experience in Western University. I have been extremely lucky to be aided by him as a tutor.

I would also like to thank my laboratory mentor, Prof. Beauchemin for his useful suggestions and support.

My deepest appreciation goes to my lovely parents for their never-ending loves sacrifices and encouragements. They constantly support me with endless inspirations and I would not be able to pass this experience without their positive energy and support even from far away.

My special thank goes to my sister, who is not only a sister, but also a wise and wonderful friend who has always given me support and confidence needed to figure out all the tasks I have met throughout my life.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

Due to the increasing number of vehicles and pedestrians, drivers are in need of the safety measures to decrease the possible risk of accidents. Advanced Driver Assistance Systems (ADASs) have significantly progressed in many aspects. By augmenting vehicles with advanced technology and informing drivers about highly dangerous circumstances or even taking automatic actions, crashes can be prevented.

As stated in [10], the main reason for accidents is human error. The main purpose of an Intelligent Transportations System (ITS) is to generally reduce fatal situations and improve the safety during the driving by providing the driver critical information about the roads and automate repetitive tasks. ITSs can be divided into two different views: Independent Systems and Driver Assistance Systems. Independent Systems are characterized by autonomous vehicles where there are a lot of ongoing research and development. However, given some of the challenges facing autonomous vehicles (including non-technical ones), their wide spread availability and use is still some years away. In contrast, vehicles today already include technology that can assist a driver and so there is still substantial effort devoted to these Advanced Driver Assistance Systems (ADASs). Some aspects of ADASs have been in use for a long time and are being improved over time. GPSs (Global Positioning Systems) were introduced in the 1990s and have become commonplace in many vehicles. Other types of ADASs have been introduced recently, such as advanced cruise control, automatic parking, collision avoidance, automatic braking, and lane departure and warning systems. More recent efforts have begun to look at the use of cameras, lidars, radars and other sensing technologies for use in even more sophisticated ADASs. Some of this work includes leverage image analysis technologies for analysis of data from onboard cameras.

One area that has gained increased interest in both ADASs and autonomous vehicles is that of traffic sign detection and recognition. Designing systems for traffic sign detection and recognition (TSDRs) is a challenging problem and has been studied over the past 30 years. As mentioned in Paclik [11], Japan was the first country where scientists started investigating automated TSDR systems. Thereafter, many different TSDR methods and their obstacles have been highlighted by researchers.

The main purpose of designing traffic signs is to control traffic by informing both drivers and pedestrians about the restrictions and circumstances of the road. In particular, signs provide drivers with necessary information about the roads and possible traffic conditions and constraints. In fact, one of the main roles for TSDR systems is to warn the driver of approaching traffic to avoid accidents.

One area that has received little study is just how observant drivers are in seeing and recognizing signs. One reason for the little work in this area is that in order to actually measure this, one needs data on actual driving sequences and data on the gaze of the driver. The assumption is that if one

can determine where a driver is looking (point of gaze) then there is a strong likelihood that the driver has at least seen the sign once.

The research presented in this thesis provides quantitative information to help answer questions about just how observant of traffic signs that drivers actually are.

However, there is no way to know if the driver has actually recognized the sign or not. The driver may keep his gaze on a traffic sign, but the sign is not recognized due to his distraction. In fact, the main objective of our work is to focus on the objects that the driver has focused on and driver's distraction cannot be measured through our work.

Data for this research comes from RoadLab [12]. RoadLab is an initiative that provides data for the development of Intelligent Advanced Driver Assistant Systems (I-ADAS). The aim is to support research for ADASs cognizant of driver behaviour, intent, surrounding traffic and general driving conditions. The RoadLab dataset provides a resource for researchers to analyze driver data with the hope of reducing the social and economic costs caused by human errors made while driving.

RoadLab data is collected by an in-vehicle laboratory instrumented with an on-board diagnostic system (OBD-II) using the CANbus protocol. The instrumentation collected video sequences of the driving environment in front of the vehicle, the ocular behaviour of the driver such as driver gaze, and other information, such as the geographic position of the vehicle via GPS and data describing the state of all the vehicle parameters such as brake pedal position, steering wheel position, etc. This information was collected on 16 drivers who drove an instrumented vehicle. Each driver followed the same route and each drive took approximately 50 minutes.

1.1 Problem Overview

Our research problem is to measure aspects of how actual drivers view traffic signs. Data from RoadLab provides driving sequences as well as data on driver gaze. Our overall methodology entails four different steps.

1. The attentional visual field of the driver is obtained by analyzing the combination of a front-view stereo imaging system and a non-contact 3D gaze tracker; some of the algorithms for this aspect of the work has been done in previous research on the RoadLab data [12]. The gaze of the driver on an image, specifically the point of gaze on the image, was determined for all driving frame sequences in our data set.
2. The second stage of the work is to analyze images in the sequences to find and identify traffic signs. This part of the work is analogous to developing a traffic sign detection and recognition system (TSDR). This requires us to analyze each image to detect regions within the image that may be traffic signs, then determine if those regions are traffic signs and if so what kind of sign.
3. For each frame, using the point of gaze and the regions corresponding to traffic signs, the intersection of the point of gaze with regions is determined. If there is an intersection with a region, then the traffic sign corresponding to that region is deemed to have been "gazed upon" by the driver. This is recorded and for each driver and for each image in the sequence

we have a corresponding data entry on signs in the image (as detected) and whether seen or not.

4. Finally, for each driver we analyze the sequence of data on signs detected and those that have intersected the driver's gaze and determine the number of signs, and whether or not the driver actually "saw" the sign (as determined by gaze). This provides information on just what the drivers did and did not see during their drives.

Using this approach, we can find not only the number of signs seen by the driver, but also the number of signs missed and the type of the sign.

1.2 Thesis Overview

A critical component of our work is the development of a TSDR system that can be used to detect and recognize traffic signs in the Canadian context. There has been a substantial amount of work in this area and we build on and adapt previous work done in this area. In Chapter 2, we provide an overview of work in this area and approaches taken in previous work. Then in Chapter 5, we outline our approach to traffic sign detection and recognition.

While there has not been a lot of work done on measuring what drivers see, there has been some work on driver behavior. We provide an overview of work in this area in Chapter 3.

In Chapter 4 we describe the data that was used for the research. This includes describing the image data that we used for training our methods for sign detection and recognition as well as more detail on the RoadLab data.

In Chapter 6 we present our method for analysis of the sign and gaze data and present the results of the analysis of analyzing that data for which signs are seen, not seen, etc. for the drivers in the data set.

Chapter 7 provides a summary of the results and conclusions and identifies further directions.

1.3 Contributions of the Thesis

We have designed a complete frame work for TSDR based on a North America Data set and used driver's gaze to determine the driver's vigilance to signs. Our system identifies signs that have been recognized or signs missed in actual driving sequences. As far as we know, previous work has focused only on traffic sign detection and recognition and have not considered drivers' gaze and actual driving sequences. Our system is the only one to add driver's gaze as an important factor in identifying driver's inattention.

Our data set comes from our instrumented vehicle and drivers' gaze is calculated by a front view stereo system installed on our vehicle. The combination of Maximally Stable Extremal Regions and color information in addition to a binary linear Support Vector Machine (SVM) and Histogram of Oriented Gradients (HOG) was used for detection. A multi Class SVM and HOG feature vectors

were combined for recognition part and our intersection method was used for the analysis of driver's gaze. This leads to our analysis of the number of signs those drivers have or have not seen.

Chapter 2
Related Work on Traffic Sign Detection and Recognition

Traffic Sign Detection and Recognition systems detect and recognize the signs coming from sequences of recorded frames by a camera mounted on a vehicle. In most of the TSDR systems, cameras are mounted in front of the vehicle, however in recent TSDR systems, there may be one more camera at the rear or the side of the vehicle that can record the signs located behind or beside the vehicle. In Figure 2.1, a vehicle equipped by TSDR systems is illustrated. This vehicle contains a mounted camera to record the images.

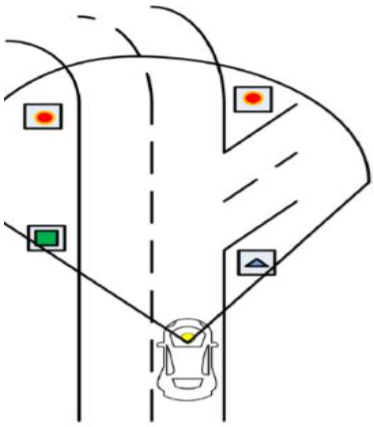


Figure 2. 1 *Vehicle with a TSDR system*

Although various countries have designed different traffic signs based on their rules and regulations, there are similar sets of traffic signs based on their shape and color. Circular, Octagonal, Triangular, Rectangular, Square and Pentagonal are the common shapes. Yellow, red, blue, green and white are distinctive colors used to design traffic signs. In spite of the noticeable shape and colors used in traffic signs, they are often not noticed, intentionally or unintentionally, by drivers. For example, due to high speeds, a driver may not have enough concentration.

Due to different designs of traffic signs from one region to another and also different designs based on their purpose, most TDSRs narrow their focus down to sign types in a specific country. The border between detection and recognition part is not obvious, because some detection parts are more practical in providing information for the next stage than others.

In the detection part, the possible location of the sign is first determined. The system searches over the images to find the possible locations of road signs. In the classification part the exact type of

the signs is identified through the evaluation of the regions found by detection stage. In some cases, these two tasks can be considered separately. However, in most cases, the recognition part is dependent on the detection part to supply information such as color and shape.

Traffic signs are different in color, shape and written content. The meaning of each sign is expressed by the combination of these features. Hence, these properties constitute the primary core of designing TSDRs. Generally, TSDR systems are studied in four different stages:

- **Image acquisition**: The basic stage in designing TSDR systems is to create a reliable data set. The data set can be captured by a mounted camera gathering the images of road signs during the driving.
- **Segmentation**: Segmentation is the initial stage in TSDR systems. In this stage, the possible regions of interest in the image are localized by applying different approaches. Since traffic signs contain comparatively constant colors (white, red yellow, blue), color information is the most appropriate method in this stage. The goal of this stage is to reduce the search area for the detection step by finding the possible locations of the signs. Segmentation mostly is based on color, though some authors do not utilize this step in their work and directly start detection.
- **Detection**: Traffic sign detection is the most complicated step in TSDR systems. In this stage the existence of the sign in the region of interest is explored. Use of color-based, shape-based or combination of both are widely used approaches.
- **Classification**: The last step in designing a TSDR system is to determine the exact type of the road signs. A fixed set of possible traffic signs is selected and different methods, from applying simple template matching to sophisticated machine learning approaches, are applied to classify a possible sign.

Due to the effect of a number of factors, the design of TSDR systems can be challenging. Some of these factors have been highlighted below:

- **Illumination changes**: Time of the day and weather conditions, such as rain, clouds and fog, have effects on the illumination of signs (see Figure 2.2). The poorer the illumination is, the harder it is to detect and identify signs.

Figure 2.2 *Examples of poorly illuminated and foggy weather [1]*

- **Shadow**: The shadows of other exterior objects also decreases the visibility of signs. Some parts of the sign may not receive enough light to be recognized.
- **Image blur and car vibration:** Images taken from a low-quality camera suffer motion blur. In addition, due to holes or muddy conditions of roads, cameras are not stable and vibration while driving makes the taken images blurry.
- **Defective signs:** Colors, patterns, text paint of the sign become blurry due to age, sunlight or rain or natural disasters, such as flood or earthquakes, or other damages (see Figure 2.3).

Figure 2. 3 *Damaged road sign [2]*

- **Sign occlusion:** Traffic signs can be occluded due to exterior objects such as trees, traffic lights, buildings, advertisement banners or other signs (Figure 2.4 shows an example of occluded signs).

  .

Figure 2.4 *Occluded road sign [3]*

- **Similar object:** Other objects which are similar to the road signs in shape and color, may confuse the recognition part. In addition, some signs have the same patterns or shape, while they are different in text. These issues can cause misrecognition.

Given these problems, TSDRs need to be robust to the problems mentioned above. During the design of TSDR applications, both driver conditions and exterior environmental situations are considered. The following sections explore detection and recognition separately in detail.

2.1 Detection methods

In this section, we go through different methods used for detection of traffic signs. In this stage, the main task is to make candidate regions which are probably traffic signs. Hence, the methods need to search over the images and extract regions of interest to make the images ready for the following stage (recognition). There is a myriad of methods used for this purpose. In general, these methods take advantage of the obvious characteristics of the signs, such as shape and colors. Researchers have utilized these two characteristics and there are many kinds of approaches based on these two characteristics.

Generally, the main detection methods can be divided into groups:

- Color-based detection
- Shape-based detection
- Hybrid methods detection (the combination of different methods)
- Neural network
- Other approaches

2.1.1 Color-based detection methods

Undoubtedly, color is one of the most important and popular characteristics for designing TSDR systems. Colors used in traffic signs are selected by different countries and mostly include main colors (red, blue, white, yellow, green) such that they can be easily noticeable by the drivers. These kinds of colors provide information about the features of regions of interest, search based on specific color is straightforward. Color-based detection are mostly used to segment various colors in images. In fact, they divide the images into subsets of pixels similar in colors and a label is assigned to each pixel so that the pixels with the same labels have similar features. Hence, distinguishing regions of interest can be done by focusing on different colors. Commonly, by thresholding the color space or other methods for segmentation such as neural networks, methods can find the regions containing specific colors. In thresholding method, each pixel with a value above a threshold is considered with appropriate label. In other words, possible candidate locations of traffic signs are separated from background by pixels value [13]. The simplicity is the most important advantage of this method. In fact, complicated computational operations are not needed to perform color-based methods and is easy and straightforward [14].

In thresholding method, the pixels are classified into two classes: traffic sign pixel and non-traffic sign pixel or background. A pixel is considered as a traffic sign pixel if its color is close to the reference color otherwise it is considered as a background pixel [15]. [16] is an example of using color thresholding and shape analysis to detect traffic sign.

Using the most proper color space is important due to the significant effect on detection stage. Different types of color spaces have been used in different methods, e.g., RGB, HSV, YUV, YCbCr, CIELab and are discussed in this part [17] [18-25].

2.1.1.1 RGB color space

RGB is the most commonly used color space. it is based on human color perception and used in computer vision and image processing applications. Each pixel is assigned the intensity value from 0 to 255. There are three color channels, so the value of intensity can be combined and shown as 16,777,216 colors in total.

The RGB color space is illustrated in Figure 2.5.

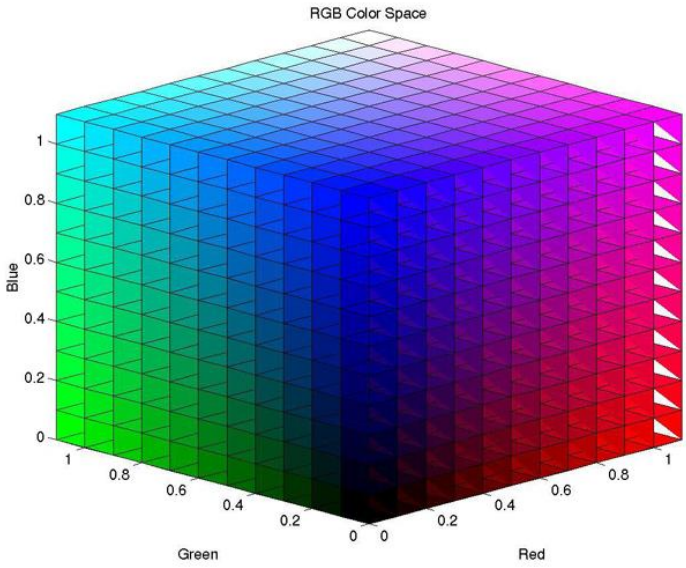


Figure 2.5 *RGB color space [4]*

RGB color space is a basic color model such that other color spaces are obtained by RGB transformation. What is important about RGB color space is the threshold associated with these channels. An example of RGB thresholding is shown in Figure 2.6.

Figure 2.6 *RGB segmentation. (a), (b): original image, (c)(d): segmented image [5]*

There are many applications that use RGB color space segmentations for detection. In [26], red color enhancement is used in RGB color space and the threshold was selected empirically. Authors in [27] examined segmentation with YUV and RGB color spaces to see which worked better. They selected RGB due to its less computation operation and fewer number of false positives. In [28] the authors proved that the conversion from RGB color space to HSV, is not linear and has high computational costs. M.Benallal and J. Meunier [29] explored the behavior of RGB elements in different traffic signs from sunrise to sunset. According to their result, there is a little difference between RGB components of the traffic signs colors. In [30] the average of both green and red components is calculated. Then a G-R histogram is created to show the subtraction the Green value from Red value. This histogram expresses the gap between two components and the pixels for each gap are counted [31]. However, some information about the color is missed due to the transformation from RGB image to a grayscale one. Zadeh [32] studied the nature of variation in pixels values for similar colors in the RGB space and created sub-spaces which enclose most of the variations of each color. They defined a sub-space as a conical region placed on fairly straight lines from RGB (0,0,0) to the combination of primary colors defining each color used in the traffic signs.

In [33] the authors used chromatic and achromatic filter to detect white signs. R, G and B are the brightness representation of red, blue and green channels in order. D is the degree of extraction of an achromatic color where, R, G, and B represent the brightness of the respective color, and D is the degree of extraction of an achromatic color.

$$f(R,G,B) = \frac{(|R-G| + |G-B| + |B-R|)}{3D} \tag{2.1}$$

When $f(R,G,B)$ is less than 1, it represents achromatic color, while if it is greater than 1, it is chromatic [33]. During the years, different researchers considered that the RGB color space was not a good space for color segmentation. However, after doing many different experiments, the overall consensus is that there are not significant benefits for using other color spaces, such as HSI (see following sections) over RBG. [34]

When objects have similar chromatics properties, standard RGB color space is not useful due to the effect of environmental conditions on the segmentation threshold. The authors in [35] attempted to enhance RGB color space. They used parameter normalization for RGB. The equations of normalization are:

$$r = \frac{R}{(R+G+B)} \tag{2.2}$$

$$g = \frac{G}{(R+G+B)} \tag{2.3}$$

$$b = \frac{B}{(R+G+B)} \tag{2.4}$$

Where $r + g + b = 1$.

Normalized RGB (NRGB) color space can decrease the effect of illumination variation. They applied Look up Tables to remove unwanted colors. In spite of the advantages of NRGB, the color saturation is still susceptible to the light.

In order to reduce the sensitivity to the brightness, some researchers explored alternative color spaces. They introduced HSV (Hue, Saturation and Value) or another version of it, HSI (Hue, Saturation, Intensity); in the following we review these and some research using these. To separate the luminance and color entirely, the color segmentation process is then carried out on images that were initially converted to HSV (Hue or color, Saturation and Value lightness) space; or HSI, (Hue, Saturation, Intensity), HSB (Hue, Saturation, Brightness) and HSL (Hue, Saturation, Lightness).

2.1.1.2 HSV color space

HSV color space (see Figure 2.7) is another color space that is used for color segmentation. HSV corresponds to intuitive color properties proposed by Sminth [36] and he showed that this color space has an important role in segmentation due to its independency to the light. It is represented by three variables:

Hue: Hue has an angle scale from 0 to 360 to shows colors. The red color is the beginning point and has a 0-60 angle. Green and blue have 120-180 and 240-300 angles respectively. Other colors, such as yellow has 60-120, cyan 180-240 and magenta 300-360 angles.

Saturation: Saturation describes how white a color is. It has range from 0.0 to 1. For example, a pure red has 1 saturation value and white color has 0 saturation value.

Value: Value is also called lightness and indicates how dark the color is. The level of brightness is between 0 and 1. A value of 0 is black and 1 is white.

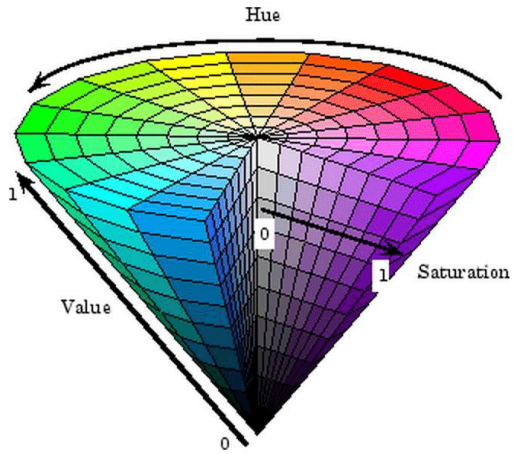


Figure 2.7 *HSV color space [6]*

Due to the ability of HSV to model human visual perception and its independency to the light variation when compared to RGB color space, many researchers have employed HSV for segmentation. In [37] the authors proposed an approach for sign detection and recognition. It is called fuzzy color-based method. First, images in RGB color space are taken by a camera mounted on a vehicle, then they are converted to HSV color space images. Hue and Saturation are extracted, and a fuzzy set of rules based on H and S are set. Images are segmented according to these rules.

Vitable [38] proposed another approach using a dynamic threshold to collect pixels in HSV color space. Using a dynamic threshold is useful since it decreases the dependency of Hue on illumination changes.

HSV transformation from RGB is another approach used in sign detection. RGB channels are transformed to HSV by using a set of equations. The advantage of this method is that it can segment signs that are badly illuminated.

The relationship between RGB and HSV (see Figure 2.8 and Figure 2.9) color spaces is expressed as follows:

$$\begin{cases} \dfrac{60 \times (G - B)}{S} & if\ V = R \\ 80 + \dfrac{(B - R)}{S} & if\ \ V = G \\ 240 + \dfrac{(R - G)}{S} & if\ V = B \end{cases} \tag{2.5}$$

Where:

$$V = \max(R, G, B) \tag{2.6}$$

$$S = \frac{\max - min}{max} \tag{2.7}$$

$$S = \frac{\max - min}{max} \tag{2.8}$$

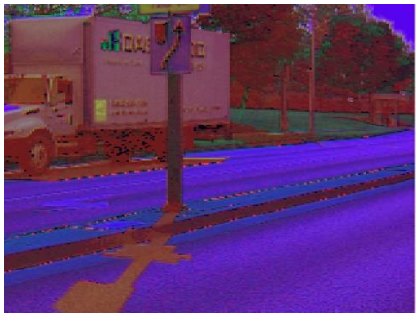


Figure 2.8 *Example of conversion of RGB to HSV*

Figure 2.9 *Example of extraction Hue, Saturation and V component*

2.1.1.3 HSI color space

Hue, Saturation, Intensity (HSI) is another color space. This color space is invariant to illumination variation [39]. As most images taken by camera are in RGB color space, they can be transformed to the HSI color space. H value represents colorless information and I represents the intensity value related to different illumination conditions. In addition, HSI color space is similar to human being colors perception [40] [41]. In HSI system color information is coded through breaking up intensity value from Hue and Saturation value to make HSI color space resistant to illumination changes. The equation to convert RGB to HIS is as follows:

$$I = \frac{R + G + B}{3} \tag{2.9}$$

$$S = 1 - \frac{\min(R, G, B)}{I} \tag{2.10}$$

$$H = across \begin{cases} \frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-G)(G-B)]^{\frac{1}{2}}} \end{cases} \quad (2.11)$$

This color space was applied in [42]. RGB images are converted to HSI, LUTs are applied for hue and saturation values for red, yellow and blue colors. In [43], the authors create a hue-saturation histogram for red road signs to obtain the value of the threshold. Then LUTs were used to improve the H and S values in HSI color space.

Authors in [44-47] attempt to find the white signs through augmenting thresholds in the HSI color space. But due to the property of white color that can be at any Hue, this color space is not practical to detect white signs. Hence, they used achromatic decomposition of an image. Their method is similar to the following equation [48]:

$$f(R,G,B) = \frac{(|R-G|+|G-B|+|B-R|)}{3D} \quad (2.12)$$

By setting D to 20, they got the best result.

Aoyaji [49] showed that due to the dependency of the Hue component on different factors such as distances, weather and the age of the sign, the HSI color space is not practical. He concluded that the segmentation step is not a sufficiently strong step to reliably detect the sign region. He concluded that based on his investigation that the HSI color space is not efficient.

Region Growing is another method proposed in [50]. This method uses a single pixel and then is enlarged to a group of pixels with similar color. This approach is performed in HSI color space. However, it has some restrictions due to the dependency on the initial pixel and the ending criteria. In the other words, when the first and last criteria are not satisfied, this approach may encounter a problem [51].

In [52], the authors HSI color space to find the region of interest and then employing a genetic algorithm to find the signs in ROI. The advantage of this method is that by applying a genetic algorithm, the algorithm works well without considering rotation, occlusion, other objects presence and unstable weather conditions. In addition, position, scale and rotation do not need to be taken into account.

2.1.1.4 YUV color space

This color space was defined by the National Television Systems Committee (NTSC) as a transmission standard. It is used for color television broadcasting. The Y component represents luminance or lightness and U and V represent for color (Chroma); the Y range is between 0 and 1, while (U V) ranges are from 0-255.

The following equation shows how to convert an RGB color space to YUV:

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{cases} Y = 0.299R + 0.587G + 0.114B \\ U = -0.147R - 0.289G + 0.436B \\ V = 0.615R - 0.515G - 0.100B \end{cases} \quad (2.13)$$

And to convert YUV to RGB**:**

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0.114 \\ 1 & -0.39 & -0.58 \\ 1 & -2.03 & 0 \end{pmatrix} \begin{pmatrix} Y \\ U \\ V \end{pmatrix} \quad (2.14)$$

There are a few researchers that have used the YUV color space. In [53] the authors examined the benefits and drawback of using HSV and YUV and the concluded that a combination of these two spaces can give better results.in fact they can compensate each other. Hue parameter form HSV color space is extracted in order to be combined with two parameters of (U, V) from YUV color space. AND operator is applied to get better result.

2.1.1.5 YCbCr color space

YCbCr or Y'CbCr (sometimes written as YCBCR or Y'CBCR), is a rarely used color space. It is applied to digital component video. Y is the illumination component. Cb and Cr are the comparison between blue and red component and offset components (see Figure 2.10)

The following equation shows how to convert an RGB color space to YCbCr:

$$\begin{cases} Y = 0.299R + 0.587G + 0.114B \\ Cb = 0.564\,(B - Y) \\ Cr = 0.713\,(R - Y) \end{cases} \quad (2.15)$$

and how to convert a YCbCr to RGB:

$$\begin{cases} R = Y + 1.402Cr \\ G = Y - 0.334Cb - 0.714VCr \\ B = Y + 1.772Cb \end{cases} \quad (2.16)$$

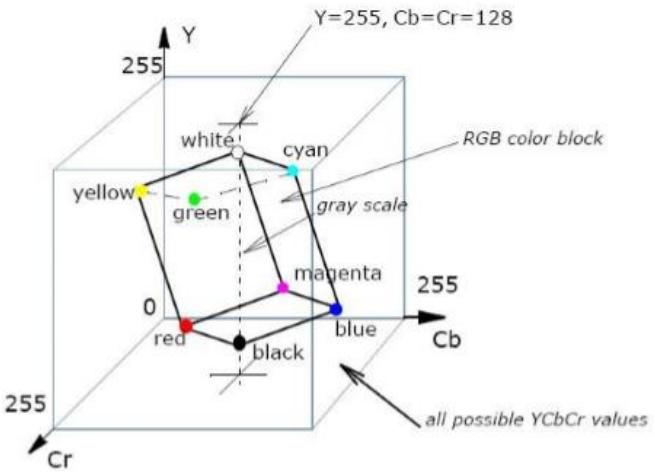


Figure 2.10 *YCbCR color space [7]*

In [54] the authors utilized this color space for road signs with different shapes. This color space is seldom used for color segmentation.

2.1.1.6 Other Color Spaces

We reviewed the use of the most popular color spaces in the segmentation of images with traffic signs. There are still other color spaces such as CIELAB is useful for localizing traffic signs when it is combined by Hue [55]. This method needs parameter adjustments during detection and is suitable for traffic sign detection under a simple background. When the environment is complex, the ability of algorithm is less to ensure high accuracy.

In [56] the authors applied CIECAM97 color space for segmentation to explore the color features of the sign without considering different conditions in angels and views.

As it was stated before, the HSI color space is practical in the intensity of illumination changes factor, however, it cannot help in changing the color temperature caused by weather conditions. Hence, some authors are not satisfied to apply this method. They used Luminosity, Chroma and Hue represented by (LCH) thresholding instead [57] [58]. This color space obtained from CIECAM97 model. This model can be used for considering the color temperature.

Color-indexing is another method in which colored objects are compared in two different images are compared based on their color histogram, without considering orientation and occlusion. The advantages of this approach are its speed, being straightforward and efficient; however, when the scenes are a bit complicated it can be time-consuming.

2.1.2 Shape-based detection

Shape is another characteristic that is widely used in traffic sign detection. Generally, traffic signs are circular, rectangular, octagonal and triangular. As it was mentioned in the previous section, one of the disadvantages of color-based methods for detection is its instability to illumination changes and weather conditions. These factors may cause unsatisfactory results. Hence, researchers prefer to use shape-based methods for road sign detection stage. There are many different approaches using shape as a feature to detect signs.

In this section, we review the most popular ones.

Among different shape-based methods, Hough Transform, Canny edge detection and template matching, are more popular than others [59]. Circular and rectangular shapes are detected by these methods. In spite of some false positive regions found through these methods, they are still useful for the detection stage.

2.1.2.1 Template matching

Template matching [60, 61] is one of the most common methods in image processing and pattern recognition. This is a machine vision technique to search images pixel by pixel to identify parts of an image that match a given pattern [62]. This approach is used in TSDR systems as well. Although template matching seems to be simple, it needs a large number of computational operations to enhance its robustness. In addition, if the number of templates increase, the pattern set becomes larger. Consequently, more computations are needed. The advantage of this approach is its high speed.

In order to avoid some of these issues, some authors proposed more alternative algorithms. In [63] the authors proposed a new algorithm based on distance. In spite of its advantages, it still needs a large number of templates.

2.1.2.2 Hough Transform

Hough Transform is a feature extraction technique used in shape-based methods. It can be employed for line, circle, and rectangle detection. The Hough transform method was first introduced in 1962 to find straight lines. The main benefits of this approach are its resistance to noise and rotation, however due to the highly complicated computations, it is not practical for real-time systems. In order to remove this drawback, authors in [64] used a Radial Symmetry algorithm based on the Hough transform, to find the symmetric edges, based on the symmetrical nature of circular or rectangular shapes, to find the most likely centers of the sign in an image [65]. Edge gradients in a circle are intersected in the center. A threshold is selected and fixed. The gradient value more than this threshold is stored, while points with lower magnitudes are ignored. Each remaining gradient element is a vote for a possible circle center and the vote is placed to the nearest pixel to this point.

The regions which include the most votes have a higher possibility to be the center of the circle. By using the Symmetry algorithm, the number of computational operations was reduced, but only circular signs can be identified. In [66] the authors used circular Hough transform to detect European speed limit signs in addition to the straight-line (rectangular shape Hough transform segmentation) Hough Transform for USA rectangular speed limit signs. Although the radial symmetry algorithm can be used in natural environment, it is not practical, since a great number of symmetrical objects exist in natural environments.

2.1.2.3 Edge detection

Edge detection is an image processing technique for finding the boundaries of objects within images. Edge detection is used for extraction of information and segmentation in computer vision and image processing. Canny edge detector is an edge detection operator to segment images and extract straight lines. In fact, it is a shape-based detection which can find a wide range of edges in an image by a multi-step algorithm. Canny edge detection uses straight line extraction and then segments them to find suitable slope and length which are useful for traffic signs. In [67] the authors used contour as a useful feature for shape-detection. Corner detection is another approach based on edge detection. The authors in [68] use corner detection for identifying triangles by searching for corners of 60 degrees to find out if there are any other corners to create triangles with equal degrees. The same approach is used to find rectangles by considering 90 degrees rather than 60. This method is rotation invariant.

In [69] Harris's corner detector was used to find triangle and rectangle-shape road signs. This method was firstly introduced in [70]. Simplicity and rotation invariant are the advantages of this approach.

2.1.2.4 Neural network (NN)

Trained Neural Network to find the color, shape or pictogram of road signs, is another approach using for shape-detection. In order to detect triangle signs, in [71] the authors used neural network by creating a back-Propagation network. In [72] the authors combined the input layer of neurons as edge detector and combine this color information.

2.1.2.5 Extraction through gradient features

Histogram of Oriented Gradients (HOG) features have attracted much attention during the past years. This approach has been successfully used in many contributions [73]. The difference between HOG and template matching is that HOG features do not need to know about the goal firstly. The first use of HOG features was for pedestrian detection [74] by Dalal and Triggs. They segmented the images into sets of blocks and computed the HOG for each block. Since HOG features have different parameters, the change over each of them may have effect on the accuracy of detection stage.

HOG features are also used in other contributions such as the detection of road signs [75,76]. Due to coarse spatial sampling, scale invariance and local contrast normalization, HOG features have been widely used in shape-based method for detection.

2.1.2.6 Other methods

In [77], template matching was enhanced by employing mathematical morphology to remove the noise and extract bones as mathematical morphology features. Since traffic sign have both internal and external borders, morphological features can be used to extract the inner shape of traffic signs. In [78], the authors used Hierarchical Spatial Feature Matching. This method uses geometrical features of shape and they search for these features in an image including traffic signs. A list of similar geometric shapes of road signs are found by the detection algorithm. This technique has been used for gray scale images only.

Similarity detection is another method used in shape-base detection approaches. A similarity factor between a segmented region and a set of samples of binary images for designating the shapes of the signs is computed assuming that the sampled and segmented image have same dimensions [79]. The researchers in [79] take advantage of color and shape for detection part. Similarity detection was applied in shape analysis.

A template hierarchy is used to take different object shapes by the Distance Transform Matching. Although this method can be practical due to accepting a small dissimilarity between templates and objects, it may need contour segmentation.

Maximally Stable External Regions is another approach applied to gain better results in different illumination levels. The MSER algorithm utilizes a stable threshold obtained from a normalized RGB [80, 81] color space.

2.1.3 Hybrid Methods

Both color-based and shape-based approaches have their own pros and cons. Color-based methods have been defined well in theory and searching a specific color in an image is not complicated and needs only a small amount of computation, in fact these kinds of methods are less problematic. However, they are highly dependent on illumination, weather changes, age, image noise and sign conditions. In addition, using thresholding for this approach is tedious [82].

Shape-based approaches are not susceptible to illumination changes, since the shape of the sign is always unique, but some parts of the signs may be occluded, or they are confused by other objects or even background. Some authors have tried to develop approaches using the combination of both shape and colors. The research in [83, 84, 85] are examples of using color-based detection method to create regions of interests and then following up by using a shape- based method. In some works, these two methods have been used simultaneously [86-91].

2.2 Classification Techniques

The next stage in TSDR systems is to classify the exact type of the sign according to their content and identify their category (speed limit, stop, etc.). There are many approaches that can be used based on the output of detection stage. However, we review the most commonly used ones, including machine learning, template matching and digit recognition for signs containing digits or texts.

2.2.1 Template and Similarity Methods

One of the most popular methods for similarity measures in traffic sign classification is normalized cross-correlation [81]. Gray scale regions of interest are considered, and the method looks for the similarity between a given template of road signs and gray scale based on normalized-cross correlation [92]. Resistance to illumination changes and simplicity are the benefits of this approach. However, it suffers from presence of non-unique pixels due to occlusion.

Template matching is another technique used for classification. To implement this approach, a huge number of samples are required. In fact, all the shapes of signs to be recognized are stored in a database. Then the obtained regions of interest (potential sign) are normalized. These signs are compared to every similar template in data set to find the category of the sign. Template matching methods is used in [93]. The first one uses 30 triangles and 10 circulars. They reached 85% accuracy. However, the latter has 60 circular and 47 triangular samples and obtained 98% accuracy. Yves et al. [94] introduced a worth noting template matching encoding that allows different templates to be combined according to the embedded color information. With this encoding, a template is constructed for an object, and a correlation computation can be defined, which serves as a measure for computing matches between the templates. The method is fast and can be easily modified to include new classes of signs. It has been implemented successfully in the classification phase of [95] and [96].

2.2.2 Machine Learning Methods

Machine learning techniques are often used in the recognition part of TSDR systems. In [97], authors employed a neural network and trained it by color and shape from a set of template signs.

In [98], the authors used a multi-layer perceptron on the Fast Fourier transform of the detected sign and a bank of filters. The concluded that neural network achieves better result than the template matching. In [99], a multi-layer neural network was used to classify the traffic signs where a feed forward neural network was employed as a classifier. In this case, conjugate gradient descent optimization was used to obtain better accuracy and achieved an accuracy for the classification stage of 91%. In [100] the authors combined the genetic algorithm and neural network. The employed genetic algorithm for detection stage and neural network for classification stage. In spite of so many advantages of using neural network, they still have some drawbacks

such as overfitting, the need of huge number of samples, and significant computational processing [101].

During the past years, use of support vector machines has been more popular. A small number of samples is needed to train this classifier and it often has better results than neural networks when it is applied in the classification stage. In [102] the authors used Invariant theory in the steps of detection and recognition traffic signs. As this method is robust to rotation and different scale. This method also can be used to extract feature vectors to SVM for classification of traffic signs. In [103] the authors proposed a new technique based on SVM and distance to borders. The patterns created by vectors describe the distances to borders of the candidates of traffic signs. The algorithm was tested on more than 300 images. The combination of distance to border and distance from Center was used in [104] as features to train SVM classifier. An accuracy of 89.98% was achieved by this algorithm for triangular signs. It is also invariant to translation, rotation and scale. Another method proposed based on SVM is [105] where they used a cascade SVM classifier that was trained by HOG features. The accuracy was 89.2% for white and 92.1% for color traffic signs.

2.2.3 Invariant Based Methods

Invariants theories have also been employed in TSDR systems. The examples of extracting features of signs to be applied in TSDR systems are: SIFT [106], invariant features [107,108], Haar Wavelet [109], Discrete Cosine Transform (DCT). These features also can also be used to train an SVM to classify the road signs. SIFT was used in [110] to find the invariant features in a given image. Then features of the training set are extracted to match the generated features by the algorithm. When the training image has the greatest number of matches, it is recognized as a sign. To enhance the accuracy, color information and orientation were added to the algorithm. They achieved a 99.3% accuracy over 149 samples. Speeded-Up Robust Features or SURF are also feature descriptors used in [111]. They used 200 images and achieved 92.7% accuracy. Optical Character Recognition [112] was also proposed for sign recognition, however due to not being able to recognize the traffic signs without signs, it is not practical to be employed for recognition stage.

Chapter 3
Related Work on Driver Behavior and Attention

Although there are several works on driver assistance systems based on driver's behavior, very little work has been done to consider driver behavior in the traffic sign detection and recognition field. In the following part, we review some driver assistance systems that have considered driver behavior.

In spite of the dramatic improvement in roads and vehicles, the total number of car accidents has increased. According to the Traffic Safety Administration (NHTSA), 56,000 sleep related road accidents happen annually [113]. There is no doubt that human error is at the heart of road fatalities.

In order to reduce the number of crashes, it is crucial to consider driver behavior while driving and alert them when they are in distracted states. Through this, crashes can be reduced 10-20 percent [114], hence, the role of a reliable driver distraction detection system to decrease the dangerous situation is critical.

Recently, excessive uses of in-vehicle information systems, such as navigation systems and mobile phones, induce visual and cognitive distraction. Distracted drivers do not pay enough attention to the roads, so they are not aware of the presence of obstacles or other objects, such as traffic signs. In addition, lane variation, slow response to road events such as wandering a pedestrian, turning the steering suddenly are other causes of driver distraction. Monitoring the driver's distraction is considered important in the design of a safe monitoring system.

To determine the drivers' distraction and inattention, it is crucial to extract information about the position of gaze and head [115]. Through the information about the position of the head, we can estimate the drivers' field of view and current focus attention. It is obvious that visual gaze direction and head position are connected intrinsically. We can take advantage of this feature when the eyes are not visible, the direction of eyes can be estimated by the head pose. In [116] the author used the combination of gaze direction and head pose to get the gaze information.

In [117] the authors proposed an approach to estimate the drivers' observation within the vehicle. They used the integration of eye-gaze tracking and road scene events to validate the driver behavior. Other factors, such as speed, fatigue, drunk driving also cause inattention. In order to determine driver's inattention truly, we need an accurate real time and driver-gaze monitoring aggregated with road scene states, what has been seen or not seen by the drivers, must be identified.

In [118] the author considered the direction of the driver's gaze on to the road scene; however, the features of the road were not determined. Jabon et al. [119] aggregated a large number of facial features in pre-accident intervals as key features to predict accidents. His approach was considered as one of the most important studies of driver-safety systems used for preventing the accidents. A similar approach was developed in EPFL university [120] where they analyzed the muscle movement and facial states to detect driver distraction. Through this analysis, they could identify whether the driver was distracted when facing hazardous situations.
In [121] a forward warning system was proposed by using information about the driver behavior to recognize a driver's distraction. In addition, they determined whether the driver was looking straight ahead or not.

In [122] the authors found that the position of the head and direction of eyes are useful factors to indicate fatigue. The correlation between eye-gaze and curvature of the road was investigated by Land and Lee [123]. In [124] the researchers used gaze and lane tracking to find their relation. They also figured out that the drivers frequently watch the oncoming traffic. Work reported in [125] found some relation between head-gaze direction and driving tasks. By considering where the driver is looking, the safety of the system can be improved by removing dispensable warnings, so that if there is no road event, such as pedestrian walking or the presence of traffic signs, no action is needed. The driver behavior before and after the road event, is considered an important factor in deciding if a warning is necessary.

As one way to monitor the drivers' behavior is the combination of eye-gaze tracking and instrumented vehicles, much research has been performed on gaze direction detection with different sensors and for a variety of applications. Research in [126] predetermined an architecture to predict path planning which uses a virtual driver model. The behavior and action of drivers are considered as inputs for driver assistance system, hence, a safe action is generated by considering the performance of the system and compatibility with the driver. Authors in [127] employed force feedback by using a steering wheel as a natural interface between human and machine to keep in a lane. Gentle resistance coming from steering wheel was used to persuade the driver to keep in the lane. Although the resistance was not strong, the driver could cancel the force to move faster. A similar approach was added to a driver assistance system [128] to control the speed by accelerator. When the speed limit is ignored due to the high speed, the accelerator pedal cannot be depressed easily. This example describes the feedback of the system to the driver in unobtrusive situations.

In [129] the authors used the combination of driver head pose as an analysis of driver behavior and road observation to driver assistance systems. In [130] the authors used the combination of the gaze and head position as an important interface between human and machines. The position of the head and gaze was accurately achieved.

Land and Lee [123] consider driving as a tracking task. This means that the driver interprets the road by what he/she sees and then decides what kinds of control signals should be applied to correctly keep the vehicle in the proper position. Summala and Nicminen [131] stated that in order to keep the vehicle in a lane, peripheral vision is enough. By using peripheral vision, drivers can verify the small motions on the road to demonstrate that the vehicle is on track. When the driver

misses the right direction, he/she receives an alert. The benefit of using the correlation of the driver's gaze and road scene is that the driver does not always need to concentrate on the road to drive safely. However, when a driver departs the lane, they will receive a visual cue to pay attention to the position of the road. If the driver is not looking at the road at the time, it reflects the driver's inattention. Land and Lee [123] showed that the location of the road scene where the driver would look to have attentive lane tracking is the correlation between the eye gaze and road curvature. Land and Lee did research on where a driver looks while entering a curve [123], finding a driver looks at the tangent of the road ahead. Maltz and Shinar [132] found that relying on peripheral vision was insufficient due to its inefficiency in detecting dangerous road situations.

In [133] the system investigated whether speed limit signs had been missed by the drivers or not. A road detection system based on vision-based eye was proposed in [134].

In [135] a solution which applies heat-maps and optical flow methods for a highly precise attention estimation was presented. The vigilance of a driver has been widely studied, as [136] shows how driver drowsiness could be measured. In [137] an IR-camera was used to measure eye closing duration, as well as eye blinking and nodding frequencies, or in [138] using time-of-flight cameras.

In [139] the authors designed a system for detecting a driver's drowsiness and distraction. These two factors cause a large number of accidents. In fact, drivers are less capable of making proper decisions and controlling the vehicle when they face hazardous situations. In order to decrease the accidents due to these factors, one way is to supervise the drivers during the driving. When they are in a drowsy and distracted state, the system can alert them. Another factor that has impact to reduce the crashes is being able to predict unsafe driving behavior in advance. Regarding to these issues, the authors discuss different methods to monitor the driver's behavior to predict unsafe behavior. To indicate drowsiness, the authors used methods to extract visual (eye closure, eye blinking, yawning, head pose, facial expression) and non-visual features (heart rate, pulse rate and brain activity), and driving performance behavior related to the vehicle-based features, psychological signals are employed to other methods to find the non-visual features, for vehicle-based features, they considered and described steering wheel movement. For distraction, the position of driver's head and the direction of drivers' gaze are taken into account. Finally, they proposed prediction methods based on facial expression to predict unsafe driving behavior.

[140], the authors conducted another study over traffic sign detection and recognition. They considered eye movement as a feature for detection and classification of the traffic signs. They considered different effects of conditions, such as age and lighting. They concluded that the more the luminance at night increases, the more response time becomes.

Traffic sign detection and recognition according to the gaze of the driver was firstly proposed in [141]. After designing a system to detect the research of interest and classify the types of the sign, we applied the point of gaze of the drivers obtained from the calculation in their laboratory to identify if the driver has paid attention to the signs or not.

Chapter 4
Data Sets

The data from RoadLab have been used in our research; we discuss this in Section 4.1. Since we must detect, recognize and classify traffic signs, we also need data sets containing images of traffic signs. We discuss the traffic sign data sets and our data set in Section 4.2.

4.1 Road Lab Data Set

RoadLab provides data for the improvement of Intelligent Driver Assistant systems. The goal is to provide information such as driver's behavior and environmental conditions to develop systems to help prevent from accidents.

RoadLab data is collected by using an in-vehicle laboratory instrumented with an on-board diagnostic system which uses the CANbus protocol [142] and provides vehicle parameters such as brake pedal pressure, steering wheel angle, etc. Video sequences of the driving environment in front of the vehicle and other optical behavior such as driver gaze as well as GPS data are collected.

Our image sequences come from a study of 16 individual drivers who were between 20 and 47 years old in the city of London, ON, Canada using RoadLab. Each driver used the RoadLab vehicle to drive over the same route. Two other observers were also present in the vehicle to both supervise the performance of the equipment and guide the driver to correctly navigate the route. More information about the environmental conditions and participants have been provided in the table (4.1).

Figure 4. 1 *The RoadLab in-vehicle laboratory: a) (left): on-board computer and LCD screen b) (center): dual stereo front visual sensors, c) (right): side stereo visual sensors [4].*

Participant	*Date*	*Time*	*Weather Conditions*	*Age*	*Gender*
1	2012-08-24	13:15	29C, Sunny	37	M
2	2012-08-24	15:30	31C, Sunny	37	M
3	2012-08-30	12:15	23C, Sunny	41	F
4	2012-08-31	11:00	24C, Sunny	41	M
5	2012-09-05	12:05	27C, Partially Cloudy	37	F
6	2012-09-10	13:00	21C, Partially Cloudy	22	F
7	2012-09-12	11:30	21C, Sunny	31	F
8	2012-09-12	14:45	27C, Sunny	21	M
9	2012-09-17	13:00	24C, Partially Cloudy	21	F
10	2012-09-19	09:30	8C, Sunny	20	M
11	2012-09-19	14:45	12C, Sunny	22	F
12	2012-09-21	11:45	18C, Partially Sunny	24	F
13	2012-09-21	14:45	19C, Partially Sunny	23	M
14	2012-09-24	11:00	7C, Sunny	47	F
15	2012-09-24	14:00	13C, Partially Sunny	44	F
16	2012-09-28	10:00	14C, Partially Sunny	25	M

Table 4. 1 *Information about Participants and Driving Conditions*

As shown in the above table, sequences of images have been recorded under different environmental conditions such as light and shadow or cloudy and sunny weather during the driving. The RoadLab system encompasses different instruments, including stereo cameras, LCD screens and GPS units. It also has cameras that are used for eye tracking to record the gaze of the driver (see Figure 4.1).

The data coming from RoadLab was recorded in real-time and under real environmental conditions. Each drive took approximately 60 minutes for each driver; the times differ based on the drivers or other road events, such as traffic. In addition, the number of frames can be seen per second is 30.

Our dataset images are recorded with a resolution of 320 by 240 through a front stereo rig mounted on the front roof of the vehicle. For 60 minutes of driving, more than 100,000 frames have been recorded approximately. Figure 4.2 illustrates some samples of images from our dataset.

Figure 4. 2 *Examples of Images from the RoadLab Dataset*

4.2 Traffic Sign Data

In order to train and test a detector to find traffic signs, we need access to a large number of samples and preferably a data set with images under different environmental situations, such as weather conditions, different illumination levels and occlusion and notations. A number of research groups have made data sets available for the community. The most popular open source dataset is the German Traffic Sign Data set (GTSD) (see Table 4.2), divided into two parts, for detection and recognition.

Among all the datasets introduced, BTS and GTS have a huge number of samples for both classification and detection; the datasets include annotated images. In addition, and BTSD contains video tracks; the Steriopolis and Swedish Traffic sign Detection data sets also encompass

sequences of images, and hence can be used for tracking. Every five frames are annotated in STSD. Details on these data sets are summarized in the following table:

paper	Data set	Purpose	classes	Total image	country
[148]	GTSRB	Classification	43	50000	German
[148]	GTSDB	Detection		900	German
[149]	BTSCB	Classification	62	7000	Belgium
[149]	BTSDB	Detection		10000	Belgium
[150]	STSD	Detection, Classification		20000	Sweden
[150]	FTSD	Detection, Classification		3000	Sweden
[151]	Stereopolis	Detection, Classification	10	847	France

Table 4. 2 *Examples of Common Data sets*

European traffic data sets continue to grow. In addition, due the number of samples in these data sets for both detection and recognition, a number of TSDR applications have used this data set during the years. However, traffic signs are different between Europe and North America. There is no open source data set for North America, certainly Canada. Our system is based on Canada traffic signs, so a new data set for our system was needed. There are many differences based on both shape and colors between European traffic signs and north American ones that we could not use the European data set for our system. Hence, we constructed our own data set.

For the initial step of detection stage, we used the recorded real sequences coming from the RoadLab. For the additional stage of detection stage, we need to train our binary linear Support Vector Machine, we also used RoadLab data, cropped the traffic signs in the RoadLab sequences and rotated them to have more samples. In addition, we extracted signs from the background with different size of the sign under various weather conditions and different sizes of annotation to construct a data set.

We selected 1000 sequences randomly to be cropped so that they could be employed for training. As the vehicle is moving during the driving, and the camera mounted on the top is recording the images nonstop, we have different sizes of the signs with different margins. These cropped images were categorized into 30 different classes to be supplied for multi-class classifier.

The samples of the extracted signs form background are illustrated in the Figure 4.3

Figure 4.3 *Examples of extracted traffic signs*

As it will be explained in the following parts, we need two different data sets, one for detection and one for classification. For the detection part, we need to train a linear binary classifier, hence we used a positive data set containing 950 samples and 2500 samples as a negative data set.

For the classification stage, we have 30 different classes, as our classifier is a multi-class Support Vector Machine. We use 50 samples for each class as our positive data set and use 500 negative samples as background. The total number of positive samples for the classification stage is 1500 and the total number of negative samples is 500.

We used the 80 percentage of the positive and negative samples for training stage and 20 percentage of the data set for test part.

Chapter 5
Traffic Sign Detection and Recognition Method

In this chapter, we provide details about our methods and introduce other related concepts behind our techniques. For the detection stage we describe how we applied the color information, Maximally Stable Extremal region methods and also the combination of a linear binary Support Vector Machine and Histogram of Oriented Gradients feature vectors were used for detection and recognition. For the classification we describe our HOG features descriptor and the multi class SVM classifier. Our methods are highlighted below:

1. Creating the attentional area of the driver;
2. Detection stage using color information and Maximally Stable Extremal Regions in addition to the binary Support Vector Machine and Histogram of Oriented Gradients feature descriptor;
3. Classification stage using Histogram of Oriented Gradients features and a multi-class Support Vector Machine classifier;
4. Assessment of driver attention.

5.1 Creating Attentional Area of the Driver

Most of the detection and recognition systems have operated on a chain of image sequences. However, in [141] the attentional area of the driver was identified as important factor. In this thesis we describe how to determine the attentional field of view of the driver and detect the traffic signs inside of this area to identify the number of the signs a driver has seen or missed and also analyze the drivers' behavior during the driving.

The 3D point of gaze (PoG) of the driver is obtained by relating the 3D line-of gaze (LoG) of the driver to the depth map derived of the front camera system. We took advantage of the method that was introduced and implemented in our laboratory in [144]. The coordinates of the 3D PoG on each frame of reference is derived by the method proposed in [144].

Figure 5. 1 *Remote Eye Tracking [144]*

Figure 5.1 shows the eye tracking system. In order to obtain the region of interests of the driver (see [144]), we consider a plane vertically placed at the 3D point of gaze along the 3D line of gaze. A cone in 3D space represents the region of interest; see Figure 5.2. However, when this cone is mapped to the imaging plane of stereoscopic image, it becomes a 2D ellipse. In addition, the radius of the cone of attentional area of the driver is 13 [144]. By eye position and 3D point of gaze obtained through the eye tracker system, the radius of 3D gaze is computed by following formula:

$$r = \tan(\theta) d(e,g) \tag{5.1}$$

Where e= (e_x, e_y, e_z) represents the coordinates of eye position and g= (g_x, g_y, g_z) is the coordinates of points of gaze.

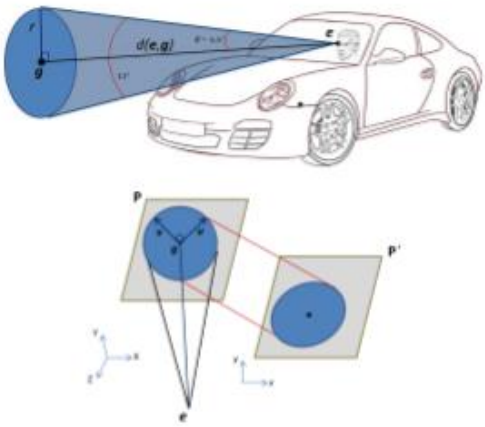


Figure 5. 2 *Attentional Gaze Cone of the Driver*

As it is proposed in [144], the gaze area has the range of 6.5-degree extension for each pitch and yaw of direction. So, we assume $\theta = 6.5\ degree$.

The Euclidian distance between the eye positions and the 3D points of gaze is obtained by equation 5.2.

$$d(e,g) = \sqrt{(e_x - g_x)^2 + (e_y - g_y)^2 + (e_t - g_t)^2} \tag{5.2}$$

Also, a 3D circle can be parametrically defined as in equation 5.3.

$$S(\varphi) = (X(\varphi), Y(\varphi), Z(\varphi))^T = g + r(\cos(\varphi)\, u + r(\sin(\varphi)\, v) \quad (5.3)$$

Where $u = (u_x, u_y, u_z)$ and $v = (v_x, v_y, v_z)$ are the coordinates of two perpendicular vectors in the plane and φ is the angles with different values from 0 to 2π.

Both eye tracker and forward stereoscopic systems have their own aspect. What we need is the transformation of the 3D circles which is equal to the attentional area of the driver to the aspect of the stereo system of the driver. The computation of parameters related to the transformation was performed earlier in our laboratory by a cross-validation process. This process was applied between the eye tracker system and stereoscopic system [145] (see equation 5.4).

$$S'(\varphi) = R^T(S(\varphi) - T) \quad (5.4)$$

Where R and T are the rotation and translation matrices respectively.

After developing the 3D circle, we need to place it on the stereo imaging plane; equation 5.5 is used for computation of this stage:

$$s'(\varphi) = \frac{1}{Z'} K S'(\varphi) \quad (5.5)$$

where K is the intrinsic calibration matrix of the scene stereo system. Finally, we obtain the attentional area of the driver which is our region of interest.

5.2 Detection Stage

Maximally Stable Extremal Region (MSER) is a method of blob detection in computer vision detecting regions in images based on brightness. In fact, this algorithm extracts from an image a number of co-variant regions called MSERs: an MSER is a stable connected component of some gray-levels of the image. In other words, MSER is according to the idea of taking regions which are left the same through a wide range of thresholds. Several different regions can be detected by this approach. In addition, MSER can be applied to all kinds of images, regardless of the texture of the image.

The MSER algorithm can be performed by sorting all the gray value pixels and then eventually adding pixels to each connected component since the threshold is changed. The regions having the

minimal variations with respect to the threshold are defined as maximally stable. In other words, assume Q is a region with threshold t, and Q' is a region with threshold $t + 1$ such that Q is not significantly larger than Q', Q is a **maximally stable region**.

As an example, we consider an arbitrary threshold (the threshold must be between 0 to 255), then all the pixels bellow a given threshold are white and the others are assumed black, we try a sequence of threshold with changing value. As the value is increased the white blobs related to white pixels are appeared and grow larger at the same time with the changes of values of thresholds. After trying a large number of thresholds, the local regions are stable and show some invariance of affine transformation of image intensities and scaling.

All the pixels in MSER have either higher (bright extremal regions) or lower (dark extremal regions) intensity than all pixels on its outer boundary. This operation is performed by storing the pixels by pixels to each connected component as the threshold is changed.

MSER processing:

The MSER extraction implements the following steps:

1. Convert an image to gray scale level.
2. Trying different threshold of intensity to change the luminance intensity of pixels from black to white or conversely.
3. Extract connected components (or extremal regions).
4. Find the threshold when an extremal region is ‘maximally stable’.
5. Keep those regions descriptor as features.
6. Back to step 2 to find more regions.

The goal of the detection stage is to find the regions that are most likely to include traffic signs. We utilized the MSER algorithm but with some changes.

Since we are interested in traffic signs, we are interested in segments of the images as candidate regions which include the colors of the sign (such as green, yellow, red, orange, etc.). Our main purpose is to extract these regions of color and then create images with different gray scales for analysis by the MSER algorithm. In addition, a preprocessing operation is employed to the images to enhance the color of the images containing the signs. Details on the preprocessing steps are covered in Section 5.2.1.

Other work [146] used similar approach for region of interest extraction for sign detection. The approach extracted the regions containing signs with red and blue colors. Other colors, such as green, yellow, orange, etc., were not considered. We use a region detector to find candidates regions using several different colors (not only blue and red); this is the main difference between our approach and other mentioned technique.

There are two types of MSER regions. Dark Connected Components on a brighter background (MSER +) and a bright one surrounded by a darker background (MSER-). we used the latter one since extracting specific color makes us to choose this kind of MSER.

In order to find and extract regions containing specific color pixels, colored pixels are converted in a way corresponding to one of the main color channels (red, blue, green). By extracting these pixels, the image sequence is transferred to gray scale and is ready to be employed by MSER region detector. Hence, by extracting these kinds of pixels, we can convert the image into bright and dark pixels. Bright pixels are considered as MSER regions. Then we take advantage of this strategy to extract regions of interest. Again, more details are presented in Section 5.2.1.

Our overall detection stage consists of the 4 main stages:

- Preprocessing.
- MSER region detection.
- Feature extraction using a HOG feature extractor.
- Training an SVM classifier and using hard negative mining.

5.2.1 Implementation Details

In this part, we explain the methods we used for detection part. The implementation consists of four steps:

1. Preprocessing stages to extract desired pixels of colors
2. Extract MSER regions and connected components which are extremal regions.
3. Using Region Properties which measure and analyze the properties of image region according to some of features related to these regions. Bounding boxes examples of property measurement used to show the location of the image. (we used connected components and bounding boxes)
4. Finding bounding boxes for each region and then merge them to get the correct one.

5.2.1.1 Preprocessing Stages for Different Colors:

In this step, different pixels of colors are extracted in order to identify the MSER regions. Three different functions based on three primary colors are used to acquire the MSER regions and connected components. In each function, the desired MSER regions are found for each specific color. They all use the median filter and contrast normalization as two different preprocessing technique to improve the result of later processing. **Median filter** is a non-linear digital filtering technique which is used to remove noise from an image or signal. Due to preserving the edges while removing noise, this technique is widely used. This approach runs through an image entry by entry pixel and then replaces each entry with the median of neighboring entries. As we know the **median** is the value separating the higher half from the lower half of a data. The pattern is called “window” sliding entry by entry. If the number of entries is odd, then the median is the middle number, and for an even number of entries, the median is defined as the mean of the two middle numbers. The general equation is 5.6.

$$median(a) = \frac{a\lfloor(\#x+1)\div 2\rfloor\ a\lceil\#(x+1)\div 2\rceil}{2} \quad (5.6)$$

he median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the *mean* of neighboring pixel values, it replaces it with the *median* of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is use.

Normalization is used to change the range of pixel intensity values to make a better quality of image. It is also used to increase the contrast of the image. Normalization transforms a n-dimensional grayscale image I: $\{X \subseteq R^n\} \rightarrow \{\text{Min, Max}\}$ into a new image $I_N : \{X \subseteq R^n\} \rightarrow \{\text{newMin, …, newMax}\}$. Gray scale normalization is computing according to equation 5.7.

$$Normalization = (I - Min)\ \frac{newMax - newMin}{Max - Min} + \text{newMin} \quad (5.7)$$

After removing noise through normalization and median filter, our images are ready for other processes.

The goal of this stage is to find the regions possibly containing the traffic signs. These road signs include all the North American traffic signs. Sequenced images are separated into the R, G and B different channels. Since road signs are affected by the illumination changes, we need to enhance the images and remove the likely noise; a median filter is applied to individual channel. The contrast normalization is employed on an individual channel and use adjusted pixel intensity to regulate the intensity value for each pixel and blur the back ground. The images are ready to extract pixels of interest.

Processing Red Pixels: In order to extract red pixels of each image for detection stage, we use formula 5.8 [146]. According to this equation, red pixels of an image is obtained by subtracting the green and blue channels.

Red pixels = max (0, Red - Green-Blue) (5.8)

We can extract red pixels from the image and convert them to gray scale; the extracted red pixels become white in gray scale and the other pixels are black. Figure 5.3 illustrates an example of extracting red pixels.

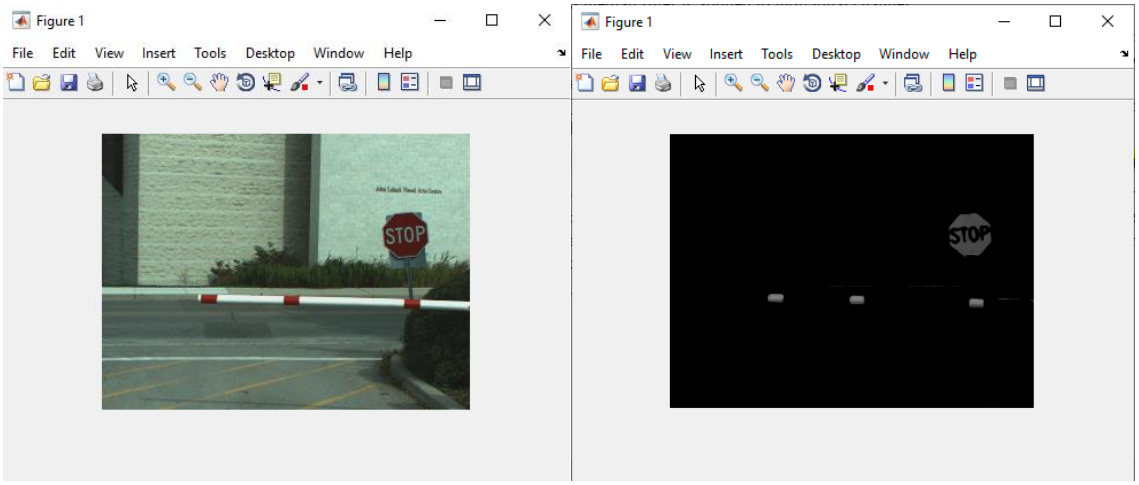
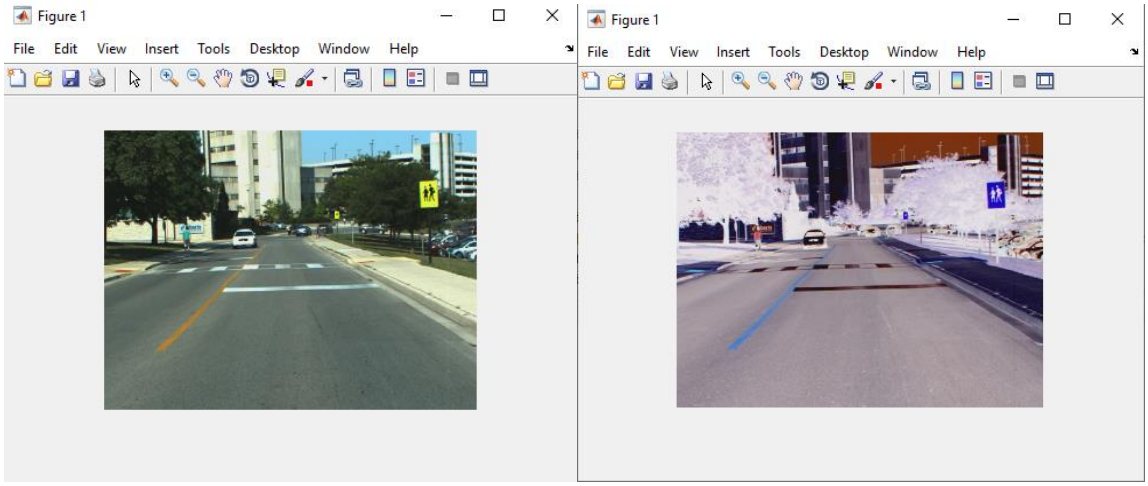


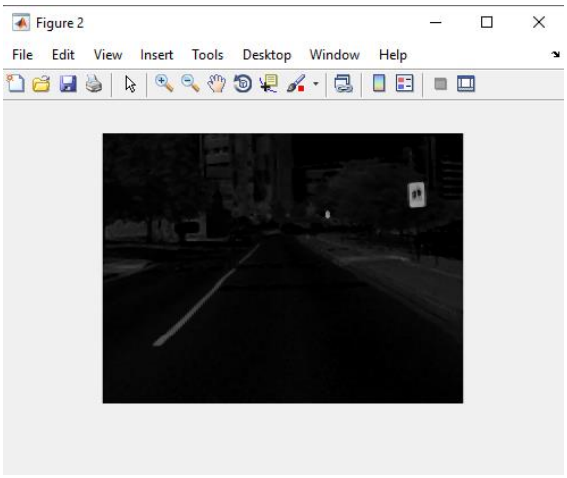
Figure 5. 3 *Extracting red pixels. Left original image, right the extracted red pixels image.*

The MSER algorithm is used to process these images and the algorithm then identifies white pixels as regions of interest.

Processing Yellow Pixels: In addition to de-noising operations over the images mentioned in the previous section, we do additional preprocessing to extract yellow pixels. In order to extract yellow pixels of an image, we first invert our image. Through this way the yellow pixels become blue in the inverted image, and then we subtract the blue channel of the image from grayscale image. Now the yellow pixels are defined. Figure 5.4a shows an image with yellow signs and Figure 5.4b shows the inverted image; Figure 5.4c shows the gray scale image.



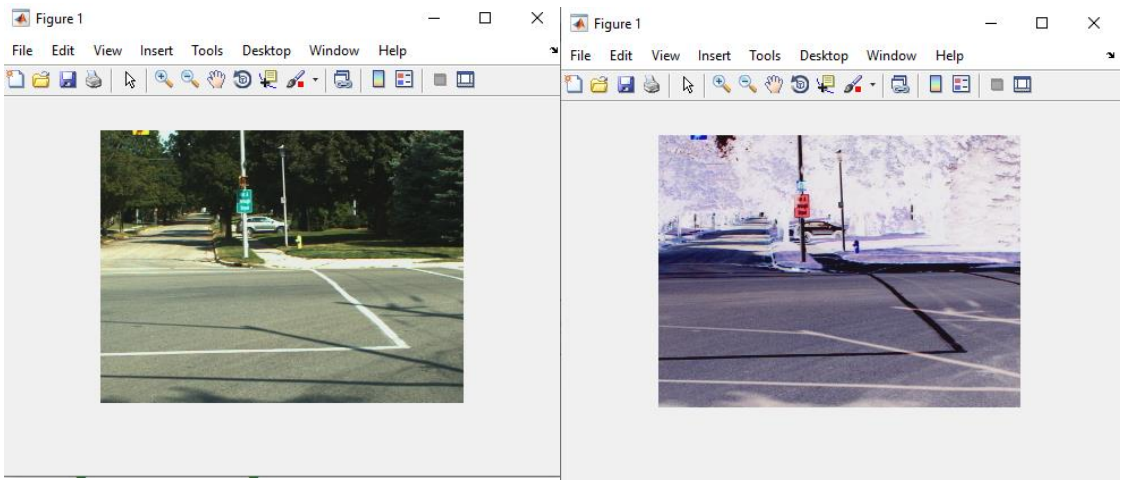
(a) *(b*



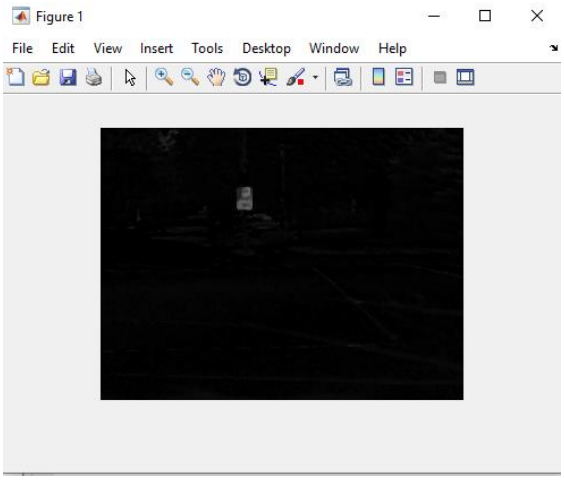
(c)

Figure 5. 4 *Finding yellow pixels, (a) the original image. (b) inverted the original image. (c) the gray- scale image.*

Processing Green Pixels: We utilize the same strategy for green pixels. After denoising our sequenced images by median filter and contrast normalization, one more preprocessing step is added. Green color pixel corresponds to red pixels in inverted images. We repeat the previous step and extract the red pixels from the gray scale of original image. Figure 5.5 illustrates the extraction of the gray scale image from an original image using green pixels. We now have a gray scale image for processing via the MSER algorithm to identify regions of interest.



(a) *(b)*



(c)

Figure 5. 5 *Finding the green pixels. (a) the original image, (b) the inverted image, (c) gray scale image.*

Processing White Pixels: For processing white pixels, once the denoising stages are completed, white pixels are extracted. Extracting white pixels (traffic signs with white back ground) was challenging since a huge number of objects with white pixels are considered as signs. Hence the number of false positives was more than for the other colors. However, by using the constraints presented in the following subsections, we can remove some of the false positive regions found by the algorithm as well as determining the size of bounding boxes.

5.2.1.2 Extracting MSER regions and Connected Components

After preprocessing operations, our images are ready for the next stage. We applied MSER as a region detector to search and find regions and connected components containing particular colored pixels. Regions with high levels of contrast in a grayscale scene are detected by MSER. As described above, by extracting specific color pixels, we can make a gray scale image and then use the MSER region detector. The regions with extracted color pixels have the high level of contrast in a gray scale image. Regardless of the shape of the regions, connected components remained unchangeable after trying different thresholds.

5.2.1.3 Using region props to measure the region properties

After finding the regions, we measure properties of the regions found in the images; these are the connected components and bounding boxes. We stored all constructed bounding boxes for each found region.

5.2.1.4 Implementation Algorithm

Firstly, the MSER regions are detected. In order to detect MSER regions, we used MSERFeatureDetector from the Matlab toolbox. This function has two important inputs: a threshold delta which is used to compute the intensity of the threshold levels. This input range of this parameter is between 0.8 to 4. We empirically chose a threshold with value 3. We chose this threshold as it can be used for the color Orange as well as Red and so the Red function can cover red and orange color pixels. For the other types of colors, such as green and yellow we tried different thresholds and finally chose the same as the Red threshold. This seemed to generate the best results for all colors of interest. Through this function we can find the MSER regions and MSER connected components.

We repeat this step for all the colors, and the connected components derived from this step for each color are stored in a matrix. The matrices are merged together in order to be evaluated in region properties measurement stage. After evaluation, the measured regions are combined, and a bounding box is placed over each region.

In order to determine whether the bounding boxes encompassing the border of the sign, we expanded the values of the edge coordinates of the bounding box. We extract all four coordinates of bounding boxes and expand them according an empirically predetermined small value which is 0.02. This value is used by the application to expand neighboring bounding boxes [147] and is employed for all the found bounding boxes. Expanding neighboring bounding boxes is a crucial step that is done in preparation of the final merge of individual bounding boxes. This expansion is necessary to detect a wide and varied areas of signs. The size of the primary sign region in proportion to the rest of the subjects in the image could also contribute to this [147]. We chose the expansion amount empirically.

Some false positives are eliminated by clipping the bounding boxes. Some images contain multiples bounding boxes overlaying on each other. We need them to be merged together to form a single bounding box around an individual region. This issue was addressed by setting an overlap ratio between all the bounding box pairs. This quantifies the distance between all pairs of sign regions so that it is possible to find the groups of neighboring sign regions by looking for non-zero overlap ratios. An overlap ratio is a value between 0 and 1 used between the pairs of bounding boxes. The best value is 1 indicating the perfect overlap. The bounding boxes around each sign is prepared to be sent to the recognition part. We used "bboxOverlapRatio" function of MATLAB toolbox to compute the pair-wise overlap ratios for all the expanded bounding boxes. The output of this function is a matrix by M rows and N columns. Each (I, J) element in this matrix corresponds to the overlap ratio between row I in the first bounding box and row J in the other one. The overlap ratio is computed according to equation 5.9:

$$overlap\ ratio = \frac{Area\ (A \cap B)}{Area\ (A \cup B)} \text{ where A and B are the bounding boxes.} \quad (5.9)$$

The bounding boxes are merged according to this overlap ratio which is based on the size of the bounding boxes. Even after this merger, there may still be more overlapping bounding boxes. In

order to address this issue, we consider one more constraint. The ratio of the height to the width or the width to the height of the bounding boxes is also considered. If the ratio of the height to the width is less 1.3 or if the ratio of the width to the height is less than 1.2, then we merge the bounding boxes to determine the bounding box over a sign.

Although the generated results through the combination of color information and MSER algorithm is satisfactory enough for our work, there are some regions wrongly considered as a sign. In order to remove this issue, we added one more step to the detection part.

Given the success of using a Histogram of Oriented Gradients feature detector [74] (see also Section 5.3) and Support vector Machine classifier, we used the combination in one more step to improve the accuracy. We constructed a set of positive and negative data (images). The positive data set includes the North America Traffic Sign samples. In order to have a satisfying number of positive samples, we added flipped, rotated and translated versions of original samples which results in better detection performance. We used 950 positive samples and 2500 negative samples.

We used 80 percentage of our data set for training and 20 parentage of the data set for testing. In addition, in order to train our classifier efficiently, we used a method called hard negative mining.

Each image without a traffic sign is considered as a negative sample. We train the SVM in iterative process. When the SVM predicts a negative sample as a positive sample (false positive), we add this sample to the SVM for next iteration as a negative sample. We repeat this process for 10 iterations to make our classifier more powerful and accurate. The steps of the algorithm are outlined in Figure 5.6.

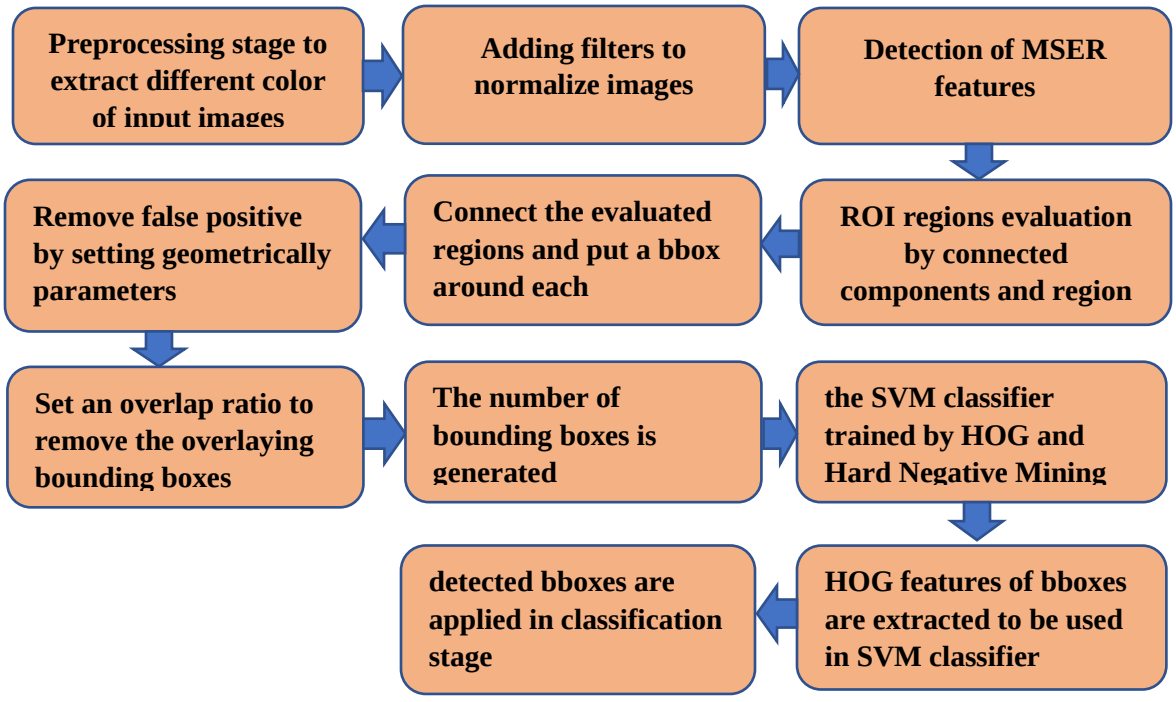


Figure 5. 6 *Implementation of the Detection Algorithm*

The images in Figure 5.4 illustrate the process of the detection stage. Width and height are coming from the bounding boxes have been produced from the region properties.

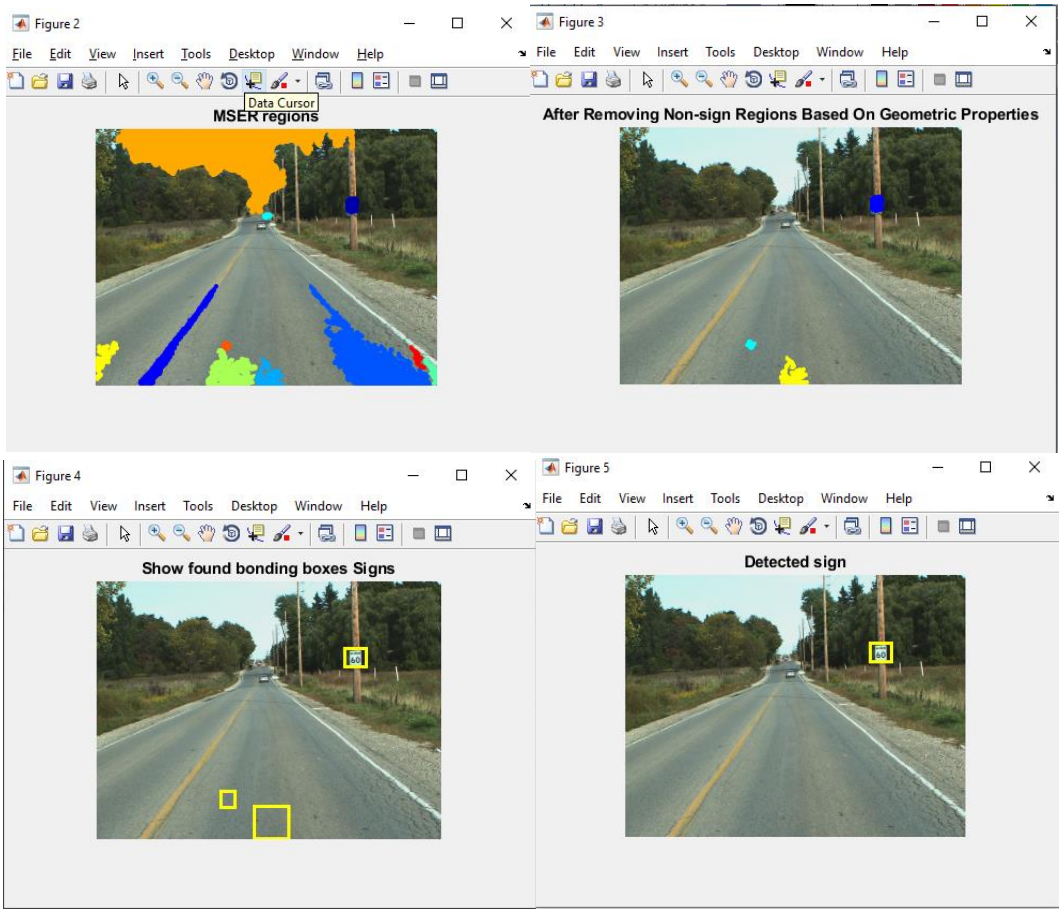


Figure 5. 7 *The processing during the detection stage. Upper left: the MSER regions are found. Upper right: after removing non-sign regions. Bottom left: found bounding boxes for regions. Bottom right: The sign is detected*

5.3 Classification stage

We first introduce the concepts used for the recognition computation, such as the Histogram of Oriented of Gradients as features and Support Vector Machine as a classifier. We also describe the architecture of system.

5.3.1 HOG Descriptors

Histogram of Oriented Gradients (HOG) are widely used in the computer vision and image processing fields. These features are suitable to acquire the features of objects. HOG features were

used in 2005 to detect pedestrians [74] by Dalal and Triggs. They compared HOG features with the features were used by others and concluded that HOG features have better performance.

As it is illustrated in Figure 5.8, the process of HOG implementation is divided into 7 steps.

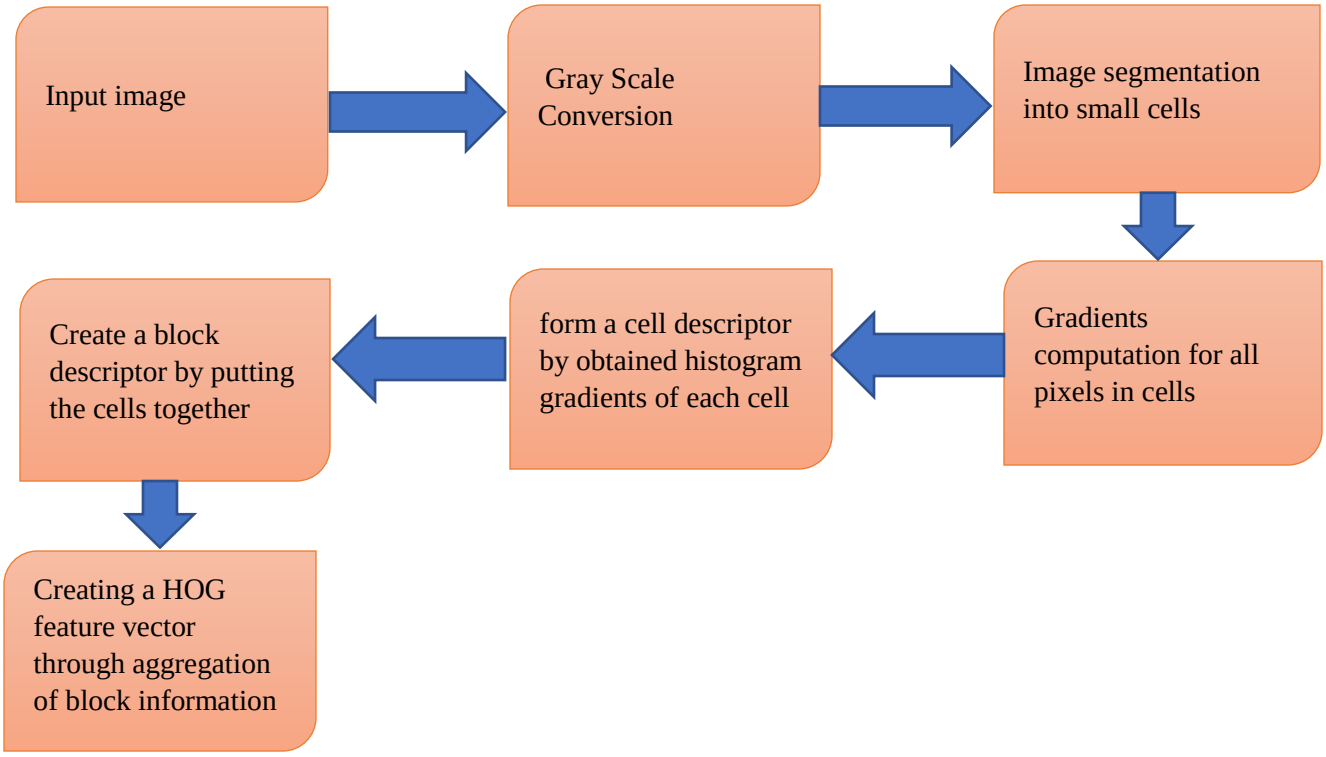


Figure 5. 8 *HOG Features Process*

5.3.1.1 Input Normalization

The color of input images is converted to gray scale levels in order to decrease the impact of some factors such as illumination variation, shadow, contrast etc. Gamma correction is the method used to do this. It also prevents noise from affecting the images.

5.3.1.2 Computation of Gradients

The gradients (both horizontal and vertical) for each pixel over the image is computed in this stage. This stage reduces the influence of light on the image as well as considering information about the contour and texture are the purposes of this stage. In order to determine the horizontal and vertical directions, a 1-D gradient operator is used. G_x is the horizontal gradient and G_y is the vertical one and $H(x, y)$ is the gray value of pixel (x, y). The gradients of each pixel can be defined as follows:

$$G_x = H(x+1, y) - H(x-1, y) \tag{5.10}$$

$$G_y = H(x, y+1) - H(x, y-1) \tag{5.11}$$

Gradient magnitude can also be obtained through the following equation:

$$G_{x,y} = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{5.12}$$

5.3.1.3 Orientation Binning

The goal in this stage is to create the cell histogram of gradient orientation in each image. The image is segmented into the pixels with predetermined size. Each pixel in a cell computes a weighted vote for a histogram channel which is based on orientation. Each vote has a value to be computed based on the magnitude gradients obtained in the previous step.

The shape of the cell can be either radial or rectangular. The histogram channels are evenly spread either 0-180 (unsigned) or 0-360 (signed). In [74] the authors showed that using unsigned gradients and 9 histogram channels has the best performance.

5.3.1.4 Descriptor Blocks

Due to the light or shadow variation and contrast changes, the range of gradient strength is very large. Hence the importance of local normalization of gradient strength to enhance system performance is undeniable. In [74] the authors proposed a method for normalization based on the combination of the individual cells to make a group of them called blocks and then normalize each block separately. Now, the HOG descriptor has become a vector consisting of all cell histograms. However, due to the block overlap, the cells may be present in the block in several times. Rectangular (R-HOG) and Circular (C-HOG) are the main shapes of blocks. Rectangular HOG, consisting of the number of cells per block. Each cell consists of a number of pixels and each histogram pixel includes a number of channels. For example, the authors in [74] found that, (16×16) block size, (8×8) cell size and 9 for histogram channel are the optimal size of parameters for human detection task.

C-HOG is another block HOG shape. It can be found in a circular center cell or in a central cell which has been divided angularly. It also has four parameters: the number of angular and radial bins, the radius of the center bin and expansion factor for the radial bin's radius [74].

In order to increase the invariance to lighting changes, Dalal and Trigge in [74] considered four different normalization methods.

$$L1 - sqrt: f = \sqrt{\frac{v}{\left(||v||\right)_1 + e}} \tag{5.13}$$

$$L1 - norm: f = \frac{v}{\left(||v||\right)_1 + e} \tag{5.14}$$

$$L2 - norm: f = \frac{v}{\sqrt{||v||_2^{\,2} + e^2}} \tag{5.15}$$

$$L2 - hys : f = \frac{v}{\left(||v||\right)_1 + e^2} \quad (v \leq 0.2) \tag{5.16}$$

In this equation, v is the original and non-normalized normal vector f and e are feature vector and a constant value respectively. $L2 - norm$ is computed before the $L2 - hys$. The result is shortened to 0.2. then process is continued by renormalizing the result and using $L2 - norm$ to obtain the final ones. According to what Dalal and Trigge present in [74], the performance of three blocks normalization $L2 - hys$, $L2 - norm$ and $L1 - sqrt$ is the same, while the efficiency of $L1 - norm$ is less than others. Generally, all these block normalizations have excellent influence on non-normalized data.

When the block normalization step is finished, all the histograms are aggregated and combined into a single feature vector. Examples of extracted feature vectors are illustrated in Figure 5.9.

In fact, the feature vectors represent the images in a simple way by extracting and keeping the useful information and removing the useless information. A feature descriptor converts an image with three dimensions into an array.

Figure 5. 9 *Examples of HOG Features on Images [8]*

5.3.2 Support Vector Machine:

Support Vector Machines are one of the most powerful supervised learning algorithms in machine learning. In a supervised learning method, prediction is based on desired labeled data. In this method, the algorithm searches among the labeled data to find the most proper pattern and then uses that one to do the prediction of new examples.

SVM is a classifier searching for an optimal separating hyperplane, which maximizes the margin to classify unseen examples based on labeled training data. Margin is the space is the space between the hyperplane and the closet data points on each side to the decision boundary. These kinds of points called support vectors.

A binary SVM separates the data into two different classes through subsets of samples called Support Vectors, while, a multi class SVM, categorizes the sample in to different classes. When a new test sample goes through the SVM, it is categorized based on its distance to the support vector. Dalal and Trigge [73] used the combination of SVM classifier and normalized HOG features in pedestrian detection.

This method is a useful binary classifier to categorize a data set. A binary SVM can be divided to one-one classifier or one-all classifier. An example of one-one classifier is illustrated in Figure 5.10. Class1 and Class2 are two different samples. Different lines separate these classes, however selecting the best line is an issue. The line separating the classes most properly is called a hyperplane (in a 3D space).

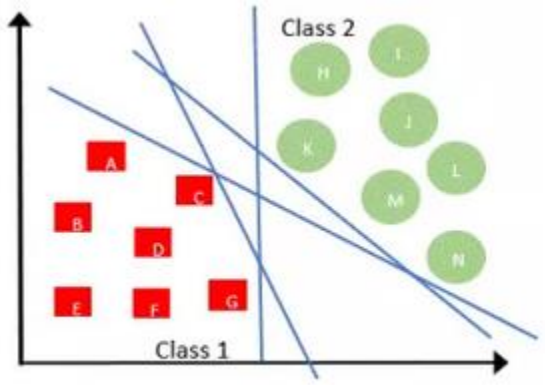


Figure 5. 10 *SVM Classifier [9]*

We have two different sets of samples and want to classify them. Suppose our k data samples can be represented by $(x_1 y_1) \ldots. (x_k y_k)$, the label of each pair is -1 for negative samples and +1 for positive samples. A hyperplane is found and selected as a decision function to separate these two classes. A hyperplane is found with the following equation:

$$w.x_i + b = 0 \tag{5.17}$$

b and *w* are called bias and a normal direction to a hyperplane.

As it is obvious in Figure 5.6, the samples have been separated by several planes. Hence, there are many possible hyperplanes that could be chosen. We choose two planes defined as follows:

$$w.x_i + b = 1 \tag{5.18}$$

$$w.x_i + b = -1 \tag{5.19}$$

These two hyperplanes are parallel and the distance between them must be as great as possible. In fact, we are looking for a hyperplane containing the maximum distance between data points of both classes. The distance between two hyperplanes called margin and is obtained after some computational operation as follows:

$$\frac{2}{||w||} \tag{5.20}$$

In order to maximize the margin, we need to minimize ||w||. In addition, we need to find the optimal hyperplane. A hyperplane is optimal if the distance between a point and the hyperplane is larger in comparison to other planes. Since a hyperplane is close to a point, it suffers from the noise. Hence, an optimal hyperplane has the greatest distance from points. The optimal hyperplane is mostly placed in the middle of the two predefined hyperplanes. In fact, margin is maximized by an optimal hyperplane. An optimal hyperplane has been depicted in Figure 5.7.

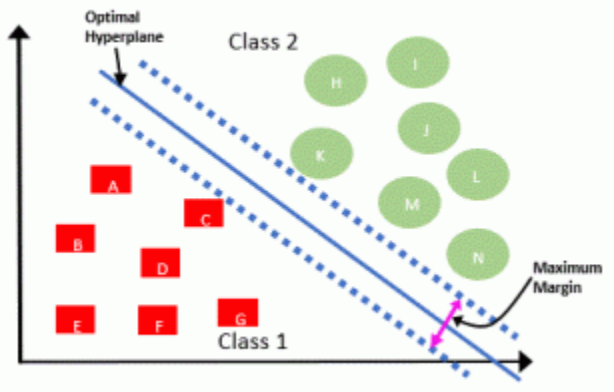


Figure 5. 11 *Example of a hyper plane*

In addition to the above, in order to have an optimal hyperplane, the samples should not fall into the margin. We truncate our states in the following equations and conditions:

$$if \quad y_i = +1, \quad w.x_i + b \geq 1 \tag{5.21}$$

$$if \quad y_i = +1, \quad w.x_i + b \geq 1 \tag{5.22}$$

By the combination of this formula we can find most of the satisfy the following formula:

$y_i(\ w.x_i + b) \geq 1$ and i is the number of samples.

Now to maximize the margin,

$$\max \frac{2}{||w||} \ \ while\ y_i(\ w.x_i + b) \geq 1 \tag{5.23}$$

Or in the other words, the equivalent to the above equation can be represented as follows by quadratic programming:

$$\min \frac{1}{2} \left|\left|w^2\right|\right| \ \ while\ \ y_i(\ w.x_i + b) \geq 1 \tag{5.24}$$

The maximum margin classifier works well for the data distributed such that they can be separated linearly. When the datasets are not separated linearly, there are some mislabeled examples. Hence, we need to use a modified version of maximum margin. In this way, a slack variable ξ is defined to identify how much points are in a wrong side, also allows a certain degree of fault tolerance:

$$y_i(w.x_i + b) \geq 1 - \xi_i \qquad \forall_i \tag{5.25}$$

The value range of slack is another issue. Since if the value is more than 1, it means that there are some points are not classified, while when the value is between 0 and 1, it represents some data points are fall into the margin. To guarantee that our classier not to suffer from these problems, we need to consider a new term called margin constant C. so our equation has become as the following:

$$\left\{\min \frac{1}{2}||w^2|| + C\sum_{i=1}^{n} \xi_i\right\} while\ y_i(w.x_i + b) \geq 1 - \xi_i\ \forall_{i=1\ldots n}, \xi_i \geq 0 \tag{5.26}$$

The constraint of above equation can be solved by introducing a Lagrange theory to transform our formula to dual maximization problem:

$$w = \sum_i \alpha_i x_i y_i \tag{5.27}$$

$$\max\big(\mathrm{w}(\ \alpha)\big) = \left\{\sum_i \alpha_i - \tfrac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i\, y_j K\langle x_i x_j\rangle\right\} \tag{5.28}$$

$$while\ \forall_i \sum_i \alpha_i y_i = 0\ , 0 \leq \alpha_i \leq C \quad (4.23) \tag{5.29}$$

The solution of the SVM is dependent on the kernel function $K\langle x_i x_j\rangle$

5.3.3 Hard Negative Mining

Hard negative mining is an approach to reduce the false positives in training a classifier. In order to train a linear classifier, we need both negative and positive samples. After training the classifier, when it runs over the training samples to test the efficiency of the classifier. Sometimes the result obtained is not very good. In fact, it throws a bunch of false positives (regions detected, which are not actually traffic signs.). Hard negative is when a falsely detected patch is taken and explicitly create a negative example out of that patch, this negative is added to the training set. When a classifier is retrained, it should perform better with this extra knowledge and not make as many false positives.

5.3.4 Implementation Details

In this part, we classify the different types of the road signs. The advanced learning methods we used are introduced here. HOG features and SVM classifier are used to implement this stage. Any classification system needs a set of samples to be used in training process. We selected 30 different classes each containing 50 samples with different sizes to improve the accuracy.

The input images are evaluated, and the labels of each class are processed. Then the HOG features of each class are extracted to be used as features to train the SVM classifier, hence, all the image samples are trained with their own labels. The function used to extract bounding boxes over detected sigs are called to provide the detected road signs. The size of the bounding boxes is resized to 64× 64 to fit the HOG window size. The HOG features are extracted from each bounding box. So as not to miss any small-scale detail, the cell sizes are selected 5×5. Also, block sizes are selected with 4× 4 .Finally, the test images are fed into the SVM classifier. The label of each sign is recognized.

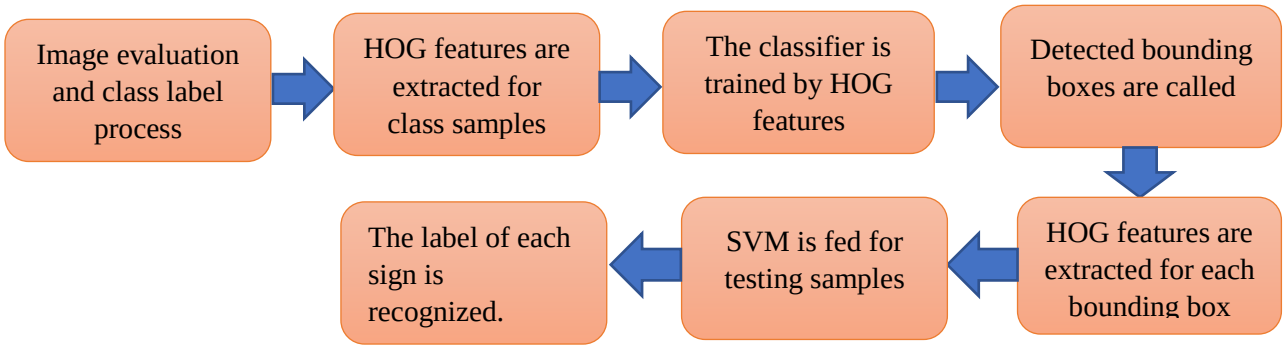


Figure 5. 12 ***Recognition Process***

As we have two different SVM classifiers, the approach of labeling is different. For binary SVM, we labeled our positive samples with value 1 and negative samples with value -1, however, in multi class SVM, we labeled all of classes with value 1, and by indexing, we can obtain the class

name, which is necessary for the prediction of type of traffic signs. The abstraction of our implementation algorithm is depicted in Figure 5.11

As stated before, for training the classifier, we considered 30 different classes containing 50 samples of each class. These samples are extracted from our recorded images from the camera mounted on the car. We extracted samples from different distances, annotations and lighting conditions to obtain more accurate data (see Section 4.2). Examples of different recognized traffic signs are illustrated in Figure 5.12.

Figure 5. 13 *The samples of recognition stage*

5.4 Assessment of Driver Attention based on Gaze

After detecting the regions of interest and classifying the signs, the last stage is to combine the driver's gaze into our detection system. This is done using the computed attentional area of the

driver (see Section 5.1). In this section, we provide information about the algorithm used to evaluate the driver attention based on gaze. To do this, we want to determine:

1. If the detected and recognized sign is in the attentional area of the driver. In other words, if the driver has gazed the sign or not (if he/she has gazed at least once over a traffic sign, we conclude that the sign has been seen otherwise, it has been missed).
2. What is the type of the sign, which is missed or seen.

In order to determine if the driver has seen or missed a sign, we make the assumption that if the center of the bounding box of a sign is inside the attentional area of the driver then the driver has seen the sign. The attentional area of the driver is a 2D ellipse. By using the equation of the 2D ellipse (see equation 5.14), we can identify whether the sign is inside the attentional area of the drive or not.

$$\varepsilon = \frac{((x_{bb}-c_x)\times\cos(\theta)+(y_{bb}-c_y)\times\sin(\theta))^2}{a^2} + \frac{((x_{bb}-c_x)\times\sin(\theta)-(y_{bb}-c_y)\times\cos(\theta))^2}{b^2} \quad (5.17)$$

Where $(x_{bb.}y_{bb})$ is the center coordinate of the bounding box, $(c_{x,}c_y)$ is the center coordinate of the gaze, θ is the gaze angle, (a, b) are the radiuses of the ellipsoid of the attentional area. Then a sign is seen or not by the driver given equation 5.15.

$$\begin{cases} if\ \varepsilon \leq 1 & sign\ is\ seen \\ else & sign\ is\ not\ seen \end{cases} \quad (5.18)$$

The constraint means the sign has been inside the drivers' attention of area obtained from point of gaze in Section 5.1. This constraint is employed over all the detected and recognized signs inside the image frames. Some examples of the result of this step are shown in Figures 5.13, 5.14 and 5.15.

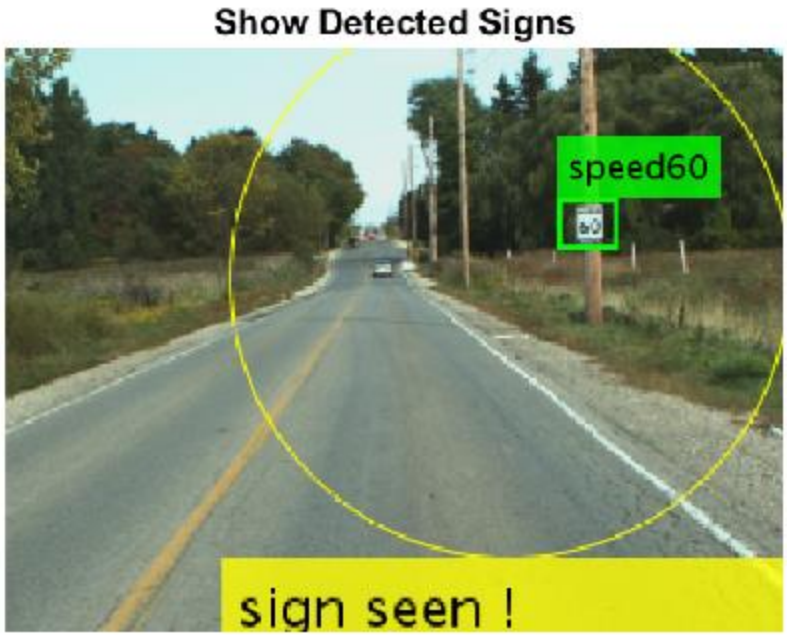


Figure 5. 14 *An example of a seen sign*

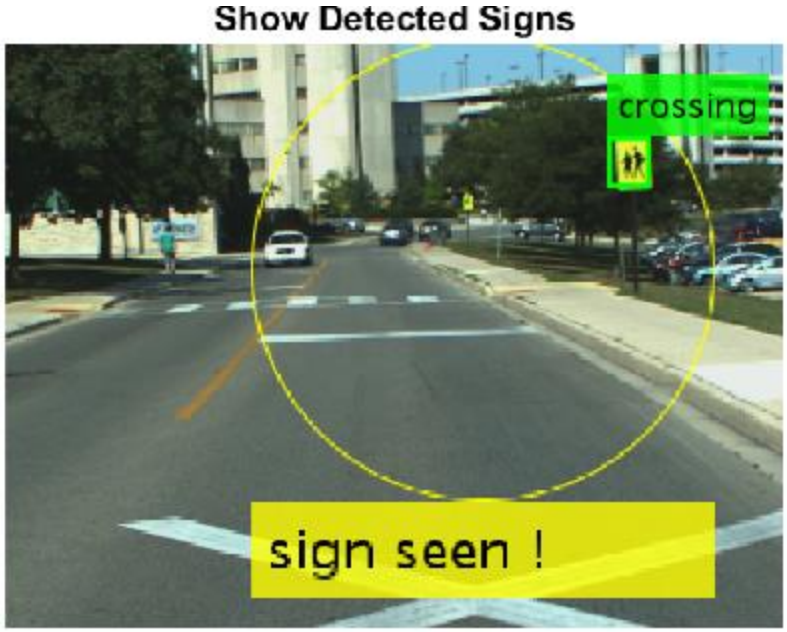


Figure 5. 15 *An example of seen sign*

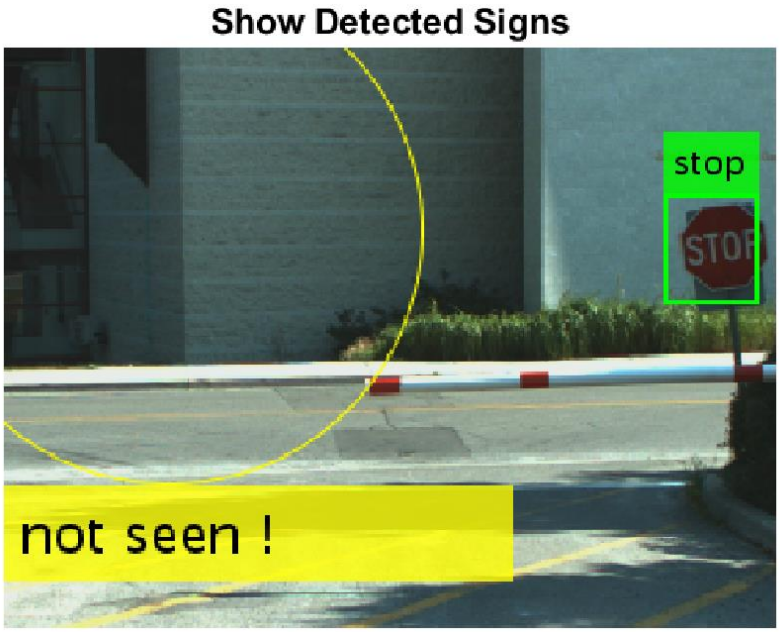


Figure 5. 16 *A sample of missed sign*

5.5 Detection and Recognition Results

We employed our system over more than 328,248 different frames. The accuracy of the detection stage is 80.98% and the accuracy for classification stage is 81.71%; the accuracy of the classification stage is based on the signs detected.

We validated the method against the images for driving sequence for one driver (driver number 14). The calculation of the number of true positive and false negative was done by hand. We obtained the accuracy rate 80.98%. The total number of frames considered was 87131 (30 frames per second) and there was a total of 20505 images of signs in those frames. Our detection algorithm identifies 16606 of these as signs (see Table 5.1).

Number of signs (frames containing signs)	20505
Number of detected signs	16606
Number of missed signs	3899
Detection accuracy	80.98%

Table 5.1. *Summary of traffic sign detection results.*

The result of our classification (recognition) is summarized in Table 5.2. Of the 16606 signs identified in the detection stage, 13568 were correctly classified.

The number of detected signs	16606
The number of correctly classified signs	13568
The number of wrongly classified signs	3038
Accuracy	81.71%

Table 5.2. *Summary of traffic sign classification results.*

Chapter 6
Analysis of Gaze

In order to show that our method is robust enough, we performed our detection system over more than 300,000 frames obtained from cameras mounted on the vehicle; the size of frames is 240 by 320 pixels. We employed our system over frames acquired from 4 different drivers. These sequences were recorded with our RoadLab vehicle.

After detection and recognition of the traffic signs, we performed one more stage to identify whether the sign has been seen by the driver or not, i.e., whether the sign is inside the attentional area of the driver or not. Using these results, we determine the signs seen by our sample of drivers and the types.

The driving sequences taken by the drivers have different number of frames. One reason is that the camera has not recorded some of the images initially. In addition, the frames with low quality have been excluded, also different drivers take different times to finish the driving task.

In order to have a reliable data set, we chose four different driving sequences taken by four different drivers with different ages and gender under different environmental conditions such as sunny weather, partially sunny, etc.

6.1 Analysis of Gaze Data

We analyzed the driving sequences of 4 different drivers and collected the data on whether drivers gazed at the signs in the images or not. This information is summarized in Tables 6.1-6.4.

Driver number 4 **Gender: M** **Weather: Sunny**			
Type of the sign	**The total number of frames containing signs**	**The number of frames with gazed signs**	**The number of frames with not gazed sign**
1. Park not allowed	151	20	131
2. Stop	185	100	85
3. Exit only	80	63	17
4. School crossing	459	457	2
5. Turn left not allowed	91	32	59
6. Traffic light	238	234	4
7. One green sign	93	35	54
8. No Truck	207	200	7
9. Yield	200	131	69
10. Speed 60	142	140	2

11. Speed 70	64	45	19
12. Speed 80	0	0	0
13. Speed 50	298	187	111
14. Speed 40	0	0	0
15. Lane for two-way left turns	0	0	0
16. Parking	0	0	0
17. Construction	422	389	33
18. No enter	22	20	2
19. Keep to the right of traffic	704	588	116
20. U turn not allowed	50	48	2
21. Closed lane	275	194	81
22. Lane ahead is closed	213	62	151
23. Truck	0	0	0
24. Turn right not allowed	0	0	0
25. Railway crossing ahead	0	0	0
26. Road work ahead	128	120	8
27. Traffic travel in one direction	0	0	0
28. Slight bend on the road	0	0	0
29. Bicycle	0	0	0
30. Pedestrian cross over	0	0	0

Table 6.1. *Frames with Signs Gazed and Not Gazed for Driver 4*

Driver number 9 **Gender: female** **Weather condition: Partially Sunny**			
Type of the sign	**The total number of frames containing the signs**	**The number of frames with gazed sign**	**The number of frames with not gazed sign**
1. Park not allowed	605	201	404
2. Stop	73	62	11
3. Exit only	0	0	0
4. School crossing	763	698	65
5. Turn left not allowed	415	103	312
6. Traffic light	1342	1280	62
7. One green sign	64	42	22
8. No Truck	1012	578	434
9. Yield	870	412	458
10. Speed 60	1218	697	521
11. Speed 70	47	39	8
12. Speed 80	31	24	7
13. Speed 50	1862	1693	169
14. Speed 40	165	143	22
15. Lane for two-way left turns	108	31	71
16. Parking	946	334	612
17. Construction	2710	2238	472
18. No enter	54	25	29

19. Keep to the right of traffic	6728	4925	1803
20. U turn not allowed	760	610	150
21. Closed lane	713	621	92
22. Lane ahead is closed	437	387	50
23. Truck	34	12	22
24. Turn right not allowed	236	193	43
25. Railway crossing ahead	237	98	139
26. Road work ahead	224	129	95
27. Traffic travel in one direction	134	65	69
28. Slight bend on the road	54	12	42
29. Bicycle	177	23	154
30. Pedestrian cross over	182	179	3

Table 6.2. *Frames with Signs Gazed and Not Gazed for Driver 9.*

Driver 11 **Gender: Female** **Weather conditions: Sunny**			
Type of the sign	**The total number of frames containing the signs**	**The number of frames with gazed sign**	**The number of frames with not gazed sign**
1. Park not allowed	1221	270	951
2. Stop	318	211	107
3. Exit only	127	88	39
4. School Crossing	821	795	26
5. Turn left not allowed	359	100	259
6. Traffic light	873	783	90
7. One green sign	272	177	95
8. No truck	55	21	34
9. Yield	100	87	13
10. Speed 60	3162	2456	706
11. Speed 70	229	120	109
12. Speed 80	0	0	0
13. Speed 50	618	506	112
14. Speed 40	157	132	25
15. Lane for two-way left turns	28	12	16
16. Parking	1055	327	728
17. Construction	2644	2531	113
18. No enter	146	98	48
19. Keep to the right of the traffic	5263	3147	2116
20. U turn not allowed	64	41	23
21. Closed lane	3139	3087	52
22. Lane ahead is closed	2594	2309	285
23. Truck	96	24	72
24. Turn right not allowed	829	234	595
25. Railway crossing ahead	2082	1743	339
26. Road work ahead	369	294	75

27. Traffic travel in one direction	0	0	0
28. Slight bend in the road	232	34	198
29. Bicycle	336	104	232
30. Pedestrian cross over	0	0	0

Table 6.3. *Frames with Signs Gazed and Not Gazed for Driver 11.*

Driver 14 **Gender: Female** **Weather condition: Sunny**			
Type of the sign	**The total number of frames containing the signs**	**The number frames with sign gazed**	**The number of frames with sign not gazed**
1. Park not allowed	547	244	303
2. Stop	464	280	184
3. Exit only	306	108	198
4. School crossing	964	701	263
5. Turn left not allowed	2573	1094	1479
6. Traffic light	944	898	46
7. One green sign	380	210	170
8. No truck	64	35	29
9. Yield	55	18	37
10. Speed 60	675	581	94
11. Speed 70	72	34	38
12. Speed 80	0	0	0
13. Speed 50	453	201	252
14. Speed 40	183	111	72
15. Lane for two-way left turns	166	32	134
16. Parking	512	231	281
17. Construction	2121	1621	500
18. No enter	113	28	85
19. Keep to the right of the traffic	7151	5892	1259
20. U turn not allowed	58	21	37
21. Closed lane	1309	1029	280
22. Lane ahead is closed	283	208	75
23. Truck	0	0	0
24. Turn right not allowed	136	34	102
25. Railway crossing ahead	184	123	61
26. Road work ahead	277	196	81
27. Traffic travel in one direction	0	0	0
28. Slight bend in the road	179	41	138
29. Bicycle	336	104	232
30. Pedestrian cross over	0	0	0

Table 6. 4. *Frames with Signs Gazed and Not Gazed for Driver 14.*

From Table 6.1-6.4, we see that different drivers have different gaze behavior during driving. Some of the signs have been detected more than others, such as different types of speed limits, however, other types of sign such as 'park not allowed' have been paid less attention.

6.2 Signs Seen and Missed

In the previous part we calculated whether the driver has gazed at images containing traffic signs in frames or not. In this step, we want to calculate the number of actual signs has been detected and recognized by the diver. In order to address this problem, we used the following strategy. If the image of a particular type of sign appears in a sequence of frames, we can take advantage of the Euclidian distance from the center of these signs. We used a threshold with value of 3.32 to compare the Euclidian Distance between the two centers of two consecutive images of signs. If the distance is less than the threshold, we consider that the two signs are the same, if not, then we consider them to be two different signs. Tables 6.5-6.8 illustrate the number of seen or missed traffic signs based on this strategy.

Driver number 4		
Type of the sign	**The number of signs seen**	**The number of missed signs**
1. Park not allowed	3	2
2. Stop	1	1
3. Exit only	1	0
4. School crossing	1	1
5. Turn left not allowed	2	1
6. Traffic light	4	1
7. One green sign	1	0
8. No Truck	1	0
9. Yield	1	0
10. Speed 60	7	1
11. Speed 70	1	0
12. Speed 80	0	0
13. Speed 50	2	1
14. Speed 40	0	0
15. Lane for two-way left turns	0	0
16. Parking	0	0
17. Construction	2	1
18. No enter	1	0
19. Keep to the right of traffic	7	4
20. U turn not allowed	1	0
21. Closed lane	3	0
22. **Lane ahead is closed**	2	0
23. Truck	0	0
24. Turn right not allowed	0	0
25. Railway crossing ahead	0	0
26. Road work ahead	1	0
27. **Traffic travel in one direction**	0	0
28. Slight bend on the road	0	0
29. Bicycle	0	0

30. **Pedestrian cross over**	0	0

Table 6. 5. *The number of individual actual signs seen and missed by Driver 4*.

Driver number 9		
Type of the sign	**The number of signs seen**	**The number of missed signs**
1. Park not allowed	3	7
2. Stop	1	1
3. Exit only	0	0
4. School crossing	6	2
5. Turn left not allowed	2	3
6. Traffic light	7	1
7. One green sign	1	0
8. No Truck	2	1
9. Yield	2	1
10. Speed 60	10	6
11. Speed 70	1	0
12. Speed 80	1	0
13. Speed 50	6	1
14. Speed 40	1	0
15. Lane for two-way left turns	1	1
16. Parking	2	5
17. Construction	7	1
18. No enter	1	1
19. Keep to the right of traffic	15	10
20. U turn not allowed	3	1
21. Closed lane	3	1
22. Lane ahead is closed	5	0
23. Truck	1	0
24. Turn right not allowed	2	0
25. Railway crossing ahead	1	1
26. Road work ahead	1	0
27. Traffic travel in one direction	1	1
28. Slight bend on the road	1	1
29. Bicycle	1	2
30. Pedestrian cross over	2	0

Table 6. 6. *The number of individual actual signs seen and missed by Driver 9.*

Driver 11		
Type of the sign	**The number of signs seen**	**The number of missed signs**
1. Park not allowed	4	9
2. Stop	1	1
3. Exit only	1	0
4. School crossing	6	3
5. Turn left not allowed	3	3

6. Traffic light	7	2
7. One green sign	1	0
8. No Truck	1	1
9. yield	1	1
10. Speed 60	12	4
11. Speed 70	1	1
12. Speed 80	0	0
13. Speed 50	5	1
14. Speed 40	2	0
15. Lane for two-way left turns	1	1
16. Parking	3	7
17. Construction	7	0
18. No enter	1	1
19. Keep to the right of the traffic	15	13
20. U turn not allowed	1	0
21. Closed lane	6	2
22. Lane ahead is closed	4	1
23. Truck	1	0
24. Turn right not allowed	1	1
25. Railway crossing ahead	2	1
26. Road work ahead	2	0
27. Traffic travel in one direction	0	0
28. Slight bend on the road	0	0
29. Bicycle	2	4
30. Pedestrian cross over	0	0

Table 6.7. *The number of individual actual signs seen and missed by Driver 11.*

Driver 14		
Type of the sign	**The number of signs seen**	**The number of missed signs**
1. Park not allowed	1	4
2. Stop	1	1
3. Exit only	1	0
4. School crossing	7	2
5. Turn left not allowed	3	2
6. Traffic light	5	0
7. One green sign	1	0
8. No truck	1	0
9. Yield	1	1
10. Speed 60	8	3
11. Speed 70	1	0
12. Speed 80	0	0
13. Speed 50	4	3
14. Speed 40	1	1
15. Lane for two-way left turns	1	0
16. Parking	4	2
17. Construction	6	1
18. No enter	1	3

19. Keep to the right of the traffic	17	9
20. U turn not allowed	1	1
21. Closed lane	5	0
22. Lane ahead is closed	4	0
23. Truck	0	0
24. Turn right not allowed	1	2
25. Railway crossing ahead	2	1
26. Road work ahead	2	0
27. Traffic travel in one direction	0	0
28. Slight bend in the road	1	1
29. Bicycle	1	2
30. Pedestrian cross over	0	0

Table 6.8. *The number of individual actual signs seen and missed by Driver 14.*

Based on the signs seen and not seen in Tables 6.5-6.8, we can see that there are a substantial number of signs not seen, at least based on the computation of the point of gaze on the sequence of driving images. Admittedly, the collection of gaze and image sequence information is not perfect, and the sign detection and classification methods are not perfect and so the counts reported may be subject to some errors. Regardless, it is clear that there are a substantial percentage of signs which drivers likely do not "see". A summary of the signs seen and missed is presented in Table 6.9.

Driver	**Signs Seen**	**Signs Missed**	**Percentage Missed**
Driver 4	42	13	30.95%
Driver 9	90	48	53.33%
Driver 11	91	57	62.64%
Driver 14	81	39	48.15%
Total	**304**	**157**	**51.64%**

Table 6.9. *The result of individual actual signs seen and missed by All Drivers.*

All Drivers		
Type of the sign	**The number of signs seen**	**The number of missed signs**
1. Park not allowed	11	22
2. Stop	4	4
3. Exit only	3	0
4. School crossing	20	8
5. Turn left not allowed	10	9
6. Traffic light	23	4
7. One green sign	4	0
8. No truck	5	2
9. Yield	5	3
10. Speed 60	37	14

11. Speed 70	4	1
12. Speed 80	1	0
13. Speed 50	17	6
14. Speed 40	4	1
15. Lane for two-way left turns	3	2
16. Parking	9	14
17. Construction	22	3
18. No enter	4	5
19. Keep to the right of the traffic	54	36
20. U turn not allowed	6	2
21. Closed lane	17	3
22. Lane ahead is closed	15	1
23. Truck	2	0
24. Turn right not allowed	4	3
25. Railway crossing ahead	5	3
26. Road work ahead	6	0
27. Traffic travel in one direction	1	1
28. Slight bend in the road	2	2
29. Bicycle	4	8
30. Pedestrian cross over	2	0

Table 6.10. *The number of individual actual signs seen and missed by All Drivers.*

The summary across all drivers for the individual signs is presented in Table 6.10. The results of the table also suggest that some signs may not be as "important" as others. For example, signs for "parking not allowed" may be missed simply because the driver is not interested in parking. Other more critical signs, such as "Road work ahead", so get noticed since they have an impact on driving maneuvers. On the other hand, "Stop" signs were also missed; perhaps because of driver familiarity or because it was obvious that one needed to stop.

Chapter 7
Conclusion and Future Work

In this thesis, we analyzed driving data to measure driver visual attention towards traffic signs. To do this we first designed a traffic sign detection and recognition system based on the North American traffic signs. The accuracy of the detection and recognition methods were 81% and 82%, respectively.

We then analyzed driving sequences that had been collected as part of the RoadLab project. These video sequences provided the actual driving environment for several drivers as well as providing gaze data acquired from a gaze tracking system installed in the laboratory vehicle. By computing the intersection of bounding boxes for detected signs and the gaze area information, we presented a novel analysis to assess the drivers' behavior during driving by considering their gaze.

For this initial study, we assumed that a traffic sign was seen if the gaze of the driver fell on the image of the traffic sign in one of the frames that had captured the image of the sign. We were able to determine if a sign had been seen or missed by the drivers. We analyzed the sequences of four drivers.in fact, we chose these four drivers in order to have data set with different attributes. The driving sequences from these four drivers are different from each other in terms of the environmental condition and gender and the age of the drivers. We were able to count the number of frames that contained images of different types of signs and determine the number of those frames that intersected the driver's point of gaze. We were then able to use this information to count the number of signs seen and not seen by drivers. The main results show that drivers do miss some signs, at least based on point of gaze computations. Though some signs may have been missed or incorrectly classified, there are sufficient numbers of signs identified from images to have confidence in the estimates of signs seen or not. Improving the accuracy is an area for future work.

Our main contributions were:

a) results about drivers' attention in "seeing" signs from actual driving sequences.
b) a novel approach to computing this information using image analysis methods.
c) novel improvements to approaches for traffic sign detection and recognition.

However, there are some limitation in designing our system.

- Methods are limited to Canadian signs, as they are color dependent; generalization of methods is something to consider.
- Our system cannot actually measure the cognition. It happens when a driver has kept his gaze on an object without recognizing that. Our system is only able to focus on driver's gaze and whether a sign has been seen over a number of frames.
- The number of frames has used in classification stage as data set is limited.

This work is only a first step in analyzing driver attention through automated means. There are several areas where further work can be done in the near term:

- The methods introduced make use of various thresholds for the different computations; some sensitivity analysis around different thresholds would be useful in evaluating the robustness of the methods.
- There are additional driving sequences that have not been analyzed and these would enhance the results.
- With more driving sequences, we can look at differences among different classes of drivers, e.g. based on age or gender, to see if any differences appear.
- While the accuracy of the detection and recognition methods are good, improvements can be made; the accuracy of these methods can improve the overall accuracy of the analysis.

The approaches and methods introduced in this thesis can also be used as a starting point for further analyses. For example, with more data about the types of signs seen or not seen, we can look at whether certain types of sign are missed more often than others. With additional data, such as the color, shape or location of the sign, we can look at whether there are "better" signs or "better" locations of signs.

Finally, in addition to traffic signs, the approaches can be generalized to consider other objects that occur in the driving environment, such as vehicles, bicycles, pedestrians, etc. These methods can be extended to encompass a more general analysis of which objects in the visual space of the driver are seen or not seen and under what circumstances. This can help provide more information for driving systems and perhaps provide additional warnings for drivers to further reduce accidents.

References

[1] https://www.foap.com/image-photo/roadside-view/page-4, https://www.wbir.com/article/ traffic/ road-conditions-remain-fair-for-east-tennessee-on- friday/51-513778576

[2] https://www.hardmansigns.com/blog/p.180327000/damaged-sign-well-come-and-fix-it/

[3] http://www.wseas.org/multimedia/journals/signal/2016/a285814-096.pdf

[4] “RGB color space.” http://photo.hanyu.iciba.com/ upload/ encyclopedia_ 2/88/95/bk_ 889574aebacda6bfd3e534e2b49b8028 _vheBLq.jpg, Online; accessed: 2016-09-30.

[5] V. Andrey and K. H. Jo, “Automatic detection and recognition of traffic signs using geometric structure analysis”, in SICE-ICASE International Joint Conference, pp. 1451–1456, 2006.

[6] https://www.mathworks.com/help/images/convert-from-hsv-to-rgb-color-space.html

[7] https://forum.videohelp.com/threads/385085-Is-there-an-actual-difference-between-YCbCr-%28YUV-%29-4-4-4-and-RGB

[8] https://ascelibrary.org/doi/10.1061/%28ASCE%29CP.1943-5487.0000491

[9] https ://analyticsrusers.blog/ 2018/07/15/machine- learning -and- credit- risk- part- 4 -support-vector-machines/

[10] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation" survey”, 2015.

[11] H. Akatsuka and S. Imai, "Road signposts recognition system", Training Journal, pp. 1211, 2013.

[12] S. Beauchemin, M. Bauer, D. Laurendeau, T. Kowsari, J. Cho, M. Hunter, and O. Mc-Carthy, “Roadlab: "An in-vehicle laboratory for developing cognitive cars", in Proc. 23rd Int. Conf. CAINE, 2010.

[13] Brki’c. Karla, "An overview of traffic sign detection methods" Department of Electronics, Microelectronics, Computer and Intelligent Systems Faculty of Electrical Engineering and Computing Unska 3, 10000 Zagreb, Croatia, 2014.

[14] A. Gudigar, Shreesha Chokkadi ·Raghavendra U, "A review on automatic detection and recognition of traffic sign"© Springer Science +Business Media New York 2014.

[15] Y.-Y. Nguwi and A.Z. Kouzani, "A Study on Automatic Recognition of Road Signs" School of Electrical Engineering and Computer Science the University of Newcastle Callaghan, NSW 2308, Australia 2006.

[16] de la Escalera, A., Moreno, L.E., Salichs, M.A., Armingol, J.M.: "Road traffic sign detection and classification". IEEE Trans. Ind. Electron. 44(6), 848–859, 1997.

[17] K. Plataniotis and A. N. Venetsanopoulos, "Color image processing and applications. Springer Science & Business Media", 2013.

[18] L. D. Lopez and O. Fuentes, "Color-based road sign detection and tracking in Image Analysis and Recognition", pp. 11381147, Springer, 2007.

[19] A. Ruta, F. Porikli, S. Watanabe, and Y. Li, "In-vehicle camera traffic sign detection and recognition", Machine Vision and Applications, vol. 22, no. 2, pp. 359-375, 2011.

[20] K. H. Lim, K. P. Seng, and L. M. Ang, "Intra color-shape classification for traffic sign recognition", in IEEE Computer Symposium (ICS), International, Tainan, China, pp. 642647, 2010.

[21] C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, and T. Koehler, "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information", in IEEE Intelligent Vehicles Symposium, pp. 255-260, 2005.

[22] H. Gomez-Moreno, S. Maldonado-Bascon, P. Gil-Jimenez, and S. Lafuente-Arroyo, "Goal evaluation of segmentation algorithms for traffic sign recognition", IEEE Transactions on Intelligent Transportation Systems, vol. 11, no. 4, pp. 917-930, 2010.

[23] T. Bui-Minh, O. Ghita, P. F. Whelan, T. Hoang, and V. Q. Truong, "Two algorithms for detection of mutually occluding traffic signs", in IEEE International Conference on Control, Automation and Information Sciences (ICCAIS), Ho Chi Minh City, Vietnam, pp. 120-125, 2012.

[24] S. Maldonado Bascon, J. Acevedo Rodriguez, S. Lafuente Arroyo, A. Fernndez Caballero, and F. Lopez- Ferreras, "An optimization on pictogram identification for the road-sign recognition task using svms", Computer Vision and Image Understanding, vol. 114, no. 3, pp. 373383, 2010.

[25] H. Fleyeh, "Shadow and highlight invariant color segmentation algorithm for traffic signs", in IEEE Conference on Cybernetics and Intelligent Systems, Bangkok, Thailand, pp. 17, 2006.

[26] Zaklouta F, Stanciulescu B "Real-time traffic sign recognition in three stages". Robot Auton Syst 62(1):16–24, 2014.

[27] A. Broggi, P. Cerri, P. Medici, P. P. Porta, and G. Ghisio, "Real time road signs recognition", in IEEE Intelligent Vehicles Symposium, pp. 981–986, 2007.

[28] de la Escalera, A. Moreno, L.E. Salichs, M.A., Armingol, J.M.: Road traffic sign detection and classification. IEEE Trans. Ind. Electron. 44(6), 848–859, 1997.

[29] M. Benallal, J. Meunier "Real-time color segmentation of road signs". In: CCECE, pp 1823–1826, 2003.

[30] A. Soetedjo., K. Yamada "A new approach on red color thresholding for traffic sign recognition system". J Japan Soc Fuzzy Theory Intell Inform 19(5):457–465, 2007.

[31] Anjan Gudigar, Shreesha Chokkadi, Raghavendra U "A review on automatic detection and recognition of traffic sign". Received: 13 March 2014 / Revised: 1 August 2014 / Accepted: 19

September 2014 Published online: 1 October 2014 Springer Science Business Media New York 2014.

[32] M. Zadeh, T. Kasvand, C. Suen "Localization and recognition of traffic road signs for automated vehicle control systems. In: Proceedings of the SPIE intelligent system and automated manufacturing", pp 272–282, 1998.

[33] S. Maldonado - Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez-Moreno, and F. Lopez-Ferreras, "Road-sign detection and recognition based on support vector machines", IEEE transactions on intelligent transportation systems, vol. 8, no. 2, pp. 264– 278, 2007.

[34] H. Gomez-Moreno, S. Maldonado-Bascon, P. Gil-Jimenez, and S. Lafuente-Arroyo, "Goal evaluation of segmentation algorithms for traffic sign recognition", IEEE Trans. Intell. Transp. Syst., vol. 11, no. 4, pp. 917–930, Dec. 2010.

[35] Greenhalgh J, Mirmehdi M "Real-time detection and recognition of road traffic signs". IEEE Trans Intell Transp Syst 13(4):1498–1506, 2012.

[36] Chiang H-H, Chen Y-L, Wang W-Q, Lee T-T "Road speed sign recognition using edge-voting principle and learning vector quantization network". In: ICS. Taiwan, pp 246–251, 2010.

[37] H. Fleyeh, "Road and traffic sign color detection and segmentation-a fuzzy approach", Red, vol. 250, p. 207, 2005.

[38] S.Vitabile, G.Pollaccia, G.Pilato and F.Sorbello, "Road signs recognition using a dynamic pixel aggregation technique in the HSV color space", in Proc. other II the Int. Conf. on Image Analysis and Processing, Palermo, Italy, pp. 572-577, 2001.

[39] W.-J. Kuo and C.-C. Lin, "Two-stage road sign detection and recognition", in IEEE International Conference on Multimedia and Expo, China, pp. 14271430, 2007.

[40] P. Paclik, J. Novovicova, P and Pudil, P.Somol, "Road Sign Classification using the Laplace Kernel Classifier", Pattern Recognition Letters, Volume 21, pp. 1165-1173, No. 13-14, 2000.

[41] Y.-Y. Nguwi, A.Z. Kouzani "A Study on Automatic Recognition of Road Signs", and School of Electrical Engineering and Computer Science the University of Newcastle Callaghan, NSW2308, Australia, 2006.

[42] S. Lafuente-Arroyo, P. Garca-Daz, F. Acevedo-Rodrguez, P. Gil-Jimenez, and S. Maldonado-Basc_on, "Traffic sign classification invariant to rotations using support vector machines", Proceedings of Advabced Concepts for Intelligent Vision Systems, Brussels, Belgium, 2004.

[43] W. Shadeed, D. Abu-Al-Nadi, and M. Mismar, "Road traffic sign detection in color images", in Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems (CECS), vol. 2, pp. 890893, 2003.

[44] P. Gil Jiménez, S. Bascón, H. Moreno, S. Arroyo, and F. Ferreras, "Traffic sign shape classification and localization based on the normalized FFT of the signature of blobs and 2D homographies", Signal Process., vol. 88, no. 12, pp. 2943–2955, Dec. 2008.

[45] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gomez- Moreno, and F. López-Ferreras, "Road-sign detection and recognition based on support vector machines", IEEE Trans. Intell. Transp. Syst., vol. 8, no. 2, pp. 264–278, Jun. 2007.

[46] S. Lafuente-Arroyo, S. Salcedo-Sanz, S. Maldonado-Bascón, J. A. Portilla-Figueras, and R. J. López-Sastre, "A decision support system for the automatic management of keep-clear signs based on support vector machines and geographic information systems", Expert Syst. Appl., vol. 37, no. 1, pp. 767–773, Jan. 2010.

[47] S. Lafuente-Arroyo, S. Salcedo-Sanz, S. Maldonado-Bascón, J. A. Portilla- Figueras, and R. J. López-Sastre, "A decision support system for the automatic management of keep-clear signs based on support vector machines and geographic information systems", Expert Syst. Appl., vol. 37, no. 1, pp. 767–773, Jan. 2010.

[48] L. Han, L. Ding, L. Qi "Real-time recognition of road traffic sign in moving scene image using genetic algorithm". In: Proc. 1st. int. conf. mach. learn. cybern, vol 2, pp 83–86, 2002.

[49] H. Liu, D. Liu, and J. Xin, "Real-time recognition of road traffic sign in motion image based on genetic algorithm", in Proc. Int. Conf. Mach. Learn. Cybern., vol. 1, pp. 83–86, Nov 2002.

[50] G. Nicchiotti, E. Ottaviani, P. Castello, and G. Piccioli, "Automatic road sign detection and classification from color image sequences", In S. Impedovo, editor, Proc. 7thMInt. Conf on Image Analysis and Processing, World Scientific, 1994, pp. 623-626, 2006.

[51] L. Priese and V. Rehrmann, "On hierarchical color segmentation and applications", In Proc. CVPR, 1993, pp 633-634, 2005.

[52] de la Escalera, A., Armingol, J., Mata, M.: "Traffic sign recognition and analysis for intelligent vehicles". Image Vis. Comput. 21, 247–258, 2003.

[53] W. Shadeed, D. Abu-Al-Nadi, and M. Mismar, "Road traffic sign detection in color images", in Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems (CECS), vol. 2, pp. 890–893, 2003.

[54] Fleyeh H, Dougherty M "Traffic sign classification using invariant features and support vector machines", In: Procs. IEEE intelligent vehicles symposium. Netherlands, pp 530–535, 2008.

[55] D. Soendoro and I. Supriana, "Tra_c sign recognition with color-based method, shapearc estimation and svm", in International Conference on Electrical Engineering and Informatics (ICEEI), pp. 1–6, 2011.

[56] X. Gao, K. Hong, P. Passmore, L. Podladchikova, and D. Shaposhnikov, "Color vision model-based approach for segmentation of traffic signs", EURASIP Journal on image and video processing, vol. 2008, no. 1, pp. 1–7, 2007.

[57] X. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova, "Recognition of traffic signs based on their colour and shape features extracted using human vision models", J. Vis. Commun. Image Represent. vol. 17, no. 4, pp. 675–685, 2006.

[58] V. Prisacariu, R. Timofte, K. Zimmermann, I. Reid, and L. Van Gool, "Integrating object detection with 3D tracking towards a better driver assistance system", pp. 3344–3347, in Proc. 20th ICPR, Aug. 2010.

[59] A. Mammeri, A. Boukerche, and M. Almulla, "Design of traffic sign detection, recognition, and transmission systems for smart vehicles", Wireless Communications, IEEE, vol. 20, pp. 36–43, December 2013.

[60] M. A. Garcia, M. A. Sotelo, and E. M. Gorostiza, "Traffic sign detection in static images using MATLAB", in IEEE Conference on Emerging Technologies and Factory Automation (ETFA), vol. 2, pp. 212–215, 2003.

[61] S.-H. Hsu and C.-L. Huang, "Road sign detection and recognition using matching pursuit method", Image and Vision Computing, vol. 19, no. 3, pp. 119–129, 2001.

[62] M. Shneier, "Road sign detection and recognition", in Defense and Security Symposium, vol. 6230, pp. 1-7, 2006.

[63] D. Gavrila, "Traffic sign recognition revisited", pp. 86–93, in DAGM-Symposium, 1999.

[64] N. Barnes and A. Zelinsky, "Real-time radial symmetry for speed sign detection", in IEEE Intelligent Vehicles Symposium, pp. 566–571, 2004.

[65] G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system", in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),vol. 1, pp. 70–75, 2004.

[66] M. A. Garcia-Garrido, M. A. Sotelo, and E. Martin-Gorostiza, "Fast traffic sign detection and recognition under changing lighting conditions", in IEEE Intelligent Transportation Systems Conference, pp. 811–816, 2006.

[67] M. ´A. Garc´ıa-Garrido, M. ´A. Sotelo, and E. Mart´ın-Gorostiza, "Fast road sign detection using hough transform for assisted driving of road vehicles", in International Conference on Computer Aided Systems Theory, pp. 543–548, Springer, 2005.

[68] G. Loy and N. Barnes, "Fast shape-based road sign detection for a driver assistance system", in IEEE/RSJ International Conference on Intelligent Robots and Systems. (IROS). Proceedings. vol. 1, pp. 70-75, 2004.

[69] K. H. Lim, K. P. Seng, and L. M. Ang, "Intra color-shape classification for traffic sign recognition", in IEEE Computer Symposium (ICS), International, Tainan, China, pp. 642-647, 2010.

[70] C. Bahlmann, Y. Zhu, V. Ramesh, M. Pellkofer, and T. Koehler, "A system for traffic sign detection, tracking, and recognition using color, shape, and motion information", in IEEE Intelligent Vehicles Symposium, pp. 255260, 2005.

[71] S. dong Zhu, Z. Yi, and L. Xiao-feng, "Detection for triangle traffic sign based on neural network", in IEEE International Conference on Vehicular Electronics and Safety, China, pp. 2528, Oct 2005.

[72] T. Bui-Minh, O. Ghita, P. F. Whelan, T. Hoang, and V. Q. Truong, "Two algorithms for detection of mutually occluding traffic signs", in IEEE International Conference on Control, Automation and Information Sciences (ICCAIS), Ho Chi Minh City, Vietnam, pp. 120-125, 2012.

[73] Y. Xie, L.-f. Liu, C.-h. Li, and Y.-y. Qu, "Unifying visual saliency with hog feature learning for traffic sign detection", in IEEE Intelligent Vehicles Symposium, pp. 24–29, 2009.

[74] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 1, pp. 886–893, 2005.

[75] X. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova, "Recognition of traffic signs based on their color and shape features extracted using human vision models", J. Vis. Commun. Image Represent, vol. 17, no. 4, pp. 675–685, 2006.

[76] Y. Xie, L.-F. Liu, C.-H. Li, and Y.-Y. Qu, "Unifying visual saliency with HOG feature learning for traffic sign detection", in Proc. IEEE Intell. Veh. Symp, pp. 24–29, Jun. 2009.

[77] J. Gangyi, Z. Yi, and C. T. Young, "Morphological skeleton analysis of traffic signs on road", in IEEE International Conference on Systems, Man, and Cybernetics, Beijing, China, vol. 1, pp. 7075, 1996.

[78] Pavel Paclik and Jana Novovicova, "Road Sign Classification without Color Information", in Proc. of6th Conf. of Advanced School of Imaging and Computing, ASCI, Lommel, Belgium, 2000.

[79] S. Vitabile, G. Pollaccia, G. Pilato, and F.Sorbello, "Road signs recognition using a dynamic pixel aggregation technique in the HSV colour space", in Proc. Int. Conf. on Image Analysis and Processing, , pp. 572-577, Italy, 2001.

[80] Greenhalgh J, Mirmehdi M, "Real-time detection and recognition of road traffic signs". IEEE Trans Intell Transp Syst 13(4):1498–1506, 2012.

[81] Li Y, Pankanti S, Guan W, "Real-time traffic sign detection: an evaluation study". In: ICPR. Istanbul, pp 3033–3036, 2010.

[82] J. Miura, T. Kanda, and Y. Shirai, "An active vision system for real-time traffic sign recognition", in IEEE Intelligent Transportation Systems, pp. 52-57, 2000.

[83] J. Torresen, J. W. Bakke, and L. Sekanina, "Efficient recognition of speed limit signs", in Intelligent Transportation Systems, pp. 652-656, 2004.

[84] C.-Y. Fang, S.-W. Chen, and C.-S. Fuh, "Road-sign detection and tracking", IEEE transactions on vehicular technology, vol. 52, no. 5, pp. 1329–1341, 2003.

[85] X. W. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova, "Recognition of traffic signs based on their color and shape features extracted using human vision models", Journal of Visual Communication and Image Representation, vol. 17, no. 4, pp. 675-685, 2006.

[86] L. Sekanina and J. Torresen, "Detection of Norwegian speed limit signs", in European Stability Mechanism, pp. 337-340, 2002.

[87] H.-M. Yang, C.-L. Liu, K.-H. Liu, and S.-M. Huang, "Traffic sign recognition in disturbing environments", in Foundations of Intelligent Systems, pp. 252261, Springer, 2003.

[88] P. Doll´ar, Z. Tu, P. Perona, and S. Belongie, "Integral channel features", 2009.

[89] C.-Y. Fang, S.-W. Chen, and C.-S. Fuh, "Road-sign detection and tracking", IEEE Transactions on Vehicular Technology, vol. 52, no. 5, pp. 13291341, 2003.

[90] L. Petersson, L. Fletcher, N. Barnes, and A. Zelinsky, "An interactive driver assistance system monitoring the scene in and out of the vehicle", in IEEE International Conference on Robotics and Automation, vol. 4, pp. 34753481, 2004.

[91] H. Fleyeh and M. Dougherty, "Road and traffic sign detection and recognition", in 10th EWGT Meeting and 16th Mini-EURO Conference, Poznan, Poland, pp. 644-653, 2005.

[92] J. F. Khan, R. R. Adhami, and S. M. Bhuiyan, "Image segmentation-based road sign detection", in IEEE Southeastcon, Atlanta, pp. 24-29, 2009.

[93] S.H Hsu, C.L. Huang, "Road sign detection and recognition using matching pursuit method Image", Vis. Compute. 19, 119–129, 2001.

[94] Y. Berube Lauziere, Gingras and Frank P. Ferrie, "A Model-Based Road Sign Identification System", in Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition, ppI-1163- 1-1170, Vol.1,2001.

[95] H. Ohara, I. Nishikawa, S. Miki, and N. Yabuki, "Detection and recognition of road signs using simple layered neural networks", in Proc. Of the gth Int. Conf. on Neural Information Processing, pp. 626-630, vol. 2, 2002.

[96] J. Torresen, J. Bakke, and L. Sekanina, "Efficient recognition of speed limit signs", in Proc. of the 7th Int. IEEE Conf on Intelligent Transportation Systems, pp. 652 – 656, 2004.

[97] Krumbiegel, D., Kraiss, K.-F., Schrieber, S. "A connectionist traffic sign recognition system for onboard driver information", In: Proceedings of the Fifth IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems, pp. 201–206, 1992.

[98] P. Douville, "Real-time classification of traffic signs". Real-Time Imaging, 6(3), 185–193, 2000.

[99] S. Vitabile, A. Gentile, and F. Sorbello, "A neural network based automatic road signs recognizer", in International Joint Conference on Neural Networks (IJCNN), vol. 3, pp. 2315–2320, 2002.

[100] Y. Aoyagi and T. Asakura, "A study on traffic sign recognition in scene image using genetic algorithms and neural networks", in Proceedings of the IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation, Taipei, China, vol. 3, pp. 1838-1843, 1996.

[101] J. J. Abukhait, "A discriminative imaging-based framework for road sign condition assessment using local features and SVM classifiers", 2012.

[102] K.-m. Wang and Z.-r. Xu, "Study on method of traffic signs recognition based on neural network and invariant moments", Application Research of Computers, vol. 3, pp. 254-256, 2004.

[103] S. Lafuente-Arroyo, P. Gil-Jimenez, R. Maldonado-Bascon, F. Lopez-Ferreras, and S. Maldonado-Bascon, "Traffic sign shape classification evaluation i: Svm using distance to borders", in IEEE Proceedings. Intelligent Vehicles Symposium, pp. 557–562, 2005.

[104] C. Kiran, L. V. Prabhu, V. A. Rahiman, K. Rajeev, and A. Sreekumar, "Support vector machine learning based traffic sign detection and shape classification using distance to borders and distance from center features", in TENCON IEEE Region 10 Conference, pp. 1–6, 2008.

[105] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs", IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 4, pp. 1498– 1506, 2012.

[106] M. Kus, M. Gokmen, S. Etaner- Uyar, "Traffic sign recognition using scale invariant feature transform and color classification". In: ISCIS. Istanbul, pp 1–6, 2008.

[107] H. Fleyeh, M. Dougherty, "Traffic sign classification using invariant features and support vector machines", In: Procs. IEEE intelligent vehicles symposium. Netherlands, pp 530–535, 2008.

[108] Y. Gu, T. Yendo, MP. Tehrani, T. Fujii, M. Tanimoto, "A new vision system for traffic sign recognition", In: IEEE intelligent vehicles symposium. USA, pp 7–12, 2010.

[109] W-J. Kuo, C-C. Lin, "Two-stage road sign detection and recognition.", In: IEEE international conference on multimedia and expo. Beijing, pp 1427–1430, 2007.

[110] M. C. Kus, M. Gokmen, and S. Etaner-Uyar, "Traffic sign recognition using scale invariant feature transform and color classification", in 23rd International Symposium on Computer and Information Sciences (ISCIS), pp. 1–6, 2008.

[111] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features", in European conference on computer vision, pp. 404–417, Springer, 2006.

[112] W. Wu, X. Chen, and J. Yang, "Detection of text on road signs from video", IEEE Transactions on Intelligent Transportation Systems, vol. 6, no. 4, pp. 378–390, 2005.

[113] U.S. Dept. of Transportation, "Traffic Safety Facts 2006: A Compilation of Motor Vehicle Crash Data from the Fatality Analysis Reporting System and the General Estimates System", tech. report DOTHS 810 818, Nat'l Highway Traffic Safety Administration, 2006.

[114] M. Bayly, B. Fildes, M. Regan, and K. Young, "Review of crash effectiveness of intelligent transport system", Traffic Accident Causation in Europe (TRACE), 2007.

[115] E. Murphy-Chutorian and M. Trivedi, "Head pose estimation in computer vision: a survey", IEEE Trans. Pattern Anal. Mach. Intell. 31(4), 607–626, 2009.

[116] S. Langton, H. Honeyman, and E. Tessler, "The Influence of Head contour and Nose Angle on the Perception of Eye-Gaze direction", Perception and Psychophysics, vol. 66, no. 5, pp. 752-771, 2004.

[117] L. Fletcher, A. Zelinsky "Driver Inattention Detection based on Eye Gaze–Road Event Correlation". Department of Information Engineering, RSISE, Australian National University Canberra, Australia, 2006.

[118] Ishikawa, T., Baker, S. and Kanade, T, "Passive driver gaze tracking with active appearance models". Proceedings of the 11th World Congress on Intelligent Transportation Systems. October 2004.

[119] M.E. Jabon, J. N. Bailenson, E. Pontikakis, L. Takayama, and C. Nass "Facial-Expression Analysis for Predicting Unsafe Driving Behavior".Stanford University, 2011.

[120] http://phys.org/news/2012-11-tracking-facial-features-safercomfortable.html#inlRlv.

[121] S.J. lee, J. Jo, H.G. Jung, K. R Park, J.Kim, "Real-Time Gaze Estimator Based on Driver's Head Orientation for Forward Collision Warning System". IEEE Transactions on Intelligent Transportation Systems, 2011.

[122] L. Haworth, T. J. Triggs, and E.M. Grey, "Driver fatigue: Concepts, measurement and crash countermeasures". Technical Report, Federal Office of Road Safety Contract Report 72, Human Factors, 2005.

[123] M. Land, and D. Lee, "Where we look when we steer". Nature, 369(6483): 742–744, July 1994.

[124] N Apostoloff. and A. Zelinsky", Vision in and out of vehicles: integrated driver and road scene monitoring". International Journal of Robotics Research, 23(4-5): 513–538, 2004.

[125] K. Takemura, et al. "Driver monitoring system based on non-contact measurement system of driver's focus of visual attention". Proceedings of the IEEE Symposium on Intelligent Vehicles, June, pp. 581–586, 2003.

[126] F. Holzmann, et al. "Introduction of a full redundant architecture into a vehicle by integration of a virtual driver". Proceedings of the IEEE Intelligent Vehicles Symposium, June, pp. 126–131, 2006.

[127] J. C. Gerdesand, E. J. Rossetter, "A unified approach to driver assistance systems based on artificial potential fields. Journal of Dynamic Systems, Measurement and Control", 123(3): 431–438, 2001.

[128] O. Carsten, and F. Tate, "Intelligent speed adaptation: The best collision avoidance system". Proceedings of the17th International Technical Conference on the Enhanced Safety of Vehicles, June, Amsterdam, 2001.

[129] M. M. Trivedi, T. Gandhi, and J. McCall, "Looking in and looking out of a vehicle: selected investigations in computer vision based enhanced vehicle safety". IEEE International Conference on Vehicular Electronics and Safety, October, pp. 29–64, 2005.

[130] Y. Matsumoto, J. Heinzman, and A. Zelinsky, "The essential components of human-friendly robots". Proceedings of the International Conference on Field and Service Robotics FSR'99, August, pp. 43–51, 1999.

[131] H. Summala, and T. Nicminen, "Maintaining lane position with peripheral vision during in-vehicle tasks". Human Factors, 38: 442–451, 1996.

[132] Maltz,M. and Shinar, D, "Imperfect in-vehicle collision avoidance warning systems can aid drivers". Human Factors, 46(2): 357–366, 2004.

[133] L. Fletcher, G. Loy, N. Barnes, and A. Zelinsky. "Correlating driver gaze with the road scene for driver assistance systems". Robotics and Autonomous Systems, 52:71–84, 2005.

[134] F. Vicente, Z. Huang, X. Xiong, F. D. la Torre, W. Zhang, and D. Levi. "Driver gaze tracking and eyes off the road detection system". IEEE Transactions on Intelligent Transportation Systems, 16(4):2014–2027, August 2015.

[135] R. Perko, M. Schwarz, and L. Paletta. "Aggregated mapping of driver attention from matched optical flow". 2014 IEEE International Conference on Image Processing, pages 214–218, January 2014.

[136] N. Haworth, T. Triggs, and E. Grey. Driver fatigue: "concepts, measurement and crash countermeasures". Technical report, Federal Office of Road Safety Contract Report 72 by Human Factors Group, Department of Psychology, Monash University, 1988.

[137] L. Bergasa, J. Nuevo, M. Sotalo, and M. Vazquez. "Real-time system for monitoring driver vigilance". In Proceedings of the IEEE Intelligent Vehicle Symposium, pages 78–83, 2004.

[138] N. Ziraknejad, P. Lawrence., and D. Romilly. "The effect of time-off light camera integration time on vehicle driver head pose tracking accuracy". In IEEE International Conference on Vehicular Electronics and Safety, pages 247–254, 2012.

[139] H.B. Kang Dept. of Digital Media Catholic University of Korea. "Various Approaches for Driver and Driving Behavior Monitoring: A Review", 2013.

[140] J. Chan, P. Gonzalez, E. M. M. Perez. Designing Traffic Signs: "A Case Study on Driver Reading Patterns and Behavior", 2016.

[141] S.J. Zabihi ; S.M. Zabihi ; S.S. Beauchemin ; M.A. Bauer , "Detection and Recognition of Traffic signs inside the attentional visual field of drivers ", 2017.

[142] S. Beauchemin, M. Bauer, D. Laurendeau, T. Kowsari, J. Cho, M. Hunter, and O. McCarthy, "Roadlab: An in-vehicle laboratory for developing cognitive cars", in Proc. 23rd Int. Conf. CAINE, 2010.

[143] J. J. Abukhait, "A discriminative imaging-based framework for road sign condition assessment using local features and svm classifiers", 2012.

[144] K. Takagi, H. Kawanaka, M. S. Bhuiyan, and K. Oguri, "Estimation of a three dimensional gaze point and the gaze target from the road images", in 14th International IEEE Conference onIntelligent Transportation Systems (ITSC), pp. 526–531, 2011.

[145] T. Kowsari, S. S. Beauchemin, M. A. Bauer, D. Laurendeau, and N. Teasdale, "Multidepth cross-calibration of remote eye gaze trackers and stereoscopic scene systems", in IEEE Intelligent Vehicles Symposium, pp. 1245–1250, 2014.

[146] S. Salti, A. Petrelli, F. Tombari, N. Fioraio and L. D. Stefano: "A Traffic Sign Detection pipeline based on interest region extraction", 2013.

[147] T. Nicole Brooks: "Exploring Geometric Property Thresholds for Filtering Non-Text Regions in A Connected Component Based Text Detection Application". Seidenberg School of CSISPace University New York, 2017.

[148] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: a multi-class classification competition", in the 2011 International Joint Conference on Neural Networks (IJCNN), pp. 1453–1460, 2011.

[149] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3d localization", Machine Vision and Applications, vol. 25, no. 3, pp. 633–647, 2014.

[150] F. Larsson and M. Felsberg, "Using fourier descriptors and spatial models for traffic sign recognition", in Scandinavian Conference on Image Analysis, pp. 238–249, Springer, 2011.

[151] R. Belaroussi, P. Foucher, J.-P. Tarel, B. Soheilian, P. Charbonnier, and N. Paparoditis, "Road sign detection in images: A case study", in 20th International Conference on Pattern Recognition (ICPR), pp. 484–488, 2010.

Curriculum Vitae

Name:	Shabnam Shabani
Post-Secondary Education and Degrees:	The University of Western Ontario London, Ontario, Canada 2017-2019 M.Sc. Computer Science
Honors and Scholarship Awards:	Western Graduate Research 2017-2019
Related Work Experience:	Teaching Assistant and Research Assistant The University of Western Ontario 2017 - 2019