

Electronic Thesis and Dissertation Repository

3-10-2017 12:00 AM

Detection and Recognition of Traffic Signs Inside the Attentional Visual Field of Drivers

Seyedjamal Zabihi, *The University of Western Ontario*

Supervisor: Steven Beauchemin, *The University of Western Ontario*

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

© Seyedjamal Zabihi 2017

Follow this and additional works at: <https://ir.lib.uwo.ca/etd>



Part of the [Artificial Intelligence and Robotics Commons](#)

Recommended Citation

Zabihi, Seyedjamal, "Detection and Recognition of Traffic Signs Inside the Attentional Visual Field of Drivers" (2017). *Electronic Thesis and Dissertation Repository*. 4421.
<https://ir.lib.uwo.ca/etd/4421>

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlsadmin@uwo.ca.

Abstract

Traffic sign detection and recognition systems are essential components of Advanced Driver Assistance Systems and self-driving vehicles. In this contribution we present a vision-based framework which detects and recognizes traffic signs inside the attentional visual field of drivers. This technique takes advantage of the driver's 3D absolute gaze point obtained through the combined use of a front-view stereo imaging system and a non-contact 3D gaze tracker. We used a linear Support Vector Machine as a classifier and a Histogram of Oriented Gradient as features for detection. Recognition is performed by using Scale Invariant Feature Transforms and color information. Our technique detects and recognizes signs which are in the field of view of the driver and also provides indication when one or more signs have been missed by the driver.

Keywords: traffic sign detection, traffic sign recognition, traffic sign dataset, Advanced Driver Assistance Systems (ADAS).

Acknowledgements

First and foremost, I am profoundly beholden to my supervisor, Dr. Steven Beauchemin for teaching me countless lessons and insights on the workings of academic research. He is not only a brilliant mentor, but also a great friend. I appreciate all his offerings of time, ideas, and thoughtful guidance to make my graduate student experience stimulating.

I would also like to thank all my laboratory colleagues and mentors, in particular, Prof. Michael A. Bauer, my brother Seyed Mohsen Zabihi, and Md Junaedur Rahman for their perceptive comments and support. The group has been a source of friendship and appropriate suggestions and cooperation.

Lastly, I would like to thank my family for their endless love, encouragement, and sacrifices. My deepest appreciation goes to my parents for constantly supporting and providing me with so much love even from 6000 miles away.

Contents

Abstract	i
Acknowledgements	ii
Table of Contents	iii
List of Figures	v
List of Tables	1
1 Introduction	2
1.1 Advanced Driver Assistance Systems	2
1.2 Research Overview	2
1.3 Problems and Issues	3
1.4 Contributions	5
1.5 Thesis Outline	5
2 Literature Review	6
2.1 Detection Methods	6
2.1.1 Detection Based on Color Information	6
RGB Color Space	7
HSV Color Space	8
HSI color space	9
YUV Color Space	11
YCbCr Color Space	11
2.1.2 Detection Based on Shape Information	12
the Hough Transform	13
Edge Detection	13
Neural Networks	13
Template Matching	14
Gradient features	14
Other methods	14
2.1.3 Detection Based on Hybrid Methods	15
2.2 Recognition Methods	15
3 Traffic Signs Dataset	18
3.1 Background	18

3.2	Differences Between North American and European Signs	19
3.3	RoadLab Traffic Sign Dataset	20
4	TSDR System Architecture Design	22
4.1	Establishing the Field of View of the Driver	22
4.2	Detection Phase	24
4.2.1	HOG Descriptors	24
	Image Normalization	25
	Gradient Calculation	25
	Orientation Binning	26
	Normalization and Descriptor Blocks	26
4.2.2	Support Vector Machine (SVM)	27
4.2.3	Implementation Details	30
4.3	Recognition Phase	31
4.3.1	Scale Invariant Feature Transform	32
	Scale Space	33
	SIFT Detector	33
	SIFT Descriptor	35
4.3.2	Implementation Details	35
5	Experimental Results	39
5.1	Traffic Sign Detection Results	39
5.2	Traffic Sign Recognition Results	43
5.3	Output Images	46
6	Conclusion and Future Work	50
	Bibliography	51
	Curriculum Vitae	57

List of Figures

1.1	<i>Effect of variable lighting conditions.</i>	3
1.2	<i>Examples of damaged and partially obscured traffic signs.</i>	4
1.3	<i>Presence of a banner on the left side of the road</i>	4
2.1	<i>RGB color space [1]</i>	7
2.2	<i>RGB color segmentation results [2]: (a,b): input images, (c,d): candidate regions</i>	8
2.3	<i>HSV color space [3]</i>	10
2.4	<i>YCbCr color space [4]</i>	12
2.5	<i>Different votes cast by a gradient element $g(p)$ when searching for different shapes at a given radius [5]</i>	13
2.6	<i>Building the distance transform image. from left to right: original image, edge image and the distance transform image [6].</i>	14
3.1	<i>Samples of European traffic signs.</i>	19
3.2	<i>Samples of speed limit signs. left: Europe right: Canada.</i>	19
3.3	<i>Samples of image sequences in our dataset.</i>	21
4.1	<i>Physical configuration (left): Remote eye tracking system. (right): RoadLAB stereoscopic vision system.</i>	23
4.2	<i>Depiction of the diver attentional gaze cone</i>	23
4.3	<i>Attentional gaze areas projected onto the forward stereo scene system of the vehicle.</i>	24
4.4	<i>HOG descriptor implementation scheme.</i>	25
4.5	<i>Examples of HOG features extracted from traffic signs.</i>	27
4.6	<i>. 2D classification sample</i>	28
4.7	<i>. Optimal hyperplane</i>	29
4.8	<i>. The effect of the soft margin constant C.</i>	30
4.9	<i>(top): Positive samples (bottom): Average image.</i>	31
4.10	<i>Hard negative mining.</i>	32
4.11	<i>Gaussian pyramid.</i>	34
4.12	<i>Searching an image for local extrema over scale and space.</i>	34
4.13	<i>The SIFT descriptor layout. The size of a spatial bin is $m\sigma$.</i>	35
4.14	<i>Examples of template signs.</i>	36
4.15	<i>HSV color space images a) (top): detected sign b) (bottom): template sign.</i>	37
4.16	<i>DoG scale space images (left): template sign (right): detected sign.</i>	38
5.1	<i>Traffic sign detection rate</i>	40

5.2	<i>The AUC of two classifiers</i>	41
5.3	<i>true positives, false positives, true negatives and false negatives</i>	42
5.4	<i>ROC curve with threshold=0.56</i>	42
5.5	<i>Traffic sign recognition rate</i>	44
5.6	<i>Confusion matrix example</i>	44
5.7	<i>Confusion matrix with accuracy of 88.9%</i>	45
5.8	<i>Detection and recognition of the stop sign. The driver has seen the sign.</i>	46
5.9	<i>Detection and recognition of the speed limit sign. The driver has seen the sign.</i>	47
5.10	<i>Detection and recognition of the traffic light ahead sign. The driver has seen the sign.</i>	47
5.11	<i>Detection and recognition of the bike lane ends sign. The driver has seen the sign.</i>	48
5.12	<i>Detection and recognition of the school zone sign. The driver has missed the sign.</i>	48
5.13	<i>Detection and recognition of the No heavy trucks permitted on this roadway sign. The driver has missed the sign.</i>	49
5.14	<i>Detection and recognition of the keep to the right of traffic island sign. The driver has missed the sign.</i>	49

List of Tables

2.1	HUE ANGLE OF A COLOR	9
2.2	THRESHOLD VALUES USED FOR COLOR SEGMENTATION	10
3.1	PUBLICLY AVAILABLE TRAFFIC SIGN DATASETS	18
5.1	DETECTION RATE AND FALSE POSITIVE PER FRAME	40
5.2	SUMMARY OF TRAFFIC SIGN DETECTION RESULTS	40
5.3	STATE OF THE ART DETECTION RESULTS ON EUROPEAN TRAFFIC SIGNS	43
5.4	SUMMARY OF TRAFFIC SIGN RECOGNITION RESULTS	43

Chapter 1

Introduction

1.1 Advanced Driver Assistance Systems

Advanced Driver Assistance Systems (ADAS) are technologies that aim to provide drivers with critical information about road and traffic environments, take control of some complex or repetitive tasks, and increase the overall safety of drivers and pedestrians.

According to the National Highway Traffic Safety Administration (NHTSA) [7], the main cause (94%) of vehicle crashes is human error. Among all possible types of driver errors, recognition errors, decision errors, and performance errors are the most frequent driver-related critical reasons for accidents. Based on this investigation, we conclude that there should be a strong motivation to develop and implement technologies that alleviate and avoid accidents. Indeed, vehicles with Advanced Driver Assistance Systems (ADAS) are common nowadays.

Many ADAS have been proposed in the past two decades. For example, Global Positioning System (GPS) navigation is the most popular technology that has been around since the 1990s. There are also other ADAS that have been developed in recent years, including adaptive cruise control, adaptive light-beam control, automatic braking, automatic parking, collision avoidance systems, blind spot detection, driver drowsiness detection, hill descent control, night vision, and lane departure warning systems.

The goal of these systems is to make roads a safer place for vehicles and pedestrians. However, these systems pay practically no attention to the modes of behaviour exhibited by the driver. In this thesis, we plan to demonstrate that driver gaze behaviour is an important aspect of safety as we design and implement a Traffic Sign Detection and Recognition (TSDR) technique capable of informing drivers if they have not seen particular traffic signs.

1.2 Research Overview

TSDR have attracted a great deal of attention over the past few years. TSDR techniques attempt to make drivers aware of incoming traffic signs on the road and warn them against possible dangers and troubles. Traffic signs include useful and essential visual information such as speed limitation, lane closings, direction, distance to destination, and dangerous or unusual conditions [8] to assist the driver while driving. If a driver fails to notice a traffic sign or understand the information provided in it, then it is reasonable to assume that driving conditions

become suboptimal. In order to mitigate this possibility, a TSDR system can significantly help the driver by detecting and recognizing those traffic signs.

In this thesis, we describe three different stages. First, we find the attentional visual field of drivers in every frame by using the 3D absolute gaze point of the driver obtained through the combined use of a front-view stereo imaging system and a non-contact 3D gaze tracker. After establishing this gaze area in every image sequence, we perform sign detection based on Histograms of Oriented Gradients (HOG) features in addition to a Support Vector Machine (SVM) classifier. The next stage consists of recognizing the exact type of detected road sign candidates which is performed by combining color information inside a reliable feature matching technique.

1.3 Problems and Issues

Developing a TSDR system is a challenging and difficult task. There are many factors that can make the process of detection and recognition of traffic signs less successful. We can divide the problems each TSDR system is facing into the following elements:

1. Inconsistent Lighting Conditions

One of the main issues in the development of a TSDR system is the problem of inconsistent lighting conditions. Different traffic signs have different colors which make them noticeable. This color information is sensitive to changes in illumination. Figure 1.1 illustrates the effect of lighting on a green information and direction sign.



Figure 1.1: *Effect of variable lighting conditions.*

2. Damaged Traffic Signs

Damaged and slightly occluded traffic signs can render both the detection and the recognition stages faulty. Figure 1.2 shows examples of damaged and partially occluded stop signs.

3. Blurring and Fading Effects



Figure 1.2: *Examples of damaged and partially obscured traffic signs.*

Another important difficulty for a TSDR system is the fading and blurring of traffic signs caused by illumination through rain or snow. These conditions can lead to increase in false detections, and reduce the effectiveness of a TSDR system.

4. Motion Artifacts

Images taken from a running vehicle may be subject to motion blur. Using a low-resolution camera is also another reason for noisy or blurred images.

5. Region Establishment

There are different objects on the road that have characteristics similar to traffic signs. These objects can cause difficulties for the system while determining the exact location of the signs. For instance, advertising banners on the road can result in incomplete target region establishment (see figure1.3).



Figure 1.3: *Presence of a banner on the left side of the road*

6. Poor Visibility

Headlights of other vehicles on the road can cause shadows which lead to poor visibility. Other factors that may reduce the visibility are rain, snow, and fog.

All of the aforementioned factors may negatively impact the performance of a TSDR system.

1.4 Contributions

While the problem of traffic sign detection may appear solved (in particular for European traffic signs), there is still room for numerous improvements for North American traffic signs. We have created a complete and reliable North American traffic sign dataset by using our experimental vehicle and front-view stereo system which is installed on top of the vehicle. Furthermore, To the best of our knowledge, this work is the first TSDR method that shows how to detect and recognize traffic signs inside the attentional visual field of drivers and, by extension, outside of it. Other approaches are just merely concerned with the identification of traffic signs within image sequences. By using the combination of Histogram of oriented gradients (HoG) and SVM classifier for the detection phase and combination of color information and Scale Invariant Feature Transform descriptors for the recognition phase, we have developed an efficient Traffic Sign Detection and Recognition system (TSDR).

Since our main focus is on the detection and recognition of those traffic signs that drivers are likely to see at any moment, the number of false detections should be less of a problem due to the reduction of the search area. In addition, by performing sign detection outside of the driver's visual attention area, a feedback can be provided when the driver fails to see a traffic sign before it reaches a certain, predefined minimal distance to the vehicle.

1.5 Thesis Outline

The rest of this Thesis is structured as follows, in Chapter 2 we focus on describing the algorithms and methods found in the recent literature. Chapter 3 introduces the RoadLAB traffic sign dataset and provides a comparison between North American and European traffic signs. Chapter 4 describes the algorithms used for establishing the visual field of drivers, the traffic sign detection, and the traffic sign recognition in detail. Chapter 5 provides the results of the detection and recognition phases. Lastly, Chapter 6 concludes the thesis and addresses future plans.

Chapter 2

Literature Review

Different traffic sign recognition methods have been proposed in the recent past. They usually consist of two sequential processes, namely a detection stage that identifies a Region of Interest (RoI), and a recognition stage that identifies the exact type of sign or rejects the identified RoI.

2.1 Detection Methods

In this Section, we review the techniques that have been used for the detection of traffic signs. As mentioned above, the main task in detection is creating candidate image regions which are probably traffic signs. Therefore, we need to search for ROIs from image sequences and pre-process them for the sign recognition stage. Several methods can be used for extracting ROIs, based on main characteristics of traffic signs such as shape and color. In general, we can divide the detection methods into three main groups:

- Detection based on color information
- Detection based on shape information
- Detection based on hybrid methods

In the following Sections, we discuss the common approaches used in each of the three groups.

2.1.1 Detection Based on Color Information

One of the prevalent approaches for sign detection is using color information. By using thresholding or advanced segmentation methods, finding the areas of an image which contains the color of interest is possible. However, the main drawback of this method is its high sensitivity to variable illumination conditions. Depending on the time of the day and weather conditions, the colors may be inconsistent. Different types of color spaces are used in the current literature. The most widely used ones are RGB, HSI/HSV, YUV, YCbCr, CIE Lab, and CIECAM97 [9].

RGB Color Space

RGB is the most frequently used color space for most applications in image processing and computer vision. The intensity values of RGB components are between 0 and 255. Since there are three color channels, a total of $256^3 = 16,777,216$ colors can be displayed. Figure 2.1 represents the RGB color space. In many systems, RGB color segmentation is used for

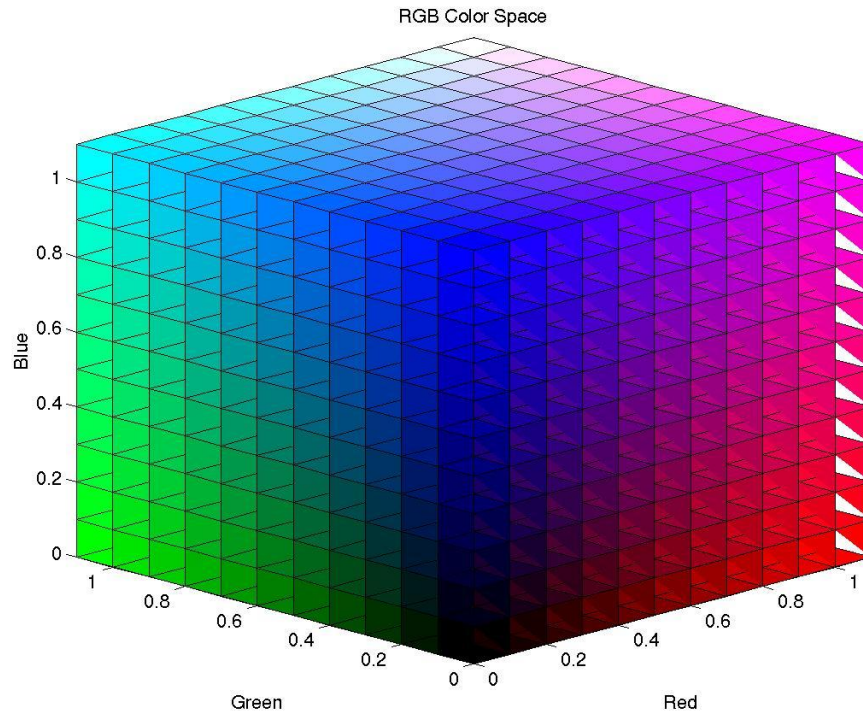


Figure 2.1: *RGB color space [1]*

the task of traffic sign detection. For example, authors in [10] tested YUV and RGB color spaces and finally took the second one because of its lower number of false positives and lower computational time. Moreover, Benallal and Meunier [11] have studied the behavior of the RGB components of many traffic signs and have found this method reliable for road signs segmentation. Many other researchers also focused on RGB segmentation such as [12], [13], and [14]. Figure 2.2 illustrates an example of color segmentation using RGB thresholding.

Authors in [15] proposed a new method for detecting white signs by using RGB color space. They used chromatic and achromatic filters in order to help them in detecting white signs. These are computed as:

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

where R , G , and B represent the brightness of the red, green, and blue channels. D is the degree of extraction of an achromatic color, and is determined experimentally. If the value of $f(R, G, B)$ is less than 1, then it is said to represent achromatic colors, and conversely if it is greater than 1 [15].

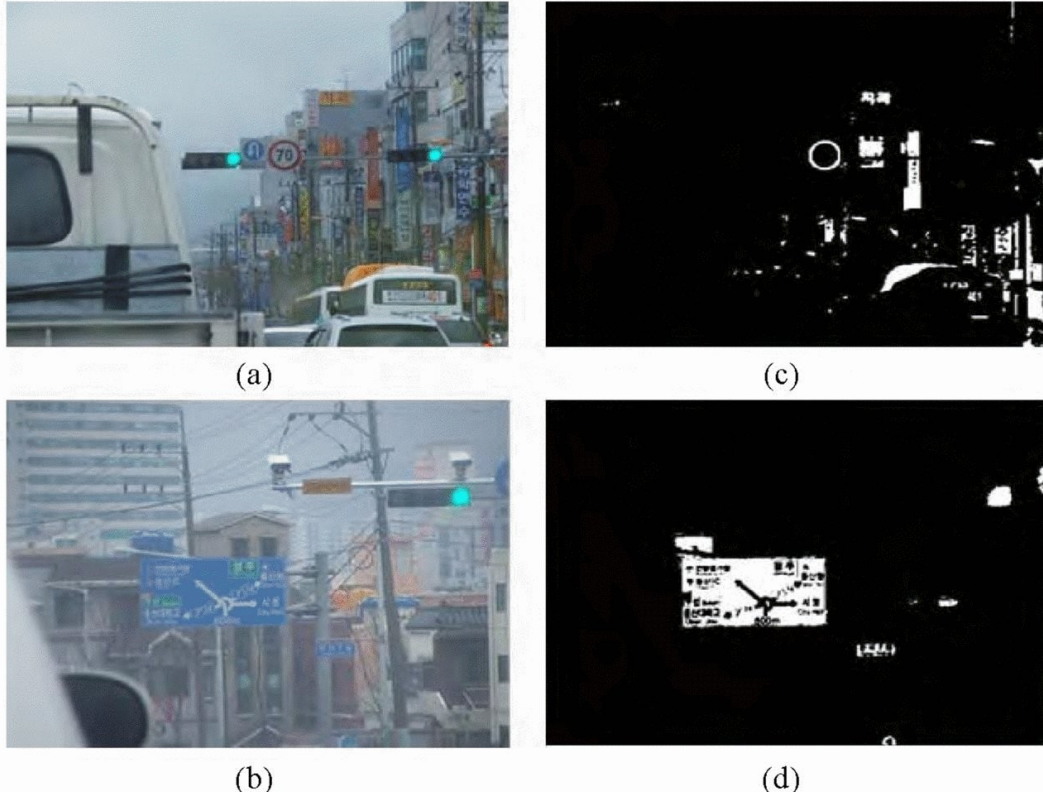


Figure 2.2: *RGB color segmentation results [2]: (a,b): input images, (c,d): candidate regions*

In [16, 17] RGB normalized thresholding was used to alleviate the problem of variable illumination conditions. Each color channel is normalized as follows:

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

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

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

where $r + g + b = 1$. The use of these normalized parameters have a minor effect in reducing the effects of illumination changes. However, this method remains sensitive to color saturation.

HSV Color Space

Another color space that has been used for road sign segmentation is known as HSV, stands for Hue, Saturation, and Value. They are the three components of this color model, defined as follows:

- **Hue:** Hue is an angle between 0 and 360 degrees and it represents the color. Table 2.1 shows that different hue angles result in different colors.

Angle	Color
0-60	Red
60-120	Yellow
120-180	Green
180-240	Cyan
240-300	Blue
300-360	magenta

Table 2.1: HUE ANGLE OF A COLOR

- **Saturation:** This value represents the range of gray in color space. It varies from 0 to 1. 0 means the color is gray, and 1 means the color is a primary color. For example, the saturation value of white is 0.
- **Value:** This value which is also called lighting varies from 0 to 1. It indicates how dark or how bright a color is. 0 means completely dark and 1, completely bright.

Figure 2.3 depicts the HSV color space. Since the HSV color space is less sensitive to variable lighting conditions, many researchers considered using this method for traffic sign segmentation. For instance, Paclik *et al.* [18] used this color space due to its similarity to human perception of colors. Authors in [8] also applied the HSV color space. They converted the RGB color space to the HSV color space following these equations:

$$H = \begin{cases} (G - B) * 60/S & \text{if } V = R \\ 80 + (B - R) * 60/S & \text{if } V = G \\ 240 + (R - G) * 60/S & \text{if } V = B \end{cases} \quad (2.5)$$

$$S = \begin{cases} (V - \min(R, G, B)) * 255/V & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (2.6)$$

$$V = \max(R, G, B) \quad (2.7)$$

Hasan Fleyeh [19] proposed a different solution for color detection and segmentation of road signs based on fuzzy sets. RGB images were first taken by a digital camera mounted on a vehicle and then converted to the HSV color space. Following this, they were segmented using a set of fuzzy rules depending on the hue and saturation values of each pixel in the HSV color space. According to [20], the HSV color space is appropriate for color segmentation since the Hue component of this space is invariant to changing light intensity.

HSI color space

Another popular color space for traffic sign segmentation is HSI (Hue, Saturation, Intensity). Chiang *et al.* [21] transferred colors in RGB color space to the HSI color space because this color model provides convenience for perceiving colors and their characteristics. They used the following equations for conversion between the two color models:

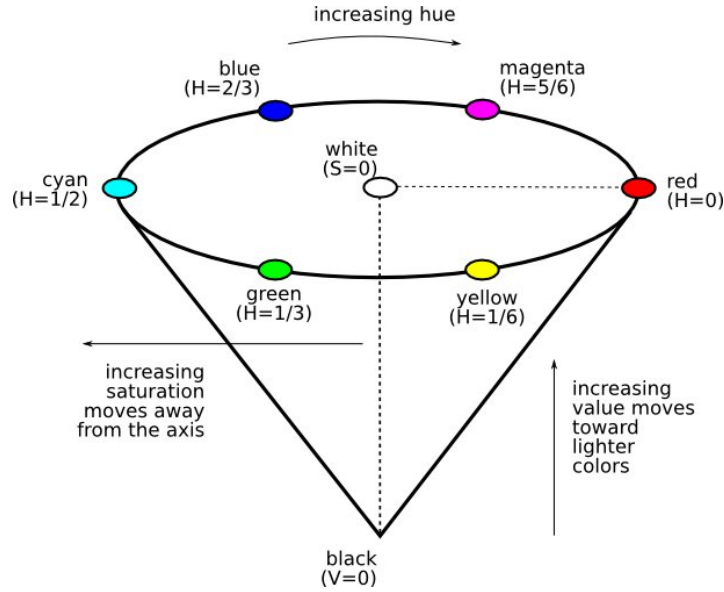


Figure 2.3: HSV color space [3]

$$H = \arccos \left(\frac{\frac{1}{2}((R - G) + (R - B))}{\sqrt{(R - G)^2 + (R - G)(R - B)}} \right) \quad (2.8)$$

$$S = 1 - \frac{3\min(R, G, B)}{R + G + B} \quad (2.9)$$

$$I = \frac{R + G + B}{3} \quad (2.10)$$

Gudigar et al. [22] believe that the hue and saturation components of the HSI color space are sufficient to isolate traffic signs. They have built the histogram of hue and saturation for red traffic signs in order to acquire threshold values. Furthermore, they used Look-Up Tables (LUTs) for enhancing the hue and saturation values. Authors in [23] also applied LUTs for enhancing the hue and saturation values of red, blue, and yellow coloured traffic signs.

Another example of using HSI color space is found in [24]. Authors adopted this color segmentation method because the HSI color space is relatively immune to inconsistent lighting conditions. Table 2.2 demonstrates the threshold values used in the color segmentation phase.

	Red	Blue
Hue	$H \geq 0$ and $H < 0.111\pi$ $H \geq 1.8\pi$ and $H < 2\pi$	$1.066\pi \leq H \leq 1.555\pi$
Saturation	$0.1 < S \leq 1$	$0.28 < S \leq 1$
Intensity	$0.12 < I < 0.8$	$0.22 < I < 0.5$

Table 2.2: THRESHOLD VALUES USED FOR COLOR SEGMENTATION

Other authors[25, 26, 27, 28] have used this color space as it models human color perception better than the RGB color space.

YUV Color Space

This color space has been used for analog color television broadcasting, as defined by National Television System Committee (NTSC). The Y component represents lightness (or luma), while the U and V components determine the color (chroma). Y varies from 0 to 1 while U and V vary from 0 to 255. Conversion formulas from RGB to YUV and from YUV to RGB are presented below:

- **RGB to YUV**

$$\begin{cases} Y = 0.299R + 0.587G + 0.114B \\ U = -0.147R - 0.289G + 0.436B \\ V = 0.615R - 0.515G - 0.100B \end{cases} \quad (2.11)$$

- **YUV to RGB**

$$\begin{cases} R = Y + 1.140V \\ G = Y - 0.395U - 0.581V \\ B = Y + 2.032U \end{cases} \quad (2.12)$$

A few researchers used the YUV color space for color segmentation [29], [30]. In [30], authors combined the results of the HSV and YUV color spaces. Their segmentation method is based on the Hue value of HSV and image chrominance (U,V) in the YUV space.

YCbCr Color Space

The YCbCr or Y'CbCr color space (sometimes written as YCBCR or Y'CBCR), is used for component digital video. YCbCr is a scaled and offset version of the YUV color space [31]. Figure 2.4 delineates the YCbCr color space and the following equations that perform the transformation between RGB and YCbCr spaces.

- **RGB to YCbCr**

$$\begin{cases} Y = 0.299R + 0.587G + 0.114B \\ Cb = 0.564(B - Y) \\ Cr = 0.713(R - Y) \end{cases} \quad (2.13)$$

- **YCbCr to RGB**

$$\begin{cases} R = Y + 1.402Cr \\ G = Y - 0.334Cb - 0.714VCr \\ B = Y + 1.772Cb \end{cases} \quad (2.14)$$

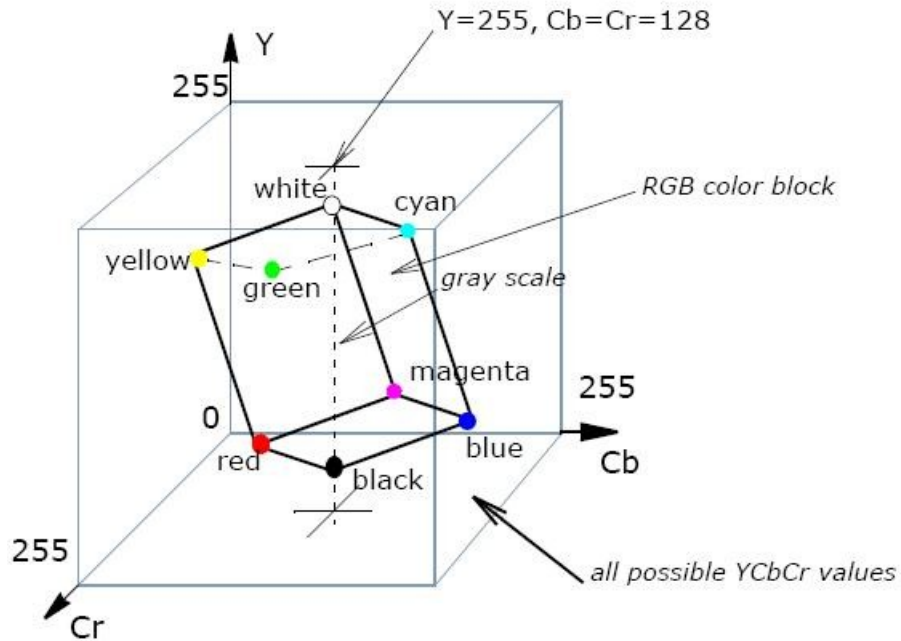


Figure 2.4: *YCbCr color space [4]*

This color space is rarely used for the task of traffic sign segmentation. The CIE Lab and YCbCr color spaces were used for different shapes of traffic signs by [32]. In particular, they have found that YCbCr space is suitable for segmentation of triangular signs.

While we reviewed the most frequently used color segmentation methods, there are still a few more color spaces that have been used in current literature. Soendoro *et al.* [33] performed color filtering using the CIELAB color space adjoined with hue, due to its efficiency on localizing traffic signs. According to [34] a color appearance model named CIECAM97 performs better than other color models such as CIELUV, CIELAB, and RGB. Others [35] also used CIECAM97 for color segmentation.

2.1.2 Detection Based on Shape Information

Another important characteristic of a traffic sign is its shape (circular, triangular, octagonal, and rectangular). While color-based detectors are popular, there are many other approaches based on the shapes of traffic signs. As we previously noted, the main drawback of color-based detectors is their sensitivity to weather conditions and variations in luminance. In contrast to color-based detectors, shape-based detectors do not suffer from weather conditions and variations in luminance. Therefore, many researchers prefer to use shape-based techniques for the detection of traffic signs. Several approaches for shape-based traffic sign detection have been proposed in the current literature. We proceed to review the most common ones.

the Hough Transform

The Hough Transform (HT) can be used for the detection of lines, circles, rectangles or other curves. This method was first introduced in 1962 and its first application was finding lines in image sequences. The main advantage of HT is its immunity to noise, scaling, and rotation.

Authors in [36] used the Hough Transform for circumference in order to detect circular signs, while for triangular-signs detection they used the Hough transform for straight lines. Loy and Barnes [5] proposed a method for traffic sign detection based on fast radial symmetry transform, and the general technique is almost identical to the Hough transform. They first create a gradient magnitude image, and then threshold the output image in order to discard points with low magnitudes. Each remaining non-zero gradient element votes for a possible circle center. The vote is placed at the closest pixel to this point. Figure 2.5 shows different votes cast by a gradient element $g(p)$ when searching for different shapes at a given radius [5]. Another example of using the Hough Transform is found in [37]. This method is suitable only for circular traffic signs.



Figure 2.5: Different votes cast by a gradient element $g(p)$ when searching for different shapes at a given radius [5]

Edge Detection

Edge detection is one of the basic techniques of image processing for finding the boundaries of different objects in images. The main application of edge detection in areas such as image processing and computer vision is segmentation when images can be divided into areas corresponding to various objects.

The Canny edge detector was used by [38] because this method preserves contours, which is necessary for shaped based traffic sign detectors. Aoyagi and Asakura [39] proposed a system that used Gaussian and Laplacian filter.

Corner-point detection is yet another technique for finding shapes of interest. Paulo and Correia [40] approached traffic sign detection based on the Harris corner detector. They identified the triangular and square shapes by finding the corners of each ROI, using the Harris corner detection. The existence of corners is then tested in six different control areas of the ROI [40]. Harris corner detector was first introduced in 1988 by Chris Harris and Mike Stephens [41]. This corner detector is popular because it is simple and fast. Furthermore, the Harris corner detector is scale, rotation and illumination variation invariant.

Neural Networks

Another approach for shape detection is the use of trained Neural Networks (NN). Zhu *et al.* [42] used a neural network for detecting triangular signs. [43] is another example of using NN for traffic sign shape detection.

Template Matching

Another common method in image processing and pattern recognition is template matching. Template Matching is a high-level machine vision technique that allows identifying the parts of an image (or multiple images) that match a given image pattern [44]. This method is also used in TSDR systems. For instance, researchers in [45], [46], and [47] all proposed image matching techniques for TSDR.

Gavrila [6] proposed a shaped-based system, based on distance transforms and template matching. The first stage of this technique is finding the edges in the original images. In the second stage, a distance transform (DT) image is created. Figure 2.6 depicts these two stages. In order to find the traffic signs inside the images, matching a template against the DT image is performed. Matching a template with the edge image is also possible, but the advantage of template matching with DT image is that the resulting similarity measure is much smoother.

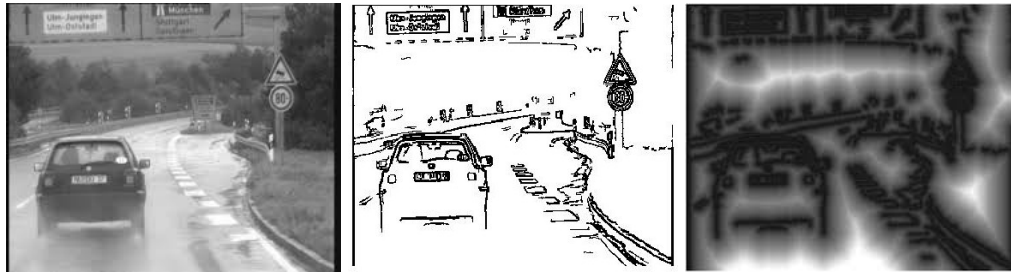


Figure 2.6: *Building the distance transform image. from left to right: original image, edge image and the distance transform image [6].*

Gradient features

In recent years, Histogram of Oriented Gradients (HoG) features [48] were used in many contributions for traffic sign feature extraction, such as [49] and [50]. We also used this method due to its robustness for object detection applications. The HoG detector was first introduced for the task of pedestrian detection. This method begins with dividing the imagery into a set of blocks. The HoG is then computed for each block. HoG has different parameters that can influence the accuracy of the detection stage. HoG features possess several advantages in comparison to other shaped-base methods, including high accuracy, scale invariance, local contrast normalization, and coarse spatial sampling.

Other methods

Authors in [51] and [52] defined a operator named Local Contour Pattern (LCP) and used it to detect circular and triangular road sign shapes. They defined this operator as a measure over binary images to find local geometrical structures. Jiang et al. [53] used mathematical morphology in order to filter noise out and extract the morphological feature of inner shapes of traffic signs, in a way that is invariant to translation.

2.1.3 Detection Based on Hybrid Methods

As previously discussed, Both color-based and shape-based methods have some advantages and disadvantages. Thus, a combination of the two reviewed methods is also prevalent among researchers. Many sign detection systems include a color segmentation stage followed by some kind of shape extraction stage. The work of Fang *et al.* [43] includes using the hue as a color feature and an edge detector method for shape feature extraction. [54] and [55] are other examples of integrating color-based methods with the ones based on shape analysis. Mathias *et al.* [56] used Integral Channel Features (ICF) for sign detection which was first established by Dollar *et al.* [57] for the task of pedestrian detection. Integral channel features include a combination of different orientation channels, color space channels, and gradient magnitude channels. Sekanina and Torresen [58] proposed an algorithm for detection of Norwegian speed limit signs. They used the RGB color space for the color segmentation stage and template matching for locating speed limit signs. A converted version of the RGB color space and a Laplacian of Gaussian (LoG) edge detector have been used by [59] for detection of triangular traffic signs.

2.2 Recognition Methods

The next stage in TSDR systems is sign recognition, which ascertains whether the detected candidate is an actual traffic sign or not. Different ways exist for recognizing the detected sign candidates. We introduce the most widely used methodologies in this Section.

We have reviewed Neural Networks for the task of sign detection. This method is also prevailing for the task of traffic sign recognition. In [60], six Neural Networks have been trained with the back propagation method for six different classes of road signs. Their proposed algorithm was tested on 200 different traffic signs. Authors in [61] also selected six categories of road signs for recognition. These are stop, yield way, no left turn, no right turn, speed limit 60, and speed limit 90 signs. In order to recognize the exact type of signs, they used a series of one to one architectural Multi Layer Perception (MLP) Neural Networks. They took advantage of Resilient Back Propagation (RP)[62] and Scaled Conjugate Gradient (SCG) [63] algorithms for training their neural networks. The average accuracy of system using RP classifiers is 91%. And the system using SCG classifiers also resulted in a 91% recognition rate.

Another multi-layer network was used in [64] to classify the extracted candidate road signs. They performed the classification based on a developed and tested feed-forward MLP neural network classifier. They also used The Conjugate Gradient Descent optimization algorithm in order to achieve better results. They obtained an average classification rate of 91%. Another example of using Neural Networks for sign classification is found in [36]. They implemented two different back propagation Neural Networks for recognition of circular and triangular road signs. One of the networks identifies the triangular signs while the other one identifies the circular ones. The authors considered the speed-limit and end-of-speed-limit, stop, forbidden-overtaking, and end-of-forbidden-overtaking signs for the circular ones, while for triangular signs, the yield way sign, and dangerous curves signs were selected. Finally, they reported a 98.5% recognition rate for speed limit signs and a 97.2% recognition rate for warning signs. Aoyagi and Asakura [39] also proposed a traffic sign recognition module using Neural Net-

works. They classified their recognition category into three classes: the speed sign, other traffic sign, and not a traffic sign. They used 324 input layer units, 15 hidden layer units, and 3 output layer units. Back propagation is also used in their learning process. It is worth taking into consideration that Neural Network based methods have some drawbacks such as, over fitting, high computational cost, and difficulty of fine-tuning the performance [65].

Another method for traffic sign classification is using the Joint Transform Correlation (JTC). JTC is one of the main techniques in pattern recognition. In a JTC, the unknown input scene and the known reference image are displayed side-by-side in the input plane known as the input joint image which is Fourier Transformed (FT)[64]. Then, these FT patterns are joined with each other in order to build a pattern called the Joint Power Spectrum (JPS). Based on JPS values between a test image and a template one, computing a correlation is possible. If two correlation peaks are detected, a match is found. Authors in [64] used this method. Miura *et al.* [29] carried out identification of traffic signs by a normalized correlation-based pattern matching using a traffic sign image database. They applied this cross correlation between test images and template images. This normalized cross-correlation made the system invariant to lighting conditions. They reached the identification rate of 46%. Perez and Javidi [66] presented a non-linear correlator that performs many correlations between an input scene and different reference targets. According to them, non-linear filters provide invariance to distortions of the target, noise robustness, and rejection of background noise.

S. Lafuente-Arroyo *et al.*[67] proposed a new method for recognition of traffic signs based on Support Vector Machines (SVMs) and Distance to Borders (DtBs). The patterns generated by the vectors describe the distances to borders of the objects candidate to be traffic signs. They have tested their algorithm on more than 300 image sequences. Based on the reported results, their system is robust under different states. Authors in [68] combined Distance from Centre (DfC) vectors with distance to border vectors for making the features for the input to the linear SVM. According to their explanation, DfC is the distance from the centre of the blob to the external edge of the blob. The main contribution of their technique is its robustness to translation, rotation, and scale. They achieved a classification success rate of 89.96% for red triangular signs and 92.54% for red circular signs.

A new method for traffic sign classification based on grayscale images has been presented in [69]. In the first stage the RGB images were converted to grayscale by using color thresholding and histogramming techniques. Then they used a method called ring partitioning in order to divide the image into several ring areas and match the image by computing the histogram for every ring. They show that the ring partitioning method performs best in the matching of rotated, shadowed, occluded, (including illumination changes) images. They achieved a 93.9% matching rate. Authors in [70] presented a classification method using Principal Component Analysis (PCA). The system was able to achieve a performance of 99.2% correct classifications of road signs. It was implemented so as to be invariant to image translation, rotation and scaling.

A classification model using SVMs was developed by authors in [71]. They performed recognition for seven categories of traffic sign shapes and five categories of speed limit signs. Binary images and Zernike moments features were used for presenting the data to the SVM for training and testing. They achieved a 100% accuracy in sign shapes classification and a 99% accuracy on speed limit signs classification. Another method that is also based on SVMs has been proposed by [72]. The authors used a cascade of SVM classifiers that were trained using

Histogram of Oriented Gradient (HoG) features for recognition. The accuracy of their system for white road signs is 89.2%, and 92.1% for color signs.

Another proposed method for traffic sign recognition uses the Scale Invariant Feature Transform (SIFT)[73]. This method finds local invariant features in a given image and matches these features to the features of images that exist in the training set. The training image that gives the maximum number of matches is the target sign. To increase the performance of the system, the authors added color and orientation information to the recognition process. For instance, the orientations of the matched features are also computed and checked. And if two orientations have a difference larger than a defined threshold, the match is considered to be false, else it is taken as a true match. They obtained a 99.3% recognition rate by recognizing 149 out of 150 traffic sign images correctly. In [74], the SURF (Sped-Up Robust Features) [75] descriptor was selected instead of SIFT. There are a few differences between SIFT and SURF. For example, for scale spacing, SIFT approximates a Laplacian of Gaussian (LoG) with a Difference of Gaussian (DoG), but SURF approximates LoG with a Box Filter. To assess the performance of the proposed method, 200 images were selected. They obtained a recognition accuracy of 92.7%.

Optical Character Recognition (OCR) is also a method used in the current literature for traffic sign recognition through text recognition of traffic signs. The OCR approach is used in [76]. The main disadvantage of this method is that it is not able to recognize traffic signs without text and only the signs with text fit to this method. Therefore, there are several signs without text that cannot be recognized.

Chapter 3

Traffic Signs Dataset

3.1 Background

Another principal element of any traffic sign detection and recognition system is the availability of a dataset. In order to train and test a detector for identifying an object based on different features and classifiers, we need to have access to a large number of samples of that object.

During the past few years, a number of research groups have worked on creating traffic sign datasets for the task of detection, recognition, and tracking. Some of these datasets are publicly available for use by the research community. The following table introduces some of these datasets.

Dataset	Description	Paper
GTSRB	Country of origin: Germany Number of images: 50000 Classes: 43	[77]
GTSDB	Country of origin: Germany Number of images: 900	[77]
BTSCB	Country of origin: Belgium Number of images: 10000 Classes: 62	[78]
BTSDB	Country of origin: Belgium Number of images: 7000	[78]
STSD	Country of origin: Sweden Number of images: more than 20000	[79]
Stereopolis	Country of origin: France Number of images: 847 Classes: 10	[80]
LISA	Country of origin: United States Number of images: 6000 Classes: 46	[81]

Table 3.1: PUBLICLY AVAILABLE TRAFFIC SIGN DATASETS

Among these datasets, the German Traffic Sign (GTS) dataset and the Belgium Traffic Sign

(BTS) dataset are the two large and famous datasets that can be used for detection and recognition. The German Traffic Sign Recognition Benchmark (GTSRB) and Belgium Traffic Sign Classification Benchmark (BTSCB) can be used for the task of both detection and recognition.

3.2 Differences Between North American and European Signs

Most of these traffic sign datasets have been recorded and created in European countries. Furthermore, it is worth taking into consideration that most of the detection and recognition algorithms that have been proposed in the past two decades are based on European traffic signs. Due to the lack of open-sourced and publicly available datasets for USA and Canada, most researchers tested their developed methods on European datasets. Figure 3.1 shows some European traffic signs.



Figure 3.1: *Samples of European traffic signs.*

The traffic signs in different countries have different colors, backgrounds, and shapes. For example, the design of traffic signs in North America is entirely different from European traffic signs. The main difference between the two traffic systems is speed limit signs.



Figure 3.2: *Samples of speed limit signs. left: Europe right: Canada.*

The rectangular shape of North American speed limit signs is completely different from circular European speed limit signs. The other main difference is related to the color of signs. In Europe, the speed limit signs are white with a red circular border, but in North America, these signs are totally white with a black rectangular border. Figure 3.2 represents samples of speed limit signs in Europe and North America. There are also other traffic signs in North America with the same shape and color as speed limit signs, including many regulatory and High Occupancy Vehicle (HOV) signs.

While the problem of detection and recognition of European traffic signs can be considered solved, there is still much to accomplish for North American traffic signs. As we reviewed the main proposed methods for detection and recognition of European signs in the previous Chapter, we found out that some researchers showed impressive and accurate detection results on the GTSDDB. But the story of detection of North American sign is different. If we apply the same method used for detection of European signs on North American signs, we will not get the same detection accuracy. This is due to the different designs of the two traffic systems. For example, color thresholding which was used extensively in segmentation of all types of European signs is not suitable for segmentation of North American speed limit signs. Since speed limit signs have no color to make them stand out from the background, color segmentation methods are unable to distinguish them from other objects. Therefore, we can consider most of the North American traffic signs hard cases for detection. This leads us to conclude that the challenge of detection of traffic signs is not completely solved.

3.3 RoadLab Traffic Sign Dataset

In this section, we introduce our own traffic sign dataset. Based on a previous project in our laboratory [82], several driving sequences were recorded by our RoadLab experimental vehicle around the city of London, ON, Canada. This vehicle is equipped with different hardware systems including stereo camera rigs. The front stereo rig which is mounted on the roof of the vehicle records stereo images with a resolution of 320 by 240. This dataset includes different types of traffic signs such as warning signs, temporary condition signs, information and direction signs and regulatory signs which makes it a complete dataset for the task of detection and recognition. Therefore, we used our own dataset for training and testing the proposed method. Figure 3.3 delineates some of the recorded sequences of this dataset.

There are also other vision systems installed on our instrumented vehicle such as a remote eye tracking system and RoadLAB stereoscopic vision systems. In the next Chapter we provide more information on these vision systems and their intended use.



Figure 3.3: *Samples of image sequences in our dataset.*

Chapter 4

TSDR System Architecture Design

In this Chapter, we focus on providing detailed information about the three main stages of our TSDR system and introducing related concepts behind our method. As mentioned before the three stages of this system are:

1. Establishing the field of view of the driver.
2. Detecting the traffic signs using a linear Support Vector Machine (SVM) and Histogram of Oriented Gradients (HOG) features.
3. Recognition of detected candidates using color information and Scale Invariant Feature Transform (SIFT) matching [83].

4.1 Establishing the Field of View of the Driver

All of the current TSDR systems are performing based on detection and recognition of traffic signs within image sequences. The main difference in our TSDR system from others is our ability to establish the visual attentional field of the driver and then detecting the signs inside that area.

In this work, our main focus lies on detection and recognition of signs within the visual field of the driver. In order to relate the 3D Line-of-Gaze (LoG) of the driver to the depth map obtained by the forward stereo camera system and derive the 3D Point-of-Gaze (PoG), we used a technique proposed in our laboratory [84] to identify the 3D PoG in absolute coordinates expressed in the frame of reference of the vehicle. Figure 4.1 depicts the remote eye tracking system and the stereoscopic vision system. By intersecting the visual cone of attention with the plane perpendicular at the 3D PoG along the 3D LoG of the driver, we are able to form a circle in 3D space which represents our region of interest. Note that this circle becomes a 2D ellipse once projected onto the imaging plane of the stereoscopic sensors. Figure 4.2 displays the driver visual cone of attention. According to [85] the radius of the cone is approximately 13° .

Given the eye position $\mathbf{e} = (e_x, e_y, e_z)$ and the 3D PoG $\mathbf{g} = (g_x, g_y, g_z)$, provided in the frame of reference of the eye tracker, we compute the radius of the 3D circular gaze area r onto the



Figure 4.1: *Physical configuration (left): Remote eye tracking system. (right): RoadLAB stereoscopic vision system.*

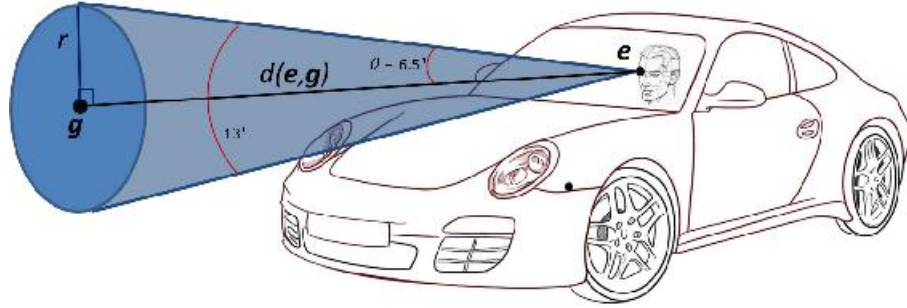


Figure 4.2: *Depiction of the diver attentional gaze cone*

plane perpendicular to the LoG and containing the PoG, according to:

$$r = \tan(\theta)d(\mathbf{e}, \mathbf{g}) \quad (4.1)$$

The area of gaze is defined as the range to extend 6.5 degrees for each pitch and yaw of gaze direction [85]. Therefore, $\theta = 6.5^\circ$, and

$$d(\mathbf{e}, \mathbf{g}) = \sqrt{(e_x - g_x)^2 + (e_y - g_y)^2 + (e_z - g_z)^2} \quad (4.2)$$

is the Euclidean distance between eye position and the PoG. The parametric form of the 3D circle can be written as:

$$S(\phi) = (X(\phi), Y(\phi), Z(\phi))^T = \mathbf{g} + r(\cos(\phi)\mathbf{u} + \sin(\phi)\mathbf{v}) \quad (4.3)$$

where $\mathbf{u} = (u_x, u_y, u_z)$ and $\mathbf{v} = (v_x, v_y, v_z)$ are the two unit orthogonal vectors within the plane and ϕ is an angle varying between 0 and 2π .

The eye tracking system and the forward stereoscopic system each operate in their own frame of reference, and we need to transform the 3D circle corresponding to the visual attentional area of the driver into the frame of reference of the stereo system of the vehicle. To accomplish this, we used a cross-calibration process developed earlier in our laboratory to

compute the parameters of the rigid transformation (a translation T and a rotation matrix R) between the eye tracker and the stereoscopic system [84]:

$$S'(\phi) = R^T(S(\phi) - T) \quad (4.4)$$

The last step is the projection of the 3D circle onto the stereo imaging plane in image coordinates, and is computed with the following equation:

$$s'(\phi) = \frac{1}{Z'(\phi)}KS'(\phi) \quad (4.5)$$

where K is the matrix of intrinsic parameters of the scene stereo system. Our region of interest is now defined and figure 4.3 depicts a few attentional gaze areas for chosen frames.



Figure 4.3: Attentional gaze areas projected onto the forward stereo scene system of the vehicle.

4.2 Detection Phase

In this Section, we provide more information regarding the algorithms used in the detection stage such as HOG features and SVM classifier.

4.2.1 HOG Descriptors

It is widely accepted that Histogram of Oriented Gradients (HOG) features are suitable for rigid object detection. These features are based on evaluating well-normalized local histograms of image gradient orientations in a dense grid. This method has been used for the first time by

Navneet Dalal and Bill Triggs in 2005 [48]. They have used it for the task of pedestrian detection. Since that year, many researchers have used HOG descriptors for different applications in the field of image processing and computer vision. These kinds of features have an excellent performance compared with others [48]. We can divide the implementation of HOG descriptors algorithm into four different stages, namely image normalization, gradient calculation, orientation binning, and normalization and descriptor blocks. We describe each stage of the HOG features extraction in detail. Figure 4.3 displays the algorithm implementation scheme.

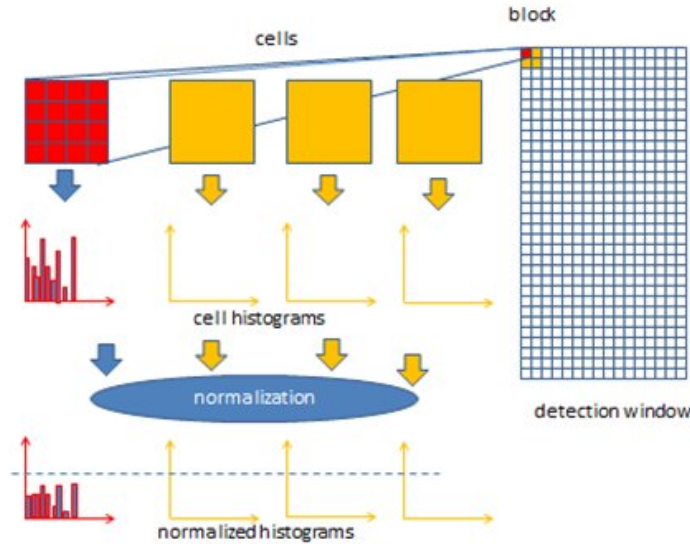


Figure 4.4: *HOG descriptor implementation scheme.*

Image Normalization

The first step in extracting HOG features is to apply an image equalization in order to reduce the impacts of inconsistent illumination. This is performed using Gamma compression by computing the square root or the log of each color channel.

Gradient Calculation

The second stage consists of computing the first order image gradients. The goal of this step is capturing some contour and texture information as well as providing more resistance to inconsistent lighting conditions. The most common method for computing the image gradients is using 1-D centred, point discrete derivative mask in both the horizontal and vertical directions. For instance, the gradient of pixel (x, y) can be defined as follows:

$$G(x) = H(x + 1, y) - H(x - 1, y) \quad (4.6)$$

$$G(y) = H(x, y + 1) - H(x, y - 1) \quad (4.7)$$

where $H(x, y)$ is the gray value of pixel (x, y) , and $G(x)$ and $G(y)$ are the gradients. Different methods may use second order image derivatives.

Orientation Binning

The third step is the creation of the cell histograms. Each pixel calculates a weighted vote for an edge orientation histogram channel based on the orientation of the gradient element centered on it, and the votes are accumulated into orientation bins over local spatial regions that we call cells [48]. These cells can be either radial or rectangular. The histogram channels are evenly spread over $0^\circ - 180^\circ$ (unsigned gradient) or $0^\circ - 360^\circ$ (signed gradient). According to [48], increasing the number of orientation bins up to about 9 can improve the performance of the system remarkably.

Normalization and Descriptor Blocks

The last step consists in the normalization of histograms. Normalization brings more invariance to illumination changes, shadowing, and edge contrast. Since the range of gradient strength variations is large, an effective local contrast normalization seems to be critical for improving the performance of the system. The proposed normalization scheme by Dalal *et al.* [48] is based on grouping cells into larger spatial blocks and contrast normalizing each block separately. Therefore, the final HOG descriptor is a vector consisting of all cell histograms. The two main block geometries are rectangular (R-HOG) and circular (C-HOG). R-HOG blocks are generally represented by three parameters: the number of cells in each block, the number of pixels in each cell, and the number of channels in each cell histogram. Dalal and Triggs [48] found 8X8 pixels cells per block with 9 histogram channels to be optimal parameters for the task of pedestrian detection.

Despite some similarities between R-HOG blocks and SIFT descriptors, they are used differently. R-HOG blocks are computed in dense grids at a single scale without dominant orientation alignment, whereas SIFT descriptors are computed at a sparse set of scale invariant key points, rotated to align their dominant orientations, and used individually [48]. The other block shape is circular. C-HOG descriptors can be found in two different variants: those with a single circular central cell and those with an angularly divided central cell. C-HOG blocks are represented by four parameters: the number of angular and radial bins, the radius of the center bin, and the expansion factor for the radius of additional radial bins [48]. For achieving an optimal performance two radial bins and four angular bins are necessary. Dalal and Triggs evaluated four different block normalization methods for R-HOG and C-HOG blocks:

$$\text{L1-sqrt: } f = \frac{v}{\sqrt{(\|v\|_1 + e)}} \quad (4.8)$$

$$\text{L1-norm: } f = \frac{v}{(\|v\|_1 + e)} \quad (4.9)$$

$$\text{L2-norm: } f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \quad (4.10)$$

$$(4.11)$$

$$\text{L2-hys: } f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \quad (v \leq 0.2) \quad (4.12)$$

where v is the non-normalized feature vector, e is a small constant, and f is the normalization factor. Based on experiments performed by Dalal and Triggs, L2-hys, L2-norm, and L1-sqrt schemes provide reliable performance, with L1-norm being less reliable. In general, all of these block normalization methods showed notable improvement over non-normalized blocks.

After block normalization, all the histograms can be combined in one feature vector for use in the window classifier. Figure 4.5 illustrates examples of HOG features extracted from a few traffic signs.



Figure 4.5: Examples of HOG features extracted from traffic signs.

4.2.2 Support Vector Machine (SVM)

This section introduces the Support Vector Machine (SVM) [86] learning algorithm. SVM is among the best supervised learning algorithms in machine learning. Supervised learning refers to a machine learning task of inferring a function based on labelled training data. A supervised learning algorithm evaluates the training data in order to infer a function for mapping new, unseen examples. SVM is a discriminative classifier formally defined by a separating hyperplane. That is to say, given labelled training data, this method produces an optimal hyperplane which classifies new examples. This classification method is highly accurate and extremely fast which is a good choice for large amount of training data.

More precisely, SVM is a binary classifier that separates two different classes by a subset of data samples called Support Vectors. Pedestrian detection is a well-known application of SVM in the field of computer vision and machine learning. Dalal *et al.* [48] proposed a robust pedestrian detection method based on a SVM classifier and HOG descriptors. We show an example of classifying two types of samples in a two-dimensional space, as illustrated in figure 4.6. Squares and circles represent two different types of samples.

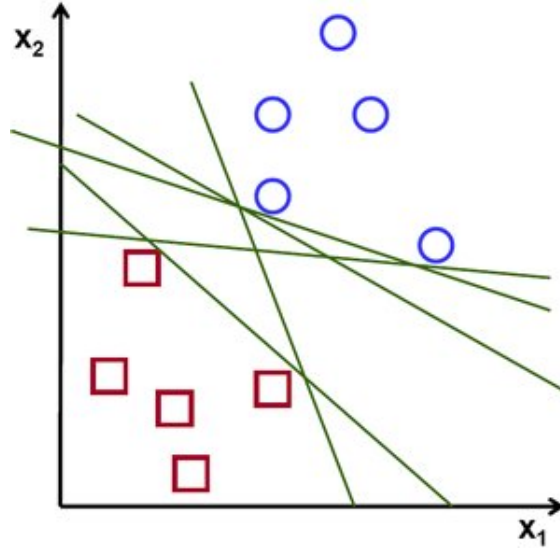


Figure 4.6: . 2D classification sample

Let's suppose that our data has the form $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$. Each pair represents the labels -1 or +1; -1 for negative samples and +1 for positive samples. A hyperplane used as a decision function can separate these two classes. Any hyperplane can be defined as a set of points \vec{x} satisfying:

$$\vec{w} \cdot \vec{x} + b = 0 \quad (4.13)$$

where \vec{w} is the normal vector to the hyperplane, and b is called the bias.

Our example from Figure 4.6 shows how the squares and circles are separated by a number of lines (or planes, in higher dimensions). We can select two parallel hyperplanes to separate our data. The distance between the two hyperplanes is made to be the greatest possible. These hyperplanes can be defined as follows:

$$\vec{w} \cdot \vec{x} + b = 1 \quad (4.14)$$

$$\vec{w} \cdot \vec{x} + b = -1 \quad (4.15)$$

The distance between these two hyperplanes is $\frac{2}{\|\vec{w}\|}$. In order to maximize the distance between the planes we need to minimize $\|\vec{w}\|$. We also need to look for the maximum-margin hyperplane which is the hyperplane that lies halfway between the two defined hyperplanes. We can also define a criterion to find out which hyperplane is the optimal one. If a line passes too close to the points, it will be sensitive to noise and it will not be a good choice for classifying the two classes. Therefore our aim is to find the line which is as far as possible from all points. The SVM algorithm is based on finding the hyperplane which gives the largest minimum distance to the training data. In SVM's theory, this distance is named margin. consequently, the optimal separating hyperplane maximizes the margin of the training samples. Figure 4.7 shows this optimal hyperplane.

Another important matter in finding the optimal hyperplane is preventing the data points from falling into the margin. By using the following constraints, we make sure the data points

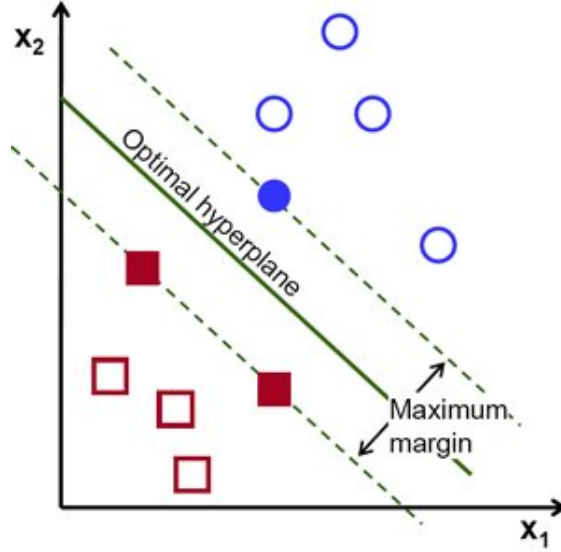


Figure 4.7: . *Optimal hyperplane*

lie on the correct side of the margin:

$$\vec{w} \cdot \vec{x}_i + b \geq 1 \quad \text{if } y_i = 1 \quad (4.16)$$

$$\vec{w} \cdot \vec{x}_i + b \leq -1 \quad \text{if } y_i = -1 \quad (4.17)$$

These constraints can be combined as:

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 \quad \text{for all } 1 \leq i \leq n \quad (4.18)$$

By joining these equations we obtain the following optimization problem:

$$\text{Minimize } \|\vec{w}\| \quad \text{subject to } y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, \quad \text{for } i = 1, \dots, n \quad (4.19)$$

The above constraints guarantee that the maximum margin classifier classifies each sample correctly when the data is linearly separable. But in those cases that the data is not linearly separable, a loss function can be introduced. That is to say, a greater margin can be achieved by enabling the classifier to misclassify some data points. To allow a certain degree of fault tolerance, slack variables are introduced:

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i \quad (4.20)$$

where $\xi_i \geq 1$ for $i = 1, \dots, n$

If slack variables are between 0 and 1 (that is, $0 \leq \xi \leq 1$) it means that data points are in the margin (margin error) and if they are greater than 1 ($\xi > 1$) some data points are misclassified. In order to penalize misclassification and margin errors, a new term is introduced which is called the soft margin constant C . The optimization problem thus becomes:

$$\text{Minimize } \frac{1}{2} \|\vec{w}\|^2 + C \sum_i \xi_i \quad \text{subject to } y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \text{for } i = 1, \dots, n \quad (4.21)$$

This formulation was presented by Cortes and Vapnik [86]. The soft-margin constant has an important effect on the decision boundary as it can be used for penalizing misclassification and margin errors. A large value for C assigns a large penalty to margin errors. As illustrated in Figure 4.8 (left), the two points that are closest to the hyperplane impact its orientation. This results in a hyperplane that is close to other data points. When C has a smaller value (Figure 4.8 (right)), those points become margin errors.

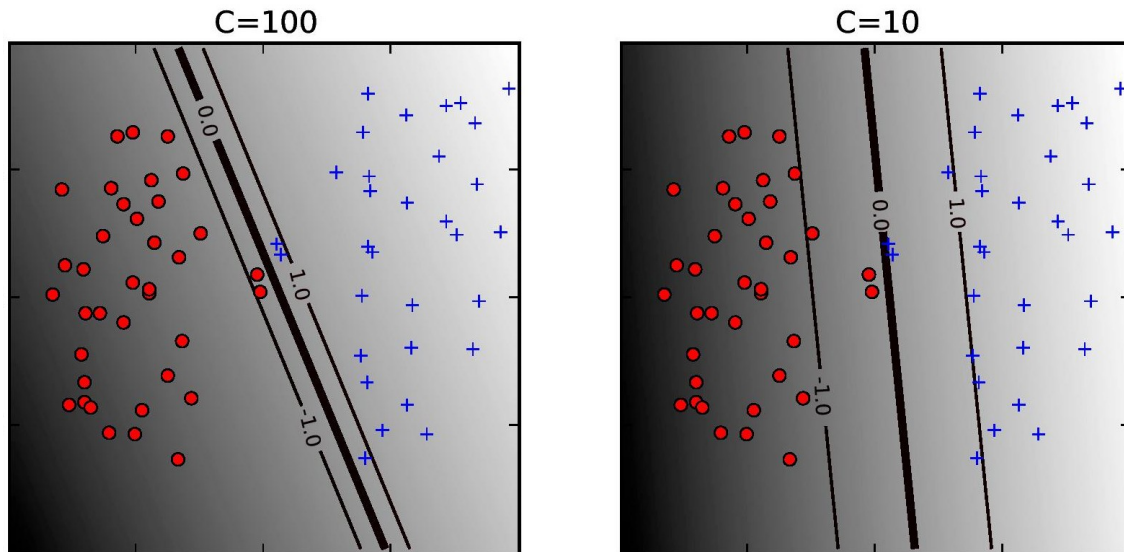


Figure 4.8: . The effect of the soft margin constant C .

4.2.3 Implementation Details

In this Section, we introduce advanced learning methods and other techniques related to our traffic sign detector. Any detection system needs a set of positive and negative samples to be used in the training process. We selected 1000 images as positive training samples. Additionally, we increased the number of positive images by adding the flipped, rotated, and translated versions of original samples which results in better detection performance. Figure 4.9 shows the image visualization of the complete list of object training examples and their average.

The initial number of negative samples (2000) are selected from the training images with the traffic signs regions cropped out. For boosting the performance of the learned classifier, we use an advanced and popular learning method called hard negative mining. So far, SVM has been learned using a small sample of negative images. Nevertheless, in essence, every single image region that does not contain a traffic sign can be considered as a negative sample. There are too many samples to be used in practice, but we are only looking for key negative samples which can be extracted from the hard negative mining stage. We train the SVM in an iterated procedure. In each iteration, the detector is applied to a new image without traffic signs. Then, we add the resulting false positives (hard negatives) to the training set for the next iteration. We performed this process for 5 iterations. Finally, the classifier is provided with more key negative samples which helps make the detection more robust. Figure 4.10 illustrates the extraction



Figure 4.9: **(top):** *Positive samples* **(bottom):** *Average image*.

of hard negative samples from a traffic sign-free image.

Once hard negative mining and training have been performed, we evaluate the model on test data. We use a sliding window over multiple scales. In order to eliminate redundant detections, a Non-Maximum Suppression (NMS) algorithm is used. NMS keeps the highest-scoring detection and removes any other detection whose overlap is greater than a threshold. We used the pascal overlap score [87] so as to establish the overlap ratio between the two bounding boxes. It is computed as:

$$a_0 = \frac{\text{area}(B_1 \cap B_2)}{\text{area}(B_1 \cup B_2)} \quad (4.22)$$

where a_0 is the overlap ratio. B_1 and B_2 are the two bounding boxes.

4.3 Recognition Phase

This stage of our technique phase is concerned with identifying the exact type of detected traffic sign candidates. Recognition uses SIFT features and color information. We introduce

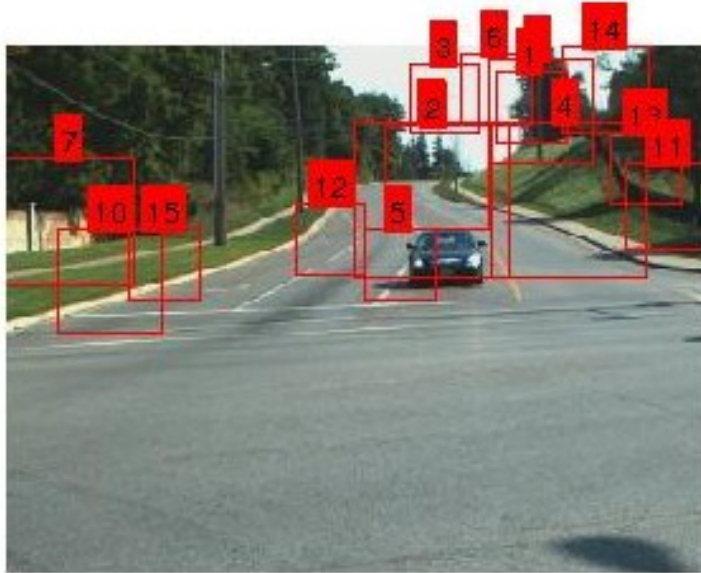


Figure 4.10: *Hard negative mining.*

the SIFT algorithm and then proceed to describe the combination of SIFT detectors and color information to build a robust traffic sign recognizer.

4.3.1 Scale Invariant Feature Transform

In order to provide a feature description for any object inside an image, we need to extract interesting points on the object. By using a feature descriptor that can be extracted from a training image, we can detect and locate the object inside an image where many other objects exist. A reliable recognition system should be able to extract features from training images even under inconsistent illumination changes and noise. Another important factor is that the relative positions between these extracted features should not vary from one image to another. For instance, if the four corners of an object are considered as features, they would work without regard to the position of the object. Moreover, features located on flexible objects would not work if the physical geometry changes between images. However, the SIFT algorithm detects a large number of features from images, which lessens the effect of errors caused by local variations. The main advantage of SIFT is its ability to identify objects under partial occlusion. Furthermore, SIFT feature descriptors are invariant to orientation, uniform scaling, and illumination changes. This Section summarizes the technical details of SIFT algorithm. We divide our description of the SIFT algorithm into the three following parts: we present the notion of scale space, then the SIFT detector followed by the SIFT descriptor.

Scale Space

The SIFT detector and descriptor are built from the Gaussian scale space of an image. The following equation introduces the Gaussian scale space function, in 1D for simplicity:

$$G(x; \sigma) = g_\sigma * I(x) \quad (4.23)$$

where I is the 1D signal, x is its coordinate, σ is the scale parameter, and g_σ is a Gaussian kernel. The SIFT algorithm uses the Difference of Gaussians (DoG) to create a scale space. This scale space is the scale derivative of the Gaussian scale space. It is defined by:

$$D(x, \sigma(s, o)) = G(x, \sigma(s + 1, o)) - G(x, \sigma(s, o)) \quad (4.24)$$

where o is the octave index and s is the sub-level index. These two parameters can be mapped to the corresponding scale σ by the formula:

$$\sigma(o, s) = \sigma_o 2^{o + \frac{s}{S}} \quad (4.25)$$

$$o \in o_{min} + [0, \dots, O - 1] \quad (4.26)$$

$$s \in [0, \dots, S - 1] \quad (4.27)$$

where σ_o is the base scale level, O is the number of octaves, S is the number of sub-levels, and o_{min} is the index of first octave. According to the above equations, the Difference of Gaussian (DoG) is obtained as the blurring of an image with two different σ . This procedure is performed for different octaves of the image in the pyramid. An example is shown in Figure 4.11.

SIFT Detector

SIFT keypoints include a set of points at the local extrema of the DoG scale-space. The selection of these points is guided by the following parameters:

- Local extrema threshold: if the value $|G(x, \sigma)|$ of local extrema is less than this threshold, then they are rejected.
- Local extrema localization threshold: if the local extrema are in a low contrast area, they are discarded.
- Boundary points removal: keypoints too close to the boundaries of the image are rejected.

The identification of keypoints is performed by comparing each pixel in the DoG images to its eight neighbors at the same scale as well as nine pixels in next scale and nine pixels in previous scale. If the pixel value is the maximum or minimum among all existing pixels (local extrema), it will be selected as a potential keypoint. Figure 4.12 demonstrates this process.

After the extraction of keypoints, the index $(x_1; x_2; s)$ is fitted to the local extremum by quadratic interpolation and a threshold on the intensity $D(x; \sigma)$, and a test on the peakedness of the extremum is applied in order to reject weak points or points on edges [88]. Any low contrast and edge keypoints are eliminated at this stage.

Following this, an orientation is allocated to every keypoint to provide invariance to image rotation. By taking a neighborhood around the location of a key point, the gradient magnitude

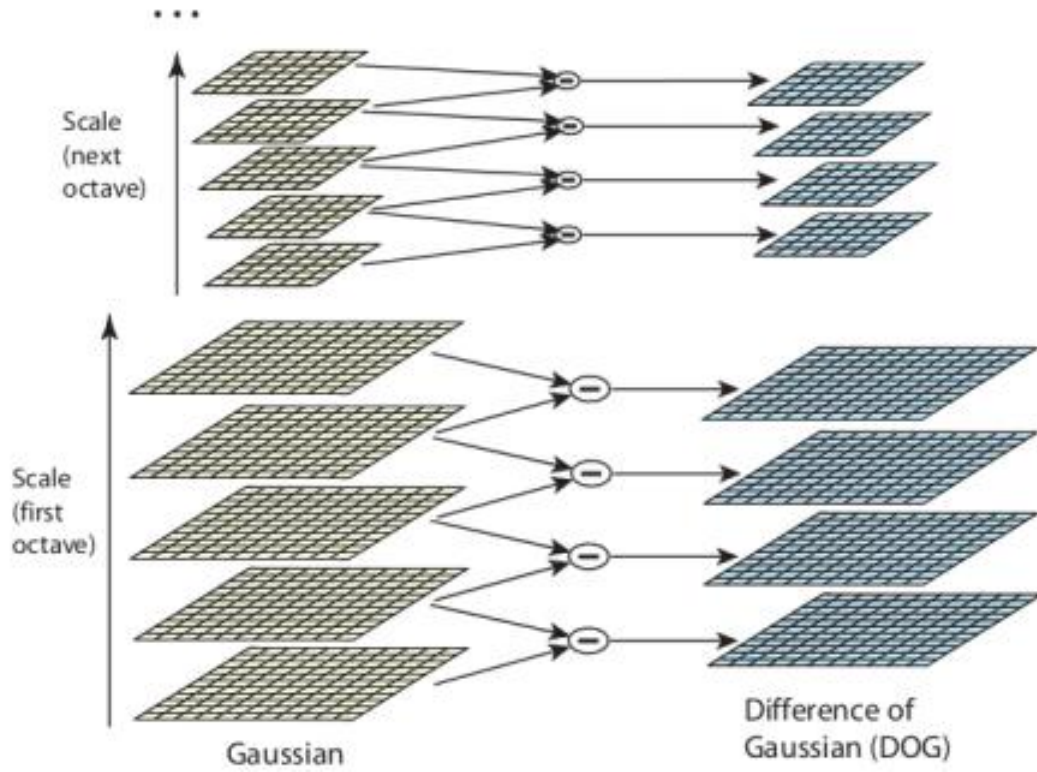


Figure 4.11: *Gaussian pyramid.*

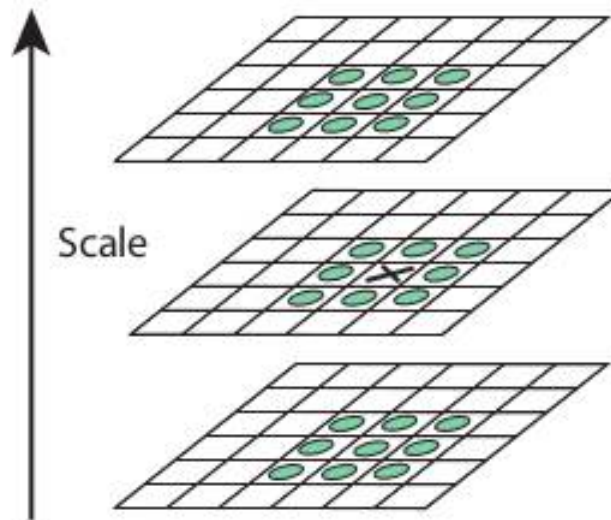


Figure 4.12: *Searching an image for local extrema over scale and space.*

and direction are computed in that area. Then an orientation histogram is created with 36 bins covering 360 degrees. The histogram is weighted both by the magnitude of the gradient and a Gaussian window centered on the key point and of deviation 1.5σ [88]. Only the highest peaks are chosen to calculate the orientation. The global maximum and any peak with a value above

80% of the highest peak is also taken for computation of the orientation.

SIFT Descriptor

The SIFT descriptor of a keypoint is an interpolated histogram of the gradient orientations and locations in a region surrounding it [88]. The parameters of the descriptor are as follows:

- The magnification factor: spatial bins in the histogram have a size of $m\sigma$ where σ is the scale of the frame.
- The number of spatial bins
- The number of orientation bins

Figure 4.13 depicts the SIFT descriptor layout. In the previous part, the orientations were assigned to keypoints. This ensured invariance to image rotation, location, and scale. In order to create a keypoint descriptor, a 16 by 16 neighborhood around the keypoint is chosen. Then it is divided into 16 sub-blocks of 4 by 4 size. 8 bin orientation histograms are created for each sub-block. Therefore, a total of 128 bin components are available which form the components of a vector. Next step is to normalize this vector in order to increase invariance to affine changes. In order to decrease the effect of non-linear illumination, a threshold of 0.2 is used and the vector is normalized again. SIFT descriptors are invariant to affine changes to some extent. It is important to note that the 16 by 16 neighborhood around the keypoint or 4 by 4 size of sub-blocks are default values of the SIFT algorithm and it has been proved that using these default values will result in generating more unique descriptors.

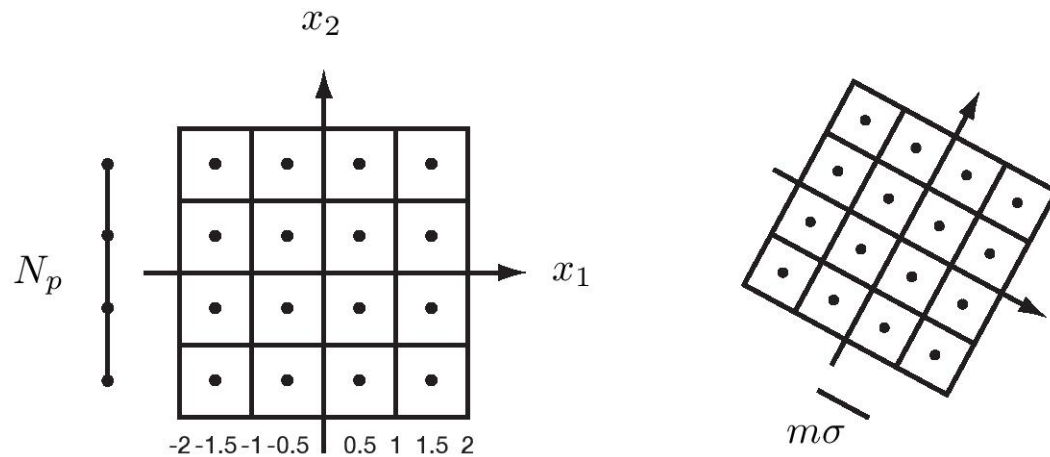


Figure 4.13: *The SIFT descriptor layout. The size of a spatial bin is $m\sigma$.*

4.3.2 Implementation Details

Since the SIFT algorithm is able to find distinctive keypoints that are invariant to location, scale, and rotation, and is robust to affine transformations, it is a judicious choice for object recognition.

First, the candidates found in the detection phase are scaled to the size of the template signs. We gathered a full set of template traffic signs to use in the recognition stage. Figure 4.14 depicts a few examples of images in the template database.



Figure 4.14: *Examples of template signs.*

The following step is to devise a way of integrating color information in order to increase the performance of the SIFT matching. We calculate the color difference between the candidate target and the template signs. We have tested HSV and lab color spaces and finally used HSV color model due to its relative insensitivity to noise and illumination. Moreover, The HSV color space outperformed Lab color space in showing more consistent color difference between the candidate target and the template signs. Figure 4.15 displays extracted H , S , and V values for the template and detected signs. Traffic signs include a wide variety of colors, and it is possible to differentiate them by using all the components of the HSV color space. Hence, we compute H , S , and V values of the detected candidate sign and all template signs. Then, we obtain the average of all values based on a defined mask. We then create the delta images:

$$\begin{aligned}\delta H &= H_{\text{channel}} - H_{\text{standard}} \\ \delta S &= S_{\text{channel}} - S_{\text{standard}} \\ \delta V &= V_{\text{channel}} - V_{\text{standard}}\end{aligned}$$

where the H_{channel} , S_{channel} , and V_{channel} are the averaged HSV color parameters of the detected candidate, and the other three are averaged HSV color parameters of the template sign. The final value which is the color difference between the two images is obtained as:

$$\delta f = \sqrt{\delta H^2 + \delta S^2 + \delta V^2} \quad (4.28)$$

where Δf is comprised between 0 and 1. If it is close to 1, we can conclude that there is a large color difference between the two images, and vice versa. We define a threshold, whose value

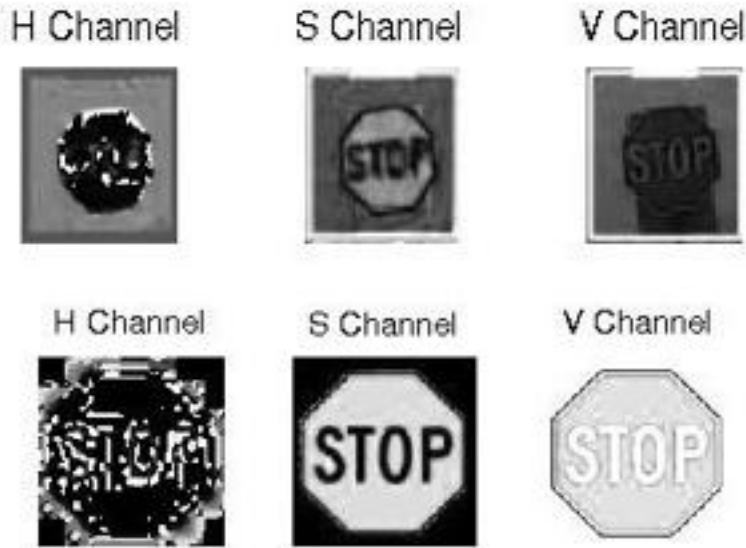


Figure 4.15: *HSV color space images a) (top): detected sign b) (bottom): template sign.*

was experimentally obtained, to determine whether the two images are color-wise similar or not. If δf is less than the defined threshold, then we select the corresponding template sign for feature matching. Hence, we perform feature matching only between the candidate image and those images in the template database whose colors are similar to the detected sign. This method removes some of the potential false matches and significantly improves performance. Once all the sample images that have a similar color to the detected target are found, we perform feature matching between those ones and the candidate image. The SIFT algorithm is used for feature matching. Figure 4.16 shows an illustration of the DoG scale space in different octaves and scales for traffic signs. After key point extraction, we calculate the descriptors of each key point. We then proceed with matching the candidate descriptors. In order to do this, we define a threshold: a descriptor d_1 is matched to a descriptor d_2 if the distance between them multiplied by a threshold is not greater than the distance of d_1 to all other descriptors. The value of this threshold was experimentally obtained and it is equal to 1.5.

The RANdom SAMple Consensus (RANSAC) algorithm [89] is also used to discard possible outliers. This is an iterative method for estimating a mathematical model from a data set that contains outliers. The basic stages of this algorithm are summarized as follows [90]:

- Randomly select the minimum number of points needed to determine the model parameters
- Solve for the parameters of the model
- Determine how many points from the set of the selected points fit the model parameters with a predefined tolerance ϵ .

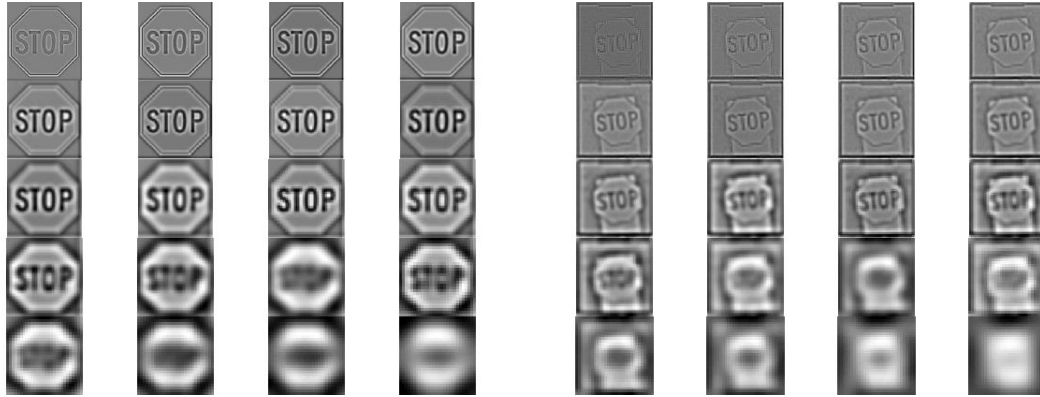


Figure 4.16: *DoG scale space images (left): template sign (right): detected sign.*

- Re-estimate the model parameters if the fraction of inliers over the total number of points in the set is lower than a predefined threshold τ .
- Repeat these steps until an adequate confidence level for the estimated model parameters is attained.

In order to filter out outliers during training, a small set of samples has been used to train a homography model. Then the samples which are within the error tolerance of the homography model are determined. These samples are considered as inliers. If the number of inliers is the largest found so far, we keep the current inlier set. This process is repeated for a number of iterations and returns the model with smallest average error among the generated models.

Chapter 5

Experimental Results

To prove the effectiveness of our framework, the proposed method was tested on 3500 frames. Among these frames, 1806 traffic signs appeared which were manually annotated. As mentioned before, these sequences were recorded with the RoadLAB experimental vehicle. The size of the recorded images is 320 by 240. While our main focus is on detection and recognition of signs within the visual field of the driver, we also performed sign detection outside this area in order to provide the driver with a response about a possibly unseen traffic sign. If the four coordinates of the bounding box are inside the driver’s field of view, we can conclude that the driver has seen the sign. Otherwise, the driver has missed the sign. Both *SEEN* and *MISSED* feedbacks are given to the driver right after the detection and recognition of signs.

In the first part of this Chapter, we present our traffic sign detection results. Recognition results are provided in the second part. Finally, some screen shots of detected and recognized traffic signs both inside and outside of the driver’s visual attentional field are provided.

5.1 Traffic Sign Detection Results

In order to assess the accuracy of sign detection we report two numbers: the Detection Rate (*DR*) and the False Positive per Frame (*FPPF*), defined as:

$$DR = \frac{TP}{TP + FN} \quad (5.1)$$

$$FPPF = \frac{FP}{F} \quad (5.2)$$

where *TP* is the number of true positives, *FN* is the number of false negatives, *FP* is the number of false positives, and *F* is the number of image frames. Table 5.1 reports on the performance of traffic sign detection. As mentioned before, the traffic sign detector was applied on 3500 images. Some images have no traffic signs and some have more than one traffic sign. Table 5.1 also demonstrates the performance of the proposed detector. Figure 5.4 depicts the information provided in Table 5.4.

Another method used for visualizing the performance of our system is the Receiver Operating Characteristics (ROC) curve. ROC curves are commonly used in medicine, radiology, biometrics, and also in machine learning and data mining research. This curve outputs True

	<i>DR</i>	<i>FPPF</i>
Proposed method	0.84	0.04

Table 5.1: DETECTION RATE AND FALSE POSITIVE PER FRAME

Number of signs	1806
Number of detected signs	1517
Number of missed signs	289

Table 5.2: SUMMARY OF TRAFFIC SIGN DETECTION RESULTS

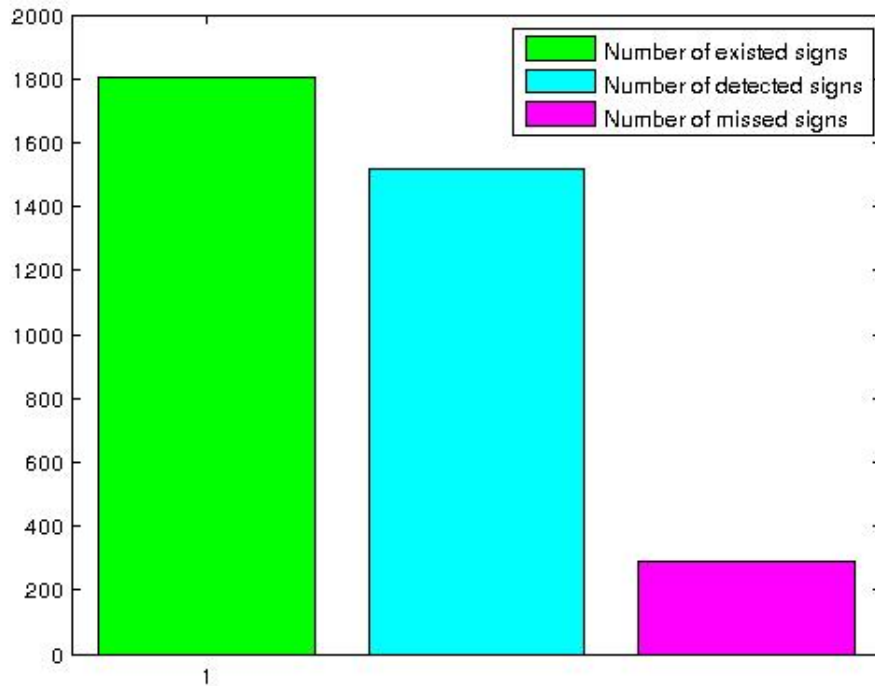


Figure 5.1: Traffic sign detection rate

Positive Rate (TPR) which is known as sensitivity, hit rate and recall against False Positive Rate (FPR) which is known as false alarm rate:

$$TPR = \frac{\text{Positives Correctly Classified}}{\text{Total Positives}} \quad (5.3)$$

$$FPR = \frac{\text{Negatives Incorrectly Classified}}{\text{Total Negatives}} \quad (5.4)$$

ROC curves can be understood as two-dimensional graphs in which *TPR* is represented on the *Y* axis and *FPR* is represented on the *X* axis.

The best possible prediction model represents 100% sensitivity (no false negatives) and 100% specificity (no false positives). That is to say, a point that reaches the coordinate (0,1) can

be called perfect classification. We can measure the accuracy by defining the Area Under the Curve (AUC). This area is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming positive ranks higher than negative) [91]. Figure 5.2 shows the area under the curve for two different classifiers. As can be noted, classifier B has a greater area under the curve and as a result a better performance. But sometimes a classifier with a greater AUC can have a lower performance in comparison to a classifier with a lower AUC. For example even though classifier B generally performs better than classifier A, classifier A, at false positive rate greater than 0.6, has a slightly better performance. A rough model for classifying the accuracy of tests based on AUC is given below.

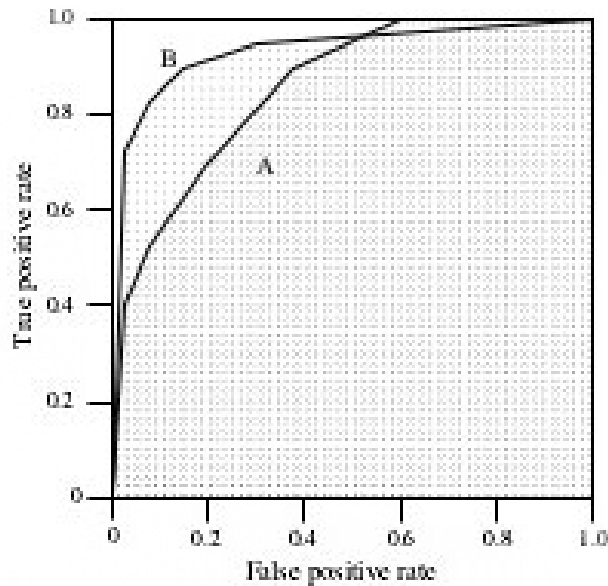


Figure 5.2: *The AUC of two classifiers*

- $0.9 \leq AUC \leq 1.0 \implies Excellent$
- $0.8 \leq AUC < 0.9 \implies Good$
- $0.7 \leq AUC < 0.8 \implies Fair$
- $0.6 \leq AUC < 0.7 \implies Poor$
- $0.5 \leq AUC < 0.6 \implies Fail$ Another important factor in plotting a ROC curve is defining a threshold. In a classification model, the classifier needs to determine a threshold value for separating the boundary between classes. Most classifiers produce a score which can be thresholded to decide on the classification. Any threshold applied to a dataset is going to produce four parameters with different values for: True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN) (see figure 5.3).

In our case, we obtain a scoring value for each detected bounding box. After normalization, the scoring values vary between 0 and 1. These scoring values can be used as threshold parameters for plotting the ROC curve. We used the ROC curve in order to obtain an optimal threshold among all scoring values for our classifier which maximizes the true positive rate, while minimizing the false positive rate. Different thresholds have been used and we found

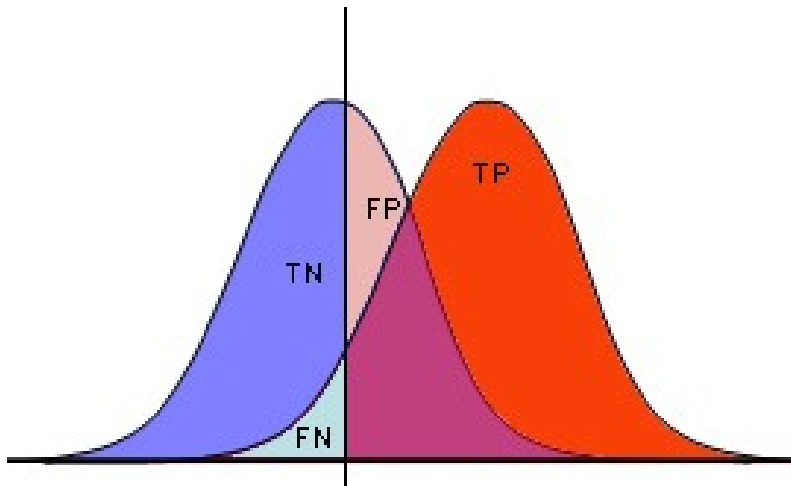


Figure 5.3: *true positives, false positives, true negatives and false negatives*

that a threshold equal to 0.56 gave us the best trade off between the true positive rate and the false positive rate. Figure 5.4 depicts this ROC curve.

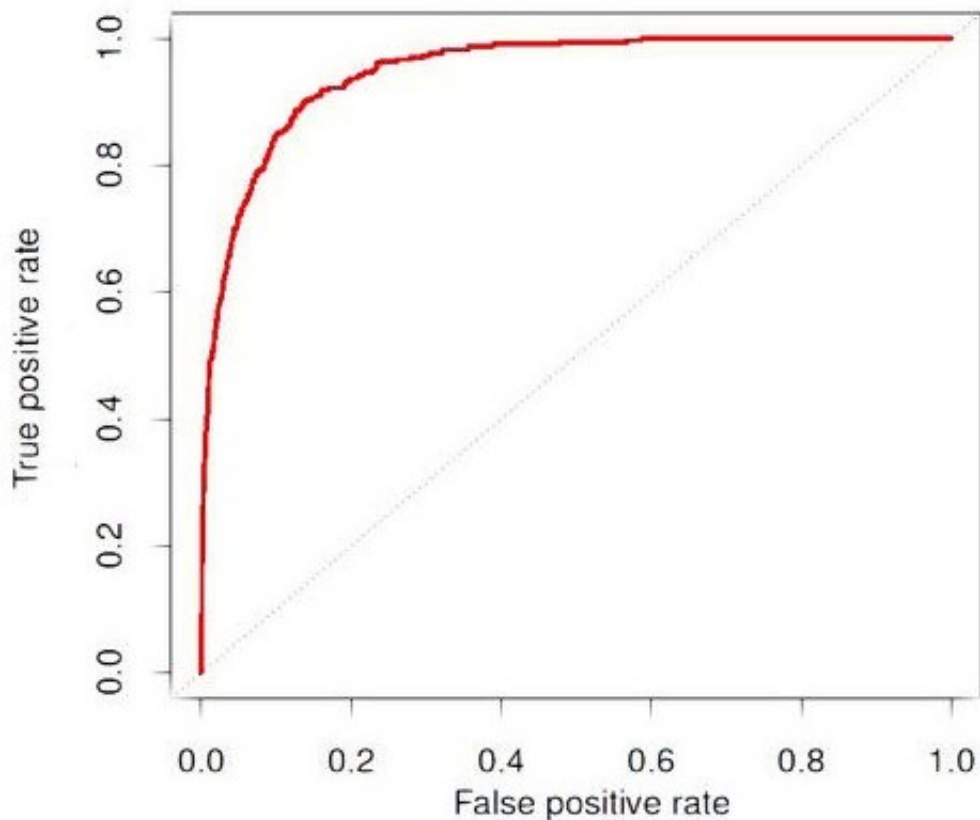


Figure 5.4: *ROC curve with threshold=0.56*

It is important to note that due to significant differences between European and North American traffic signs, providing a comparison between our proposed method and other methods

may not be meaningful. But, we just introduce a state-of-the-art detection method on European traffic sign detection. Authors in [56] have evaluated their method on the German Traffic Sign (GTS) and the Belgium Traffic Sign (BTS) datasets. Both benchmarks are split in three main super classes based on their color and shape: (M) mandatory, (D) danger, and (P) prohibitory. The following table summarizes the accuracy of their method by providing the area under curve (AUC) of their detector.

Dataset	mandatory	danger	prohibitory
GTS	96.98	100.00	100.00
BTS	94.79	96.40	86.51

Table 5.3: STATE OF THE ART DETECTION RESULTS ON EUROPEAN TRAFFIC SIGNS

Additionally, limited work has been done on the detection and recognition of North American traffic signs. For instance, authors in [92] proposed a detector only for stop, warning, and speed limit signs and provided separate accuracies for each category. In contrast, our detector detects most traffic signs, including warning, temporary conditions, information and direction, and regulatory signs

5.2 Traffic Sign Recognition Results

The recognition phase is built atop the detection phase in order to confirm the detected candidates and determine the exact type of traffic sign. Table 5.4 demonstrates the accuracy of our proposed method for recognition of traffic signs. The traffic sign recognition results are also shown in figure 5.5.

Number of detected signs	1517
Number of recognized signs	1348
Number of falsely recognized signs	169
Accuracy rate	88.9

Table 5.4: SUMMARY OF TRAFFIC SIGN RECOGNITION RESULTS

Another standard evaluation method for traffic sign recognition is the confusion matrix. In the field of machine learning and computer vision, the confusion matrix is a table that is commonly used for evaluating the performance of a classification model on a set of test data when the true values are known. Each column of this table represents the predicted values while each row represents the true values. Figure 5.6 provides an example of the confusion matrix for a two-class classifier. The meaning of the entries is explained below:

- a: the number of correct predictions that an instance is positive
- b: the number of incorrect predictions that an instance is negative
- c: the number of incorrect predictions that an instance is positive
- d: the number of correct predictions that an instance is negative

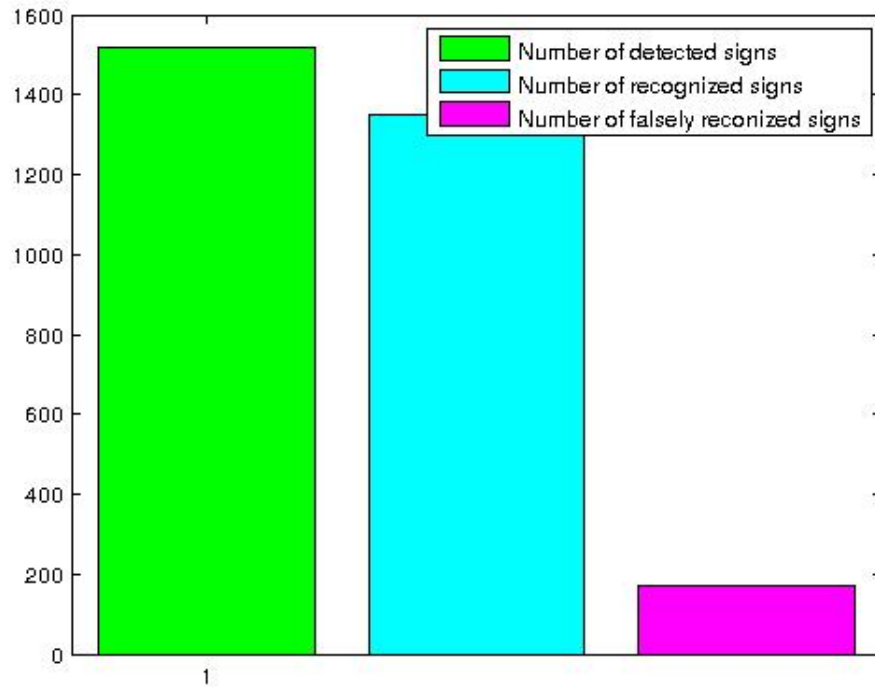


Figure 5.5: *Traffic sign recognition rate*

	Predicted	
	Positive	Negative
Actual True	a	b
Actual False	c	d

Figure 5.6: *Confusion matrix example*

Figure 5.7 provides the confusion matrix for our recognition method.

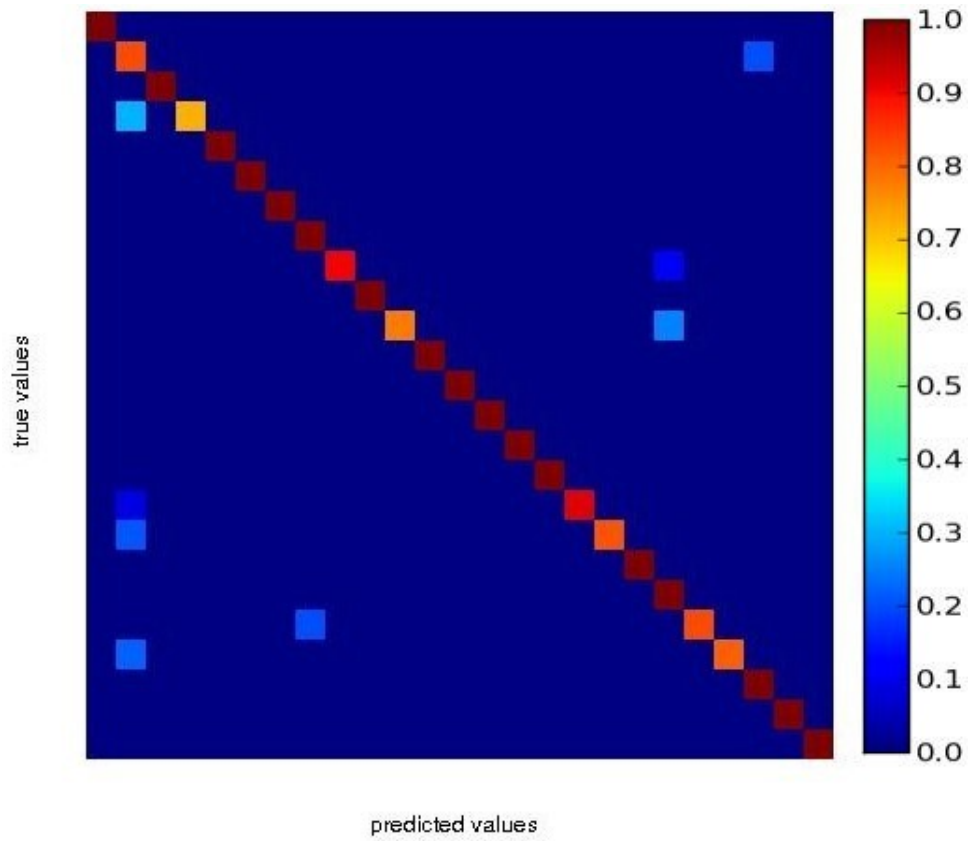


Figure 5.7: *Confusion matrix with accuracy of 88.9%*

5.3 Output Images

In this section we provide some output images from our TSDR system. The images include detected and recognized signs inside and outside of the visual field of drivers as well as *SEEN* and *MISSED* feedbacks. As it is shown in the following images, in some cases the traffic signs do not appear within the visual field of the driver and it probably means that the sign was not acknowledged. Therefore, a *MISSED* feedback is given in these cases. On the other hand, if traffic signs appear within the visual field of the driver, it also probably means they can be seen and a *SEEN* feedback is given.

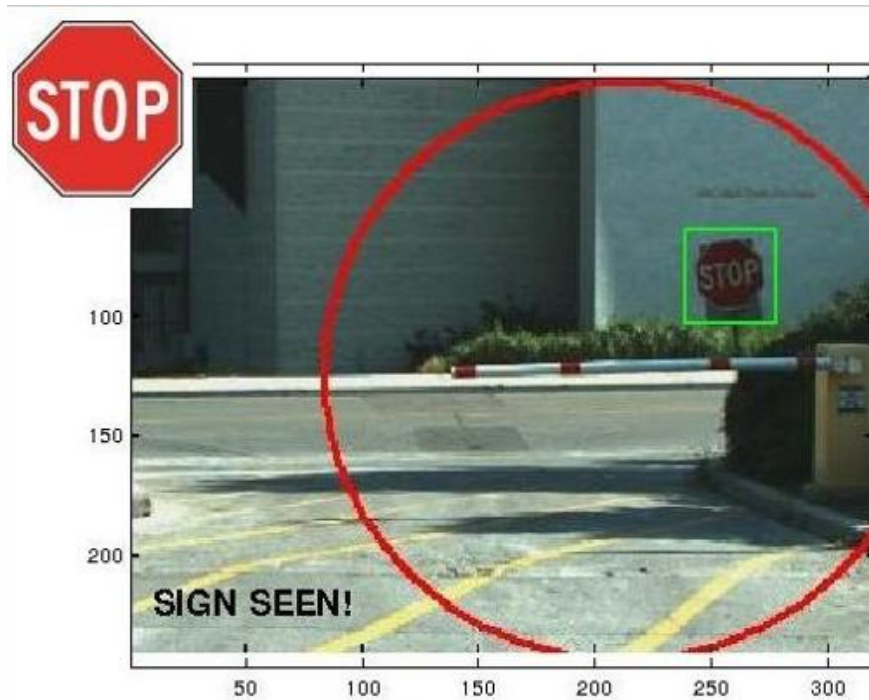


Figure 5.8: *Detection and recognition of the stop sign. The driver has seen the sign.*

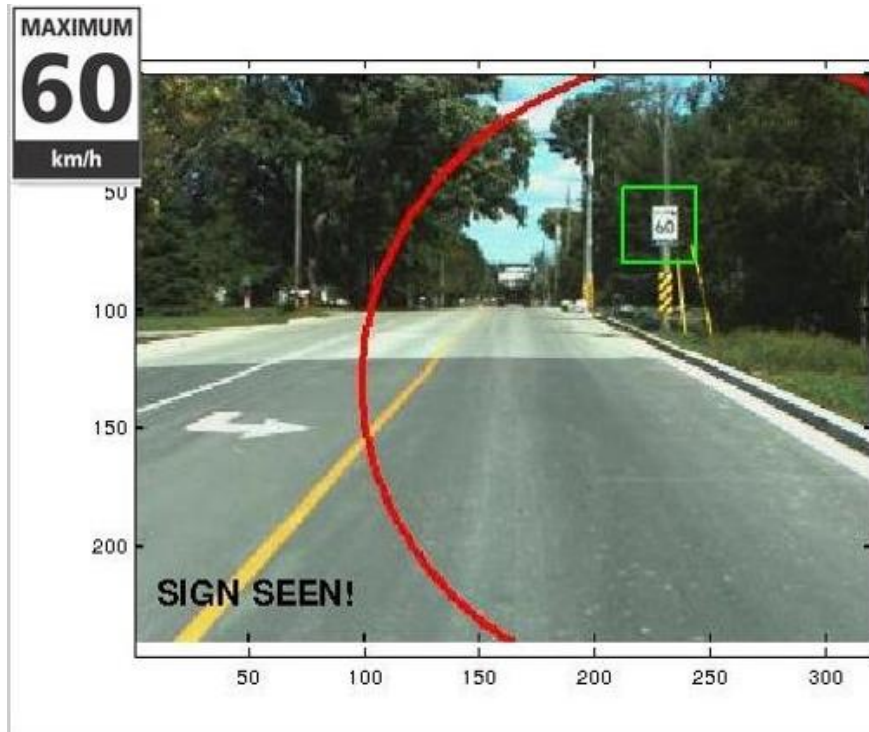


Figure 5.9: *Detection and recognition of the speed limit sign. The driver has seen the sign.*

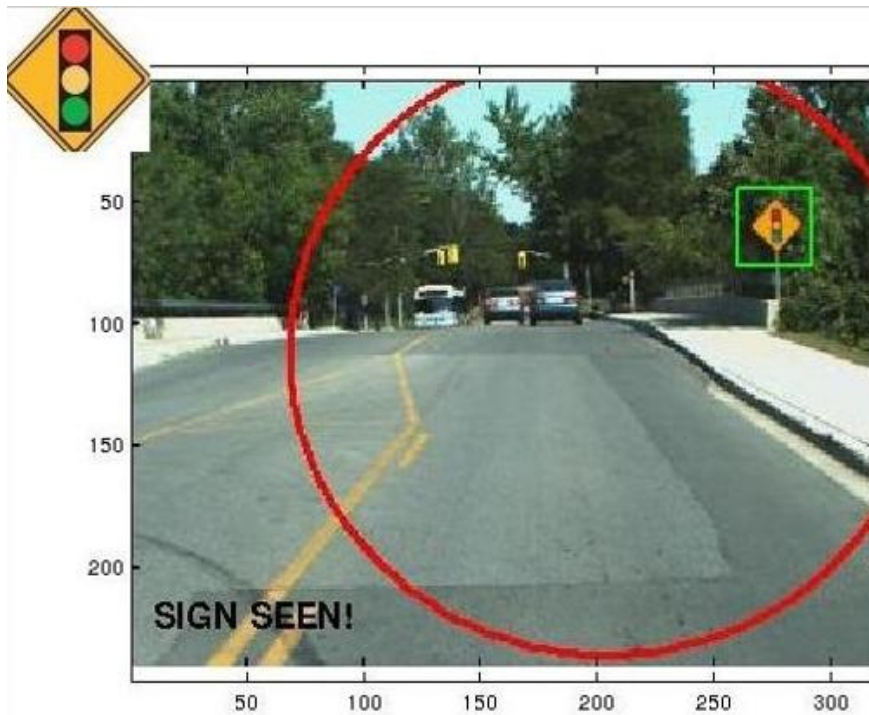


Figure 5.10: *Detection and recognition of the traffic light ahead sign. The driver has seen the sign.*

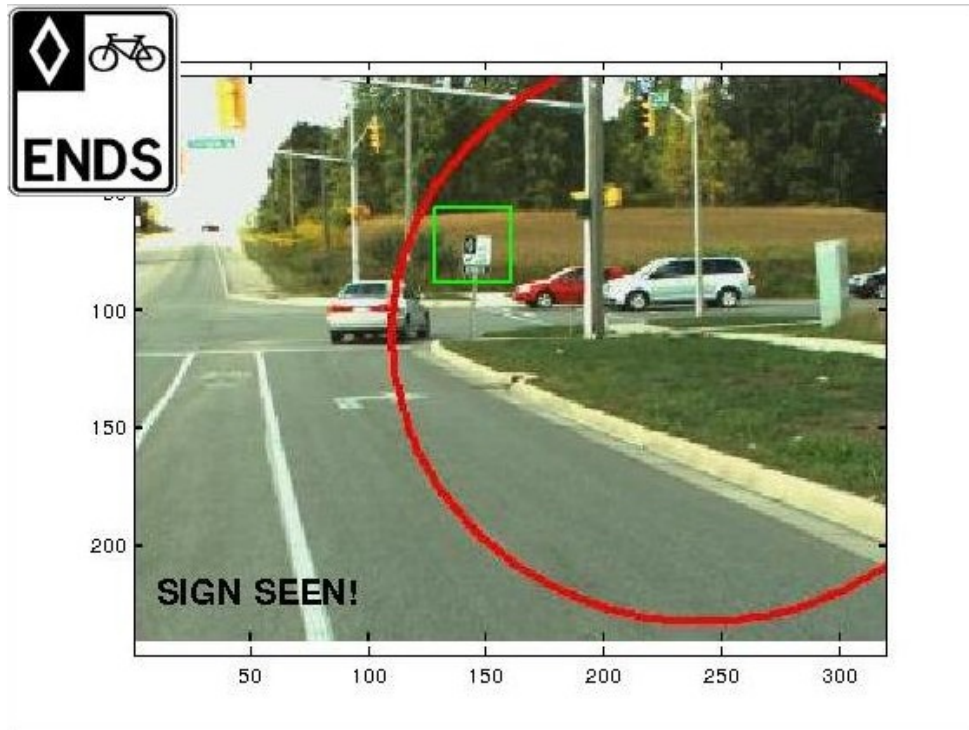


Figure 5.11: *Detection and recognition of the bike lane ends sign. The driver has seen the sign.*



Figure 5.12: *Detection and recognition of the school zone sign. The driver has missed the sign.*

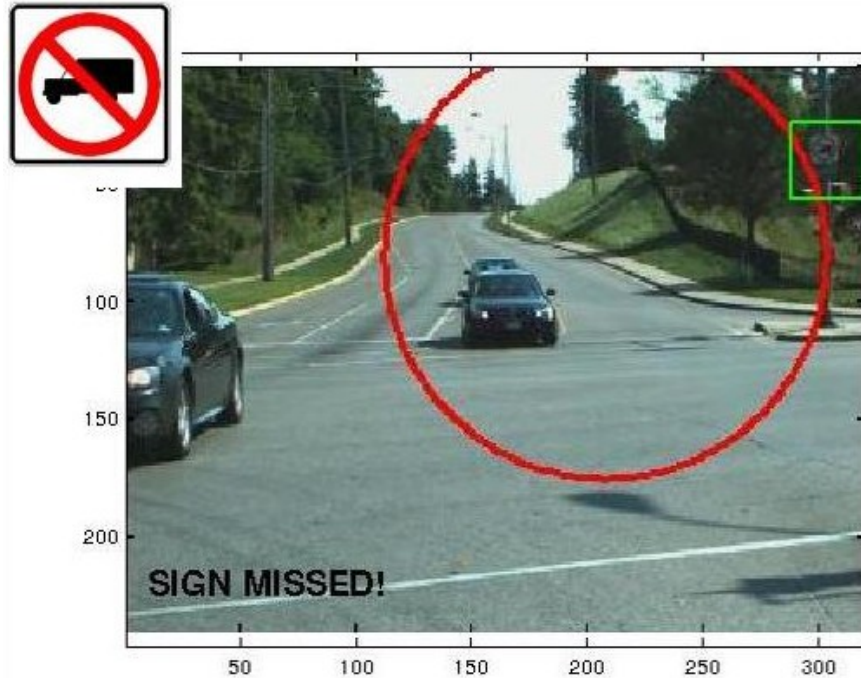


Figure 5.13: *Detection and recognition of the No heavy trucks permitted on this roadway sign. The driver has missed the sign.*



Figure 5.14: *Detection and recognition of the keep to the right of traffic island sign. The driver has missed the sign.*

Chapter 6

Conclusion and Future Work

In this thesis, we presented an efficient traffic sign detection and recognition system based on North American traffic signs. The RoadLab traffic sign dataset has been created used for training and testing the proposed method. Moreover, by using an in-vehicle, non-contact infra-red binocular gaze tracking system installed in our experimental vehicle, we were able to identify the exact gaze area of the driver. We were also able to infer whether the driver was likely to have seen the sign or not based on computing the intersection of the detected bounding box and driver gaze area. While the detection and recognition of traffic signs have come far in European countries, little attention has been given to North American signs. We tried to rectify this dissimilarity by proposing a system for detection and recognition of North American traffic signs.

This method can be extended to include other possible objects drivers attend to or encounter such as pedestrians, cyclists, and traffic lights. By identifying all objects that are inside and outside of the attentional visual field of drivers, an advanced driver assistance system that informs the driver about those objects can be developed. Such system will be valuable for safety reasons.

Another important issue which needs to be addressed is the creation of publicly available North American traffic sign datasets. Most of the proposed TSDR systems have been evaluated on European traffic signs due to a lack of publicly available North American traffic sign datasets. Based on the differences between these two traffic systems, it is necessary to pay more attention to detection and recognition of North American traffic signs.

Bibliography

- [1] “Rgb color space.” http://photo.hanyu.iciba.com/upload/encyclopedia_2/88/95/bk_889574aebacda6bfd3e534e2b49b8028_vheBLq.jpg, Online; accessed: 2016-09-30.
- [2] 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.
- [3] “Hsv color space.” <http://infohost.nmt.edu/tcc/help/pubs/colortheory/img/cone.png>, Online; accessed: 2016-09-30.
- [4] “Ycbr color space”.” https://software.intel.com/sites/default/files/did_feeds_images/0EF01A88-F874-4ECB-B2B6-3ADC38636CD4/0EF01A88-F874-4ECB-B2B6-3ADC38636CD4-imageId=FE9BEAD5-12E5-4244-80E1-E61EB8211A76.jpg, Online; accessed: 2016-09-30.
- [5] 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.
- [6] D. M. Gavrila, “Traffic sign recognition revisited,” in *Mustererkennung*, pp. 86–93, Springer, 1999.
- [7] S. Singh, “Critical reasons for crashes investigated in the national motor vehicle crash causation survey,” 2015.
- [8] S. B. Wali, M. A. Hannan, A. Hussain, and S. A. Samad, “Comparative survey on traffic sign detection and recognition: a review,” *Przeglad Elektrotechniczny*, vol. 91, no. 12, pp. 38–42, 2015.
- [9] K. Plataniotis and A. N. Venetsanopoulos, *Color image processing and applications*. Springer Science & Business Media, 2013.
- [10] 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.
- [11] M. Bénallal and J. Meunier, “Real-time color segmentation of road signs,” in *IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, vol. 3, pp. 1823–1826, 2003.

- [12] L. Estevez and N. Kehtarnavaz, "A real-time histogrammic approach to road sign recognition," in *the IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 95–100, 1996.
- [13] S. Varun, S. Singh, R. S. Kunte, R. S. Samuel, and B. Philip, "A road traffic signal recognition system based on template matching employing tree classifier," in *International Conference on Computational Intelligence and Multimedia Applications*, vol. 3, pp. 360–365, 2007.
- [14] A. Ruta, Y. Li, and X. Liu, "Detection, tracking and recognition of traffic signs from video input," in *International IEEE Conference on Intelligent Transportation Systems*, pp. 55–60, 2008.
- [15] 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 transactions on intelligent transportation systems*, vol. 8, no. 2, pp. 264–278, 2007.
- [16] H. Gómez-Moreno, S. Maldonado-Bascón, P. Gil-Jiménez, 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.
- [17] R. J. W. R. F. Stein, "Hybrid approach for traffic sign recognition," 1993.
- [18] P. Paclík, J. Novovičová, P. Pudil, and P. Somol, "Road sign classification using laplace kernel classifier," *Pattern Recognition Letters*, vol. 21, no. 13, pp. 1165–1173, 2000.
- [19] H. Fleyeh, "Road and traffic sign color detection and segmentation-a fuzzy approach," *Red*, vol. 250, p. 207, 2005.
- [20] R. Malik, J. Khurshid, and S. N. Ahmad, "Road sign detection and recognition using colour segmentation, shape analysis and template matching," in *International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3556–3560, 2007.
- [21] H.-H. Chiang, Y.-L. Chen, W.-Q. Wang, and T.-T. Lee, "Road speed sign recognition using edge-voting principle and learning vector quantization network," in *International Computer Symposium (ICS)*, pp. 246–251, 2010.
- [22] A. Gudigar, B. Jagadale, P. Mahesh, and U. Raghavendra, "Kernel based automatic traffic sign detection and recognition using svm," in *Eco-friendly computing and communication systems*, pp. 153–161, Springer, 2012.
- [23] C. Kiran, L. V. Prabhu, K. Rajeev, *et al.*, "Traffic sign detection and pattern recognition using support vector machine," in *Seventh International Conference on Advances in Pattern Recognition (ICAPR'09)*, pp. 87–90, 2009.
- [24] W.-J. Kuo and C.-C. Lin, "Two-stage road sign detection and recognition," in *IEEE International Conference on Multimedia and Expo*, pp. 1427–1430, 2007.

- [25] S. Maldonado-Bascon, S. Lafuente-Arroyo, P. Siegmann, H. Gomez-Moreno, and F. Acevedo-Rodriguez, "Traffic sign recognition system for inventory purposes," in *IEEE Intelligent Vehicles Symposium*, pp. 590–595, 2008.
- [26] X. Qingsong, S. Juan, and L. Tiantian, "A detection and recognition method for prohibition traffic signs," in *International Conference on Image Analysis and Signal Processing*, pp. 583–586, 2010.
- [27] A. De la Escalera, J. M. Armingol, and M. Mata, "Traffic sign recognition and analysis for intelligent vehicles," *Image and vision computing*, vol. 21, no. 3, pp. 247–258, 2003.
- [28] S. M. Bascón, J. A. Rodríguez, S. L. Arroyo, A. F. Caballero, and F. López-Ferreras, "An optimization on pictogram identification for the road-sign recognition task using svms," *Computer Vision and Image Understanding*, vol. 114, no. 3, pp. 373–383, 2010.
- [29] J. Miura, T. Kanda, S. Nakatani, and Y. Shirai, "An active vision system for on-line traffic sign recognition," *IEICE TRANSACTIONS on Information and Systems*, vol. 85, no. 11, pp. 1784–1792, 2002.
- [30] W. Shadeed, D. I. Abu-Al-Nadi, and M. J. Mismar, "Road traffic sign detection in color images," in *10th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, vol. 2, pp. 890–893, 2003.
- [31] "Ycber color space." <https://software.intel.com/en-us/node/503873>, Online; accessed: 2016-09-30.
- [32] I. M. Creusen, R. G. Wijnhoven, E. Herbschleb, and P. de With, "Color exploitation in hog-based traffic sign detection," in *IEEE International Conference on Image Processing*, pp. 2669–2672, 2010.
- [33] D. Soendoro and I. Supriana, "Traffic sign recognition with color-based method, shape-arc estimation and svm," in *International Conference on Electrical Engineering and Informatics (ICEEI)*, pp. 1–6, 2011.
- [34] X. Gao, K. Hong, P. Passmore, L. Podladchikova, and D. Shaposhnikov, "Colour vision model-based approach for segmentation of traffic signs," *EURASIP Journal on image and video processing*, vol. 2008, no. 1, pp. 1–7, 2007.
- [35] H. Liu, D. Liu, and J. Xin, "Real-time recognition of road traffic sign in motion image based on genetic algorithm," in *International Conference on Machine Learning and Cybernetics*, vol. 1, pp. 83–86, 2002.
- [36] 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.
- [37] N. Barnes and A. Zelinsky, "Real-time radial symmetry for speed sign detection," in *IEEE Intelligent Vehicles Symposium*, pp. 566–571, 2004.

- [38] M. Á. García-Garrido, M. Á. Sotelo, and E. Martín-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.
- [39] Y. Aoyagi and T. Asakura, “A study on traffic sign recognition in scene image using genetic algorithms and neural networks,” in *22nd IEEE International Conference on Industrial Electronics, Control, and Instrumentation (IECON)*, vol. 3, pp. 1838–1843, 1996.
- [40] C. F. Paulo and P. L. Correia, “Automatic detection and classification of traffic signs,” in *Eighth International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS’07)*, pp. 11–11, 2007.
- [41] C. Harris and M. Stephens, “A combined corner and edge detector.,” in *Alvey vision conference*, vol. 15, p. 50, Citeseer, 1988.
- [42] Z. Shuang-dong, Z. Yi, and L. Xiao-feng, “Detection for triangle traffic sign based on neural network,” in *IEEE International Conference on Vehicular Electronics and Safety*, pp. 25–28, 2005.
- [43] 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.
- [44] N. Perveen, D. Kumar, and I. Bhardwaj, “An overview on template matching methodologies and its applications,” *IJRCCT*, vol. 2, no. 10, pp. 988–995, 2013.
- [45] 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.
- [46] M. Shneier, “Road sign detection and recognition,” in *Defense and Security Symposium*, pp. 623016–623016, International Society for Optics and Photonics, 2006.
- [47] C. Grigorescu and N. Petkov, “Distance sets for shape filters and shape recognition,” *IEEE Transactions on Image Processing*, vol. 12, no. 10, pp. 1274–1286, 2003.
- [48] 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.
- [49] 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.
- [50] F. Zaklouta and B. Stanculescu, “Warning traffic sign recognition using a hog-based kd tree,” in *IEEE Intelligent Vehicles Symposium*, pp. 1019–1024, 2011.
- [51] F. Parada-Loira and J. L. Alba-Castro, “Local contour patterns for fast traffic sign detection.,” in *Intelligent Vehicles Symposium*, pp. 1–6, 2010.
- [52] I. Landesa-Vázquez, F. Parada-Loira, and J. L. Alba-Castro, “Fast real-time multiclass traffic sign detection based on novel shape and texture descriptors,” in *3th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1388–1395, 2010.

- [53] J. Gangyi, Z. Yi, and C. T. Young, “Morphological skeleton analysis of traffic signs on road,” in *EEE International Conference on Systems, Man, and Cybernetics*, vol. 1, pp. 70–75, 1996.
- [54] J.-T. Oh, H.-W. Kwak, Y.-H. Sohn, and W.-H. Kim, “Segmentation and recognition of traffic signs using shape information,” in *International Symposium on Visual Computing*, pp. 519–526, Springer, 2005.
- [55] L.-W. Tsai, J.-W. Hsieh, C.-H. Chuang, Y.-J. Tseng, K.-C. Fan, and C.-C. Lee, “Road sign detection using eigen colour,” *IET computer vision*, vol. 2, no. 3, pp. 164–177, 2008.
- [56] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, “Traffic sign recognition how far are we from the solution?,” in *The International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2013.
- [57] P. Dollár, Z. Tu, P. Perona, and S. Belongie, “Integral channel features,” 2009.
- [58] L. Sekanina and J. Tørresen, “Detection of norwegian speed limit signs.,” pp. 337–340, 2002.
- [59] H.-M. Yang, C.-L. Liu, K.-H. Liu, and S.-M. Huang, “Traffic sign recognition in disturbing environments,” in *International Symposium on Methodologies for Intelligent Systems*, pp. 252–261, Springer, 2003.
- [60] P. Medici, C. Caraffi, E. Cardarelli, P. P. Porta, and G. Ghisio, “Real time road signs classification,” in *IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, pp. 253–258, 2008.
- [61] Y.-Y. Nguwi and A. Z. Kouzani, “Automatic road sign recognition using neural networks,” in *The IEEE International Joint Conference on Neural Network Proceedings*, pp. 3955–3962, 2006.
- [62] M. Riedmiller and H. Braun, “A direct adaptive method for faster backpropagation learning: The rprop algorithm,” in *IEEE International Conference On Neural Networks*, pp. 586–591, 1993.
- [63] M. F. Møller, “A scaled conjugate gradient algorithm for fast supervised learning,” *Neural networks*, vol. 6, no. 4, pp. 525–533, 1993.
- [64] 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.
- [65] J. J. Abukhait, “A discriminative imaging-based framework for road sign condition assessment using local features and svm classifiers,” 2012.
- [66] E. Perez and B. Javidi, “Nonlinear distortion-tolerant filters for detection of road signs in background noise,” *IEEE Transactions on Vehicular Technology*, vol. 51, no. 3, pp. 567–576, 2002.

- [67] 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.
- [68] 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.
- [69] A. Soetedjo and K. Yamada, "Traffic sign classification using ring partitioned method," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 88, no. 9, pp. 2419–2426, 2005.
- [70] I. Sebanja and D. Megherbi, "Automatic detection and recognition of traffic road signs for intelligent autonomous unmanned vehicles for urban surveillance and rescue," in *IEEE International Conference on Technologies for Homeland Security (HST)*, pp. 132–138, 2010.
- [71] M. Shi, H. Wu, and H. Fleyeh, "Support vector machines for traffic signs recognition," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pp. 3820–3827, 2008.
- [72] 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.
- [73] 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.
- [74] L. Chen, Q. Li, M. Li, and Q. Mao, "Traffic sign detection and recognition for intelligent vehicle," in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 908–913, 2011.
- [75] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, pp. 404–417, Springer, 2006.
- [76] 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.
- [77] 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.
- [78] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3d localisation," *Machine Vision and Applications*, vol. 25, no. 3, pp. 633–647, 2014.

- [79] 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.
- [80] 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.
- [81] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, “Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1484–1497, 2012.
- [82] 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.
- [83] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [84] T. Kowsari, S. S. Beauchemin, M. A. Bauer, D. Laurendeau, and N. Teasdale, “Multi-depth cross-calibration of remote eye gaze trackers and stereoscopic scene systems,” in *IEEE Intelligent Vehicles Symposium*, pp. 1245–1250, 2014.
- [85] 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 on Intelligent Transportation Systems (ITSC)*, pp. 526–531, 2011.
- [86] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [87] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [88] A. Vedaldi, “An open implementation of the sift detector and descriptor,” *UCLA CSD*, 2007.
- [89] M. A. Fischler and R. C. Bolles, “A paradigm for model fitting with applications to image analysis and automated cartography (reprinted in readings in computer vision, ed. ma fischler;,” *Comm. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [90] K. G. Derpanis, “Overview of the ransac algorithm,” *Image Rochester NY*, vol. 4, no. 1, pp. 2–3, 2010.
- [91] T. Fawcett, “An introduction to roc analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [92] A. Møgelmoose, D. Liu, and M. M. Trivedi, “Traffic sign detection for us roads: Remaining challenges and a case for tracking,” in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1394–1399, 2014.

Curriculum Vitae

Name: Seyedjamal Zabihi

Post-Secondary Education and Degrees: Islamic Azad University of Mashhad (IAUM)
Mashhad, Iran
2011 - 2015 Bachelor of Computer Engineering

The University of Western Ontario
London, ON
2015 - 2016 M.Sc. Computer Science

Honours and Awards: Western Graduate Research Scholarship
2015 - 2016

Related Work Experience: Teaching Assistant and Research Assistant
The University of Western Ontario
2015 - 2016